

SIEMENS
NIXDORF

SINIX

SINIX V5.40 (MX500)

SINIX V5.41 (MX300/WX200)

Referenzhandbuch für Systemverwalter

Beschreibung

Sie haben

uns zu diesem Handbuch etwas mitzuteilen?
Schicken Sie uns bitte Ihre Anregungen unter
Angabe der Bestellnummer dieses Handbuches.

Siemens Nixdorf Informationssysteme AG
Manualredaktion STM QM 2
Otto-Hahn-Ring 6
W-8000 München 83

Fax: (089) 636-40443

email im EUnet:
man@sieqm2.uucp

SINIX V5.40 (MX500) SINIX V5.41(MX300/WX200)

Referenzhandbuch für Systemverwalter

Beschreibung

Ausgabe Mai 1992 (SINIX V5.41)

Wollen Sie mehr wissen ...

... über dieses Produkt
... oder ein anderes Thema der Informationstechnik?

Unsere Training Center stehen für Sie bereit.
Besuchen Sie uns in Berlin, Essen, Frankfurt/Main oder Hamburg,
in Hannover, Mainz, München, Stuttgart, Wien oder Zürich.

Auskunft und Informationsmaterial erhalten Sie über:

München (089) 636-2009
oder schreiben Sie an:

Siemens Nixdorf Training Center
Postfach 830951, W-8000 München 83



SINIX® Copyright © Siemens Nixdorf Informationssysteme AG 1990.
SINIX ist das UNIX® der Siemens Nixdorf Informationssysteme AG.
UNIX ist ein eingetragenes Warenzeichen von UNIX System Laboratories, Inc.

Copyright © Siemens Nixdorf Informationssysteme AG, 1992. Alle Rechte vorbehalten.
Weitergabe sowie Vervielfältigung oder Übersetzung dieser Unterlage, Verwertung und
Mitteilung ihres Inhaltes nicht gestattet, soweit nicht ausdrücklich zugestanden.
Zu widerhandlungen verpflichten zu Schadenersatz.
Liefermöglichkeiten und technische Änderungen vorbehalten.

Inhalt

1. Kommandos

intro	1-1
accept, reject	1-3
acct	1-4
acctcms	1-6
acctcon, acctcon1, acctcon2	1-8
acctmerg	1-10
acctprc, acctprc1, acctprc2	1-11
acctsh, chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, runacct, shutacct, startup, turnacct	1-13
addbad (MX500)	1-16
arp	1-18
automount (NFS)	1-19
autopush	1-24
backup	1-25
biod (NFS)	1-29
bkexcept	1-30
bkhhistory	1-34
bkoper	1-36
bkreg	1-38
bkstatus	1-47
boot (MX300)	1-50
boot (MX500)	1-55
boot (WX200)	1-60
bootflags (MX500)	1-67
bp (MX500)	1-69
bpchk (MX500)	1-70
brc, bcheckrc	1-71
captainfo	1-72
checkfsys	1-73
chroot	1-75
chrtbl	1-76
ckbinarsys	1-81
ckbupscd	1-82
ckdate, errdate, helpdate, valdate	1-84
ckgid, errgid, helpgid, valgid	1-87
ckint, errint, helpint, valint	1-89
ckitem, errintem, helpitem	1-91
ckkeywd	1-94

ckpath, errpath, helppath, valpath	1-96
ckrange, errrange, helprange, valrange	1-99
ckstr, errstr, helpstr, valstr	1-101
cktime, errtime, helptime, valtime	1-103
ckuid, erruid, helpuid, valuid	1-105
ckyorn, erryorn, helpyorn, valyorn	1-107
colltbl	1-109
comsat, in.comsat	1-113
consoletype	1-114
crash	1-115
cron	1-123
custom	1-125
dcopy (generisch)	1-128
dcopy (s5)	1-129
dd	1-130
delsysadm	1-132
devattr	1-134
devfree	1-135
devnm	1-136
devreserv	1-137
df (generisch)	1-139
df (s5)	1-141
df (ufs)	1-142
dfmounts (DFS)	1-143
dfmounts (NFS)	1-144
dfshares (DFS)	1-145
dfshares (NFS)	1-146
diskadd (MX300)	1-147
diskadd (MX500)	1-149
diskadd (WX200)	1-151
disksetup	1-153
diskusg	1-156
dispadmin	1-158
dispgid	1-160
dispuid	1-161
du	1-162
dumping (WX200)	1-163
edquota (ufs)	1-164
edsysadm	1-165
edvtoc (MX300, MX500)	1-168

edvtoc (WX200).....	1-169
fconv (MX300, ufs)	1-170
fdisk (MX300, MX500)	1-171
fdisk (WX200).....	1-173
fdp	1-175
ff (generisch)	1-178
ff (s5).....	1-179
ff (ufs)	1-181
ffile	1-182
fimage	1-186
fingerd, in.fingerd	1-190
fixperm	1-191
fromsmtp	1-194
fsba (s5)	1-195
fsck (generisch)	1-196
fsck (bfs).....	1-198
fsck (s5)	1-199
fsck (ufs)	1-202
fsdb (generisch)	1-204
fsdb (s5)	1-205
fsdb (ufs)	1-208
fstyp (generisch)	1-209
ftpd, in.ftpd	1-210
fuser	1-214
fwtmp, wtmpfix	1-216
gencc	1-217
getdev	1-218
getdgrp	1-220
gettable	1-222
getty	1-223
getvol	1-225
groupadd	1-227
groupdel	1-228
groupmod	1-229
htable	1-230
id	1-231
idbuild	1-232
idcheck	1-234
idconfig	1-236
iddefaults (MX300)	1-238

iddefaults (MX500)	1-240
iddefaults (WX200)	1-242
idinstall	1-244
idmkinit	1-247
idmnod	1-249
idmkunix	1-252
idspace	1-253
idtune	1-255
ifconfig	1-256
incfile	1-259
inetd	1-263
infocmp	1-264
init, telinit	1-268
install	1-272
killall	1-274
labelit (generisch)	1-275
labelit (s5)	1-276
labelit (ufs)	1-277
ldsysdump	1-278
link, unlink	1-279
listdgrp	1-280
listen	1-281
logins	1-283
lpadmin	1-285
lpfilter	1-297
lpforms	1-302
lpsched, lpshut, lpmove	1-309
lpsystem	1-310
lpusers	1-313
mail_pipe	1-314
makefsys	1-316
migration	1-317
mkfifo	1-320
mkfs (generisch)	1-321
mkfs (bfs)	1-323
mkfs (s5)	1-324
mkfs (ufs)	1-327
mnod	1-329
mkpart (MX300)	1-330
mkpart (WX200)	1-334

monitor (MX500).....	1-339
montbl	1-343
mount, umount	1-346
mount (bfs)	1-348
mount (nfs)	1-349
mount (s5)	1-352
mount (ufs)	1-353
mountall, mountallp, umountall	1-355
mountd (NFS)	1-357
mountfsys, umountfsys	1-358
mkdir	1-359
named, in.named	1-360
ncheck (generisch)	1-363
ncheck (s5)	1-364
ncheck (ufs)	1-365
netstat	1-366
newgrp	1-369
nfsd (NFS)	1-370
nlsadmin	1-371
nslookup	1-376
nvrn (MX300)	1-381
offline (MX500).....	1-385
online (MX500).....	1-386
passmgmt	1-387
ping	1-389
pkgadd	1-390
pkgask	1-392
pkgchk	1-393
pkginfo	1-395
pkgparam	1-397
pkgrm	1-398
pkgtrans	1-399
pmadm	1-401
profiler.....	1-406
prvtoc	1-407
putdev	1-410
putdgrp	1-414
pwck, grpck	1-416
pwconv	1-417
quot (ufs)	1-418

quota (ufs)	1-419
quotacheck (ufs)	1-420
quotaon, quotaoff (ufs)	1-421
rarpd	1-422
rc0	1-423
rc2	1-425
rc6	1-427
rdate	1-428
relogin	1-429
rep (WX200)	1-430
repquota (ufs)	1-436
restore	1-437
rexecd, in.rexecd	1-440
rlogind, in.rlogind	1-442
rootcp (MX300)	1-444
route	1-446
routed, in.routed	1-448
rshd, in.rshd	1-451
runacct	1-454
rwhod, in.rwhod	1-456
sac	1-458
sacadm	1-461
sar	1-466
savecore (MX300).....	1-468
savecore (WX200)	1-471
scsigen (MX300, MX500)	1-472
scsigen (WX200).....	1-474
setclk	1-476
setmnt	1-477
setuname	1-478
setup	1-479
share (DFS)	1-480
share (NFS)	1-481
shareall, unshareall (DFS)	1-483
shutdown	1-484
slink	1-486
smtp	1-490
smtpd	1-491
smtpqer	1-493
smtpsched	1-495

strace	1-496
strclean	1-498
strerr	1-499
sttydefs.....	1-501
su	1-503
sulogin	1-505
swap	1-506
sync	1-508
sysadm (MX300)	1-509
sysadm (MX500).....	1-519
sysadm (WX200).....	1-528
sysrestore (MX300)	1-538
syssave (MX300)	1-541
talkd, in.talkd	1-546
telnetd, in.telnetd	1-547
tftpd, in.tftpd	1-549
tic	1-550
tnamed, in.tnamed	1-552
tosmtp	1-553
trpt	1-554
ttyadm.....	1-556
ttymon.....	1-558
tunefs (ufs)	1-561
uadmin	1-563
ufsdump (ufs)	1-564
ufsrestore (ufs)	1-567
unshare (DFS).....	1-572
unshare (NFS)	1-573
useradd	1-574
userdel	1-577
usermod	1-578
ucheck	1-581
uucico	1-582
uucleanup	1-584
uusched	1-586
Uutry	1-587
uuxqt	1-588
volcopy (generisch)	1-589
volcopy (s5)	1-590
volcopy (ufs)	1-591

wall	1-592
whodo	1-593
xfscck	1-595
xinstall	1-598
xrestore, xrestor	1-599
xts	1-601
xtt	1-602
zdump	1-603
zic	1-604

4. Dateiformate

intro	4-1
acct	4-2
admin	4-4
ar	4-7
archives	4-10
binarsys	4-13
cron	4-14
dfstab	4-15
dir (s5)	4-16
dir (ufs)	4-17
dirent	4-18
.environ, .pref, .variables	4-19
ethers	4-21
fd	4-22
filehdr	4-23
fs (bfs)	4-25
fs (s5)	4-26
fs (ufs)	4-29
fspec	4-32
fstypes	4-34
gettydefs	4-35
group	4-37
hosts	4-38
hosts.equiv, .rhosts	4-39
inetd.conf	4-40
inittab	4-42
inode (bfs)	4-46
inode (s5)	4-47

inode (ufs)	4-48
issue	4-50
limits	4-51
login	4-53
loginlog	4-55
mailcnfg	4-56
mailsurr	4-59
mdevice (MX300, MX500)	4-67
mdevice (WX200)	4-72
mfsys	4-76
mnttab	4-77
mtune	4-78
netconfig	4-79
netmasks	4-83
netrc	4-84
networks	4-86
.ott	4-87
passwd	4-88
pathalias	4-89
pnch	4-90
/proc	4-91
profile	4-104
protocols	4-106
resolv.conf	4-107
routing	4-108
rt_dptbl	4-110
savetab (MX300)	4-113
sccsfile	4-115
sdevice (MX300, MX500)	4-118
sdevice (WX200)	4-125
services	4-127
sfsys	4-128
shadow	4-129
sharetab	4-130
strcf	4-131
strftime	4-136
stune	4-138
su	4-139
term	4-140
terminfo	4-145

timezone.....	4-194
ts_dptbl.....	4-195
ttysrch.....	4-199
unistd.....	4-201
utmp, wtmp.....	4-204
utmpx, wtmpx.....	4-206
vfstab.....	4-208

5. Sonstiges

intro.....	5-1
ascii.....	5-2
environ.....	5-4
fcntl.....	5-10
iconv.....	5-12
jagent.....	5-24
langinfo.....	5-25
layers.....	5-27
nl_types.....	5-30
regex.....	5-31
siginfo.....	5-36
signal.....	5-38
stat.....	5-42
term.....	5-44
xtproto.....	5-48

7. Gerätedateien

intro	7-1
acc	7-2
adp32 (MX300, MX500)	7-3
aha (WX200)	7-4
ARP	7-5
cdr (MX300, WX200)	7-8
cram (WX200)	7-9
ct (MX500)	7-10
digi (WX200)	7-14
disk	7-15
display (WX200).....	7-17
exa (MX300, MX500)	7-33

exa (WX200)	7-35
fas (WX200)	7-37
fd	7-38
filesystem	7-41
hd (MX300, MX500)	7-45
hd (WX200).....	7-47
ICMP	7-51
inet	7-53
IP	7-56
keyboard (WX200)	7-60
lad	7-75
lo	7-77
lp (WX200).....	7-78
mbad (MX500)	7-79
mem, kmem	7-82
null	7-83
prf	7-84
rtc (WX200)	7-85
shd (MX300, MX500)	7-86
shd (WX200)	7-89
si (MX300, MX500)	7-92
sr (MX300, MX500)	7-95
ssd (MX500)	7-96
sx (MX300, MX500)	7-98
sxt	7-99
TCP	7-101
termio	7-104
termiox	7-121
ttcompat	7-127
tty	7-134
UDP	7-135
xmt (MX300, MX500)	7-138
xt.....	7-141
zero	7-144

0 Einleitung

Einleitung

0-1

Einleitung

Im vorliegenden *Referenzhandbuch für Systemverwalter* werden Kommandos, Dateiformate und sonstige Einrichtungen beschrieben, die der Verwalter eines Systems SINIX V5.40/V5.41 verwenden kann. Das Handbuch basiert größtenteils auf der AT&T-Beschreibung von UNIX System V. Garantiert werden ausschließlich die SINIX API-Schnittstellen.

Bei Einträgen, die nur für die MX300 (SINIX V5.41) gelten, steht in der Kopfzeile zentriert (MX300). Einträge, die nur für die WX200 (SINIX V5.41) gelten, sind in der Kopfzeile mit (WX200) gekennzeichnet. Und Einträge, die nur für die MX500 (SINIX V5.40) gelten, sind in der Kopfzeile mit (MX500) gekennzeichnet. Beschreibungen ohne Angabe einer Hardware-Plattform, treffen sowohl für SINIX V5.40 als auch für SINIX V5.41 zu.

Weitere wichtige Informationen sind in anderen, verwandten Handbüchern zu finden:

- Der *Leitfaden für Systemverwalter* enthält Prozeduren für Verwaltungsaufgaben und Erläuterungen dazu.
- Der *Leitfaden für Benutzer* gibt einen Überblick über das Betriebssystem SINIX und enthält eine Anleitung zur Verwendung von Editoren, zum Automatisieren sich wiederholender Aufgaben sowie zum Senden von Informationen an andere Benutzer.
- Die Bände *Kommandos* beschreiben die Kommandos, die zur grundlegenden Software gehören, die auf dem System vorhanden ist.
- Das Handbuch *CES, Leitfaden und Werkzeuge für die Programmierung mit C* gibt einen Überblick über die Programmierungsumgebung des Betriebssystems SINIX sowie Anleitungen zu den verschiedenen Programmierwerkzeugen.
- Im *Referenzhandbuch für Programmierer* werden die von Programmierern verwendeten Kommandos, Systemaufrufe, Unterroutinen, Bibliotheken, Dateiformate und sonstigen Einrichtungen beschrieben.

Das *Referenzhandbuch für Systemverwalter* ist in vier Kapitel gegliedert:

- (1) Kommandos zur Systempflege und Anwendungsprogramme
- (4) Dateiformate
- (5) Sonstiges
- (7) Gerätedateien

Zahlen, die nach einem Kommandonamen erscheinen, sind Querverweise. Ist nach einem Kommando (1M) oder (7) angegeben, steht dieses Kommando im entsprechenden Kapitel dieses Handbuchs. Folgt einem Kommando eine (4) oder (5), befindet es sich in dem entsprechenden Kapitel des vorliegenden Handbuchs und des *Referenzhandbuchs für Programmierer*. Steht nach dem Kommando (1), (1C) oder (1G), ist es gewöhnlich in einem Band der *Kommandos* zu finden. Kommandos aus Kapitel 1, die zur Verwendung durch Programmierer geeignet sind, enthält das *Referenzhandbuch für Programmierer*. Kommandos, die mit (2) oder (3) gekennzeichnet sind, stehen im entsprechenden Kapitel des *Referenzhandbuchs für Programmierer*. Mit (6) sind Spiele gekennzeichnet. Sie werden nicht beschrieben.

Kapitel 1M, "Kommandos zur Systempflege und Anwendungsprogramme", enthält Kommandos und Programme, die zur Verwaltung eines SINIX-Systems benötigt werden.

Kapitel 4, "Dateiformate", dokumentiert die Struktur besonderer Dateiarnten. So wird beispielsweise das Format der Datei `/etc/passwd` unter `passwd(4)` angegeben, und der Inhalt der Datei `/etc/profile` wird unter `profile(4)` erläutert. Wenn eine Struktur der Programmiersprache C einem Dateiformat entspricht, gilt allgemein, daß sie im Verzeichnis `/usr/include` oder `/usr/include/sys` zu finden ist.

Kapitel 5, "Sonstiges", enthält verschiedene Informationen. So findet sich z.B. unter `ascii(5)` eine Tabelle der oktalen und hexadezimalen Entsprechungen des ASCII-Zeichensatzes, Shell-Umgebungsvariablen wie `HOME`, `PATH`, `LANGUAGE` usw. sind unter `environ(5)` beschrieben, und die Namen gängiger Terminals sind unter `term(5)` aufgeführt.

Kapitel 7, "Gerätedateien" beschreibt die Merkmale von Systemdateien, die sich auf Ein-/Ausgabegeräte beziehen. Die Namen in diesem Kapitel beziehen sich im allgemeinen eher auf die Hardware-Gerätenamen als auf die Namen der Gerätedateien.

Alle Kapitel beginnen mit einer Einführungsseite mit dem Titel `intro`. Die Einträge nach der Seite `intro` sind alphabetisch geordnet und können aus mehreren Seiten bestehen. Manche Einträge beschreiben mehrere Routinen, Kommandos usw. In solchen Fällen erscheint der Eintrag nur einmal alphabetisch eingeordnet unter seinem "Primärnamen". Ein Beispiel für einen solchen Eintrag ist `mount(1M)`, in dem auch das Kommando `umount` beschrieben wird.

Am Anfang des Handbuchs finden Sie ein Inhaltsverzeichnis.

Alle Einträge sind nach einem allgemeinen Format gestaltet, allerdings enthält nicht jeder Eintrag alle Elemente.

- **BEZEICHNUNG** gibt den/die Namen des Eintrags an und enthält eine Kurzbeschreibung.
- **ÜBERSICHT** faßt die Verwendung des beschriebenen Programms zusammen. Hierbei gelten einige Konventionen, insbesondere in Kapitel 1M (*Kommandos*):
 - Nicht-Proportionalschrift kennzeichnet Literale, die genau so eingegeben werden müssen, wie sie im Handbuch erscheinen.
 - *Kursiv* gedruckte Zeichenketten geben normalerweise zu ersetzende Argumente an oder Programmnamen, die an anderer Stelle im Handbuch zu finden sind. In der maschinegeschriebenen Version der Einträge sind diese Elemente unterstrichen.
 - Eckige Klammern `[]` um ein Argument zeigen an, daß dieses Argument optional ist. Wird ein Argument als "name" oder "datei" angegeben, ist damit immer ein Dateiname gemeint.
 - Auslassungspunkte `...` zeigen an, daß das vorherige Argument wiederholt werden darf.
- **BESCHREIBUNG** enthält eine Erläuterung des Kommandos.
- **BEISPIEL(E)** gibt ein oder mehrere Anwendungsbeispiele an, soweit dies sinnvoll ist.
- **DATEIEN** gibt die Dateien an, die zu dem Programm gehören.
- **SIEHE AUCH** verweist auf Referenzinformationen.

- **FEHLERMELDUNGEN** erläutert die Fehlermeldungen, die vorkommen können. Meldungen, die nicht erklärungsbedürftig sind, werden nicht aufgeführt.
- **HINWEIS** weist auf mögliche Fehlerquellen, Fehler oder Mängel hin.

1 Kommandos

**Kommandos zur Systempflege und
Anwendungsprogramme**

1-1

BEZEICHNUNG

intro – Einführung zu den Wartungskommandos und Verwaltungsprogrammen

BESCHREIBUNG

Dieses Kapitel enthält Kommandos in alphabetischer Reihenfolge, die hauptsächlich zur Systemwartung und Systemverwaltung verwendet werden. Die Kommandos in diesem Kapitel sind zusammen mit denen in Kapitel 1 der *Kommandos* und den Kapiteln 1, 2, 3, 4 und 5 des *Referenzhandbuchs für Programmierer* zu verwenden. Verweise mit der Form *Name*(1), (2), (3), (4) und (5) beziehen sich auf Einträge in diesen Handbüchern. Verweise mit der Form *Name*(1M), *Name*(7) oder *Name*(8) beziehen sich auf Einträge im vorliegenden Handbuch.

Da die Kommandos für die Architektur des virtuellen Dateisystems neu strukturiert wurden, erscheinen an einigen Stellen mehrere Handbuchseiten unter demselben Namen. Beispielsweise gibt es vier Handbuchseiten namens *mount*(1M). In diesen Fällen werden auf der ersten Seite Syntax und Optionen des eigentlichen (generischen) Kommandos beschrieben, d.h. die Optionen des Kommandos, die für alle Dateisystemtypen gelten. Die folgenden Einträge beschreiben die Funktionen der dateisystem-spezifischen Module des Kommandos. Bei diesen Einträgen ist der Name des Dateisystemtyps, zu dem sie gehören, in der Kopfzeile der Seite zentriert und in Klammern angegeben. Beachten Sie bitte, daß der Systemverwalter nicht versuchen sollte, diese Module direkt aufzurufen. Das generische Kommando bietet eine gemeinsame Schnittstelle für alle Module. Daher dürfen die dateisystemtyp-spezifischen Handbucheinträge nicht als Beschreibungen separater Kommandos, sondern als Ergänzung zu diesen Aspekten eines Kommandos gesehen werden.

KOMMANDOSYNTAX

Sofern nicht anderweitig vermerkt, können zu den Kommandos in diesem Kapitel Optionen und Argumente in der folgenden Syntax angegeben werden:

name [*option*(*en*)] [*kmdarg*(*e*)]

Dabei ist:

name Der Name einer ausführbaren Datei

option – *keinargbuchst*(*n*) oder
– *argbuchst*<>*optarg*
wobei <> ein optionales Zwischenraumzeichen ist

keinargbuchst Ein Einzelbuchstabe, der eine Option ohne Argument darstellt

argbuchst Ein Einzelbuchstabe, der für eine Option steht, die ein Argument benötigt

optarg Argument (Zeichenkette) des vorausgegangenen *argbuchst*

kmdarg Pfadname (oder anderes Kommandoargument), der *nicht* mit einem Bindestrich beginnt, oder der Bindestrich allein bezeichnet die Standardeingabe

SIEHE AUCH

getopt(1) in den *Kommandos*.

getopt(3C) im *Referenzhandbuch für Programmierer*.

FEHLERMELDUNGEN

Konnte ein Kommando vollständig ausgeführt werden, wird bei Beendigung eine 0 zurückgegeben; wenn aber Fehler wie falsche Parameter, fehlerhafte oder nicht zugängliche Daten oder andere Schwierigkeiten beim Ausführen der aktuellen Aufgabe auftraten, wird ein anderer Wert zurückgegeben. Dieser Wert wird als "Exit-Code," "Exit-Status" oder "Return-Code" bezeichnet und nur dann eigens beschrieben, wenn es sich um besondere Konventionen handelt.

HINWEIS

Leider orientieren sich nicht alle Kommandos an der Standardsyntax.

accept (1M)

accept (1M)

BEZEICHNUNG

accept, reject – Annehmen oder Abweisen von Aufträgen zum Drucken

ÜBERSICHT

accept *ziele*

reject [*-r grund*] *ziele*

BESCHREIBUNG

Mit **accept** können Aufträge zum Drucken für die benannten *ziele* in eine Warteschlange eingereiht werden. Ein *ziel* kann ein Drucker oder eine Druckerklasse sein. Mit **lpstat -a** läßt sich der Status der *ziele* feststellen.

reject verhindert, daß Druckaufträge für die benannten *ziele* in eine Warteschlange gestellt werden. Ein *ziel* kann ein Drucker oder eine Druckerklasse sein. Mit **lpstat -a** läßt sich der Status der *ziele* feststellen. Die folgende Option ist nur beim Kommando **reject** zu verwenden.

-r grund Für die Abweisung von Anforderungen wird ein *grund* angegeben. Dieser *grund* gilt für alle angegebenen *ziele*. Der *grund* wird mit **lpstat -a** angezeigt. Enthält diese Angabe Leerzeichen, muß sie in Anführungszeichen stehen. Der Standardwert für bestehende Ziele ist *unknown reason* (unbekannter Grund), für Ziele, die gerade in das System eingefügt wurden, aber noch keine Aufträge annehmen, *new destination* (neues Ziel).

DATEIEN

/var/spool/lp/*

SIEHE AUCH

lpadmin(1M), **lpsched(1M)**.

enable(1), **lp(1)**, **lpstat(1)** in den *Kommandos*.

BEZEICHNUNG

acct: acctdisk, acctdusg, accton, acctwtmp closewtmp, utmp2wtmp -
Allgemeines über Abrechnung und verschiedene Abrechnungskommandos

ÜBERSICHT

```
/usr/lib/acct/acctdisk  
/usr/lib/acct/acctdusg [-u datei] [-p datei]  
/usr/lib/acct/accton [datei]  
/usr/lib/acct/acctwtmp "grund"  
/usr/lib/acct/closewtmp  
/usr/lib/acct/utmp2wtmp
```

BESCHREIBUNG

Abrechnungs-Software besteht aus verschiedenen Werkzeugen (C-Programmen ebenso wie Shell-Prozeduren), aus denen sich Abrechnungssysteme aufbauen lassen. Unter acctsh(1M) sind die Shell-Prozeduren beschrieben, die auf den C-Programmen aufbauen.

Die Abrechnung der Verbindungszeiten wird von verschiedenen Programmen durchgeführt, die Datensätze in /var/adm/wtmp schreiben, wie in utmp(4) beschrieben. Die unter acctcon(1M) erläuterten Programme wandeln diese Datei in Sitzungs- und Rechnungsdatensätze um, die anschließend mit acctmrg(1M) zusammengefaßt werden.

Die Prozeßabrechnung übernimmt der SINIX-Systemkern. Nach dem Abschluß eines Prozesses wird ein Datensatz pro Prozeß in eine Datei geschrieben (normalerweise /var/adm/pacct). Mit den in acctprc(1M) aufgeführten Programmen werden diese Daten zur Rechnungsstellung zusammengefaßt; mit acctcms(1M) wird die Benutzung der Kommandos zusammengefaßt. Zum Überprüfen der aktuellen Prozeßdaten steht acctcom(1) zur Verfügung.

Prozeß- und Verbindungszeitabrechnung (oder andere Abrechnungsdatensätze im tacct-Format, wie in acct(4) beschrieben) können mit acctmrg zu Gesamtabrechnungssätzen kombiniert und zusammengefaßt werden (siehe das tacct-Format in acct(4)). prtacct (siehe acctsh(1M)) dient zum Formatieren einzelner oder aller Abrechnungsdatensätze.

acctdisk liest Zeilen, die Benutzernummer, Benutzername und die Anzahl der Plattenblöcke enthalten, und wandelt sie in Gesamtabrechnungssätze um, die mit anderen Abrechnungssätzen gemischt werden können.

acctdusg liest die Standardeingabe (gewöhnlich aus find / -print) und berechnet den beanspruchten Plattenspeicherplatz (einschließlich der Zeigerblöcke) für Benutzernamen. Über die Angabe -u werden Datensätze mit den Dateinamen, die acctdusg keinem Benutzer in Rechnung stellt, in die *datei* geschrieben (eine Möglichkeit, Benutzer zu finden, die einer Belastung für beanspruchten Plattenspeicherplatz entgehen wollen). Bei der Angabe -p ist *datei* der Name der Paßwortdatei. Handelt es sich bei der Paßwortdatei um /etc/passwd, muß diese Option nicht angegeben werden. (Näheres finden Sie unter diskusg(1M).)

Nur mit `accton` wird die Prozeßabrechnung ausgeschaltet. Als *datei* kann der Name einer bestehenden Datei angegeben werden, in die der Systemkern die Prozeßabrechnungssätze schreibt (siehe `acct(2)` und `acct(4)`).

`acctwtmp` schreibt einen `utmp(4)`-Satz in die Standardausgabe. Der Datensatz enthält die aktuelle Uhrzeit und eine Zeichenkette für *grund*. Der Satztyp `ACCOUNTING` wird zugewiesen (siehe `utmp(4)`). *grund* muß eine Zeichenkette aus maximal 11 Zeichen, Ziffern, Dollarzeichen \$ oder Leerzeichen sein. Es folgen zwei Beispiele, die in `reboot`- bzw. `shutdown`-Prozeduren verwendet werden können:

```
acctwtmp "acctg on" >> /var/adm/wtmp
acctwtmp "acctg off" >> /var/adm/wtmp
```

Für jeden aktuell angemeldeten Benutzer schreibt `closewtmp` einen falschen `DEAD_PROCESS`-Satz in die Datei `/var/adm/wtmp`. `runacct` (siehe `runacct(1M)`) arbeitet mit diesem falschen `DEAD_PROCESS`-Satz, so daß mit den Prozeduren zur Verbindungsabrechnung die Zeit festgestellt werden kann, die die Benutzer vor dem Aufrufen von `runacct` verbraucht haben.

Für jeden aktuell angemeldeten Benutzer legt `runacct` mit `utmp2wtmp` einen Eintrag in der mit `runacct` erstellten Datei `/var/adm/wtmp` an. Mit den Einträgen in `/var/adm/wtmp` kann `runacct` bei der späteren Aktivierung die Verbindungszeiten der aktuell angemeldeten Benutzer erfassen.

DATEIEN

<code>/etc/passwd</code>	verwendet für die Konvertierung von Benutzernamen zu Benutzernummern
<code>/usr/lib/acct</code>	enthält alle in Abschnitt 1M dieses Handbuchs verzeichneten Abrechnungsskripts
<code>/var/adm/pacct</code>	aktuelle Prozeßabrechnungsdatei
<code>/var/adm/wtmp</code>	An-/Abmelde-Protokolldatei

SIEHE AUCH

`acctcms(1M)`, `acctcon(1M)`, `acctmerg(1M)`, `acctprc(1M)`, `acctsh(1M)`, `diskusg(1M)`, `fwtmp(1M)`, `runacct(1M)`, `acct(4)`, `utmp(4)`
`acctcom(1)` in den *Kommandos*
`acct(2)` im *Referenzhandbuch für Programmierer*

BEZEICHNUNG

`acctcms` - ermittelt die Kommandonutzung aus den Abrechnungssätzen pro Prozeß

ÜBERSICHT

`/usr/lib/acct/acctcms [-a [-p] [-o]] [-c] [-j] [-n] [-s] [-t] dateien`

BESCHREIBUNG

`acctcms` liest eine oder mehrere *dateien*, normalerweise in der in `acct(4)` beschriebenen Form. Es faßt alle Sätze für Prozesse zusammen, mit denen Kommandos gleichen Namens ausgeführt wurden, sortiert sie und schreibt sie, normalerweise in einem internen Format, in die Standardausgabe. Folgende Optionen stehen zur Verfügung:

- a Ausgabe im ASCII-Format anstatt im internen Format. Angegeben werden: Name des Kommandos, Anzahl der Aktivierungen, kcore-Zeit in Minuten, CPU-Zeit in Minuten, reale Zeit in Minuten, durchschnittliche Größe (in K), durchschnittliche CPU-Zeit pro Aufruf, Monopolisierungsfaktor, Anzahl der übertragenen Zeichen sowie Anzahl der gelesenen und geschriebenen Blöcke, wie in `acctcom(1)`. Die Ausgabe wird in der Regel nach der Gesamt-kcore-Zeit (in Minuten) sortiert.
- c Sortieren nach Gesamt-CPU-Zeit statt Gesamt-kcore-Zeit.
- j Zusammenfassung aller nur einmal aufgerufenen Kommandos unter "***other".
- n Sortieren nach Anzahl der Aktivierungen.
- s Alle nachfolgend gefundenen Dateinamen stehen bereits im internen Format.
- t Alle Datensätze werden als Gesamtabrechnungssätze verarbeitet. Das standardmäßige interne Format unterteilt jedes Feld in Haupt- und Nebena-brechnungszeiten. Bei dieser Option werden Haupt- und Nebena-brechnungszeit in einem Feld zusammengefaßt; diese Datensätze sind mit `acctcms`-Datensätzen im alten internen Format (d.h. SINIX-ODT und SINIX V5.2) kompatibel.

Die folgenden Optionen dürfen nur zusammen mit der Option `-a` benutzt werden.

- p Es wird eine Kommandoabrechnung nur für die Hauptabrechnungszeit ausgegeben.
- o Es wird eine Kommandoabrechnung nur für die Nebena-brechnungszeit ausgegeben.

Werden `-p` und `-o` zusammen verwendet, wird ein kombinierter Bericht für Haupt- und Nebena-brechnungszeit erstellt. Alle Summen in der Ausgabe geben die Gesamtnutzung an, mit Ausnahme der Anzahl der Aktivierungen, der CPU-Zeit und der realen Zeit (in Minuten), die nach Haupt- und Nebena-brechnungszeit unterteilt werden.

BEISPIEL

Eine typische Kommandofolge zum Erstellen der täglichen Abrechnung der Kommandonutzung und zur Fortschreibung der Gesamtsummen lautet:

```
acctcms file ... > today
cp total previoustotal
acctcms -s today previoustotal > total
acctcms -a -s today
```

SIEHE AUCH

acct(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M),
runacct(1M), acct(4), utmp(4)
acctcom(1) in den *Kommandos*
acct(2) im *Referenzhandbuch für Programmierer*

HINWEIS

Es kann zu unvorhersehbaren Ergebnissen kommen, wenn die Option `-t` bei Dateien mit dem neuen internen Format verwendet wird bzw. wenn sie mit dem älteren internen Format nicht verwendet wird.

BEZEICHNUNG

acctcon, acctcon1, acctcon2 – Abrechnen der Verbindungszeit

ÜBERSICHT

```
/usr/lib/acct/acctcon [-l datei] [-o datei]
/usr/lib/acct/acctcon1 [-l datei] [-o datei] [-p] [-t]
/usr/lib/acct/acctcon2
```

BESCHREIBUNG

Mit `acctcon` wird eine Folge von An-/Abmeldedatensätzen in Gesamtabrechnungssätze umgewandelt (siehe das Format `tacct` in `acct(4)`). Die An-/Abmeldesätze werden aus der Standardeingabe gelesen. Dies ist gewöhnlich die Datei `/var/adm/wtmp`. Da diese Datei jedoch beschädigte Daten oder Änderungen des Systemdatums enthalten kann, muß sie vor dem Lesen mit `wtmpfix` berichtigt werden. Die korrigierte Version der Datei `/var/adm/wtmp` kann dann an `acctcon` umgeleitet werden. Die `tacct`-Datensätze werden in die Standardausgabe geschrieben. Für `acctcon` stehen folgende Optionen zur Verfügung:

`-l datei` *datei* dient zur Aufzeichnung der Leitungsbelegung und enthält die Leitungskennung, die Belegungszeit in Minuten, den Anteil der insgesamt verbrauchten Zeit in Prozent, die Anzahl der berechneten Sitzungen, die Anzahl der Anmeldungen und die Anzahl der Abmeldungen. Mit Hilfe dieser Datei lassen sich die Auslastung der Leitungen, fehlerhafte Leitungen sowie Unregelmäßigkeiten in Software oder Hardware feststellen. Hängenzustände, der Abbruch von `login(1)` und der Abbruch der Anmelde-Shell generieren Abmeldesätze. Dadurch ist die Anzahl der Abmeldungen oft drei- bis viermal so hoch wie die Anzahl der Sitzungen. Siehe `init(1M)` und `utmp(4)`.

`-o datei` *datei* enthält einen globalen Datensatz für den Abrechnungszeitraum mit den Angaben Anfangs- und Abschlußzeit, Anzahl der Neustarts, Anzahl der Datumsänderungen.

`acctcon` ist eine Kombination der Programme `acctcon1` und `acctcon2`. `acctcon1` konvertiert An-/Abmeldesätze aus der korrigierten Datei `/var/adm/wtmp` in eine ASCII-Ausgabe. `acctcon2` liest die von `acctcon1` erstellten ASCII-Datensätze und wandelt sie in `tacct`-Datensätze um. `acctcon1` kann mit den oben beschriebenen Optionen `-l` und `-o` sowie mit den folgenden Optionen verwendet werden:

`-p` Nur die Eingabe mit Leitungskennung, Benutzername und Zeit (sowohl im numerischen als auch im Datum/Zeit-Format) ausgeben.

`-t` `acctcon1` pflegt eine Liste der Leitungen, auf denen Benutzer angemeldet sind. Am Ende der Eingabe wird für jede noch aktive Leitung ein Sitzungsdatensatz ausgegeben. Das Programm geht normalerweise davon aus, daß es sich bei der Eingabe um eine aktuelle Datei handelt, so daß es die aktuelle Zeit als Abschlußzeit für die noch aktiven Sitzungen verwendet. Die Option `-t` bewirkt dagegen, daß es die letzte in der Eingabe gefundene Zeitangabe verwendet und so für nicht aktuelle Dateien sinnvolle und wiederholbare Werte erzeugt.

BEISPIELE

Das Kommando `acctcon` wird normalerweise so verwendet:

```
acctcon -l lineuse -o reboots < tmpwtmp > ctacct
```

Die Kommandos `acctcon1` und `acctcon2` werden in der Regel folgendermaßen verwendet:

```
acctcon1 -l lineuse -o reboots < tmpwtmp | sort +1n +2 > ctmp  
acctcon2 < ctmp > ctacct
```

DATEIEN

`/var/adm/wtmp`

SIEHE AUCH

`acct(1M)`, `acctcms(1M)`, `acctmerge(1M)`, `acctprc(1M)`, `acctsh(1M)`, `fwtmp(1M)`,
`init(1M)`, `runacct(1M)`, `acct(4)`, `utmp(4)`
`acctcom(1)`, `login(1)` in den *Kommandos*
`acct(2)` im *Referenzhandbuch für Programmierer*

HINWEIS

Durch Datumsänderungen wird die Abrechnung der Leitungsbelegung ungenau.
Dies kann mit `wtmpfix` (siehe `fwtmp(1M)`) mit dem Dateinamen `/var/adm/wtmp` als
Argument bereinigt werden.

BEZEICHNUNG

acctmerg – Zusammenfassen oder Addieren von Gesamtabrechnungsdateien

ÜBERSICHT

/usr/lib/acct/acctmerg [-a] [-i] [-p] [-t] [-u] [-v] [datei ...]

BESCHREIBUNG

acctmerg liest die Standardeingabe und bis zu neun weitere Dateien, alle im tacct-Format (siehe acct(4)) oder in einer ASCII-Version davon. Es faßt diese Eingaben zusammen, indem es Datensätze mit identischen Schlüsseln (meist Benutzernummer und Benutzername) addiert. Es geht davon aus, daß die Eingabe nach diesen Schlüsseln sortiert ist. Folgende Optionen stehen zur Verfügung:

- a Erstellen der Ausgabe in der ASCII-Version von tacct.
- i Das Format der Eingabedateien ist die ASCII-Version von tacct.
- p Eingabe ausgeben, ohne sie zu verarbeiten.
- t Einen einzigen Datensatz zur Zusammenfassung der gesamten Eingabe erstellen.
- u Zusammenfassung nur nach Benutzernummer anstatt nach Benutzernummer und -namen.
- v Ausgabe in ausführlichem ASCII-Format mit präziserer Schreibweise von Gleitkommazahlen erstellen.

BEISPIELE

Das Beispiel zeigt, wie eine Abrechnungsdatei korrigiert werden kann. Dazu wird diese Datei mit dem ersten Kommando ins ASCII-Format umgewandelt:

```
acctmerg -v <datei1 > datei2
```

datei2 wie gewünscht bearbeiten ... Mit dem folgenden Kommando wird die Datei wieder ins tacct-Format umgewandelt:

```
acctmerg -i <datei2 > datei1
```

SIEHE AUCH

acct(1M), acctcms(1M), acctcon(1M), acctprc(1M), acctsh(1M), fwtmpt(1M), runacct(1M), acct(4), utmp(4)
acctcom(1) in den Kommandos
acct(2) im Referenzhandbuch für Programmierer

BEZEICHNUNG

acctprc, acctprc1, acctprc2 – Prozeßabrechnung

ÜBERSICHT

```
/usr/lib/acct/acctprc
/usr/lib/acct/acctprc1 [ctmp]
/usr/lib/acct/acctprc2
```

BESCHREIBUNG

acctprc liest Daten von der Standardeingabe in der unter acct(4) beschriebenen Form. Es wandelt diese in Gesamtabrechnungssätze um (siehe den tacct-Satz in acct(4)). acctprc unterteilt die CPU-Zeit in Hauptabrechnungszeit und Nebena-brechnungszeit und bestimmt die durchschnittliche Speichergröße (in Speichersegmenteinheiten). Dann faßt acctprc die tacct-Sätze nach Benutzernummern zusammen und fügt die Benutzernamen entsprechend den Benutzernummern hinzu. Die zusammengefaßten Datensätze werden nun in die Standardausgabe geschrieben. acctprc1 liest die Eingabe in der unter acct(4) beschriebenen Form, fügt Benutzernamen entsprechend den Benutzernummern hinzu und gibt dann für jeden Prozeß eine ASCII-Zeile mit Benutzernummer, Benutzername, Haupt-CPU-Zeit (Tics), Neben-CPU-Zeit (Tics) und durchschnittlicher Speichergröße (in Speichersegmenteinheiten) aus. Ist die Datei ctmp vorhanden, muß sie eine nach Benutzernummer und Benutzernamen sortierte Liste der Benutzersitzungen enthalten. Steht diese Datei nicht zur Verfügung, werden die Benutzernamen aus der Paßwortdatei entnommen wie bei acctprc. Anhand der Daten in ctmp kann man zwischen verschiedenen Benutzernamen unterscheiden, die eine gemeinsame Benutzernummer haben.

Aus der Standardeingabe liest acctprc2 Datensätze in der Form, wie acctprc1 sie schreibt, faßt sie nach Benutzernummer und -namen zusammen und schreibt die sortierten Zusammenfassungen als Gesamtabrechnungssätze in die Standardausgabe.

BEISPIELE

Das Kommando acctprc wird gewöhnlich verwendet, wie im folgenden dargestellt:

```
acctprc < /var/adm/pacct > ptacct
```

Die Kommandos acctprc1 und acctprc2 werden in der Regel folgendermaßen verwendet:

```
acctprc1 ctmp </var/adm/pacct | acctprc2 >ptacct
```

DATEIEN

```
/etc/passwd
```

SIEHE AUCH

acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctsh(1M), cron(1M), fwtmp(1M), runacct(1M), acct(4), utmp(4)
acctcom(1) in den *Kommandos*
acct(2) im *Referenzhandbuch für Programmierer*

HINWEIS

`acctprc1` kann Benutzernamen mit gemeinsamer Benutzernummer voneinander unterscheiden, sofern die Kommandos normal ausgeführt werden. Bei Kommandos, die z.B. über `cron(1M)` ausgeführt werden, ist dies ungleich schwieriger. Eine genauere Umwandlung ist mit dem Programm `acctwtm` in `acct(1M)` möglich. `acctprc` unterscheidet nicht zwischen Benutzern mit gleicher Benutzernummer.

Ein Speichersegment der durchschnittlichen Speichergröße ist die Maßeinheit für die Byte-Anzahl in einem logischen Speichersegment eines bestimmten Prozessors.

BEZEICHNUNG

chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, runacct, shutacct, startup, turnacct - Shell-Prozeduren für Abrechnung

ÜBERSICHT

```
/usr/lib/acct/chargefee benutzername anzahl
/usr/lib/acct/ckpacct [blöcke]
/usr/lib/acct/dodisk [-o] [dateien ...]
/usr/lib/acct/lastlogin
/usr/lib/acct/monacct nummer
/usr/lib/acct/nulladm datei
/usr/lib/acct/prctmp
/usr/lib/acct/prdaily [-l] [-c] [mmtt]
/usr/lib/acct/prtacct datei [ "kopf" ]
/usr/lib/acct/runacct [mmtt] [mmtt status]
/usr/lib/acct/shutacct [ "grund" ]
/usr/lib/acct/startup
/usr/lib/acct/turnacct on | off | switch
```

BESCHREIBUNG

Mit chargefee kann dem *benutzername* eine *anzahl* von Einheiten in Rechnung gestellt werden. Ein entsprechender Datensatz wird in die Datei */var/adm/fee* geschrieben und von runacct mit anderen Abrechnungssätzen gemischt.

ckpacct muß über cron(1M) aktiviert werden, um die Größe von */var/adm/pacct* in regelmäßigen Abständen zu überprüfen. Wird die Datei größer als *blöcke* (Anzahl der Blöcke - Standardwert 1000), wird turnacct mit dem Argument *switch* aufgerufen. Sinkt die Anzahl der freien Plattenblöcke im Dateisystem */var* auf unter 500, deaktiviert ckpacct das Sammeln von Prozeßabrechnungssätzen automatisch über das Argument *off* von turnacct. Sind wieder mindestens 500 Blöcke verfügbar, wird die Abrechnungsfunktion beim nächsten Aufruf von ckpacct wieder aktiviert. Diese Funktion hängt davon ab, wie oft ckpacct (gewöhnlich von cron) ausgeführt wird.

dodisk muß von cron zum Aktivieren der Plattenabrechnung aufgerufen werden. Standardmäßig ruft dodisk diskusg (siehe diskusg(1M)) auf, um die Plattenabrechnung im S5-Dateisystem im Dateiverzeichnis */etc/vfstab* durchzuführen. Die Option *-o* bewirkt, daß mit acctdusg (siehe acct(1M)) eine langsamere Version der Plattenabrechnung aktiviert wird, die nach Login-Verzeichnissen abrechnet. *dateien* gibt den oder die Namen des Dateisystems/der Dateisysteme an, für das/die die Plattenabrechnung durchgeführt werden soll. Werden *dateien* benannt, erfolgt die Plattenabrechnung ausschließlich für diese Dateisysteme. Wird die Option *-o* verwendet, muß es sich bei *dateien* um Einhängepunkte eingängter

Dateisysteme handeln. Wird die Option `-o` nicht angegeben, muß es sich bei *dateien* um die Gerätedateinamen einhängbarer Dateisysteme handeln.

`lastlogin` wird von `runacct` aktiviert, um die Datei `/var/adm/acct/sum/loginlog` zu aktualisieren, die anzeigt, wann sich die einzelnen Benutzer zuletzt angemeldet haben.

`monacct` muß einmal im Monat oder einmal pro Abrechnungszeitraum aktiviert werden. *nummer* gibt an, um welchen Monat oder Zeitraum es sich handelt. Wird *nummer* nicht angegeben, gilt als Standardwert der laufende Monat (01-12). Dieser Standardwert ist nützlich, wenn `monacct` über `cron(1M)` an jedem Monatsersten ausgeführt werden soll. `monacct` erstellt Übersichtsdateien in `/var/adm/acct/fiscal` und reaktiviert die Erstellung der Übersichtsdateien in `/var/adm/acct/sum`.

`nulladm` erstellt eine *datei* mit Modus 664 und sorgt dafür, daß Eigentümer und Gruppe `adm` sind. `nulladm` wird von verschiedenen Abrechnungsprozeduren aufgerufen.

`prctmp` kann zum Ausgeben der Abrechnungsdatei einer Sitzung verwendet werden (normalerweise die von `acctcon1` (siehe `acctcon(1M)`) erstellte Datei `/var/adm/acct/nite/ctmp`).

`prdaily` wird von `runacct` zum Formatieren eines Berichts mit den Abrechnungsdaten des Vortags aufgerufen. Der Bericht steht in `/var/adm/acct/sum/rprt/mmtt`, und *mmtt* gibt das Datum des Berichts mit Monat und Tag an. Zum Ausgeben der aktuellen Tagesberichte wird `prdaily` eingegeben. Zum Ausgeben der Abrechnungsberichte für die Vortage wird mit der Option *mmtt* das Datum des gewünschten Berichts genau angegeben. Mit der Option `-l` wird ein Bericht über außergewöhnliche Auslastung nach Benutzernummer für das angegebene Datum ausgegeben. Frühere Tagesberichte werden nach jeder Aktivierung von `monacct` bereinigt und stehen daher nicht mehr zur Verfügung. Mit der Option `-c` wird ein Bericht über außergewöhnliche Ressourcenauslastung pro Kommando erstellt. Diese Option darf nur für die Abrechnungsdaten des laufenden Tages verwendet werden.

`prtacct` wird zum Formatieren und Ausgeben aller Gesamtabrechnungsdateien (*tacct*-Dateien) verwendet.

Mit `runacct` werden eine tägliche kumulierte Verbindungs-, Prozeß-, Gebühren- und Plattenabrechnung sowie Übersichten zur Kommandonutzung erstellt. Weitere Informationen finden Sie unter `runacct(1M)`.

`shutacct` wird bei einem Systemabschluß aufgerufen, um die Prozeßabrechnung auszuschalten und einen Datensatz "grund" an `/var/adm/wtmp` anzufügen.

`startup` kann aufgerufen werden, um die Prozeßabrechnungsfunktion zu aktivieren, wenn das System auf Mehrbenutzerbetrieb umgeschaltet wird.

`turnacct` ist eine Schnittstelle zu `accton` (siehe `acct(1M)`) und dient zum Ein-(On) bzw. Ausschalten (Off) der Prozeßabrechnung. Mit dem Argument `switch` wird der aktuellen `/var/adm/pacct`-Datei der nächste freie Name in `/var/adm/pacctinkr` zugewiesen, wobei *inkr* eine Zahl ist; die Zählung beginnt bei

acctsh (1M)**acctsh (1M)**

1 und erhöht sich mit jeder weiteren `pacct`-Datei um 1, anschließend wird die Abrechnung wieder aktiviert. Diese Prozedur wird von `ckpacct` aufgerufen und kann daher von dem Dämon `cron` gesteuert werden. Die Datei `pacct` wird dadurch auf eine sinnvolle Größe begrenzt. `shutacct` schaltet mit `turnacct` die Prozeßabrechnung aus. `startup` schaltet mit `turnacct` die Prozeßabrechnung ein.

DATEIEN

<code>/var/adm/fee</code>	Gesamtgebührendatei
<code>/var/adm/pacct</code>	Aktuelle Datei für Abrechnung pro Prozeß
<code>/var/adm/pacct.incr</code>	Wird verwendet, wenn <code>pacct</code> zu groß wird, und während der täglichen Abrechnungsprozedur
<code>/var/adm/wtmp</code>	An-/Abmelde-Übersicht
<code>/usr/lib/acct/ptelus.awk</code>	Enthält die Grenzwerte für außergewöhnliche Auslastung nach Benutzernummern
<code>/usr/lib/acct/ptecms.awk</code>	Enthält die Grenzwerte für außergewöhnliche Auslastung nach Kommandonamen
<code>/var/adm/acct/nite</code>	aktuelles Verzeichnis
<code>/usr/lib/acct</code>	Enthält alle in Kapitel 1M dieses Handbuchs aufgeführten Abrechnungskommandos
<code>/var/adm/acct/sum</code>	Übersichtsverzeichnis, enthält Daten für <code>monacct</code>
<code>var/adm/acct/fiscal</code>	Verzeichnis für Finanzberichte

SIEHE AUCH

`acct(1M)`, `acctcms(1M)`, `acctcon(1M)`, `acctmerg(1M)`, `acctprc(1M)`, `cron(1M)`, `diskusg(1M)`, `fwtmp(1M)`, `runacct(1M)`, `acct(4)`, `utmp(4)`
`acctcom(1)` in den *Kommandos*
`acct(2)` im *Referenzhandbuch für Programmierer*

BEZEICHNUNG

addbad – Behandlung von fehlerhaften Blöcken

ÜBERSICHT

/sbin/addbad *rgerät* [-m] [*block#* ...]

BESCHREIBUNG

Mit addbad können Sie als Systemverwalter Listen von fehlerhaften Plattenblöcken ausgeben. Dazu gehören sowohl Blöcke, die bereits vom Hersteller für unbrauchbar erklärt wurden, als auch später dazugekommene fehlerhafte Blöcke. Sie können sich die fehlerhaften Blöcke folgender Plattentypen auflisten lassen:

- SCSI-Platten am SCSI Peripheral Adapter (SPA),
- SCSI-Platten am NCR Host Adapter (ADP32 Controller),
- ESDI-Platten am Interphase Storer.

Weiter können Sie mit addbad einem fehlerhaften Block einen Ersatzblock zuweisen. Jeder Controller verwendet dabei eine andere Vorgehensweise.

Am Interphase Storer enthält jede Spur einen zusätzlichen, unbenutzten Block, der als Ersatz für einen fehlerhaften dienen kann. Die Plattenperformance bleibt in diesem Fall unverändert. Sind mehrere fehlerhafte Blöcke zu ersetzen, dann wird der Spur mit den fehlerhaften Blöcken eine Ersatzspur zugeordnet.

Bei SCSI-Platten ist das Ersetzen von fehlerhaften Blöcken unter dem SCSI Interface verborgen.

Im allgemeinen versucht addbad, Daten zu sichern, die sich auf fehlerhaften Blöcken befinden, indem es diese mehrmals liest und die Daten in Ersatzblöcke schreibt. Sie ersetzen deshalb sinnvollerweise fehlerhafte Blöcke, wenn aus der Protokolldatei ersichtlich ist, daß mehrmals versucht wurde, auf einen bestimmten Block zuzugreifen.

Schlimmstenfalls gelingt es nicht, die Daten zu sichern, und ein Dateiverzeichnis oder das gesamte Dateisystem gehen dabei verloren. Auf jeden Fall sollten Sie eine Konsistenzprüfung des Dateisystems durchführen (z.B. mit `fsck -y /dev/rdisk/c?d?s?`), wenn Sie fehlerhafte Blöcke ersetzt haben.

addbad sollte nur im Einbenutzer-Modus aufgerufen werden.

Bei addbad stehen folgende Argumente und Parameter zur Verfügung:

rgerät Name der Gerätedatei (Raw-Modus), die zu der Platte gehört, deren fehlerhafte Blöcke aufgelistet oder ersetzt werden sollen. `/dev/rdisk/c?d?s0` bezeichnet z.B. eine SCSI-Platte am SCSI Peripheral Adapter oder eine ESDI-Platte am Interphase Storer. `/dev/rdisk/shd?t?s0` steht für eine SCSI-Platte am NCR Host Adapter. addbad ohne Parameter listet die fehlerhaften Blöcke der angegebenen Platte auf, und zwar nur solche Blöcke, die nach dem Einbau der Platte unbrauchbar wurden.

addbad (1M)

(MX500)

addbad (1M)

-m addbad listet die fehlerhaften Blöcke der angegebenen Platte auf, und zwar nur solche Blöcke, die bereits der Hersteller für unbrauchbar erklärt hat.

block# Geben Sie beim Aufruf von addbad eine Liste von logischen Blocknummern an, die fehlerhafte Blöcke bezeichnen, dann werden allen betroffenen Blöcken Ersatzblöcke zugewiesen.

BEISPIEL

Der *ssd*-Treiber einer SCSI-Platte, die am SCSI Peripheral Adapter angeschlossen ist, entdeckt einen fehlerhaften Block und gibt eine Fehlermeldung aus:

```
ssd(5,3) - cmd read: block 11097 - hardware error
```

5 bezeichnet den Kanal, 3 die Targetnummer (target number) des Laufwerks und 11097 die logische Blocknummer eines unbrauchbaren Blocks. Diesem soll ein Ersatzblock zugewiesen werden. Der Aufruf von addbad lautet dann:

```
addbad /dev/rdisk/c5d3s0 11097
```

DATEIEN

```
/dev/rdisk/c?d?s0
```

```
/dev/rdisk/shd?t?s0
```

SIEHE AUCH

```
fsck(1M), mkpart(1M)
```

HINWEIS

Auf der MX300 und auf der WX200 ist statt addbad *mkpart*(1M) zu verwenden.

BEZEICHNUNG

arp – Anzeige und Steuerung der Adreßauflösung

ÜBERSICHT

arp *hostname*

arp -a [*unix* [*kspeich*]]

arp -d *hostname*

arp -s *hostname ether_adresse* [*temp*] [*pub*] [*trail*]

arp -f *dateiname*

BESCHREIBUNG

Mit dem Programm arp lassen sich die Internet-zu-Ethernet-Adreßumsetzungstabellen, die vom Adreßauflösungsprotokoll (arp(7)) verwendet werden, anzeigen und ändern.

Ohne Optionen zeigt das Programm den aktuellen ARP-Eintrag für *hostname* an. Der Host kann mit Namen oder Nummer in der Internet-Punktschreibweise angegeben werden.

Folgende Optionen stehen zur Verfügung:

- a Anzeigen aller aktuellen ARP-Einträge. Dazu wird die Tabelle aus der Datei *kspeich* (Standard */dev/kmem*) gelesen, die auf der Systemkerndatei *unix* (Standard */stand/unix*) basiert.
- d Löschen eines Eintrags für den *hostnamen*. Diese Option darf nur vom Systemverwalter verwendet werden.
- s Erstellen eines ARP-Eintrags für den *hostnamen* mit der Ethernet-Adresse *ether_adresse*. Die Ethernet-Adresse wird in sechs hexadezimalen, durch Doppelpunkt getrennten Byte angegeben. Der Eintrag ist permanent, sofern im Kommando nicht das Wort *temp* angegeben wird. Wenn das Wort *pub* angegeben wird, wird der Eintrag öffentlich. Dieses System reagiert dann beispielsweise auf ARP-Anforderungen für *hostname*, auch wenn es sich nicht um den eigenen Hostnamen handelt. Das Wort *trail* gibt an, daß IP-Trailer-Einbettungen (IP-Trailer-Encapsulations) an diesen lokalen Rechner gesendet werden können.
- f Lesen der Datei *dateiname* und Einstellen mehrerer Einträge in ARP-Tabellen. Einträge in die Datei müssen die Form

hostname ether_adresse [*temp*] [*pub*] [*trail*]

 haben; die Argumente entsprechen den obigen Erläuterungen.

SIEHE AUCH

ifconfig(1M), arp(7).

BEZEICHNUNG

automount – automatisches Einhängen von NFS-Dateisystemen

ÜBERSICHT

```
automount [-nTv] [-D name=wert] [-M einhängeverzeichnis]
          [-t unteroptionen] [verzeichnis abbildung [-einhängeoptionen] ]
```

BESCHREIBUNG

automount ist ein Dämon, der nach Bedarf ein NFS-Dateisystem automatisch und transparent einhängt. Er überwacht Zugriffsversuche auf Verzeichnisse, die mit einer automount-Abbildung verbunden sind, sowie alle Verzeichnisse oder Dateien, die darunter stehen. Soll auf eine Datei zugegriffen werden, hängt der Dämon das entsprechende NFS-Dateisystem ein. Sie können einem Verzeichnis mit einem Eintrag in einer direkten automount-Abbildung oder durch Angeben einer indirekten Abbildung in der Kommandozeile eine Abbildung zuweisen.

automount verwendet eine Abbildung zum Suchen eines geeigneten NFS-Servers, eines exportierten Dateisystems und der Einhängeoptionen. Dann hängt es das Dateisystem in eine temporäre Position ein und ersetzt den Dateisystemeintrag für das Verzeichnis oder Unterverzeichnis durch einen symbolischen Verweis auf die temporäre Position. Wird auf das Dateisystem nicht innerhalb eines bestimmten Zeitraums (Standardeinstellung 5 Minuten) zugegriffen, hängt der Dämon das Dateisystem aus und löscht den symbolischen Verweis. Existiert das angegebene Verzeichnis noch nicht, wird es von dem Dämon angelegt und beim Beenden des Dämons wieder gelöscht.

Da die Bindung Name-zu-Position dynamisch ist, sind Aktualisierungen einer automount-Abbildung für den Benutzer transparent. Dadurch erübrigt es sich, gemeinsam benutzte Dateisysteme für Anwendungen mit fest codierten Verweisen auf Dateien vorab einzuhängen.

Wird das Pseudo-Dateiverzeichnis /- angegeben, behandelt automount das nachfolgende Argument *abbildung* wie den Namen einer direkten Abbildung. In einer direkten Abbildung verbindet jeder Eintrag den vollständigen Pfadnamen eines Einhängpunkts mit einem einzuhängenden fernen Dateisystem.

Ist das Argument *verzeichnis* ein Pfadname, zeigt das Argument *abbildung* auf eine als indirekte Abbildung bezeichnete Datei. Eine indirekte Abbildung enthält eine Liste der Unterverzeichnisse des angegebenen *verzeichnisses*. Bei einer indirekten Abbildung werden gerade diese Unterverzeichnisse automatisch eingehängt. Das Argument *abbildung* muß einen vollständigen Pfadnamen angeben.

Das Argument *-einhängeoptionen* enthält eine Liste durch Kommas getrennter und mit einem Bindestrich (-) eingeleiteter Optionen zu mount(1M). Sind jedoch in der angegebenen Abbildung ebenfalls Optionen angegeben, haben diese Priorität.

Folgende Optionen stehen zur Verfügung:

- n Deaktivieren des dynamischen Einhängens. Bei dieser Option werden Verweise durch den Dämon `automount` nur dann erfolgreich aufgelöst, wenn das Zielsystem bereits vorher eingehängt wurde. Mit dieser Funktion kann man verhindern, daß sich NFS-Server gegenseitig einhängen.
- T Überwachung. Zeigt jeden NFS-Aufruf in ausführlicher Form auf der Standardausgabe an.
- v Ausführlich. Protokollierung von Statusmeldungen auf der Konsole.
- D *name=wert*
Zuweisung von *wert* zur angegebenen `automount`-(Umgebungs-)Variablen.
- M *einhängerverzeichnis*
Einhängen temporärer Dateisysteme in das benannte Verzeichnis anstelle von `/tmp_mnt.`
- t *unteroptionen*
unteroptionen wird als eine durch Kommas getrennte Liste mit beliebigen Kombinationen der folgenden Werte spezifiziert:
 - l *dauer*
Hier wird die *dauer* in Sekunden angegeben, während der ein nicht gebrauchtes Dateisystem eingehängt bleibt. Der Standardwert ist 5 Minuten.
 - m *pause*
Damit wird angegeben, wie lange die *pause* (in Sekunden) zwischen zwei Versuchen, ein Dateisystem einzuhängen, sein soll. Der Standardwert ist 30 Sekunden.
 - w *pause*
Hier wird angegeben, wie lange die *pause* (in Sekunden) zwischen zwei Versuchen sein soll, Dateisysteme auszuhängen, deren Cache-Zeit überschritten ist. Der Standardwert ist 1 Minute.

UMGEBUNG

Umgebungsvariablen können innerhalb einer `automount`-Abbildung verwendet werden. Erschienen beispielsweise `$HOME` in einer Abbildung, würde es von `automount` auf den aktuellen Wert für die Variable `HOME` erweitert.

Dürfen an einen Verweis (reference) keine Zeichen angehängt werden, muß der Variablenname in geschweifte Klammern gesetzt werden.

VERWENDUNG

Eingabeformat für direkte/indirekte Abbildungen

Ein einfacher Abbildungseintrag hat diese Form:

verzeichnis [*-einhängeoptionen*] *position* ...

dabei ist *verzeichnis* der vollständige Pfadname des einzuhängenden Verzeichnisses, wenn er in einer direkten Abbildung verwendet werden soll, oder der Basisname eines Unterverzeichnisses in einer indirekten Abbildung. *einhängeoptionen* ist eine Liste durch Kommas getrennter mount-Optionen, und *position* gibt ein fernes Dateisystem an, von dem aus das Dateiverzeichnis geladen werden kann. Im einfachsten Fall hat *position* folgende Form:

host : *pfadname*

Es können mehrere Felder für *position* angegeben werden; in diesem Falle sendet automount mehrere mount-Anforderungen. automount hängt das Dateisystem von dem ersten Host ein, der auf die mount-Anforderung reagiert. Diese Anforderung wird zunächst an das lokale Netz oder Teilnetz gerichtet. Wird sie nicht beantwortet, darf jeder angeschlossene Server antworten.

Ist *position* in der Form:

host : *pfad* : *unterverzeichnis*

angegeben, ist *host* der Name des Host, von dem das Dateisystem eingehängt werden soll, *pfad* der Pfadname des einzuhängenden Verzeichnisses und *unterverzeichnis*, sofern angegeben, der Name eines Unterverzeichnisses, auf den der symbolische Verweis (link) zeigt. Damit kann doppeltes Einhängen verhindert werden, wenn auf mehrere Verzeichnisse in demselben fernen Dateisystem zugegriffen werden darf. Lautet die Abbildung für /home zum Beispiel:

```
able homeboy : /home/homeboy : able
baker homeboy : /home/homeboy : baker
```

und versucht ein Benutzer, auf eine Datei in /home/able zuzugreifen, hängt automount homeboy : /home/homeboy ein, erzeugt jedoch einen symbolischen Verweis namens /home/able auf das Unterverzeichnis able in dem vorübergehend eingehängten Dateisystem. Versucht ein Benutzer unmittelbar danach, auf eine Datei in /home/baker zuzugreifen, braucht automount nur einen symbolischen Verweis auf das Unterverzeichnis baker anzulegen; /home/homeboy ist bereits eingehängt. Bei der folgenden Abbildung:

```
able homeboy : /home/homeboy / able
baker homeboy : /home/homeboy / baker
```

müßte automount das Dateisystem zweimal einhängen.

Ein Abbildungseintrag kann sich über mehrere Eingabezeilen erstrecken, wobei die Bedeutung des Zeilenende-Zeichens durch einen Gegenschrägstrich aufgehoben wird. Kommentare beginnen mit dem Zeichen # und enden beim nächsten Zeilenende-Zeichen.

Mustervergleich bei Verzeichnissen

Das Zeichen & wird für den Eintrag, in dem es steht, auf den Wert des *Verzeichnisfeldes* erweitert. Im Falle

```
able homeboy:/home/homeboy:&
```

wird das Zeichen & auf *able* erweitert.

Das Zeichen * als *Verzeichnisfeld* dient dazu, alle Einträge aufzufangen, für die bis dahin noch keine Entsprechung gefunden wurde. Erschiene die folgende Angabe in der indirekten Abbildung für /home:

```
* &:/home/&
```

dürfte in /home jedes ferne Dateisystem automatisch eingehängt werden, dessen Position sich folgendermaßen angeben ließe:

```
hostname:/home/hostname
```

Hierarchische Abbildungseinträge

Ein hierarchischer Abbildungseintrag hat die Form:

```
verzeichnis [ / [unterverzeichnis] ] [-einhängeoptionen] position...
[ / [unterverzeichnis] [-einhängeoptionen] position...]...
```

Das erste */[unterverzeichnis]* ist für die erste Positionsliste optional und für alle folgenden Listen obligatorisch. Das optionale *unterverzeichnis* wird gegenüber dem *verzeichnis* als Dateiname aufgefaßt. Wird *unterverzeichnis* bei der ersten Angabe ausgelassen, bezieht sich / auf das Dateiverzeichnis selbst.

Beim folgenden direkten Abbildungseintrag:

```
/arch/src \
/ -ro,intr arch:/arch/src alt:/arch/src \
/1.0 -ro,intr alt:/arch/src/1.0 arch:/arch/src/1.0 \
/1.0/man -ro,intr arch:/arch/src/1.0/man alt:/arch/src/1.0/man
```

hängt automount automatisch je nach Bedarf /arch/src, /arch/src/1.0 und /arch/src/1.0/man entweder aus arch oder alt ein, und zwar je nachdem, welcher Host zuerst antwortet.

Direkte Abbildungen

Eine direkte Abbildung enthält Abbildungseinträge für eine beliebige Zahl von Verzeichnissen. Jedes in der Abbildung aufgeführte Verzeichnis wird bei Bedarf automatisch eingehängt. Die direkte Abbildung insgesamt ist nicht einem einzelnen Verzeichnis zugeordnet.

Indirekte Abbildungen

Mit einer indirekten Abbildung können Abbildungseinträge für die Unterverzeichnisse eingegeben werden, die unter dem in der Kommandozeile angegebenen *verzeichnis* eingehängt werden sollen. Auch überdeckt die indirekte Abbildung lokale Unterverzeichnisse, für die kein Abbildungseintrag angegeben ist. In einer indirekten Abbildung besteht jedes *verzeichnis*-Feld aus dem Basisnamen eines Unterverzeichnisses, das bei Bedarf eingehängt werden soll.

automount (1M)

(NFS)

automount (1M)

Eingebundene Abbildungen

In eine Abbildung kann der Inhalt einer weiteren Abbildung durch einen Eintrag mit der folgenden Form eingebunden werden:

+abbildungsname

wobei *abbildungsname* ein Dateiname ist.

Sonderabbildungen

Die Abbildung `-null` ist derzeit die einzige verfügbare Sonderabbildung. Wird in der Kommandozeile die Abbildung `-null` angegeben, wird damit eine vorherige Abbildung für das angegebene Verzeichnis gelöscht.

DATEIEN

`/tmp_mnt`

Vaterverzeichnis für dynamisch eingehängte Dateisysteme

SIEHE AUCH

`df(1M)`, `mount(1M)`, `passwd(4)`.

HINWEIS

Wenn `automount` Signal Nr. 1 empfängt, liest es die Datei `/etc/mnttab` erneut, um seinen internen Datensatz für gegenwärtig eingehängte Dateisysteme zu aktualisieren. Wird ein mit `automount` eingehängtes Dateisystem mit einem `umount`-Kommando ausgehängt, muß dafür gesorgt werden, daß `automount` die Datei neu liest.

Die Dateinamenerweiterung der Shell gilt nicht für aktuell nicht eingehängte Objekte.

Da `automount` mit nur einem Pfad versehen ist, verzögert jede Anforderung, die durch einen langsamen oder nicht antwortenden NFS-Server verzögert wird, alle folgenden Anforderungen für automatisches Einhängen, bis sie abgeschlossen ist.

Für Programme, die `/etc/mnttab` lesen und dann auf Dateien zugreifen, die unter automatischen Einhängepunkten stehen, werden weitere Einträge in der Datei vorgenommen.

BEZEICHNUNG

autopush – Konfigurieren von Listen automatisch gepushter STREAMS Module

ÜBERSICHT

```
autopush -f datei
autopush -r -M klasse -m nummer
autopush -g -M klasse -m nummer
```

BESCHREIBUNG

Mit diesem Kommando läßt sich die Liste der Module konfigurieren, die beim Öffnen eines Geräts automatisch gepusht werden. Auch können damit frühere Einstellungen gelöscht oder Informationen zu einer Einstellung abgerufen werden.

Für autopush stehen die folgenden Optionen zur Verfügung:

- f Mit dieser Option kann die autopush-Konfiguration für alle Treiber nach den Daten in der angegebenen Datei definiert werden. Eine autopush-Datei besteht aus Zeilen zu mindestens vier Feldern, die durch ein Leerzeichen getrennt sind, wie nachfolgend dargestellt:
klasse _nummer _letzte_nummer _mod1 mod2 ... modn
 Die ersten drei Felder enthalten ganze Zahlen für die Geräteklassennummer, die Gerätenummer und die letzte Gerätenummer. Die weiteren Felder enthalten Modulbezeichnungen. Ist *nummer* gleich -1, werden alle Geräte eines mit *klasse* angegebenen Geräteklassentreibers konfiguriert, und der Wert für *letzte_nummer* wird ignoriert. Ist *letzte_nummer* gleich 0, wird nur ein einziges Gerät konfiguriert. Um mehrere Geräte für eine bestimmte Geräteklasse zu konfigurieren, muß *nummer* kleiner sein als *letzte_nummer*.
 Die letzten Felder einer Zeile in der autopush-Datei bilden die Liste der durch Leerzeichen getrennten Modulnamen. Es dürfen definitionsgemäß bis zu acht Module angegeben werden, die automatisch gepusht werden sollen. Die Module werden in der Reihenfolge gepusht, in der sie in der Liste angegeben sind. Kommentarzeilen beginnen mit dem Zeichen #.
- r Mit dieser Option wird die vorherige Konfiguration einer bestimmten Geräteklassen- und Gerätenummer, die mit den Optionen -M und -m angegeben wurden, gelöscht. Entsprechen die Werte für *klasse* und *nummer* dem Wert für einen Bereich von Geräten, und entspricht *nummer* der ersten Gerätenummer des Bereichs, wird die Konfiguration für den gesamten Bereich gelöscht.
- g Mit dieser Option wird die aktuelle Konfiguration für eine bestimmte Geräteklassen- und Gerätenummer angezeigt, die mit den Optionen -M bzw. -m angegeben werden. Auch wird die Gerätenummer des ersten Geräts zurückgegeben, wenn die Anforderung der Konfiguration eines Gerätebereichs entspricht (wie bei Option -f beschrieben).

DATEIEN

/etc/ap/chan.ap

SIEHE AUCH

streamio(7)
 in CES, Leitfaden und Werkzeuge für die Programmierung mit C.

backup (1M)

backup (1M)

BEZEICHNUNG

backup – Starten oder Steuern einer Systemsicherungssitzung

ÜBERSICHT

```
backup -i [-t tabelle] [-o name] [-m benutzer] [-ne] [-s | -v]
        [-c woche:tag | demand]
```

```
backup [-a] [-t tabelle] [-o name] [-m benutzer] [-ne] [-c woche:tag | demand]
```

```
backup -S | -R | -C [-u benutzer | -A | -j auftragsnummer]
```

BESCHREIBUNG

Ohne Optionen führt das Kommando `backup` alle für den aktuellen Tag und die aktuelle Woche des Sicherungsturnus im Sicherungsregister vorgesehenen Sicherungsoperationen aus. Diese Sicherungsoperationen gelten als eine Aufgabe und erhalten eine `backup`-Auftragsnummer, mit der sich die einzelnen Stadien der Sitzung steuern lassen. Während die Sicherungsoperationen ausgeführt werden, wird ihr Status überwacht. (Siehe `bkstatus(1M)`). Beim Abschluß der Sicherungsoperationen werden sie im Sicherungsprotokoll aufgezeichnet.

`backup` darf nur von einem Benutzer ausgeführt werden, der über Zugriffsrechte wie der Systemverwalter verfügt.

Zum Steuern eines Sicherungsauftrags gibt es drei Möglichkeiten. Er kann annulliert, unterbrochen oder nach der Unterbrechung wiederaufgenommen werden.

Modi für Bedienereingriffe

Bei Sicherungsoperationen muß bisweilen der Bediener bestimmte Aufgaben ausführen, z.B. Datenträger in Geräte einlegen oder die Richtigkeit der Datenträgerkennsätze bestätigen. `backup` bietet drei Modi für den Dialog mit dem Bediener.

Bei `backup` ohne Optionen wird angenommen, daß ein Bediener zur Verfügung steht, allerdings nicht an dem Terminal, an dem das Kommando `backup` abgesetzt wurde. Dieser Modus sendet Post an den Bediener. Darin werden das Gerät, das bedient werden muß, und der benötigte Datenträger angegeben. Der Bediener liest die Meldung, ruft das Kommando `bkoper` auf, entspricht den Eingabeaufforderungen, und die Sicherungsoperation wird fortgesetzt.

`backup -i` aktiviert den Dialogmodus, bei dem vorausgesetzt wird, daß sich ein Bediener an dem Terminal befindet, an dem das Kommando `backup` ausgegeben wurde. In diesem Modus wird `bkoper` automatisch an dem Terminal aufgerufen, an dem das Kommando `backup` eingegeben wurde. Der Bediener beantwortet die Eingabeaufforderungen nacheinander, wie sie erscheinen.

`backup -a` aktiviert den automatischen Modus, bei dem vorausgesetzt wird, daß kein Bediener vorhanden ist. In diesem Modus schlägt jede Sicherungsoperation fehl, die einen Bedienereingriff erfordert. Sicherungsvorgänge, bei denen der eingelegte Datenträger ausreicht, werden ausgeführt.

Registerprüfungen

Bevor die in einem Sicherungsregister verzeichneten Sicherungsoperationen durchgeführt werden können, müssen mehrere Backup-Service-Datenbanken konsistent sein. Ob dies der Fall ist, kann nur beim Initialisieren des Kommandos `backup` überprüft werden. Besteht eine Inkonsistenz, wird `backup` abgebrochen.

Wird `backup -ne` angegeben, werden nicht nur die vorgesehenen Sicherungsoperationen angezeigt, sondern auch diese Konsistenzprüfungen durchgeführt. Folgendes wird überprüft:

1. Bei der Sicherungsmethode muß es sich um eine Standardmethode oder um eine ausführbare Datei in `/bkup/method` handeln.
2. Alle Bedingungen für eine Eingabe sind im Register definiert. Zirkuläre Bedingungen (z.B. Eintrag `abc` hängt von Eintrag `def` ab; Eintrag `def` hängt von Eintrag `abc` ab) sind zulässig.
3. Die Gerätegruppe für ein Ziel muß in der Gerätegruppentabelle `/etc/dgroup.tab` definiert sein. (Siehe *Geräteverwaltung*).

Optionen

- a Aktiviert im automatischen Modus alle Sicherungsoperationen; eine Aufforderung zum Einlegen von Datenträgern an einen Bediener wird nicht ausgegeben.
- c *woche:tag* | *demand* Wählt aus dem Sicherungsregister nur die Sicherungsoperationen für die angegebene Woche und den angegebenen Tag des Sicherungsturnus aus, nicht die Operationen für den aktuellen Tag oder die laufende Woche des Turnus. Wird *demand* angegeben, werden nur die Sicherungsoperationen ausgewählt, die auf Anfrage ausgeführt werden.
- e Mit dieser Option wird angezeigt, wieviele Datenträger voraussichtlich für jede Sicherungsoperation benötigt werden.
- i Auswahl des Dialogbetriebs.
- j *auftragsnummer* Steuert nur den mit *auftragsnummer* gekennzeichneten Sicherungsauftrag. *auftragsnummer* ist eine `backup`-Auftragsnummer.
- m *benutzer* Sendet Post an den genannten *benutzer*, wenn alle Sicherungsoperationen für den Sicherungsauftrag abgeschlossen sind.
- n Zeigt die auszuführenden Sicherungsoperationen an, führt sie aber nicht aus. Die Anzeige ist nach den im Sicherungsregister angegebenen Bedingungen und Prioritäten geordnet.
- o *name* Aktiviert Sicherungsoperationen nur für das genannte Ursprungsobjekt. Das Element *name* hat folgende Form:
uname | *ugerät*
- s Zeigt nach der Übertragung von 100 Blöcken (zu 512 Byte) zum Zielgerät einen Punkt an. Damit kann der Verlauf der Sicherungsoperation verfolgt werden.
- t *tabelle* Startet Sicherungsoperationen, die nicht im Standardregister `etc/bkup/bkreg.tab`, sondern in dem angegebenen Register definiert sind. *tabelle* ist ein Sicherungsregister.

backup (1M)

backup (1M)

-u *benutzer*

Steuert die von dem genannten *benutzer* gestarteten Sicherungsaufträge und nicht die Aufträge, die der Benutzer aktivierte, von dem das Kommando stammte. *benutzer* ist eine gültige Benutzernummer.

-v

Während der Sicherungsoperationen wird der Name jeder einzelnen Datei oder jedes Verzeichnisses angezeigt, sobald die Übertragung zum Zielgerät abgeschlossen ist.

-A

Steuert Sicherungsaufträge für alle Benutzer, nicht nur solche, die vom Benutzer gestartet wurden, der das Kommando aufgerufen hat.

-C

Storniert Sicherungsaufträge.

-R

Nimmt unterbrochene Sicherungsaufträge wieder auf.

-S

Unterbricht Sicherungsaufträge.

FEHLERMELDUNGEN

Die Exit-Codes von `backup` haben folgende Bedeutung:

0 Aufgabe erfolgreich abgeschlossen.

1 Ein oder mehrere Parameter zu `backup` sind ungültig.

2 Es ist ein Fehler aufgetreten; deshalb konnte `backup` nicht *alle* Teile dieses Auftrags ausführen.

BEISPIELE

Beispiel 1:

```
backup -i -v -c 2:1 -m admin3
```

aktiviert die Sicherungen, die für den Montag der zweiten Woche des Sicherungszeitraums vorgesehen sind, anstelle der Sicherungen für den aktuellen Tag und die laufende Woche. Führt die Sicherung im Dialogmodus durch und gibt auf der Standardausgabe die Namen aller Dateien, Verzeichnisse, Dateisystem-Partitionen oder Datenpartitionen aus, sobald die Übertragung zum Zielgerät beendet ist. Wenn alle Sicherungen abgeschlossen wurden, wird eine Benachrichtigung an den Benutzer mit der Benutzerkennung `admin3` gesendet.

Beispiel 2:

```
backup -o/usr:/dev/rdisk/0s2:usr
```

aktiviert nur die Sicherungen für das Dateisystem *usr*, das auf dem Ursprungsgerät `/dev/rdisk/0s2` eingehängt ist und den Kennsatz `usr` trägt.

Beispiel 3:

```
backup -S
```

Unterbricht die vom aufrufenden Benutzer angeforderten Sicherungsaufträge.

Beispiel 4:

```
backup -R -j back-359
```

nimmt die Sicherungsoperationen mit der Auftragsnummer back-359 wieder auf.

DATEIEN

```
/etc/bkup/method/*  
/etc/bkup/bkreg.tab  
/etc/device.tab  
/etc/dgroup.tab
```

SIEHE AUCH

bkhistory(1M), bkoper(1M), bkreg(1M), bkstatus(1M)

biod (1M)

(NFS)

biod (1M)

BEZEICHNUNG

biod – NFS-Dämon

ÜBERSICHT

biod[*server*]

BESCHREIBUNG

biod startet Dämonen für die asynchrone blockorientierte Ein-/Ausgabe. Dieses Kommando wird auf einem NFS-Client als Lese-/Schreibpuffer (read-ahead, write-behind) verwendet. Die übliche Anzahl für *server* ist vier.

Die biod-Dämonen werden automatisch aktiviert, wenn das System in Betriebsstufe 3 arbeitet.

SIEHE AUCH

mountd(1M), nfsd(1M), sharetab(4).

BEZEICHNUNG

`bkexcept` – Ändern oder Anzeigen einer Ausnahmeliste für Sicherungen

ÜBERSICHT

```
bkexcept [-t datei] [-d muster]
bkexcept [-t datei] -a|-r muster
bkexcept -C [datei...]
```

BESCHREIBUNG

Mit dem Kommando `bkexcept` wird eine Liste von Mustern angezeigt. Diese Muster dienen der Bezeichnung von Dateien, die bei Sicherungsoperationen mit `incfile` oder `ffile` nicht mitgesichert werden. Diese Liste wird als "Ausnahmeliste" bezeichnet.

`bkexcept` darf nur von Benutzern ausgeführt werden, die über Zugriffsrechte wie der Systemverwalter verfügen.

Mit `bkexcept -a` werden weitere Muster in die Liste aufgenommen.

`bkexcept -d` dient zum Anzeigen von Mustern aus der Liste.

Mit `bkexcept -r` werden Muster aus der Liste gelöscht.

Muster

Muster stehen für einzelne Pfadnamen oder Gruppen von Pfadnamen. Diese Muster müssen den unter DEFINITIONEN in `intro(2)` genannten Konventionen für die Angabe von Pfadnamen entsprechen. Ein Muster wird als Dateiname verstanden und entsprechend `cpio` interpretiert. Ein Muster kann die Sonderzeichen der Shell, `*`, `?` und `[]` enthalten. Stern (`*`) und Fragezeichen (`?`) entsprechen Punkt (`.`) und Schrägstrich (`/`). Da es sich hierbei um Sonderzeichen der Shell handelt, müssen sie in der Kommandozeile außer Kraft gesetzt werden.

Einträge der Ausnahmeliste lassen sich auf drei Arten angeben:

- Um alle Dateien unter einem bestimmten Verzeichnis anzugeben, wird der Name des Verzeichnisses (und aller gewünschten Unterverzeichnisse) und danach ein Stern eingegeben:

```
/verzeichnis/unterverzeichnis/ *
```

- Um alle Dateien mit demselben Namen unabhängig vom Verzeichnis zu erfassen, wird zuerst ein Stern und dann der Dateiname eingegeben:

```
*/dateiname
```

- Um ein bestimmtes Exemplar einer bestimmten Datei zu erfassen, wird der gesamte Pfadname der Datei eingegeben:

```
/verzeichnis/unterverzeichnis/dateiname
```

Besteht *muster* aus einem Bindestrich (`-`), wird in der Standardeingabe nach einer Liste von Mustern gesucht (eins pro Zeile bis EOF), die hinzugefügt oder gelöscht werden sollen.

Kompatibilität

Bei früheren Versionen des Backup-Service wurden Ausnahmelisten mit der `ed`-Syntax erstellt. `bkexcept -C` enthält eine Umwandlungsfunktion für Ausnahmelisten, die mit `ed` erstellt wurden. Diese Umwandlung ist unvollständig; nicht für alle `ed`-Muster gibt es in `cpio` Entsprechungen. Bei Mustern, für die keine direkte Entsprechung vorhanden ist, wird eine Umwandlung versucht, und das Ergebnis wird mit der Marke `QUESTIONABLE` (fraglich) gekennzeichnet. Die umgewandelte Version der Ausnahmeliste wird in die Standardausgabe geschrieben. Nun muß die Standardausgabe in eine Umwandlungsdatei umgeleitet werden, deren Inhalt wird überprüft (dabei werden fehlerhaft umgewandelte Einträge korrigiert und die `QUESTIONABLE`-Marken entfernt), und das Ergebnis wird als Eingabe für ein nachfolgendes Kommando `bkexcept -a` verwendet. Wurde die umgewandelte Datei z.B. `pruefdatei` genannt, erschiene die Option `-a` so:

```
bkexcept -a - < pruefdatei
```

Optionen

`-t datei`

Die Datei wird anstelle der Standarddatei angegeben.

`-a muster...`

Fügt *muster* der Ausnahmeliste hinzu; *muster* besteht aus einem oder mehreren Mustern, die durch Kommas oder Leerzeichen getrennt und in Anführungszeichen eingeschlossen sind, und Gruppen von Pfaden angeben.

`-d muster...`

Zeigt Einträge in der Ausnahmeliste an. Beginnt *muster* mit einem Schrägstrich (/), werden mit `-d` alle Einträge angezeigt, deren Namen mit *muster* beginnen. Beginnt *muster* nicht mit einem Schrägstrich, zeigt `-d` alle Einträge an, in denen *muster* an irgendeiner Stelle vorkommt. Ist *muster* ein Bindestrich (-), wird die Eingabe aus der Standardeingabe gelesen. *muster* ist selbst kein Muster - es vergleicht Muster. *muster* `a*b` entspricht `/a*b`, aber nicht `/adb`. Bei Dateien, die ein Wagenrücklaufzeichen enthalten, wird eine leere Ausnahmeliste zurückgegeben. Bei Dateien mit Länge null (keine Zeichen) wird ein Fehler zurückgegeben (Durchsuchung der Tabelle fehlgeschlagen).

Die Einträge werden in ASCII-Sortierfolge (Sonderzeichen, Zahlen, dann alphabetisch) angezeigt.

`-r muster...`

Löscht *muster* aus der Ausnahmeliste. *muster* ist ein Muster oder eine Liste von Mustern (durch Komma oder Leerzeichen getrennt und in Anführungszeichen eingeschlossen), die Gruppen von Pfaden angeben. *muster* muß einem Eintrag in der Ausnahmeliste genau entsprechen, damit *muster* entfernt wird. Muster, die entfernt werden, werden über die Standardausgabe angezeigt.

`-C [datei...]`

Gibt die Umwandlung jeder *datei* (Ausnahmeliste aus einer früheren Version) in die neue Syntax auf der Standardausgabe aus. Jede *datei* enthält pro Zeile ein `ed`-Muster.

Wird *datei* nicht angegeben, wird die standardmäßige SINIX-Ausnahmeliste `/etc/save.d/except` umgewandelt. Wird für *datei* ein Bindestrich (-) angegeben, wird die Eingabe zeilenweise aus der Standardeingabe entnommen.

FEHLERMELDUNGEN

Die Exit-Codes von `bkexcept` haben folgende Bedeutung:

- 0 = Aufgabe erfolgreich beendet.
- 1 = Ein oder mehrere Parameter zu `bkexcept` sind ungültig.
- 2 = Es ist ein Fehler aufgetreten. Dadurch konnte `bkexcept` nicht alle Teile der Aufgabe ausführen.

BEISPIELE

Beispiel 1:

```
bkexcept -a /tmp/*,/var/tmp/*,/usr/rje/*,*/trash
```

Fügt die vier Dateigruppen (alle Dateien unter `/tmp`, alle Dateien unter `/var/tmp`, alle Dateien unter `/usr/rje` und alle Dateien namens `trash` im System) in die Ausnahmeliste ein.

Beispiel 2:

```
bkexcept -d /tmp
```

Zeigt von den Mustern, die in Beispiel 1 in die Ausnahmeliste eingetragen wurden, folgende Muster an.

```
/tmp/*
```

```
bkexcept -d tmp
```

Zeigt von den Mustern, die in Beispiel 1 in die Ausnahmeliste eingetragen wurden, folgende Muster an.

```
/tmp/*, /var/tmp/*
```

Zeilenweise Anzeige mit Kopfzeile.

Beispiel 3:

```
bkexcept -r /var/tmp/*,/usr/rje/*
```

Löscht die beiden Muster aus der Ausnahmeliste.

Beispiel 4:

```
bkexcept -C /save.d/old.except > trans.except
```

Wandelt die Datei `/save.d/old.except` aus dem `ed`-Format in das `cpio`-Format um und sendet die Ergebnisse an die Datei `trans.except`. Die Umwandlungen von `/save.d/old.except` können mit `bkexcept -a` folgendermaßen in der aktuellen Ausnahmeliste hinzugefügt werden

```
bkexcept -a - < trans.except
```

bkexcept(1M)

bkexcept(1M)

DATEIEN

/etc/bkup/bkexcept.tab
/etc/save.d/except

SIEHE AUCH

backup(1M), incfile(1M)
cpio(1), ed(1), sh(1) in den *Kommandos*
intro(2) im *Referenzhandbuch für Programmierer*
Das Kapitel "Sichern und Wiederherstellen von Daten" im *Leitfaden für System-*
verwalter.

BEZEICHNUNG

bkhistory – Bericht über abgeschlossene Sicherungsoperationen

ÜBERSICHT

bkhistory [-hl] [-f *feldtrennzeichen*] [-d *daten*] [-o *namen*] [-t *marken*]

bkhistory -p *zeitraum*

BESCHREIBUNG

Mit **bkhistory** ohne Optionen wird eine Übersicht über den Inhalt des Sicherungsprotokolls *bkhist.tab* angegeben. Die Sicherungsoperationen sind alphabetisch nach Marken sortiert. Für die einzelnen Marken sind die Operationen chronologisch von der letzten bis zur am weitesten zurückliegenden geordnet. **backup(1M)** aktualisiert dieses Protokoll nach jeder erfolgreichen Sicherungsoperation.

bkhistory darf nur von Benutzern verwendet werden, die über Zugriffsrechte wie der Systemverwalter verfügen.

Mit **bkhistory** -p wird ein Berichtszeitraum *zeitraum* (in Wochen) für das Sicherungsprotokoll festgelegt; alle Einträge, die vor der angegebenen Woche erstellt wurden, werden aus dem Protokoll gelöscht. Der Standardwert hierfür ist eine (1) Woche.

Optionen

-d *daten*

Begrenzt den Bericht auf Sicherungsoperationen, die an den angegebenen Daten durchgeführt wurden. Die *daten* stehen im *date* Format. *tag*, *stunde*, *minute* und *jahr* sind wahlfrei und werden ignoriert. Die *daten* in der Liste sind durch Kommas oder Leerzeichen getrennt und in Anführungszeichen eingeschlossen.

-f *feldtrennzeichen*

Unterdrückt den Feldumbruch in der Anzeige und gibt das Trennzeichen an, das für die Ausgabefelder verwendet werden soll. Der Wert von *feldtrennzeichen* ist das Zeichen, das in der Anzeige als Feldtrennzeichen erscheint. Aus Gründen der Klarheit empfiehlt es sich, als Trennzeichen kein Zeichen zu verwenden, das wahrscheinlich auch in einem Feld vorkommt. So sollte beispielsweise der Doppelpunkt nicht als Feldtrennzeichen verwendet werden, wenn die Anzeige Daten enthält, in denen Stunden und Minuten durch Doppelpunkt getrennt werden. Um das Standard-Feldtrennzeichen (Tabulatorzeichen) zu verwenden, geben Sie für *feldtrennzeichen* das Nullzeichen (""") ein.

-h Unterdrückt die Ausgabe von Kopfzeilen in den Berichten.

-l Zeigt eine ausführliche Form des Berichts an. Damit wird eine **ls -l**-Liste der Dateien im Sicherungsarchiv erstellt, sofern on-line-Inhaltsverzeichnisse der Sicherungen verfügbar sind.

-o *namen*

Beschränkt den Bericht auf die angegebenen Ursprungsobjekte (Dateisysteme oder Daten-Partitionen). *namen* ist eine Liste von Namen von Ursprungsobjekten *unamen* und/oder Ursprungsgeräten *ugeräte*. (Siehe **bkreg(1M)**).

bkhistory (1M)

bkhistory (1M)

Die Namen in der Liste sind durch Kommas oder Leerzeichen getrennt und in Anführungszeichen eingeschlossen.

-p *zeitraum*

Dient zum Festlegen der Dauer (in Wochen), für die Daten im Sicherungsprotokoll gespeichert werden. Der Mindestwert und gleichzeitig der Standardwert für *zeitraum* ist 1.

-t *marken*

Beschränkt den Bericht auf Sicherungskopien mit den angegebenen Marken *marken*. *marken* ist eine Liste der im Sicherungsregister angegebenen Marken. Die einzelnen *marken* sind durch Kommas oder Leerzeichen getrennt und in Anführungszeichen eingeschlossen.

FEHLERMELDUNGEN

Die Exit-Codes von **bkhistory** haben folgende Bedeutung:

- 0 Die Aufgabe wurde ordnungsgemäß ausgeführt.
- 1 Ein oder mehrere Parameter zu **bkhistory** sind ungültig.
- 2 Es ist ein Fehler aufgetreten. Daher konnte **bkhistory** nicht alle Teile der Aufgabe ausführen.

BEISPIELE

Beispiel 1:

```
bkhistory -p 3
```

Definiert als Berichtszeitraum für das Sicherungsprotokoll drei Wochen. Einträge, die älter sind als drei Wochen, werden aus dem Protokoll gelöscht.

Beispiel 2:

```
bkhistory -t SpoolDai,UsrDaily,TPubsWed
```

Zeigt einen Bericht über die abgeschlossenen Sicherungsoperationen für die drei genannten Marken an.

Beispiel 3:

```
bkhistory -l -o /usr
```

Zeigt eine **ls -l**-Liste der Dateien an, die aus **/usr**, dem Ursprungsobjekt, gesichert wurden, sofern ein Inhaltsverzeichnis vorhanden ist.

DATEIEN

/etc/bkup/bkhist.tab das Sicherungsprotokoll, das Daten über ordnungsgemäß ausgeführte Sicherungsoperationen enthält

/etc/bkup/bkreg.tab Beschreibung der vom Systemverwalter festgelegten Sicherungsprinzipien

/var/sadm/bkup/toc Liste der Verzeichnisse mit On-line-Inhaltsverzeichnissen

SIEHE AUCH

backup(1M), **bkreg(1M)**.
date(1), **ls(1)** in den *Kommandos*.

BEZEICHNUNG

`bkoper` – Dialog bei Sicherungsoperationen; Aufforderungen zum Einlegen von Datenträgern werden ausgegeben.

ÜBERSICHT

`bkoper [-u benutzer...]`

BESCHREIBUNG

Bei Sicherungsoperationen kann es vorkommen, daß ein Bediener Datenträger einlegen und die Richtigkeit der Datenträgerkennsätze bestätigen muß. Das Kommando `bkoper` bietet eine `mailx`-artige Schnittstelle für diese Interaktion mit dem Bediener. Zu Beginn wird eine Liste der Kopfsätze ausgegeben. Jeder Kopfsatz beschreibt eine Sicherungsoperation, die einen Eingriff des Bedieners erfordert, das betreffende Gerät, einschließlich des benötigten Datenträgertyps und des Kennsatzes für den einzulegenden Datenträger (siehe **BEISPIEL**). Das System gibt Eingabeaufforderungen aus, und der Bediener setzt Kommandos ab, um die Sicherungsoperation fortzusetzen. Wird die Eingabetaste betätigt, erscheint der aktuelle Kopfsatz. Wurde noch kein Kopfsatz bearbeitet, ist der aktuelle Kopfsatz der erste in der Liste. Wurde ein Kopfsatz ausgewählt und bearbeitet, ist der aktuelle Kopfsatz der nächste.

`bkoper` darf nur von Benutzern verwendet werden, die über Zugriffsrechte wie der Systemverwalter verfügen. Standardmäßig darf der Bediener nur mit den Sicherungsoperationen kommunizieren, die von derselben Benutzernummer gestartet wurden.

Wird die Option `-u benutzer...` angegeben, ist der Dialog nur mit Sicherungsoperationen möglich, die von dem/den angegebenen *benutzer(n)* aktiviert wurden.

Kommandos

`!shell-kommando`

Sub-Shell wird aufgerufen. Der Text nach dem Ausrufezeichen (!) wird an die `SINIX`-Shell (`sh`) gesendet und dort als Kommando interpretiert.

`=` Gibt die Nummer der aktuellen Sicherungsoperation aus.

`?` Gibt eine Zusammenfassung der Kommandos aus.

`[p|t] [n]` Die Optionen `p` und `t` erfüllen die gleiche Funktion. Sie beeinflussen die Sicherungsoperation, die mit dem *n*ten Kopfsatz beschrieben wird. *n* entspricht standardmäßig der Nummer des aktuellen Kopfsatzes.

`h` Gibt die Liste der Sicherungsoperationen aus.

`q` Dient zum Verlassen von `bkoper`.

FEHLERMELDUNGEN

Die Exit-Codes von `bkoper` haben folgende Bedeutung:

- 0 Ordnungsgemäßer Abschluß der Aufgabe.
- 1 Ein oder mehrere Parameter zu `bkoper` sind ungültig.
- 2 Es ist ein Fehler aufgetreten. Deshalb konnte `bkoper` nicht alle Teile der Aufgabe ausführen.

BEISPIEL

Ein Beispiel für einen Kopfsatz finden Sie unten. Die einzelnen Elemente dieses Kopfsatzes sind in der folgenden Reihenfolge angegeben: Kopfsatznummer, Auftragsnummer, Marke, Ursprungsgerät, Zielgruppe, Zielgerät, Kennsätze des Zieldatenträgers. (Eine Beschreibung der einzelnen Elemente finden Sie unter `bkreg(1M)`.) Nicht jeder Kopfsatz enthält Werte für alle Felder; ist in `/etc/bkup/bkreg.tab` keine Zielgruppe angegeben, erscheint auch im Kopfsatz kein Wert dafür.

```
1 back-111 usrsun /dev/dsk/c1d0s1 disk /dev/dsk/c2d1s9 usrsave
2 back-112 fs2daily /dev/dsk/c1d0s8 ctape /dev/ctape/c4d0s2 -
```

Die Kopfsätze für Sicherungsoperationen werden nach Eingang numeriert; der älteste Kopfsatz hat die niedrigste Nummer. Ist für das Zielgerät kein Datenträgerkennsatz vorhanden, wird in dem Kopfsatz ein Bindestrich angezeigt.

SIEHE AUCH

`bkreg(1M)`, `bkstatus(1M)`, `getvol(1M)`, `mailx(1)`.

BEZEICHNUNG

bkreg – Ändern oder Anzeigen des Inhalts eines Sicherungsregisters

ÜBERSICHT

```
bkreg -p zeitraum [-w awoche] [-t tabelle]

bkreg -a marke -o ursprg -c wochen:tage | demand -d zielger
      -m methode | migration [-b äoptionen] [-t tabelle] [-D vorher] [-P prio]

bkreg -e marke [-o ursprg] [-c wochen:tage | demand] [-m methode | migration]
      [-d zielger] [-t tabelle] [-b äoptionen] [-D vorher] [-P prio]

bkreg -r marke [-t tabelle]

bkreg [-A|-O|-R] [-hsv] [-t tabelle] [-c wochen[:tage] | demand]

bkreg -C felder [-hv] [-t tabelle] [-c wochen[:tage] | demand] [-f c]
```

BESCHREIBUNG

Ein Sicherungsregister ist eine Datei mit Beschreibungen von Sicherungsoperationen, die auf einem SINIX-System vorgenommen werden sollen. Das Standard-Sicherungsregister steht in `/etc/bkup/bkreg.tab`. Weitere Sicherungsregister können angelegt werden.

Das Kommando **bkreg** darf nur von einem Benutzer mit Systemverwalter-Zugriffsrechten benutzt werden.

Alle Einträge in ein Sicherungsregister beschreiben Sicherungsoperationen, die für ein bestimmtes Plattenobjekt (das sogenannte Ursprungsobjekt) für bestimmte Tage und Wochen während eines Turnus durchgeführt werden sollen. Das Register kann mehrere Einträge für ein Objekt enthalten, aber welche Sicherungsoperationen für ein Objekt an einem bestimmten Tag einer bestimmten Woche des Turnus ausgeführt werden sollen, darf nur in einem Eintrag definiert werden. Der Eintrag gibt das Objekt, die zu verwendende Sicherungsmethode zur Archivierung des Objekts sowie die Zieldatenträger zur Speicherung des Archivs an. Jeder Eintrag ist mit einer eindeutigen *marke* gekennzeichnet. Die Angabe *marke* muß den Dateinamen-Konventionen entsprechen.

Sicherungsturnus

Sicherungen werden in einem Sicherungsturnus durchgeführt, der in Wochen angegeben wird. Wenn ein Turnus beendet ist, beginnt der nächste. Der Turnus beginnt immer mit einem Sonntag. Die Standarddauer eines Turnus ist eine Woche.

Ursprungsobjekte

Ein Ursprungsobjekt ist entweder eine Partition für Daten im Raw-Modus oder ein Dateisystem. Es wird durch seinen Ursprungsobjektnamen, den Gerätenamen und wahlfreie Datenträgerkennsätze bezeichnet.

Wenn die entsprechenden Prioritäten und Abhängigkeiten angegeben werden, können auch mehrere Sicherungsoperationen für unterschiedliche Ursprungsobjekte gleichzeitig durchgeführt werden. Während einer Sicherungssitzung werden Operationen mit höherer Priorität früher ausgeführt als Operationen mit

niedrigerer Priorität. Alle Sicherungsoperationen mit einer bestimmten Priorität dürfen gleichzeitig ablaufen, solange keine Abhängigkeiten angegeben werden. Wird festgelegt, daß eine Sicherungsoperation von anderen abhängig ist, wird sie erst gestartet, wenn alle übergeordneten Operationen ordnungsgemäß abgeschlossen sind.

Zielgeräte

Alle Sicherungsarchive werden auf einem Satz von Datenträgern gespeichert, die in ein Zielgerät eingelegt werden. Ein Zielgerät kann durch eine Zielgerätegruppe, einen Zielgerätenamen, Datenträgermerkmale und Datenträgerkennsätze gekennzeichnet sein. Die Standardmerkmale für einen Datenträger (die in der Gerätetabelle aufgelistet sind) können geändert werden (siehe hierzu das Kapitel "Verwalten von Speichergeräten" im *Leitfaden für Systemverwalter*).

Sicherungsmethoden

Ein Ursprungsobjekt wird nach einer bestimmten Methode in ein Zielgerätearchiv gesichert. Die Methode bestimmt, wieviele Daten gesichert und wie sie dargestellt werden. Für ein bestimmtes Ursprungsobjekt können an verschiedenen Tagen des Turnus unterschiedliche Methoden verwendet werden. Für jede Methode stehen spezifische Optionen zur Verfügung.

Mit dem Backup-Service werden einige Standardmethoden bereitgestellt. Weitere Methoden können über eine SINIX-Installation hinzugefügt werden. Beschreibungen der Standard-Sicherungsmethoden finden Sie unter `incfile(1M)`, `ffile(1M)`, `fdisk(1M)`, `finage(1M)` und `fdp(1M)`.

Wird als Sicherungsmethode `migration` angegeben, kann ein Archiv auch zu einem anderen Ziel umgelagert werden. Der Gerätenamen des Ursprungsobjekts für die Migration muß als Zielgerät bei einer früheren, erfolgreich abgeschlossenen Sicherungsoperation fungiert haben. Diese Form der Sicherung archiviert das Ursprungsobjekt nicht neu. Dabei wird ein Archiv von einem Ziel zum anderen kopiert, und gleichzeitig werden die Datenbanken des Backup-Service aktualisiert, so daß die Daten weiterhin automatisch wiederhergestellt werden können.

Registerüberprüfung

Es gibt Elemente, die innerhalb einzelner Registereinträge und in mehreren Registereinträgen konsistent sein müssen, damit der Backup-Service eine Sicherungssitzung ordnungsgemäß durchführen kann. Dazu stehen Konsistenzprüfungen zur Verfügung. Einige davon werden durchgeführt, während das Sicherungsregister angelegt oder geändert wird. Andere lassen sich nur überprüfen, wenn das Kommando `backup(1M)` mit dem Sicherungsregister arbeitet. Eine vollständige Liste der Prüfungen findet sich unter `backup(1M)`.

Modi

Das Kommando `bkgreg` kann in zwei Modi arbeiten: den Inhalt eines Sicherungsregisters ändern und den Inhalt eines Sicherungsregisters anzeigen.

Ändern

- bkreg -p Dient zum Ändern des Sicherungsturnus für ein Sicherungsregister. Der Standardturnus ist eine Woche.
- bkreg -a Stellt einen Eintrag in das Sicherungsregister. Zusammen mit dieser Option müssen weitere Optionen angegeben werden. Diese sind nachstehend im Abschnitt **Optionen** aufgeführt.
- bkreg -e Dient zum Ändern eines bestehenden Eintrags in einem Sicherungsregister.
- bkreg -r Dient zum Löschen eines Eintrags aus dem Sicherungsregister.

Anzeigen

- bkreg -C Erstellt eine angepasste Anzeige eines Registerinhalts.
- bkreg [-A|-R|-O]
Erstellt eine Übersicht über einen Registerinhalt.

Optionen

- a Fügt einen neuen Eintrag in das Standard-Sicherungsregister ein. Folgende Optionen müssen zusammen mit -a angegeben werden: *marke*, *ursprg*, *wochen:tage*, *zielger*, *methode*. Wird nichts anderes angegeben, gelten folgende Standardwerte: Verwendet wird das Standard-Sicherungsregister, Optionen zu Methoden entfallen, die Priorität ist 0, zwischen den Einträgen bestehen keine Abhängigkeiten.

-b *äoptionen*

Jede Sicherungsmethode unterstützt eigene Optionen, die ihre Arbeitsweise modifizieren. *äoptionen* erscheint als Liste von Optionen, die durch Leerzeichen getrennt und in Anführungszeichen eingeschlossen sind. Die hier eingegebene Argumentenfolge wird unverändert und genau wie eingegeben an die Methode übergeben. Listen gültiger Optionen finden Sie in dem Kapitel "Sichern und Wiederherstellen von Daten" im *Leitfaden für Systemverwalter* und unter den folgenden Einträgen im vorliegenden Handbuch: *fdisk(1M)*, *fdp(1M)*, *ffile(1M)*, *fimage(1M)* und *incfile(1M)*.

-c *wochen:tage* | *demand*

Definiert die Woche(n) und den Tag (die Tage) des Sicherungsturnus, in/an denen eine Sicherungsoperation ausgeführt oder eine Anzeige erstellt werden soll.

wochen ist eine Zahlenreihe von 1 bis 52. Der Wert von *wochen* darf nicht höher sein als der Wert von *-pzeitraum*. *wochen* wird als Kombination von Einzelwochen und Zeiträumen angegeben (die durch Kommas oder Leerzeichen getrennt und in Anführungszeichen eingeschlossen sind). Ein Beispiel ist

```
' '1 3-10,13''
```

Hier werden die erste Woche, die dritte bis einschließlich zehnte und die dreizehnte Woche des Sicherungsturnus angegeben.

tage ist entweder eine Reihe von Zahlen zwischen 0 (Sonntag) und 6 (Samstag) oder eine Reihe von Abkürzungen von s (Sonntag) bis sa (Samstag). Darüber hinaus wird *tage* als Kombination von Einzeltagen und Zeiträumen angegeben (die durch Kommas oder Leerzeichen getrennt und in Anführungszeichen eingeschlossen sind).

demand gibt an, daß ein Eintrag nur benutzt wird, wenn er von

```
backup -c demand
```

ausdrücklich angefordert wird.

-d *zielger*

Gibt *zielger* als Zielgerät für die Sicherungsoperation an. *zielger* hat die Form:

```
[ggruppe] [: [zielgerät] [:zchar] [:zname] ]
```

Dabei muß entweder *ggruppe* oder *zielgerät* angegeben werden, und *zchar* und *zname* sind wahlfrei. *ggruppe* und *zielgerät* können zusammen angegeben werden. Die Doppelpunkte markieren die Feldgrenzen und müssen angegeben werden wie oben gezeigt.

ggruppe ist die Gerätegruppe für das Zielgerät. (Siehe *devgroup.tab(4)*.) Fehlt sie, muß *zielgerät* angegeben werden.

zielgerät ist der Gerätenamen eines bestimmten Zielgeräts. (Siehe *device.tab(4)*.) Fehlt er, muß *ggruppe* angegeben werden, und jedes verfügbare Gerät der *ggruppe* kann benutzt werden.

zchar beschreibt Gerätemerkmale. Sind solche Charakteristika angegeben, setzen sie die Standardwerte für Gerät und Gruppe außer Kraft. *zchar* hat die Form:

```
schlüsselwort=wert
```

Dabei ist *schlüsselwort* ein gültiges Schlüsselwort für Gerätemerkmale (wie in der Gerätetabelle angegeben.) *zchar*-Einträge können durch Kommas oder Leerzeichen getrennt werden. Sind sie durch Leerzeichen getrennt, muß die gesamte Argumentenkette zu *zielger* in Anführungszeichen stehen.

zname ist eine Liste von Datenträgerkennzeichen der Zieldatenträger. Die Elemente der Liste von *zname* müssen durch Kommas oder Leerzeichen getrennt sein. Sind sie durch Leerzeichen getrennt, muß das gesamte Argument *zielger* in Anführungszeichen stehen. Jedes *zname* entspricht einem *datenträgernamen*, wie im Kommando *labelit* angegeben. Wird *zname* nicht angegeben, überprüfen *backup* und *restore* die Datenträgerkennsätze dieses Eintrags nicht.

- e Dient zum Bearbeiten eines bestehenden Eintrags. Ist eine der Optionen -b, -c, -d, -m, -o, -D oder -P vorhanden, ersetzen sie die aktuellen Werte für den genannten Eintrag im Register.

- f *c* Setzt das Standard-Trennzeichen für Ausgabefelder außer Kraft. Das Zeichen *c* erscheint in diesem Beispiel als neues Feldtrennzeichen in der Bildschirmausgabe. Das Standard-Trennzeichen für Ausgabefelder ist der Doppelpunkt (:).
- h Unterdrückt Kopfsätze beim Erstellen von Anzeigen.
- m *methode* | *migration*
Führt die Sicherungsoperation nach der angegebenen *methode* aus. Die Standardmethoden sind: *incfile*, *ffile*, *fdisk*, *fimage* und *fdp*. Handelt es sich bei der angegebenen Methode nicht um eine Standardmethode, muß sie als ausführbare Datei in dem Verzeichnis für Standardmethoden */etc/bkup/method* stehen. *migration* gibt an, daß der Wert von *ursprg* (nach der Option -o) dem Wert von *zielger* während einer früheren Sicherungsoperation entspricht. Das Ursprungsobjekt wird nicht neu archiviert; es wird nur an die mit *zielger* (nach der Option -d) angegebene Position kopiert. Das Sicherungsprotokoll (sofern vorhanden) und die Inhaltsverzeichnisse (sofern vorhanden) werden aktualisiert, so daß sie das neue Ziel des Ursprungsarchivs enthalten.
- o *ursprg*
Gibt mit *ursprg* das Ursprungsobjekt für die Sicherungsoperation an. *ursprg* wird im folgenden Format dargestellt:

uname:*ugerät*[:*ukenn*]

Dabei ist *uname* der Name eines Ursprungsobjekts. Bei Dateisystem-Partitionen ist dies der Name des Knotens, an dem das Dateisystem normalerweise eingehängt ist (*mount*). Bei Daten-Partitionen kann jeder gültige Pfadname verwendet werden. Dieser Wert wird an die Sicherungsmethode übergeben und von *backup* überprüft. Die Standard-Sicherungsmethoden für Daten-Partitionen, *fdp* und *fdisk*, überprüfen diesen Namen nicht.

ugerät ist der Gerätenamen für das Ursprungsobjekt. In allen Fällen ist dies der Gerätenamen einer Partition im Raw-Modus. Für SINIX-Rechner wird dieser Name im folgenden Format angegeben: */dev/rdisk/c?d?s?*.

ukenn ist der Datenträgerkennsatz für das Ursprungsobjekt. Bei Dateisystem-Partitionen entspricht er dem *datenträgernamen*, der mit dem Kommando *labelit* angezeigt wird. Eine Daten-Partition kann einen zugehörigen Datenträgernamen haben, der nur außen auf dem Datenträger erscheint, auf dem sie gespeichert ist; mit *getvol* kann veranlaßt werden, daß ein Bediener den Namen überprüft.

Bei SINIX-Rechnern bezeichnet die Partition */dev/rdisk/c?d?s6* eine ganze Festplatte. Sie wird verwendet, wenn die Festplatte formatiert oder neu partitioniert werden soll, um das Inhaltsverzeichnis der Festplatte zu bezeichnen (VTOC). (Siehe *prtvtoctoc(1M)*.) *backup* überprüft diese spezielle Partition anhand des Datenträgernamens der Festplatte, der beim Partitionieren der Festplatte angegeben wurde. Wird der Datenträgername der Festplatte nicht angegeben, überprüft *backup* die Datenträgerkennsätze für dieses Ursprungsobjekt nicht.

- p *zeitraum*
Setzt den Sicherungsturnus (in Wochen) für das Sicherungsregister auf *zeitraum*. Der Mindestwert ist 1; der Höchstwert ist 52. Die laufende Woche des Sicherungsturnus hat standardmäßig den Wert 1.
- r Entfernt die angegebenen Einträge aus dem Register.
- s Unterdrückt den Umbruch beim Erstellen von Anzeigen. Normalerweise werden lange Werte innerhalb eines Feldes umbrochen.
- t *tabelle*
Verwendet *tabelle* anstelle des Standardregisters *bkreg.tab*.
- v Erstellt Anzeigen mit (senkrechten) Spalten anstelle (waagrechter) Zeilen. Dadurch können mehr Daten angezeigt werden, ohne daß die mit dem Anzeigen langer Zeilen verbundenen Probleme auftreten.
- w *awoche*
Setzt das Standardverhalten außer Kraft und definiert die laufende Woche des Sicherungsturnus als *awoche*. *awoche* ist eine Ganzzahl zwischen 1 und dem Wert von *zeitraum*. Der Standardwert ist 1.
- A Zeigt einen Bericht an, in dem alle Felder des Registers aufgeführt sind. Die mit dieser Option erstellte Anzeige eignet sich besonders als Eingabe in einen Filter, da im Horizontalmodus extrem lange Zeilen erzeugt werden.
- C *felder*
Gibt den Inhalt eines Sicherungsregisters in einer Anzeige aus. Diese ist auf die angegebenen Felder beschränkt. Die Ausgabe besteht aus je einem Satz Zeilen pro Registereintrag. Die einzelnen Zeilen enthalten die gewünschten Felder, die durch ein Feldtrennzeichen getrennt sind. *felder* ist eine Liste der durch Kommas oder Leerzeichen getrennten und in Anführungszeichen eingeschlossenen Namen der gewünschten Felder. Die gültigen Feldnamen lauten *period* (*zeitraum*), *cweek* (*awoche*), *tag* (*marke*), *oname* (*uname*), *odevice* (*ugerät*), *olabel* (*ukenn*), *weeks* (*wochen*), *days* (*tage*), *method* (*methode*), *moptions* (*äoptionen*), *prio*, *depend* (*vorher*), *dgroup* (*ggruppe*), *ddevice* (*zielgerät*), *dchar* (*zchar*) und *zname*.
- D *vorher*
Gibt eine Reihe von Sicherungsoperationen an, die erfolgreich abgeschlossen sein müssen, bevor diese Operation beginnen darf. *vorher* ist eine Liste von Marken, die durch Kommas oder Leerzeichen getrennt und in Anführungszeichen eingeschlossen sind und die vorausgehenden Sicherungsoperationen bezeichnen.
- f *c*
Setzt das Standard-Feldtrennzeichen außer Kraft. *c* ist das Zeichen, das in der Anzeige als neues Feldtrennzeichen erscheint. Das Standard-Feldtrennzeichen ist ein Doppelpunkt (":").
- O Zeigt eine Zusammenfassung aller Ursprungsobjekte mit Einträgen im Register an.

-P *prio*

Setzt die Priorität von *prio* für diese Sicherungsoperation. Der Standardwert ist 0; die höchste Priorität ist 100. Alle Sicherungsoperationen mit derselben Priorität dürfen gleichzeitig ausgeführt werden, sofern die Priorität nicht 0 ist. Alle Sicherungsoperationen mit Priorität 0 laufen nacheinander ohne eine bestimmte Reihenfolge ab.

-R Zeigt eine Übersicht über alle Zielgeräte mit Einträgen im Register an.

FEHLERMELDUNGEN

Die Exit-Codes von **bkreg** haben folgende Bedeutung:

- 0 Die Aufgabe wurde ordnungsgemäß abgeschlossen.
- 1 Einer oder mehrere Parameter zu **bkreg** sind ungültig.
- 2 Es ist ein Fehler aufgetreten. Daher konnte **bkreg** nicht alle Teile der Aufgabe ausführen.

Fehler erscheinen in der Standardfehlerausgabe, wenn eine der folgenden Bedingungen erfüllt wird:

- 1. Die in **bkreg -e** oder **bkreg -r** angegebene *marke* existiert im Sicherungsregister nicht.
- 2. Die in **bkreg -a** angegebene Marke besteht im Register bereits.

BEISPIELE

Beispiel 1:

```
bkreg -p 15 -w 3
```

Definiert für das Standard-Sicherungsregister einen Turnus von 15 Wochen und die laufende Woche als dritte Woche dieses Turnus.

Beispiel 2:

```
bkreg -a acct5 -t wklybu.tab \
-o /usr:/dev/rdisk/c1d0s2:usr -c "2 4-6 8 10:0,2,5" \
-m incfile -b -txE \
-d diskette:capacity=1404:acctwkly1,acctwkly2,acctwkly3
```

Fügt den Eintrag *acct5* in das Sicherungsregister *wklybu.tab* ein. Existiert *wklybu.tab* noch nicht, wird es nun angelegt. Das zu sichernde Ursprungsobjekt ist das Dateisystem */usr* auf dem Gerät */dev/rdisk/c1d0s2*, das als *usr* bezeichnet wird. Gesichert wird immer am Sonntag, Dienstag und Freitag der zweiten, vierten bis sechsten, achten und zehnten Woche des Sicherungsturnus mit der *incfile*-Methode (Teilsicherung). Die Optionen für die Sicherungsmethoden bestimmen, daß ein Inhaltsverzeichnis auf einem Zusatzmedium und nicht im Sicherungsprotokoll erstellt, die Ausnahmeliste ignoriert und eine Schätzung der benötigten Datenträger für das Archiv vor der Sicherungsoperation ausgegeben wird. Gesichert wird auf dem zuerst verfügbaren Diskettengerät auf die drei Disketten *acctwkly1*, *acctwkly2* und *acctwkly3*. Diese Datenträger haben eine Kapazität von je 1404 Blöcken.

Beispiel 3:

```
bkreg -e services2 -t wklybu.tab \
-o /back:/dev/rdisk/c1d0s8:back -m migration \
-c demand -d ctape:/dev/rdisk/c4d0s3
```

Ändert die Angaben für die Sicherungsoperation `services2` in der Sicherungstabelle `wklybu.tab`. Jedesmal wenn das Kommando `backup -c demand` ausgeführt wird, erfolgt nun eine Migration der auf dem Zielgerät `back:/dev/rdisk/c1d0s8:back` (das nun das Ursprungsgerät ist) erstellten Sicherungen auf einer Magnetbandkassette.

Beispiel 4:

```
bkreg -e pubsfri -P 10 -D develfri,marketfri,acctfri
```

Ändert die Priorität für die Sicherungsoperation `pubsfri` auf 10 und macht diese Sicherungsoperation abhängig von den drei Sicherungsoperationen `develfri`, `marketfri` und `acctfri`. Die Operation `pubsfri` wird nur ausgeführt, wenn alle Sicherungsoperationen mit einer höheren Priorität als 10 begonnen haben und die Operationen `develfri`, `marketfri` und `acctfri` erfolgreich abgeschlossen wurden.

Beispiel 5:

```
bkreg -c 1-8:0-6
```

Liefert die Standardanzeige des Inhalts des Standard-Sicherungsregisters für alle Wochentage für die erste bis achte Woche des Sicherungsturnus. Die Daten im Register werden in der folgenden Form angezeigt:

```
Rotation Period = 10      Current Week = 4
(Sicherungsurnus)      (Aktuelle Woche)
```

```
Originating Device: / /dev/root
(Ursprungsgerät)
```

Tag (Marke)	Weeks (Wochen)	Days (Tage)	Method (Methode)	Options (Optionen)	Pri (Priorität)	Dgroup (Ggruppe)
rootdai	1-8	1-6	incfile			diskette
rootsp	1-8	0	ffile	-bxt	20	ctape

```
Originating Device: /usr /dev/dsk/c1d0s2
(Ursprungsgerät)
```

Tag (Marke)	Weeks (Wochen)	Days (Tage)	Method (Methode)	Options (Optionen)	Pri (Priorität)	Dgroup (Ggruppe)
usrdai	1-8	1-5	incfile			diskette
usrsp	1-8	0	ffile	-bxt	15	ctape

bkreg (1M)

bkreg (1M)

DATEIEN

/etc/bkup/method/*	
/etc/bkup/bkreg.tab	enthält Angaben zu dem vom Systemverwalter definierten Sicherungssystem
/etc/dgroup.tab	enthält eine Liste der vom Systemverwalter festgelegten logischen Gerätegruppen
/etc/device.tab	enthält Angaben zu bestimmten Geräten und ihren Attributen

SIEHE AUCH

backup(1M), fdisk(1M), fdp(1M), incfile(1M), ffile(1M), fimage(1M),
getvol(1M), labelit(1M), mkfs(1M), mount(1M), prtvtoc(1M), restore(1M)

BEZEICHNUNG

`bkstatus` – Status von Sicherungsoperationen anzeigen

ÜBERSICHT

`bkstatus` `[-h] [-f feldtrennzeichen] [-j auftragsnummern] [-s status | -a]
[-u benutzer]`

`bkstatus` `-p zeitraum`

BESCHREIBUNG

Gibt man das Kommando `bkstatus` ohne Optionen ein, wird der Status der Sicherungsoperationen angezeigt, die gerade ablaufen. Als Status wird *active* (aktiv), *pending* (anstehend), *waiting* (im Wartestatus) oder *suspended* (unterbrochen) angegeben. Wird das Kommando mit der Option `-a` verwendet, dann werden auch fehlgeschlagene (*failed*) und abgeschlossene (*completed*) Sicherungsoperationen mit angezeigt.

`bkstatus -p` bestimmt, wie lange Statusinformationen für die Anzeige gesichert werden.

`bkstatus` kann nur von Benutzern verwendet werden, die über Zugriffsrechte wie der Systemverwalter verfügen.

Jede Sicherungsoperation durchläuft verschiedene Stadien, die im folgenden näher erläutert werden. Die Buchstaben, die nach jedem Status in Klammern aufgelistet sind, werden mit der Option `-s` verwendet und erscheinen auch auf der Anzeige.

pending(p) - anstehend

`backup` wurde aufgerufen, und die Operationen im Sicherungsregister für den angegebenen Tag sind für den Ablauf vorbereitet.

active(a) - aktiv

Der Sicherungsoperation wurde ein Zielgerät zugeordnet, und die Archivierung läuft gerade. Oder eine unterbrochene (*suspended*) Sicherungsoperation wurde wieder aufgenommen.

waiting(w) - im Wartestatus

Die Sicherungsoperation wartet auf einen Eingriff des Bedieners; zum Beispiel, daß der richtige Datenträger eingelegt wird.

suspended(s) - unterbrochen

Die Sicherungsoperation wurde durch den Aufruf von `backup -S` unterbrochen.

failed(f) - fehlgeschlagen

Die Sicherungsoperation ist fehlgeschlagen oder wurde abgebrochen.

completed(c) - abgeschlossen

Die Sicherungsoperation wurde erfolgreich abgeschlossen.

Die Optionen `-a` und `-s` schließen sich gegenseitig aus.

Optionen

- a Fehlgeschlagene (*failed*) und abgeschlossene (*completed*) Sicherungsoperationen werden in die Anzeige mit eingeschlossen. Alle Sicherungsoperationen, die innerhalb des Turnus auftraten, werden angezeigt.
- f *feldtrennzeichen* Unterdrückt in der Anzeige den Feldumbruch und gibt an, daß ein Feldtrennzeichen benutzt wird. Der Wert von *feldtrennzeichen* ist das Zeichen, das als Feldtrennzeichen in der angezeigten Ausgabe erscheint. Damit die Ausgabe übersichtlich ist, sollte man als Trennzeichen ein Zeichen verwenden, das mit großer Wahrscheinlichkeit nicht in den Feldern selbst auftritt. Zum Beispiel sollte man keinen Doppelpunkt als Feldtrennzeichen verwenden, wenn in der Ausgabe Daten enthalten sind, bei denen zur Trennung von Stunden und Minuten ein Doppelpunkt verwendet wird. Um das Standardfeldtrennzeichen (Tabulatorzeichen) zu verwenden, gibt man das Nullzeichen (" ") als Wert für *feldtrennzeichen* an.
- h Unterdrückt den Kopfsatz auf der Ausgabe.
- j *auftragsnummern* Beschränkt die Anzeige auf die angegebene Liste von backup-Auftragsnummern, die entweder durch Komma oder durch Leerstellen getrennt und in Anführungszeichen eingeschlossen werden. (Siehe *backup(1M)*).
- p *zeitraum* Gibt an, wie lange Statusinformationen über Sicherungen gespeichert werden. Der *zeitraum* gibt an, wieviele Wochen diese Informationen in der Datei */bkup/bkstatus.tab* gesichert werden. Statusinformationen, die älter sind als *zeitraum* (in Wochen), werden aus der Statusabelle gelöscht. Der kleinste gültige Eintrag ist 1, der höchste 52. Als Standardwert gilt 1 Woche.
- s *status* Beschränkt die Liste der Sicherungsoperationen auf solche mit dem angegebenen *status*. *status* ist eine Liste aus Buchstaben, die den Status angeben (verkettet, durch Komma oder durch Leerstellen getrennt und in Anführungszeichen eingeschlossen). Zum Beispiel geben
 - apf
 - a,p,f
 - "a p f"
 an, daß die Liste nur Sicherungsoperationen einschließen soll, die *active* (aktiv), *pending* (anstehend) oder *failed* (fehlgeschlagen) sind.
- u *benutzer* Beschränkt die Anzeige auf Sicherungsoperationen, die von den in der Liste angegebenen *benutzern* gestartet wurden. Die Einträge für die Benutzer werden dabei entweder durch Komma oder durch Leerstellen getrennt und in Anführungszeichen eingeschlossen. *benutzer* muß in der Datei *passwd* enthalten sein.

FEHLERMELDUNGEN

Die Exit-Codes von `bkstatus` haben folgende Bedeutung:

- 0 = Aufgabe erfolgreich ausgeführt.
- 1 = Einer oder mehrere Parameter für `bkstatus` ungültig.
- 2 = Wegen eines Fehlers konnte `bkstatus` seine Aufgabe nicht *vollständig* ausführen.

BEISPIELE

Beispiel 1:

```
bkstatus -p 4
```

Gibt an, daß die Statusinformationen über Sicherungen vier Wochen lang gespeichert werden sollen. Alle älteren Statusinformationen werden im System gelöscht.

Beispiel 2:

```
bkstatus -a -j back-459,back-395
```

Erzeugt eine Ausgabe, die den Status der beiden angegebenen Sicherungsaufträge enthält, selbst wenn sie abgeschlossen (*completed*) oder fehlgeschlagen (*failed*) sind.

Beispiel 3:

```
bkstatus -s a,c -u "oper3 oper4"
```

Erzeugt eine Ausgabe, die nur die Sicherungsaufträge enthält, die von den Benutzern `oper3` und `oper4` gestartet wurden und den Status *active* (aktiv) oder *completed* (abgeschlossen) haben.

DATEIEN

- | | |
|-------------------------------------|---|
| <code>/etc/bkup/bkstatus.tab</code> | listet den aktuellen Status von Sicherungen auf, die ausgeführt wurden oder noch laufen |
| <code>/etc/bkup/bkreg.tab</code> | beschreibt das Sicherungssystem, für das sich der Systemverwalter entschieden hat |

SIEHE AUCH

`backup(1M)`, `bkhist(1M)`, `bkreg(1M)`

BEZEICHNUNG

boot – Ladeprogramm des SINIX-Systems

BESCHREIBUNG

Das Programm `boot` lädt interaktiv Programme, die eigenständig d.h. ohne Unterstützung durch ein Betriebssystem, auf der Hardware ablaufen können, und veranlaßt deren Ausführung. Typischerweise wird `boot` zum Laden und Ausführen des Systemkerns benutzt. Im Prinzip kann aber auch ein beliebiges anderes Programm geladen und ausgeführt werden, vorausgesetzt, es ist für eine eigenständige Ausführung gebunden worden (siehe `ld(1)`).

Eine weitere wichtige Funktion von `boot` ist das Kopieren des Hauptspeichers auf ein Massenspeicher-Medium, z.B. eine Festplatte oder ein Magnetband. Dieser Programmteil wird nur unter bestimmten Umständen (s.u.) aktiviert und im folgenden als "dump" bezeichnet.

`boot` selbst ist ein eigenständig ausführbares Programm und befindet sich auf dem Zylinder 0 der ersten Festplatte an einer vorgegebenen Adresse. Außerdem wird auf Diskette SINIX0 ein speziell auf die Installation von SINIX zugeschnittenes `boot` mitgeliefert. Das Programm `boot` wird durch die Firmware der Maschine geladen. Standardmäßig werden zuerst die Diskettenlaufwerke und dann die erste Festplatte überprüft, ob sie `boot` enthalten.

Ablauf von boot

Beim Starten des Systems ist im NVRAM (non-volatile RAM) die Information hinterlegt, ob die letzte Systembeendigung regulär oder irregulär war. Im letzteren Fall wird normalerweise ein Dump gezogen. Ist `AUTODUMP` (siehe "Boot-Optionen") auf YES gesetzt, geschieht das automatisch. Bei `AUTODUMP=NO` fordert `boot` eine interaktive Bestätigung an:

```
Do you want a memory dump [yn]?
```

Bei Eingabe RETURN wird der Standardwert "y" angenommen, ebenso, wenn die TIMEOUT-Zeit (siehe "Boot-Optionen"). abgelaufen ist. In diesem Fall wird `dump` aktiviert. `dump` stellt zunächst fest, ob an seiner vorgegebenen Zieladresse bereits ein älterer Speicherabzug existiert. Wenn ja, muß der neue Speicherabzug explizit bestätigt werden:

```
dump: Dump already in default slice,  
Do you want to overwrite it?  
yes: type 'y<ret>' for overwriting previous dump  
no : type 'n<ret>' for no dump
```

Nach der Eingabe von "y" wird der Speicherinhalt auf eine vorgegebene Adresse kopiert. Dabei wird in 0.25 MB-Intervallen angezeigt, wieviel schon kopiert wurde.

Nach Beendigung von `dump` wird abhängig von `AUTOBOOT` mit den Standardwerten automatisch geladen, oder aber die Eingabeaufforderung

Enter name of program to boot:

erscheint an der Konsole. Bei RETURN oder abgelaufener TIMEOUT-Zeit wird `DEFBOOTSTR` (siehe "Boot-Optionen"). als Eingabe angenommen. Wenn das Programm erfolgreich lokalisiert werden kann, wird es geladen. Währenddessen erscheinen auf der Konsole die Ausgaben

loading text segment...

loading data segment...

Zu diesem Zeitpunkt kann der Ladevorgang durch Drücken der DEL-Taste unterbrochen werden. Kann das Programm aus irgendwelchen Gründen beim ersten Versuch nicht erfolgreich geladen werden, wird unabhängig von `AUTOBOOT` eine Eingabe verlangt:

Enter name of program to boot:

Beim zweiten und allen folgenden Versuchen ist die TIMEOUT-Uhr nicht mehr aktiv, so daß auf jeden Fall eine Benutzereingabe erforderlich ist.

Boot-Optionen

`boot` wird durch eine Reihe von Optionen gesteuert. Standardwerte dafür stehen entweder in einer Datei, üblicherweise `/stand/boot`, oder werden von `boot` vorgelegt. Die Optionen werden unverändert an das Programm weitergereicht, sofern sie nicht durch die Benutzereingabe überschrieben werden (s.u.). Über die Optionen `AUTOBOOT`, `AUTODUMP` und insbesondere `TIMEOUT` kann der komplette Lade- und Dumpvorgang automatisiert werden.

`boot` kennt folgende Optionen:

<code>AUTOBOOT=YES</code> oder <code>NO</code>	Legt fest, ob <code>boot</code> den Systemkern sofort lädt (YES) oder erst eine Eingabeaufforderung an der Konsole ausgibt (NO). Standardwert: YES.
<code>AUTODUMP=YES</code> oder <code>NO</code>	Gibt an, ob ein Speicherabzug (Dump) bei jeder nicht regulären Systembeendigung (Reset, Spannungsausfall, Systemabsturz) automatisch zu ziehen ist (YES) oder nur nach vorheriger Bestätigung durch den Systemverwalter (NO). Standardwert: YES.
<code>BOOTPROMPT=string</code>	Ändert die standardmäßige Eingabeaufforderung des Ladeprogramms zu der Zeichenkette <i>string</i> . Standardwert: Enter name of program to boot:.
<code>DEFBOOTSTR=bootstring</code>	Der Name des zu ladenden Programms (und Parameter) werden auf <i>bootstring</i> voreingestellt. Dieser Bootstring wird verwendet, wenn Sie die Eingabeaufforderung von <code>boot</code> nur mit RETURN quittieren oder aber wenn die Wartezeit gemäß <code>TIMEOUT</code> abläuft. Standardwert: <code>hd(10,0)unix root=hd(1) swap=hd(2).</code>

MEMRANGE=*bereich[,bereich...]*

Teilt dem Programm `boot` diejenigen Speicherbereiche mit, deren Vorhandensein getestet werden soll. *bereich* hat das Format *start-ende*:0. *start* und *ende* sind Dezimaladressen, die durch einen Bindestrich getrennt sind (zum Beispiel 1M-4M, 1K-65536). Aufsteigende (z.B. 15M-16M) und absteigende (z.B. 16M-15M) Adreßbereiche werden unterstützt. Die Startadresse zählt mit zum Speicherbereich (inklusive), die Endadresse nicht (exklusive). Standardwert: 0-64M:0.

MEMREQ=*größe*

Weist `boot` an, eine Mindestspeichergöße von *größe* anzufordern. Findet `boot` nicht mindestens *größe* Byte Speicherplatz, gibt es eine Meldung aus und beendet sich. *größe* hat dasselbe Format wie die Adressen bei MEMRANGE. Standardwert: 0.

MREQMSG1=*zeichenkette*

Diese Option legt die Nachricht fest, die ausgegeben wird, wenn die in MEMREQ geforderte Speichergöße nicht verfügbar ist. Standardwert: Insufficient memory for a useful system.

MREQMSG2=*zeichenkette*

Gibt eine optionale zweite Zeile als Ergänzung zur Nachricht MREQMSG1 an. Standardwert: leer.

TIMEOUT=*anz*

Wenn TIMEOUT gesetzt ist, wartet `boot` genau *anz* Sekunden auf eine Benutzereingabe. Nach Ablauf dieser Zeit verwendet `boot` stattdessen den Bootstring, der DEFBOOTSTR zugewiesen ist. Ist *anz* gleich 0, dann ist die Uhr zum Warten nicht gesetzt. Standardwert: 0.

Eingaben für boot (Bootstring)

Die wichtigsten Eingaben für `boot` sind Adresse und Name des zu ladenden Programms sowie Parameter dafür. Adresse und Name werden im "Standalone-Format" angegeben. Die Syntax der Eingabe lautet wie folgt:

gtyp(nr[,dist])[name] [par1] [par2] ... [parn]

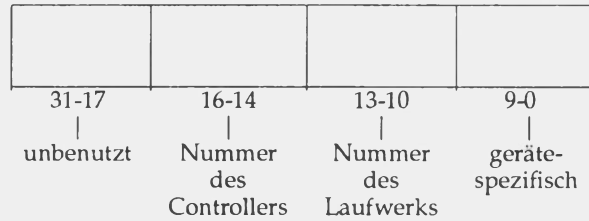
Dabei gilt:

- *gtyp* gibt an, von welchem Gerätetyp das Programm zu laden ist. Die unterstützten Gerätetypen sind *hd* (Festplatte) und *fd* (Diskette).
- *nr* bezeichnet die Gerätenummer (minor device number) des Geräts. Sie wählt ein bestimmtes Gerät des angegebenen Typs aus und bestimmt darüber hinaus eine Adresse auf diesem Gerät. Sie wird nach dem folgenden Schema angegeben:

boot(1M)

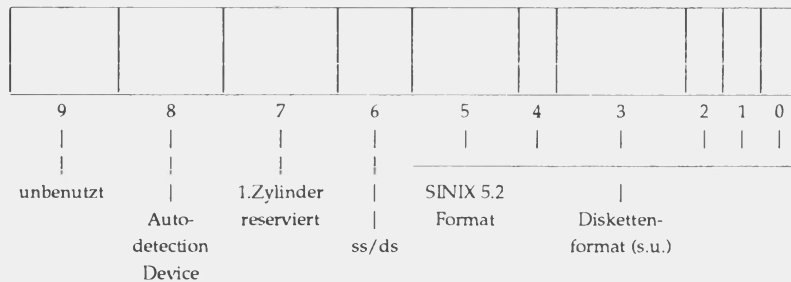
(MX300)

boot(1M)



Die Bits 9-0 geben im Fall einer Festplatte eine Slice-Nummer an. Die Numerierung von Controllern, Laufwerken und Slices beginnt immer bei 0. So bezeichnet z. B. `hd(1034)` die Slice Nr. 10 auf der zweiten Platte am ersten Controller.

Im Fall eines Diskettenlaufwerks sind die Bits 9-0 wie folgt zu interpretieren:



Bits 4-0 geben das Diskettenformat an. Folgende Disketten werden unterstützt:

Format-Nr	Bytes/Sektor	Spuren/Zoll	Sektoren/Spur	
0	512	96/135	15	ss/ds
1	512	48	9	ss/ds
2	512	48	8	ss/ds
3	1024	48	4	ss/ds
4	256	48	16	ss/ds
5	512	96	9	ss/ds
6	512	135	9	ss/ds
7	512	135	18	ss/ds
8	512	96	8	ss
32	256	96	16	SINIX 5.2 Format
33	512	96	15	SINIX 5.2 Format

- *dist* gibt eine Distanz in KB an, die zu der durch *nr* ermittelten Adresse addiert wird. Wenn *dist* weggelassen wird, wird der Wert 0 angenommen.

- *name* bezeichnet den Namen des zu ladenden Programms in dem Dateisystem, dessen Startadresse durch *nr* und *dist* ermittelt wird. Ist *name* angegeben, sollte *dist* 0 sein. Wenn kein Name angegeben ist, wird das Programm direkt von der durch *nr* und *dist* bezeichneten Geräteadresse geladen. Bezeichnet *name* ein Dateiverzeichnis, so werden die Namen der darin enthaltenen Dateien angezeigt. Das Root-Verzeichnis wird durch den Namen “.” angegeben.
- *pari* sind Parameter (s.u.) für das zu ladende Programm. Im Fall des SINIX-Kerns sind diese Parameter Zuweisungen der Form *vari=wert*. Für diese Zuweisungen kann es auch Einträge in der Standardwerte-Datei `/stand/boot` geben. Ist ein Parameter sowohl in der Standardwerte-Datei als auch in der Eingabezeile definiert, wird die Definition in der Eingabezeile benutzt.

Parameter für den SINIX-Kern

Im Bootstring können Sie folgende Parameter für den SINIX-Kern angeben:

<code>root=gtyp(nr)</code>	legt fest, wo das Root-Dateisystem liegt. <i>gtyp</i> und <i>nr</i> werden vom Kern interpretiert. Die Angaben haben die gleiche Bedeutung wie oben beschrieben. Im Regelfall gilt <code>root=hd(1)</code> für einen Kern auf der Festplatte (erste Platten-Partition) und <code>root=rand(0)</code> für einen Installationskern (Root-Dateisystem liegt in einem logisch als Platte verwalteten Hauptspeicherbereich).
<code>swap=gtyp(nr)</code>	legt den Swapbereich fest. Typisch sind <code>swap=hd(2)</code> für einen regulären Kern und <code>swap=rand(1)</code> für einen Installationskern. <i>gtyp</i> und <i>nr</i> werden wie beschrieben vom Kern interpretiert.
<code>rootfstype=fstype</code>	informiert, welcher Dateisystem-Typ als Root-Dateisystem zu verwenden ist. Erlaubt sind “s5” und “ufs”. <i>fstype</i> muß natürlich mit dem Typ des Dateisystems übereinstimmen, das als <i>root</i> angegeben ist.

DATEIEN

`/stand/boot` Datei mit Standardwerten zum Laden

SIEHE AUCH

`fdisk(1M)`, `init(1M)`, `fd(7)`, `hd(7)`.

BEZEICHNUNG

boot – Ladeprogramm des SINIX-Systems

BESCHREIBUNG

Das Programm `boot` lädt interaktiv den SINIX-Systemkern und führt ihn aus.

Ein Teil des Programms `boot` wird vom Einschaltmonitor (Power On Monitor) in den Hauptspeicher geladen und von einer CPU ausgeführt. Der Rest des Programms `boot` wird von einem der weiter unten genannten Geräte geladen.

Das Programm `boot` wird zwar hauptsächlich zum Laden und Ausführen des SINIX-Systemkerns benutzt, es kann jedoch auch jedes andere Programm laden und ausführen, das für die eigenständige Ausführung gebunden ist. Das Programm `boot` ist unverzichtbarer Bestandteil des SINIX-Betriebssystems und muß auf einer Platte oder dem Installationsband vorhanden sein, damit der SINIX-Systemkern erfolgreich geladen werden kann. Beachten Sie bitte, daß bei der Installation des SINIX-Betriebssystems eine Kopie des Programms `boot` auf die Systemplatte (die Bootplatte) geschrieben wird. Das Programm `boot` befindet sich auf Zylinder 0 der Festplatte und ist das Standardladeprogramm für Prozeduren, die von der Festplatte laden.

Ablauf von boot

Wenn der Rechner gestartet wird, wird das Programm `boot` ausgeführt. Die Firmware des Einschaltmonitors prüft zunächst die Ladezeichenkette (Bootstring) im NVRAM (non-volatile RAM). Sie hinterlegen den Bootstring dort mit dem Kommando `bootflags`, das Sie während einer vorangegangenen SINIX-Sitzung aufgerufen haben. Ist ein gültiger Bootstring im NVRAM vorhanden und ist `AUTO-BOOT` eingeschaltet, dann wird das System gemäß diesem Bootstring geladen.

Ist `AUTOBOOT` nicht eingeschaltet, dann schreibt der Einschaltmonitor einen Prompt auf die Konsole und erwartet, daß Sie das Kommando `b` und den Bootstring eingeben. `b sp(24,1)unix` ist z.B. eine korrekte Eingabe. Der Bootstring `sp(24,1)unix` muß genauso aufgebaut sein wie einer, der im NVRAM hinterlegt ist.

Das Laden ist von folgenden Geräten möglich:

Firmwarename	Kernelname	Anzahl	Beschreibung
ts	ct	2	SCSI-Bandlaufwerk am SCED Board
sp	ssd	24	SCSI-Platte am SCSI Peripheral Adapter (SPA)
is	hd	12	ESDI-Platte am Interphase Storerger am MBAD

Eingaben für boot (Bootstring)

Im Bootstring steht nach dem Kommandobuchstaben, wie `b` oder `b80`, zuerst der Firmwarename des Geräts, von dem geladen werden soll. An diesen Namen sind in runden Klammern zwei Zahlen angehängt, die je nach Gerät verschiedene Bedeutungen haben und nach unterschiedlichen Formeln berechnet werden. Folgen innerhalb der Ladezeichenkette weitere Geräteangaben (z.B. für den Root- oder Swapbereich), so ist bei diesen Angaben der Kernelname des Geräts zu verwenden (siehe "Parameter für den Kern"). Dem Kernelnamen folgen in Klammern ein oder

zwei Zahlen, die die Gerätenummer (minor device number) und - falls vorhanden - den Offset darstellen. Grund für die unterschiedlichen Gerätebezeichnungen ist, daß die erste Gerätebezeichnung innerhalb des Bootstrings von der Firmware des Einschaltmonitors ausgewertet wird, die weiteren Gerätebezeichnungen werden dagegen an den Kern übergeben und von diesem ausgewertet.

Laden von Geräten am SCED Board

Am SCED Board werden die Geräte entsprechend ihrer SCSI-Targetadresse (SCSI-Targetnummer, SCSI target number) ausgewählt. Die SCSI-Targetadressen bezeichnen Gerätedateien, die im Dateiverzeichnis `/etc/conf/sdevice.d/ct` (für `ts`-Bandlaufwerke) stehen. Standardmäßig ist die Targetadresse des ersten Bandlaufwerks 4, die des zweiten 5.

Beim Laden von Band beginnt die Ladezeichenkette mit `ts(erste_zahl,zweite_zahl)`. `erste_zahl` wird berechnet als

$$scsi_target_number \times 8$$

`zweite_zahl` bezeichnet die Dateinummer auf dem Installationsband. Auf dem Installationsband sind die Dateinummern folgendermaßen vergeben:

Datei	Inhalt
0	boot Programm
1	boot Kern
2	Inhaltsverzeichnis
3	root-Dateisystem, das auf die RAM-Platte geladen werden soll
4	Installationsarchiv des Grundsystems in <code>cpio</code> -Format
5	Installations-Paket in <code>pkadd</code> -Format

Laden von Geräten am SPA Board

Am SPA Board werden die Geräte ebenfalls entsprechend ihrer SCSI-Targetadresse ausgewählt. Die SCSI-Targetadressen bezeichnen Gerätedateien, die im Dateiverzeichnis `/etc/conf/sdevice.d/ssd` (für `ssd`-Platten) stehen. Bei Geräten, die am SPA angeschlossen sind, beginnt die Ladezeichenkette mit `sp(erste_zahl,zweite_zahl)`. `erste_zahl` wird berechnet als

$$\begin{aligned} & kanal \times 64 \\ & + scsi_target_number \times 8 \end{aligned}$$

`kanal` ist dabei einer von 8 möglichen Kanälen, die sich an den beiden SPA Boards befinden. `zweite_zahl` bezeichnet die Nummer der ausgewählten Slice (siehe `disk(7)`). Normalerweise bezeichnet 1 die `root`-Slice, 10 die `/stand`-Slice.

Sollen sich `root` oder `swap` nicht auf Standardgeräten befinden oder das root-Dateisystem vom Standardtyp abweichen, dann haben zusätzliche Angaben in der Ladezeichenkette die Form

```
root[dev]  = ssd(dritte_zahl[,offset])
swap[dev]  = ssd(dritte_zahl[,offset])
rootfs[type] = (ufs|s5)
```

dritte_zahl bezeichnet die Gerätenummer (minor device number), die nach der folgenden Formel berechnet wird:

$$\begin{aligned} & kanal \times 1024 \\ & + scsi_target_number \times 128 \\ & + slice \end{aligned}$$

Die Standardvorgaben für `root` usw. stehen in der Datei `/etc/conf/cf.d/sassign`.

Laden von Geräten am MBAD

Bei Geräten, die am MBAD angeschlossen sind, beginnt die Ladezeichenkette mit `is(erste_zahl,zweite_zahl)`. *erste_zahl* wird berechnet als

$$\begin{aligned} & mbad \times 512 \\ & + controller \times 8 \\ & + platte \end{aligned}$$

mbad hat den Wert 0 für den ersten, 1 für den zweiten oder 2 für den dritten MB-I Adapter. *controller* ist 0 für den ersten oder 1 für den zweiten Interphase Storage Controller an jedem MB-I Adapter. *platte* ist entweder 0 oder 1. 0 bezeichnet die erste Platte, 1 die zweite Platte, die seriell an jedem Controller angeschlossen ist. *zweite_zahl* bezeichnet die Nummer der ausgewählten Slice.

Sollen sich `root` oder `swap` nicht auf Standardgeräten befinden oder das root-Dateisystem vom Standardtyp abweichen, dann haben zusätzliche Angaben in der Ladezeichenkette die Form

```
root[dev]  = hd(dritte_zahl[,offset])
swap[dev]  = hd(dritte_zahl[,offset])
rootfs[type] = (ufs|s5)
```

dritte_zahl bezeichnet die Gerätenummer (minor device number), die nach folgender Formel berechnet wird:

$$\begin{aligned} & platte \times 1024 \\ & + controller \times 16384 \\ & + slice \end{aligned}$$

Die Standardvorgaben stehen in der Datei `/etc/conf/cf.d/sassign`.

Beispiele

Übliche Eingaben nach dem Prompt des Einschaltmonitors sind:

b ts(32,1) root=rand(0) swap=rand(1)

Diese Eingabe bewirkt: Laden des Installationskerns vom Installationsband. Die zusätzlichen Angaben zum root- und swap-Bereich müssen Sie machen, da sich diese nicht auf den Standardgeräten befinden. Die Standardvorgaben stehen in der Datei `/etc/conf/cf.d/sassign`. Dieser Bootstring ist ein Standardbeispiel zum Installieren des Kerns vom Band statt von der Platte.

b sp(0,10)unix root=ssd(1) swap=ssd(2) rootfs=ufs

Diese Eingabe bewirkt: Laden des Kerns `/unix` aus dem Dateisystem `/stand`, das sich auf einer SCSI-Platte am SPA Board befindet. Die Platte ist die erste, die über den ersten Kanal am ersten SPA Controller angeschlossen ist. Dieser Bootstring ist ein Standardbeispiel zum Installieren des Kerns von der Platte.

b80 sp(0,0) sp(0,6) [0 /dev/dump]

Diese Eingabe bewirkt: Laden eines eigenständigen Boot-Programms, ohne vorher den Speicher zu löschen. Ein Speicherabzug (core dump) wird nach Slice 6 (`/dev/dump`) geschrieben. Der Offset innerhalb der Slice beträgt 0. Die Platte ist die erste, die über den ersten Kanal am ersten SPA Controller angeschlossen ist.

b is(1,10)unix root=hd(1025) swap=hd(1026) rootfs=ufs

Diese Eingabe bewirkt: Laden des Kerns `/unix` aus dem Dateisystem `/stand`, das sich auf der zweiten ESDI-Platte befindet, die am ersten Controller am ersten MBAD angeschlossen ist.

Kommandosyntax

Das folgende Kommando können Sie an den Boot Command Line Interpreter (CLI) übergeben. Es steht am Anfang einer Eingabe, die Sie machen, wenn der Prompt des Einschaltmonitors am Bildschirm erscheint.

b[oot] [gerät] [datei]

Lädt die Datei *datei* von Gerät *gerät*.

gerät hat jeweils die Form *xx(erste_zahl, zweite_zahl)* mit

xx ist der Firmwarename eines Geräts, *ts*, *sp* oder *is*.

erste_zahl

ist die Nummer einer Targetadresse oder einer Einheit. Weiter oben ist beschrieben, wie sich diese Zahl berechnet.

zweite_zahl

bezeichnet eine Slice oder eine Datei. Weiter oben ist beschrieben, wie sich diese Zahl berechnet.

datei ist entweder ein Dateiname, oder der volle Pfadname einer Datei eines eingehängten Dateisystems oder eine Datei auf einem Gerät mit sequentiellem Zugriff (z.B. *ts(32,2)*).

Parameter für den Kern

Das Programm `boot` übergibt alle nach dem Prompt eingegebenen Zeichenketten an den Kern. Dieser liest sie, um festzustellen, welche Peripheriegeräte jeweils für `root`, `pipe`, `swap` und `dump` zu nehmen sind. Geben Sie dafür keine Geräte an, dann werden die in Datei `/etc/conf/cf.d/sassign` angegebenen Standardgeräte benutzt. Enthält der Bootstring Parameter für den Kern, so setzen diese die Standardwerte außer Kraft. Die Parameter für den Kern haben folgendes Format:

`gerät=xx(dritte_zahl[,offset])`

Dabei steht

- `gerät` für das gewünschte Systemgerät (`root[dev]`, `pipe[dev]`, `swap[dev]` oder `dump[dev]`),
- `xx` für den Gerätenamen (`ssd`, `hd` für eine Festplatte oder `ramd` für eine RAM Platte),
- `dritte_zahl` für die Gerätenummer, die sich nach den weiter oben stehenden Formeln berechnet und
- `offset` für den Offset in der Partition (in der Regel 0).

Werden `root`, `pipe`, `swap` oder `dumpdev` angegeben, befinden sich diese auf dem jeweils genannten Gerät, während nicht angegebene Systemgeräte die Standardgeräte benutzen.

DATEIEN

`/etc/conf/cf.d/sassign`

Standardwerte für die Systemgeräte. In der Datei steht:

`swap ssd 2 /dev/swap`

`dump ssd 6`

`root ssd 1`

`pipe ssd 1`

`/etc/conf/sdevice.d/ct`

Dateiverzeichnis der Gerätdateien für Bandlaufwerke am SCED-Board

`/etc/conf/sdevice.d/ssd`

Dateiverzeichnis der Gerätdateien für SCSI-Platten am SPA

`/etc/conf/sdevice.d/hd`

Dateiverzeichnis der Gerätdateien für ESDI-Platten am MBAD

SIEHE AUCH

`ssd(7)`, `ct(7)`, `hd(7)`.

BEZEICHNUNG

boot – Ladeprogramm des SINIX-Systems

BESCHREIBUNG

Das Programm `boot` lädt interaktiv Programme, die eigenständig d.h. ohne Unterstützung durch ein Betriebssystem, auf der Hardware ablaufen können, und veranlaßt deren Ausführung. Typischerweise wird `boot` zum Laden und Ausführen des Systemkerns benutzt. Im Prinzip kann aber auch ein beliebiges anderes Programm geladen und ausgeführt werden, vorausgesetzt, es ist für eine eigenständige Ausführung gebunden worden (siehe `ld(1)`). Das Programm `boot` ist unverzichtbarer Bestandteil des SINIX-Betriebssystems und muß in einer eigenen Plattenslice vorhanden sein, damit der SINIX-Systemkern erfolgreich geladen werden kann. Beachten Sie bitte, daß bei der Installation des SINIX-Betriebssystems ein Programm `masterboot` auf die Systemplatte (die Bootplatte) geschrieben wird. Das Programm `masterboot` befindet sich auf Zylinder 0 der Festplatte und ist das Standardladeprogramm für Prozeduren, die von der Festplatte laden.

Das System ruft das Programm `boot` immer auf, wenn der Rechner gestartet wird. Zunächst versucht das System, das Programm `boot` auf einer Diskette im Diskettenlaufwerk zu lokalisieren. Ist das Diskettenlaufwerk leer, aktiviert das System die Ladeprozedur von Festplatte. Der Ablauf der Ladeprozedur hängt davon ab, ob von Diskette oder von Festplatte geladen wird.

Wird von Diskette geladen, dann besteht die Ladeprozedur aus zwei Phasen:

1. Der Ladeblock in Sektor 0 des Dateisystems lädt `boot`.
2. `boot` wird ausgeführt, und der Benutzer wird zu den notwendigen Eingaben aufgefordert.

Wird von Festplatte geladen, dann besteht die Ladeprozedur aus drei Phasen:

1. Der Block `masterboot`, der sich in Sektor 0 auf der Festplatte befindet, wird vom ROM geladen.
2. Der Ladeblock von `masterboot` lädt dann den Ladeblock der Partition vom Sektor 0 der aktiven Partition. (Siehe `fdisk(1M)`).
3. Der Rest des Programms `boot` wird von den nächsten 29 Sektoren der Festplatte geladen.

Ablauf von boot

Wenn das Kommando `boot` das erste Mal aufgerufen wird, erscheint die folgende Statusmeldung:

```
Booting the UNIX System...
```

Soll `boot` den Standardkern und die Standardwerte benutzen, die in der Standardladedatei `/stand/boot` angegeben sind, dann machen Sie keine Eingabe. Wenn Sie eine beliebige Taste drücken, dann macht `boot` eine Pause, gibt eine Eingabeaufforderung (den Bootprompt) aus und fordert Sie zur Eingabe weiterer Angaben auf. Haben Sie das Programm `boot` von der Installationsdiskette geladen, dann drücken Sie keine Taste, damit `boot` die Standardwerte verwendet.

Wollen Sie ein anderes als das Standardprogramm laden, dann drücken Sie irgendeine Taste und unterbrechen auf diese Weise `boot`. Das Programm `boot` pausiert und fordert Sie anhand der folgenden Meldung auf, den Namen des Programms einzugeben, das geladen werden soll:

Enter the name of a kernel to boot:

Dann wartet `boot`, bis Sie den Namen des Programms eingegeben und die Taste RETURN gedrückt haben. Wie lange `boot` wartet, richtet sich nach der Anzahl der Sekunden, die bei der Option `TIMEOUT` in `/stand/boot` angegeben sind (siehe "Boot-Optionen"). Haben Sie nach der angegebenen Anzahl von Sekunden noch nichts eingegeben, und ist `AUTOBOOT` in der Datei `/stand/boot` auf `YES` gesetzt, dann verhält sich `boot` so, als ob Sie RETURN gedrückt hätten. Das Programm `boot` durchläuft weiter den Ladevorgang.

Wollen Sie nicht den SINIX-Systemkern, sondern ein anderes eigenständig ablaufbares Programm von der Installationsdiskette laden, dann müssen Sie spezifizieren, wo dieses Programm steht. Dazu geben Sie den vollständigen Pfadnamen der Datei an, die das Programm enthält - vorausgesetzt, die Datei befindet sich auf dem Standardladegerät. Sie verwenden also das folgende Format:

dateiname

Dabei steht *dateiname* für den standardmäßigen Pfadnamen im SINIX-System. Geben Sie *dateiname* als einziges Argument bei der Ladeeingabeaufforderung (dem Bootprompt) an, sucht `boot` auf dem Standardladegerät nach *dateiname* und versucht, von dort zu laden.

Boot-Optionen

`boot` wird durch eine Reihe von Optionen gesteuert. Standardwerte dafür stehen in der Datei `/stand/boot` oder werden von `boot` vorgelegt. Die Optionen werden unverändert an das Programm weitergereicht, sofern sie nicht durch die Benutzereingabe überschrieben werden (s.u.).

`boot` kennt folgende Optionen:

<code>AUTOBOOT=YES</code> oder <code>NO</code>	Legt fest, ob <code>boot</code> den Systemkern sofort lädt (<code>YES</code>) oder erst eine Eingabeaufforderung an der Konsole ausgibt (<code>NO</code>). Standardwert: <code>YES</code> .
<code>BOOTMSG=zeichenkette</code>	Die standardmäßige Lademeldung wird in <i>zeichenkette</i> geändert. Standardwert: Booting the UNIX System...
<code>BOOTPROMPT=zeichenkette</code>	Ändert die standardmäßige Eingabeaufforderung des Ladeprogramms zu der Zeichenkette <i>zeichenkette</i> . Standardwert: Enter the name of a kernel to boot:.
<code>DEFBOOTSTR=bootstring</code>	Der Name des zu ladenden Programms (und Parameter) werden auf <i>bootstring</i> voreingestellt. Dieser Bootstring wird verwendet, wenn Sie die Eingabeaufforderung von <code>boot</code> nur mit RETURN quittieren oder wenn die Wartezeit gemäß <code>TIMEOUT</code> abläuft. Standardwert: <code>/unix</code> .

boot (1M)**(WX200)****boot (1M)****INITPROG=pfad**

Gibt ein Initialisierungsprogramm an, das geladen und ausgeführt werden soll, bevor **boot** die Speichergröße testet.

MEMRANGE=bereich[,bereich...]

Teilt dem Programm **boot** diejenigen Speicherbereiche mit, deren Vorhandensein getestet werden soll. *bereich* hat das Format *start-ende:byte*. *start* und *ende* sind Dezimaladressen, die durch einen Bindestrich getrennt sind (zum Beispiel 1M-4M, 1K-65536). Aufsteigende (z.B. 15M-16M) und absteigende (z.B. 16M-15M) Adreßbereiche werden unterstützt. Die Startadresse zählt mit zum Speicherbereich (inklusive), die Endadresse nicht (exklusive). *byte* kann entweder die Zahl 0 oder 1 sein. 0 bezeichnet einen Speicherbereich ohne besondere Eigenschaften. 1 bezeichnet einen Speicherbereich, für den DMA nicht erlaubt ist. Standardwert: 0-640K:0, 1M-64M:0.

MEMREQ=größe

Weist **boot** an, eine Mindestspeichergröße von *größe* anzufordern. Findet **boot** nicht mindestens *größe* Byte Speicherplatz, gibt es eine Meldung aus und beendet sich. *größe* hat dasselbe Format wie die Adressen bei **MEMRANGE**. Standardwert: 0.

MREQMSG1=zeichenkette

Diese Option legt die Nachricht fest, die ausgegeben wird, wenn die in **MEMREQ** geforderte Speichergröße nicht verfügbar ist. Standardwert: Your system does not have the required minimum of 4 Megabytes of memory.

MREQMSG2=zeichenkette

Gibt eine optionale zweite Zeile als Ergänzung zur Nachricht **MREQMSG1** an. Standardwert: You must power down the machine and add more memory.

TIMEOUT=anz

Wenn **TIMEOUT** gesetzt ist, wartet **boot** genau *anz* Sekunden auf eine Benutzereingabe. Nach Ablauf dieser Zeit verwendet **boot** stattdessen den Bootstring, der **DEFBOOTSTR** zugewiesen ist. Ist *anz* gleich 0, dann ist die Uhr zum Warten nicht gesetzt. Standardwert: 30.

Anpassen des Ladeprozesses

Sie können den Ladeprozeß automatisieren. Dazu setzen Sie in der Datei **/stand/boot** die Option **AUTOBOOT** auf **YES**. **boot** verwendet dann die Boot-Optionen, die in der Datei **/stand/boot** stehen oder standardmäßig gesetzt sind, und läuft automatisch ab. **boot** zeigt die Lademeldung **BOOTMSG** an und lädt das Programm, das standardmäßig geladen werden soll. Tritt ein Fehler auf oder drücken Sie während des automatischen Ladens eine Taste, dann gibt **boot** die Ladeeingabeaufforderung aus und versucht erneut, das Programm zu laden. Das Programm **boot**, das auf der Installationsdiskette des SINIX-Betriebssystems vorhanden ist, führt einen solchen automatischen Ladevorgang aus.

Ist AUTOBOOT in der Datei /stand/boot auf NO gesetzt, dann können Sie einen Bootstring eingeben, bevor boot mit dem Laden beginnt. Wenn Sie keinen Bootstring eingeben, dann geht boot davon aus, daß Sie die Standardkonfiguration wünschen. boot verhält sich dann so, als sei AUTOBOOT in der Datei /stand/boot auf YES gesetzt.

Parameter für den Kern

Das Programm boot übergibt Zeichenketten aus /stand/boot an den Kern. Dieser liest sie, um festzustellen, welche Peripheriegeräte jeweils für root, pipe, swap und dump zu nehmen sind. Werden in der Datei /stand/boot keine Geräte dafür angegeben, dann werden die Standardgeräte benutzt. Die Parameter für den Kern haben folgendes Format:

ger=xx(n,o)

Dabei steht

- *ger* für das gewünschte Systemgerät (root[dev], pipe[dev], swap[dev] oder dumpdev),
- *xx* für den Gerätenamen ("hd" für Festplatte oder "fd" für Diskette),
- *n* für die Gerätenummer und
- *o* für den Offset in der Partition (in der Regel 0).

Werden root, pipe, swap oder dumpdev angegeben, befinden sich diese auf dem angegebenen Gerät, während die nicht angegebenen Systemgeräte die Standardgeräte benutzen. Die Definition eines Geräts beeinflusst die Standardwerte für die anderen Systemgeräte nicht.

Auf der Kommandozeile können nach dem Namen des Systemkerns noch weitere Argumente, durch Strichpunkt getrennt, übergeben werden. Zur Zeit wird der Parameter initargs=-s unterstützt. Dieser Parameter veranlaßt, daß init(1M) in den Einbenutzermodus wechselt, unabhängig davon welche Betriebsstufe (run level) in der Datei /etc/inittab voreingestellt ist.

DATEIEN

/stand/boot Datei mit Standardwerten zum Laden

SIEHE AUCH

fdisk(1M), init(1M), fd(7), hd(7).

FEHLERMELDUNGEN

Die Programme masterboot und boot haben unterschiedliche Fehlermeldungen. Das Programm masterboot zeigt eine Fehlermeldung an und sperrt das System. Es folgt eine Liste der häufigsten Meldungen des Programms masterboot und ihrer Bedeutungen:

- | | |
|---------|---|
| IO ERR | Ein Fehler trat beim Lesen des Ladeprogramms auf der Partition mit dem aktiven Betriebssystem auf. |
| BAD TBL | Der Anzeiger für die ladbare Partition von mindestens einem der Betriebssysteme in der Tabelle fdisk enthält unverständlichen Code. |

NO OS Ein nicht behebbaren Fehler trat bei dem Versuch auf, das Ladeprogramm der aktiven Partition des Betriebssystems auszuführen.

Das Programm `boot` zeigt eine Fehlermeldung an. Dann erscheint wieder die Eingabeaufforderung. Einige Meldungen von `boot` zeigen schwere Fehler an, die dazu führen, daß das System stoppt und neu geladen werden muß. Andere Meldungen von `boot` zeigen keine schweren Fehler an, weisen aber darauf hin, daß das Programm `boot` nicht korrekt läuft.

Die folgenden Meldungen bedeuten schwere Fehler. Wenn eine dieser Meldungen ausgegeben wird, muß das in der Meldung genannte Problem gelöst und das System neu geladen werden.

Error reading bootstrap

(Fehler beim Lesen des Urladers)

Das Programm `boot` konnte den Urlader nicht finden, oder der Urlader ist nicht lesbar. Sorgen Sie dafür, daß der Urlader sich an der korrekten Stelle auf dem angegebenen Ladegerät befindet und mit dem Kern kompatibel ist, der geladen werden soll. Dann müssen Sie das System neu laden.

No active partition on hard disk

(Keine aktive Partition auf der Festplatte)

Im Augenblick gibt es keine aktive Partition, von der aus das Programm `boot` gestartet werden kann. Ein geeignetes logisches Plattenlaufwerk muß aktiviert und dann das System neu geladen werden.

No file system to boot

(Kein Dateisystem zum Laden)

Das Programm `boot` konnte auf dem angegebenen Ladegerät kein Dateisystem `/stand` oder `/ (root)` finden. Sorgen Sie dafür, daß das Ladegerät eine Datei `/stand` oder `/ (root)` enthält, und laden Sie das System neu.

Folgende Liste beschreibt die Warnungen des Programms `boot`. Wenn eine dieser Warnungen angezeigt wird, muß das in der Warnung angegebene Problem behoben werden. Dann kann das Programm `boot` neu gestartet werden.

Cannot load initprog

(Initialisierungsprogramm kann nicht geladen werden)

Das Programm `boot` kann das mit der Option `INITPROG` angegebene Initialisierungsprogramm nicht finden, oder das Initialisierungsprogramm ist nicht korrekt installiert und kann nicht ausgeführt werden. Sorgen Sie dafür, daß das Argument *pfad* der Option `INITPROG` einen gültigen Pfad angibt und die Datei eine ausführbare Datei ist. Dann kann `boot` neu geladen werden.

Cannot open defaults file

(Kann Standarddatei nicht öffnen)

Das Programm `boot` kann die Datei `/stand/boot` auf dem Ladegerät nicht lokalisieren, oder die Datei ist nicht lesbar. Sorgen Sie dafür, daß die Datei `/stand/boot` auf dem Ladegerät vorhanden und lesbar ist. Dann kann `boot` neu geladen werden.

command argument missing or incorrect

(*kommando* Argument fehlt oder nicht korrekt)

An das Programm *boot* wurde ein Kommando ohne Argument oder mit einem ungültigen Argument übergeben. Sorgen Sie dafür, daß *kommando* in der Datei */stand/boot* die richtige Anzahl von Argumenten hat und alle Argumente gültig sind. Dann kann *boot* neu gestartet werden.

Cannot load *datei*; file not opened

(*datei* nicht ladbar, da nicht geöffnet)

Das Programm *boot* kann *datei* auf dem angegebenen Gerät nicht finden, oder *datei* ist nicht korrekt installiert, so daß eine Ausführung nicht möglich ist. Sorgen Sie dafür, daß *datei* auf dem angegebenen Gerät vorhanden ist. Dann kann *boot* neu gestartet werden.

Cannot load *datei*; cannot read COFF header

(*datei* nicht ladbar, da COFF-Kopfsatz nicht lesbar)

Die angegebene COFF-Datei (Common Object File Format) enthält keinen Dateikopf, oder der Dateikopf ist nicht lesbar. Sorgen Sie dafür, daß *datei* einen lesbaren Dateikopf enthält. Dann kann *boot* neu geladen werden.

Cannot load *datei*; not an 80386 ELF or COFF binary

(*datei* nicht ladbar, da nicht 80386 ELF oder COFF-Binärformat)

Die angegebene Datei ist keine 80386 ELF- oder COFF-Binärdatei. Sorgen Sie dafür, daß die Datei, die Sie laden wollen, eine gültige COFF-Binärdatei ist und daß sie mit 80386-Systemen kompatibel ist. Dann kann *boot* neu gestartet werden.

Cannot load *datei*; cannot read segment/sections

(*datei* nicht ladbar, da Segment/Abschnitte nicht lesbar)

Die angegebene Datei enthält keinen Abschnittskopf, oder der Abschnittskopf ist nicht lesbar. Sorgen Sie dafür, daß *datei* einen lesbaren Abschnittskopf enthält. Dann kann *boot* neu gestartet werden.

Cannot load *datei*; cannot read BKI section

(*datei* nicht ladbar, da BKI-Abschnitt nicht lesbar)

Die angegebene Datei enthält keinen BKI-Abschnitt (Bootstrap-Kernel Interface), oder der BKI-Abschnitt ist nicht lesbar. Sorgen Sie dafür, daß der BKI-Abschnitt von *datei* für die Kern- und Urladeversion korrekt ist. Dann kann *boot* neu gestartet werden.

Cannot load *datei*; BKI too old

(*datei* nicht ladbar, BKI zu alt)

Die BKI des aktuellen Urladeprogramms ist nicht mit der BKI des Programms (*datei*) kompatibel, das geladen werden soll. Sorgen Sie dafür, daß die BKI des Urladeprogramms und *datei* kompatibel sind. Dann kann *boot* neu gestartet werden.

Cannot load *datei*; BKI too new

(*datei* nicht ladbar, BKI zu neu)

Die BKI des aktuellen Urladeprogramms ist nicht mit der BKI des Programms (*datei*) kompatibel, das geladen werden soll. Sorgen Sie dafür, daß die BKI des Urladeprogramms und *datei* kompatibel sind. Dann kann boot neu gestartet werden.

Cannot load *datei*; missing text or data segment

(*datei* nicht ladbar, Text- oder Datensegment fehlt)

In der angegebenen Datei fehlt ein notwendiges Text- oder Datensegment. Sorgen Sie dafür, daß *datei* die benötigten Text- und Datensegmente enthält. Dann kann boot neu gestartet werden.

Cannot load *datei*; missing BKI segment

(*datei* nicht ladbar, BKI-Segment fehlt)

Die angegebene Datei enthält das BKI-Segment nicht. Sorgen Sie dafür, daß das BKI-Segment in *datei* vorhanden und mit der BKI des Urladeprogramms kompatibel ist.

Cannot load *datei*

(*datei* nicht ladbar, nicht genügend Speicher für Kern)

Für den Kern steht kein oder nicht genügend Hauptspeicher zur Verfügung. Sorgen Sie dafür, daß für den Kern, der geladen werden soll, genügend Speicher zur Verfügung gestellt werden kann. Dann kann boot neu gestartet werden.

Too many lines in defaults file; extra lines ignored

(Zu viele Zeilen in Standarddatei, überzählige Zeilen ignoriert)

Die Datei file/stand/boot enthält zu viele Zeilen. Alle überzähligen Zeilen werden ignoriert.

Too many arguments on command line, extra argument ignored

(Zu viele Argumente in der Kommandozeile, überzähliges Argument ignoriert)

In der Kommandozeile sind zu viele Argumente angegeben.

HINWEIS

Der Rechner versucht immer, zunächst von einer Diskette im Diskettenlaufwerk zu laden. Enthält die Diskette kein gültiges Urladeprogramm, treten Fehler auf.

Das Programm boot kann nicht dazu benutzt werden, Programme zu laden, die nicht für die eigenständige Ausführung gebunden wurden. Um eigenständige Programme zu erstellen, verwenden Sie die Option des SINIX-Systembinders (ld (1)) sowie spezielle eigenständige Bibliotheken.

Obwohl eigenständige Programme im realen oder im geschützten Modus laufen können, darf es sich bei diesen nicht um Large-Model- oder Huge-Model-Programme handeln. Programme im Realmodus (real mode) können die E/A-Routinen des Start-ROM benutzen.

BEZEICHNUNG

bootflags – Ladeoptionen des Betriebssystems anzeigen/ändern

ÜBERSICHT

```
bootflags [ -p ] [ -f ladeoptionen ] [ -b lade_zeichenkette ] [ -a lade_zeichenkette ]
bootflags [ -p ]
```

BESCHREIBUNG

Mit bootflags können Sie Werte anzeigen und ändern, die das Laden des Betriebssystems und Aktionen zur Fehlersuche steuern. Zu diesen Werten gehören die Ladeoptionen (Bootflags) und die Ladezeichenkette (Bootstring).

Wird bootflags ohne Optionen aufgerufen, dann werden die aktuellen Ladeoptionen (Bootflags) und die aktuelle Ladezeichenkette (Bootstring) angezeigt.

- p Es werden die Werte (powerup values) angezeigt oder geändert, die nach dem Einschalten des Stroms gelten sollen. Ohne diese Option werden die Werte (next-reboot values) angezeigt/geändert, die beim nächsten Wiederhochfahren des Systems aktuell sein sollen.
- f Die angegebenen ladeoptionen (Bootflags) werden eingestellt. Die Bootflags sind weiter unten beschrieben.
- b Die angegebene lade_zeichenkette (Bootstring) wird eingestellt. Die Syntax eines Bootstrings ist beim Kommando boot beschrieben.
- a Die angegebene lade_zeichenkette (Bootstring) für den "auxiliary boot" wird eingestellt. Die Syntax eines Bootstrings ist beim Kommando boot beschrieben.

Bootflags

Für die Bootflags (Lademarken) sind die folgenden Werte definiert:

```
#define BF_AUTOBOOT      0x000
/* Flags für automatisches
Hochfahren des Systems */
#define BF_ASKNAME      0x001
/* Nach dem Dateinamen fragen, von dem
das System hochgefahren werden soll */
#define BF_SINGLE      0x002
/* In den Einbenutzerbetrieb
hochfahren */
#define BF_NOSYNC      0x004
/* ohne sync-Aufruf hochfahren */
#define BF_HALT      0x008
/* nicht hochfahren, stattdessen
anhalten */
#define BF_INITNAME      0x010
#define BF_NO_CTRL      0x020
#define BF_NO_INIT      0x040
```

bootflags(1M)**(MX500)****bootflags(1M)**

```
#define BF_DUMP          0x080
                        /* von zusätzlichem Boot-Namen
                        hochfahren */
#define BF_CONFIG        0x100
#define BF_NO_CACHE      0x200
                        /* Cache-Speicher auf dem Board
                        nicht einschalten */
```

BEISPIEL

Der folgende Aufruf ändert den Bootstring so, daß der Kern `/unix` aus dem Dateisystem `/stand` geladen wird, das sich auf einer SCSI-Platte am SPA Board befindet. Die Platte ist die erste, die über den ersten Kanal am ersten SPA Controller angeschlossen ist. Dieser Bootstring ist ein Standardbeispiel zum Installieren des Kerns von der Platte:

```
bootflags -p -b 'sp(0,10)unix root=ssd(1) swap=ssd(2) rootfs=ufs'
```

Die Anführungszeichen sind nötig, um den Bootstring mit seinen Argumenten abzugrenzen gegenüber den Optionen des Kommandos `bootflags`.

HINWEIS

Wenn Sie eine Festplattenpartition mit einem Dateisystem angeben, auf das ein Speicherabzug geschrieben werden soll, dann ist dieses Dateisystem nach einem Systemabsturz zerstört.

DATEIEN

`/dev/sm_cons`

Batterie-gepuffertes RAM am SCED-Board zur Speicherung der Ladeoptionen.

SIEHE AUCH

`boot(1M)`

BEZEICHNUNG

bp – Dämon zum periodischen Überprüfen der Betriebsbedingungen

ÜBERSICHT

/sbin/bp [-d] [-s *sekunden*]

BESCHREIBUNG

Der Dämon bp liest in regelmäßigen Zeitabständen ein Gerätereister auf dem SCED-Board, um den Zustand der Lüfter in Haupt- und Erweiterungsgehäuse und der unterbrechungsfreien Stromversorgung (UPS) zu überprüfen.

Falls einer der Lüfter bei zwei aufeinanderfolgenden Überprüfungen nicht läuft, fährt bp die Maschine herunter.

Falls die Hauptstromversorgung bei zwei aufeinanderfolgenden Überprüfungen ausfällt und eine unterbrechungsfreie Stromversorgung installiert ist, wird das Shellskript /etc/poweroff ausgeführt. Dieses Shellskript fährt normalerweise die Maschine herunter. Es kann jedoch so abgeändert werden, daß es innerhalb des Shellskripts /sbin/bpchk aufgerufen wird. Ein Hardware-Zeitgeber unterbricht die Verbindung zur unterbrechungsfreien Stromversorgung nach 8 Minuten.

Wird bp ohne Optionen aufgerufen, so findet alle 20 Sekunden eine Überprüfung statt.

-s *sekunden*

Abstand der Überprüfungen auf *sekunden* Sekunden einstellen. *sekunden* muß dabei im Bereich von 1 bis 86400 liegen.

-d Diagnosemodus einstellen.

DATEIEN

/sbin/bpchk

BEZEICHNUNG

bpchk – Periodische Überprüfung der Betriebsbedingungen

ÜBERSICHT

/sbin/bpchk [-d]

BESCHREIBUNG

bpchk liest ein Gerätereister auf dem SCED-Board, um den Zustand der Lüfter in Haupt- und Erweiterungsgehäuse und der unterbrechungsfreien Stromversorgung (UPS) zu überprüfen.

bpchk kennt die folgende Option:

-d Diagnosemodus einstellen.

EXIT-CODES

Rufen Sie bpchk ohne Optionen auf, so wird ein Exit-Code mit folgender Bedeutung zurückgegeben:

- 3 Der Lüfter im Erweiterungsgehäuse ist ausgefallen.
- 4 Der Lüfter im Hauptgehäuse ist ausgefallen.
- 6 Die Hauptstromversorgung ist ausgefallen und die unterbrechungsfreie Stromversorgung ist angeschaltet.

DATEIEN

/sbin/bp

BEZEICHNUNG

`brc`, `bcheckrc` – Prozeduren zur Systeminitialisierung

ÜBERSICHT

`/sbin/brc`

`/sbin/bcheckrc`

BESCHREIBUNG

Diese Shell-Prozeduren werden von `init` über Einträge in der Datei `/etc/inittab` jedesmal dann ausgeführt, wenn das System geladen wird.

Zunächst überprüft die Prozedur `bcheckrc` den Status des Dateisystems `/` (`root`). Erweist sich dieses Dateisystem als fehlerhaft, wird es von `bcheckrc` wiederhergestellt.

Dann hängt `bcheckrc` folgende Dateisysteme - sofern vorhanden - ein: `/dev/fd` und `/proc` ohne Überprüfung, sowie `/var` nach vorheriger Überprüfung. `/var` kann als Verzeichnis im Dateisystem `/` oder als separates Dateisystem vorliegen.

Das Skript `brc` führt Verwaltungsaufgaben für Dateien am lokalen Rechner durch, auf die von fernen Rechnern aus zugegriffen werden kann.

Beachten Sie bitte, daß `bcheckrc` immer vor `brc` ausgeführt werden muß.

SIEHE AUCH

`fsck(1M)`, `init(1M)`, `rc2(1M)`, `shutdown(1M)`, `inittab(4)`, `mnttab(4)`.

BEZEICHNUNG

captoinfo - konvertiert eine termcap-Beschreibung in eine terminfo-Beschreibung

ÜBERSICHT

captoinfo [-v ...] [-V] [-1] [-w *anzahl*] *datei* ...

BESCHREIBUNG

captoinfo sucht in *datei* nach termcap-Beschreibungen. Für jede gefundene Beschreibung wird eine äquivalente terminfo-Beschreibung zusammen mit allen dazugehörigen Kommentaren (sofern vorhanden) in die Standardausgabe geschrieben. Eine Beschreibung, die in Relation zu einer anderen Beschreibung ausgedrückt ist (wie in termcap `tc = field` angegeben), wird vor der Ausgabe auf die kleinste übergeordnete Menge reduziert.

Wird *datei* nicht angegeben, wird die Umgebungsvariable `TERMCAP` für den Dateinamen oder Eintrag benutzt. Ist `TERMCAP` der vollständige Pfadname einer Datei, wird nur das Terminal, dessen Name in der Umgebungsvariablen `TERM` angegeben ist, aus der Datei extrahiert. Ist die Umgebungsvariable `TERMCAP` nicht gesetzt, wird die Datei `/usr/share/lib/termcap` gelesen.

- v Gibt während des Programmlaufs Ablaufinformationen auf der Standardfehlerausgabe aus. Zusätzliche Optionen dieser Art (-v) bewirken, daß die Ausgabe detaillierter wird.
- V Gibt auf der Standardfehlerausgabe die benutzte Programmversion aus und verläßt das Programm.
- 1 Bewirkt, daß immer nur ein Feld pro Zeile ausgegeben wird. Andernfalls werden mehrere Felder pro Zeile ausgegeben, und zwar bis zu einer Zeilenlänge von höchstens 60 Zeichen.
- w Ändert die Ausgabe auf *anzahl* Zeichen.

DATEIEN

`/usr/share/lib/terminfo/?/*`

Kompilierte Datenbank mit Terminalbeschreibungen

HINWEIS

Mit captoinfo sollen termcap-Einträge in terminfo-Einträge konvertiert werden. Die termcap-Datenbank aus früheren SINIX Versionen ist in zukünftigen Versionen nicht mehr vorhanden.

SIEHE AUCH

curses(3X), infocmp(1M), terminfo(4).

BEZEICHNUNG

checkfsys – überprüft ein Dateisystem

ÜBERSICHT

checkfsys

BESCHREIBUNG

Mit dem Kommando `checkfsys` kann ein Dateisystem auf Beschädigungen hin überprüft und auf Wunsch wiederhergestellt werden. Das Kommando ruft eine Menü-Schnittstelle auf, die auch über das Kommando `sysadm` zur Verfügung steht. Zunächst wird man dazu aufgefordert, das Gerät zu wählen, welches das betreffende Dateisystem enthält. Dann muß angegeben werden, was für eine Prüfung durchgeführt werden soll. Folgende Möglichkeiten stehen zur Verfügung:

`check only`

(nur prüfen). Das Dateisystem wird überprüft, aber nicht wiederhergestellt.

`interactive fix`

(interaktive Wiederherstellung). Das Dateisystem wird im Dialog mit dem Benutzer wiederhergestellt, d.h. der Benutzer wird auf jede Beschädigung aufmerksam gemacht und gefragt, ob sie behoben werden soll.

`automatic fix`

(automatische Wiederherstellung). Das Dateisystem wird automatisch wiederhergestellt. Alle Beschädigungen werden nach einem Standardverfahren behoben.

Über das Menü `sysadm` steht dieselbe Funktion zur Verfügung:

`sysadm check`

FEHLERMELDUNGEN

Die Exit-Codes von `checkfsys` haben folgende Bedeutung:

- 0 Normales Ende.
- 2 Ungültige Kommandosyntax. Eine Meldung zur Verwendung des Kommandos wird angezeigt.
- 7 Die Menü-Schnittstelle für dieses Kommando steht nicht zur Verfügung, weil `fmli` nicht aufgerufen werden kann. Das Programmpaket FMLI ist nicht installiert oder beschädigt.

checkfsys (1M)

checkfsys (1M)

SIEHE AUCH

`fsck(1M)`, `makefsys(1M)`, `mount fsys(1M)`, `sysadm(1M)`.

HINWEIS

Automatische Überprüfungen und Überprüfungen im Dialog sind zwar in der Regel erfolgreich, es kann jedoch gelegentlich eine Datei oder ein Dateiname verlorengehen. Dateien mit Inhalt, aber ohne Namen, werden in das Verzeichnis *dateisystem/lost+found* eingetragen.

Wenn unter keinen Umständen Daten verloren gehen dürfen, sollte zuerst das Dateisystem auf mögliche Beschädigungen hin überprüft werden. Liegen Beschädigungen vor, sollte eine der Wiederherstellungsfunktionen verwendet werden.

BEZEICHNUNG

`chroot` – ändert das Root-Verzeichnis für ein Kommando

ÜBERSICHT

`/usr/sbin/chroot neuesroot kommando`

BESCHREIBUNG

`chroot` bewirkt, daß das eingegebene Kommando relativ zum neuen Root-Verzeichnis *neuesroot* ausgeführt wird. Die Bedeutung der Schrägstriche am Anfang von Pfadnamen (/) ändert sich. Für das Kommando und seine Kindprozesse bezeichnen diese Schrägstriche das Verzeichnis *neuesroot*. Wenn das Kommando ausgeführt wird, ist das Arbeitsverzeichnis zunächst *neuesroot*.

Wird jedoch die Ausgabe des Kommandos mit

`chroot neuesroot kommando >x`

in eine Datei *x* umgeleitet, so wird *x* relativ zum ursprünglichen und nicht relativ zum neuen Root-Verzeichnis angelegt.

Der Pfadname des neuen Root-Verzeichnisses gilt immer relativ zum aktuellen Root-Verzeichnis. Auch wenn `chroot` gerade aktiv ist, ist das Argument *neuesroot* relativ zum aktuellen Root-Verzeichnis des gerade laufenden Prozesses.

Dieses Kommando darf nur vom Systemverwalter ausgeführt werden.

SIEHE AUCH

`cd(1)` im in den *Kommandos*.

`chroot(2)` im *Referenzhandbuch für Programmierer*.

HINWEIS

Gehen Sie mit äußerster Vorsicht vor, wenn Sie im neuen Root-Dateisystem auf Gerätedateien verweisen!

BEZEICHNUNG

chrtbl – generiert Zeichenklassifikations- und -umsetzungstabellen

ÜBERSICHT

chrtbl [datei]

BESCHREIBUNG

Das Kommando `chrtbl` legt zwei Tabellen an, die Informationen zur Zeichenklassifikation, zur Umwandlung von Groß- in Kleinbuchstaben und umgekehrt, zur Zeichensatzbreite und zur Formatierung von Zahlen enthalten. Eine Tabelle ist ein verschlüsseltes Feld (Array) von $(257*2) + 7$ Byte. Über Suchfunktionen in dieser Tabelle kann die Zeichenklasse eines Zeichens festgestellt, ein Zeichen konvertiert (siehe `ctype(3C)`) und die Byte- und Bildschirmgröße eines Zeichens in einer der zusätzlichen Zeichensatztabellen gesucht werden. Die andere Tabelle enthält Informationen zum Format numerischer Werte, die keine Währungsangaben sind: Das erste Byte gibt das Dezimaltrennzeichen an, das zweite Byte das Tausendertrennzeichen, die restlichen Byte stellen eine mit dem Nullzeichen abgeschlossene Zeichenkette dar, die die Gruppe angibt (jedes Element der Zeichenkette ist eine ganze Zahl, die die Anzahl der Ziffern angibt, aus denen die aktuelle Gruppe in einer formatierten numerischen Menge besteht, die keine Währung angibt).

`chrtbl` liest die vom Benutzer definierten Zeichenklassifikations- und Umsetzungsinformationen in *datei* ein und legt im aktuellen Verzeichnis drei Ausgabedateien an. Um *datei* anzulegen, benutzt man die Datei in `/usr/lib/locale/C/chrtbl_C` als Ausgangspunkt. Es können Einträge hinzugefügt werden, aber die ursprünglichen Werte für das System dürfen nicht geändert werden. Zum Beispiel kann es vorkommen, daß für andere Standorte 8 Bit umfassende Einträge zu den ASCII-Definitionen in dieser Datei hinzugefügt werden.

Eine Ausgabedatei mit dem Namen `ctype.c` (eine Quelldatei in C), enthält ein Feld von $(257*2)+7$ Byte, das durch die Verarbeitung der Daten in *datei* generiert wurde. Überprüfen Sie den Inhalt der Datei `ctype.c`, um sicherzustellen, daß das Feld Ihren Wünschen entspricht. Zusätzlich kann `ctype.c` auch von einem Anwendungsprogramm verwendet werden. Die ersten 257 Byte des Felds in `ctype.c` werden für die Zeichenklassifikation benutzt. Die Zeichen, die für die Initialisierung dieser Byte des Felds verwendet werden, stellen Zeichenklassifikationen dar, die in der Datei `/usr/include/ctype.h` definiert sind. Zum Beispiel bedeutet `_L`, daß ein Zeichen ein Kleinbuchstabe ist, und `_S|_B`, daß das Zeichen sowohl ein Zwischenraum als auch ein Leerzeichen ist. Die zweiten 257 Byte des Felds werden für die Zeichenumsetzung verwendet. Die Byte des Felds werden so initialisiert, daß Zeichen, für die keine Umsetzungsinformationen zur Verfügung stehen, in sich selbst konvertiert werden. Stehen Umsetzungsinformationen zur Verfügung, wird der erste Wert des Wertepaares dort gespeichert, wo normalerweise der zweite gespeichert würde, und umgekehrt. Stellt man zum Beispiel das Wertepaar `<0x41 0x61>` zur Verfügung, wird `0x61` dort gespeichert, wo im Normalfall `0x41` gespeichert würde, und `0x41` dort, wo im Normalfall `0x61` gespeichert würde. Die letzten 7 Byte werden für Informationen über Zeichenbreiten für bis zu drei zusätzliche Zeichensätze verwendet.

Die zweite Ausgabedatei, eine Datendatei, enthält dieselben Informationen, ist aber so strukturiert, daß sie von den Zeichenklassifikations- und -umsetzungsroutinen effizient benutzt werden kann (siehe `ctype(3C)`). Als Name dieser Ausgabedatei wird der Wert benutzt, den man dem Schlüsselwort `LC_CTYPE` zuweist. Dieses Schlüsselwort wird aus *datei* eingelesen. Bevor diese Datei von den Zeichenklassifikations- und -umsetzungsroutinen benutzt werden kann, muß sie unter dem Namen `LC_CTYPE` im Verzeichnis `/usr/lib/locale/standort` installiert werden, und zwar vom Systemverwalter oder einem Mitglied der Gruppe `bin`. Lesezugriff auf diese Datei benötigen Benutzer (`user`), Gruppe (`group`) und Andere (`other`); andere Berechtigungen sollten nicht vergeben werden. Um die Zeichenklassifikations- und -umsetzungstabellen in dieser Datei benutzen zu können, muß die Umgebungsvariable `LC_CTYPE` entsprechend gesetzt werden (siehe `environ(5)` oder `setlocale(3C)`).

Die dritte Ausgabedatei, eine Datendatei, wird nur dann angelegt, wenn Informationen zur Formatierung von Zahlen in der Eingabedatei enthalten sind. Als Name dieser Ausgabedatei wird der Wert verwendet, den man dem Schlüsselwort `LC_NUMERIC` zuweist, das aus *datei* eingelesen wird. Bevor diese Datei verwendet werden kann, muß sie unter dem Namen `LC_NUMERIC` im Verzeichnis `/usr/lib/locale/standort` installiert werden, und zwar vom Systemverwalter oder einem Mitglied der Gruppe `bin`. Lesezugriff auf diese Datei benötigen Benutzer (`user`), Gruppe (`group`) und Andere (`other`); eine andere Berechtigung sollte nicht gesetzt werden. Damit die Informationen zur Formatierung von Zahlen in dieser Datei verwendet werden können, muß die Umgebungsvariable `LC_NUMERIC` entsprechend gesetzt werden (siehe `environ(5)` oder `setlocale(3C)`).

Der Name des Standorts, an dem die Dateien `LC_CTYPE` und `LC_NUMERIC` installiert werden, sollte den Konventionen entsprechen, die in *datei* definiert sind. Hat man zum Beispiel französische Konventionen definiert und lautet der Name des französischen Standorts auf dem System `french`, dann sollten die Dateien im Verzeichnis `/usr/lib/locale/french` installiert werden.

Wenn keine Eingabedatei angegeben wird oder das Argument `"-"` vorhanden ist, liest `chrtbl` von der Standardeingabe.

Die Syntax von *datei* ermöglicht es dem Benutzer, die Namen der Datendateien zu definieren, die von `chrtbl` angelegt werden, ebenso die Zuordnung von Zeichen zu Zeichenklassifikationen, die Beziehung zwischen Groß- und Kleinbuchstaben, die Byte- und Bildschirmgrößen von bis zu drei zusätzlichen Zeichensätzen sowie drei Angaben zur Formatierung von Zahlen: das Dezimaltrennzeichen, das Tausendertrennzeichen und die Gruppierung. Folgende Schlüsselwörter werden von `chrtbl` erkannt:

<code>LC_CTYPE</code>	Name der Datendatei, die von <code>chrtbl</code> angelegt wird und Informationen über Zeichenklassifikation, Zeichenumsetzung und Zeichenbreiten aufnimmt
<code>isupper</code>	Zeichencodes, die als Großbuchstaben gelten sollen
<code>islower</code>	Zeichencodes, die als Kleinbuchstaben gelten sollen

isdigit	Zeichencodes, die als Zahlen gelten sollen
isspace	Zeichencodes, die als Vorlauf- bzw. Trennzeichen gelten sollen
ispunct	Zeichencodes, die als Interpunktionszeichen gelten sollen
isctrl	Zeichencodes, die als Steuerzeichen gelten sollen
isblank	Zeichencodes, die als Leerzeichen gelten sollen
isxdigit	Zeichencodes, die als Hexadezimalziffern gelten sollen
ul	Beziehung zwischen Groß- und Kleinbuchstaben
cswidth	Bytegrößen- und Bildschirmgrößeninformationen (standardmäßig ein Zeichen breit)
LC_NUMERIC	Name der Datendatei, die von chrtbl angelegt wird, um Informationen zur Formatierung von Zahlen aufzunehmen
decimal_point	Dezimaltrennzeichen
thousands_sep	Tausendertrennzeichen
grouping	Zeichenkette, in der jedes Element ein ganze Zahl darstellt, die die Anzahl der Ziffern in der aktuellen Gruppe spezifiziert, und zwar in einer formatierten numerischen Angabe, die nicht für eine Währung steht

Alle Zeilen mit dem Zeichen # in der ersten Spalte gelten als Kommentar und werden ignoriert. Auch Leerzeilen werden ignoriert.

Die Zeichen für isupper, islower, isdigit, isspace, ispunct, isctrl, isblank, isxdigit und ul können als hexadezimale oder oktale Konstante dargestellt werden, d.h. der Buchstabe a kann zum Beispiel hexadezimal als 0x61 oder oktal als 0141 wiedergegeben werden. Hexadezimale und oktale Konstanten können durch ein oder mehrere Leerzeichen und/oder Tabulatorzeichen getrennt werden.

Mit dem Gedankenstrich (-) kann ein Bereich aufeinanderfolgender Zahlen angegeben werden. Der Gedankenstrich kann durch ein oder mehrere Leerzeichen von den Zahlen getrennt oder ohne Leerzeichen an die Zahlen angeschlossen werden.

Mit dem Gegenschrägstrich (\) wird angezeigt, daß die Zeile fortgesetzt wird. Nach einem Gegenschrägstrich ist nur ein Wagenrücklaufzeichen erlaubt.

Die Zuordnung von Groß- und Kleinbuchstaben zueinander (ul) wird in Form geordneter Paare von oktalen oder hexadezimalen Konstanten ausgedrückt: <großbuchstabe kleinbuchstabe>. Diese beiden Konstanten können durch ein oder mehrere Leerzeichen getrennt werden. Die spitzen Klammern (< >) können durch ein oder mehrere Leerzeichen von den Zahlen getrennt oder direkt an die Zahlen angeschlossen werden.

Es folgt das Format einer Eingabespezifikation für `cswidth`:

`n1:s1,n2:s2,n3:s3`

Dabei stehen

- `n1` Bytegröße für zusätzlichen Zeichensatz 1 (erforderlich)
- `s1` Bildschirmgröße für zusätzlichen Zeichensatz 1
- `n2` Bytegröße für zusätzlichen Zeichensatz 2
- `s2` Bildschirmgröße für zusätzlichen Zeichensatz 2
- `n3` Bytegröße für zusätzlichen Zeichensatz 3
- `s3` Bildschirmgröße für zusätzlichen Zeichensatz 3

`decimal_point` und `thousands_sep` werden durch ein einzelnes Zeichen angegeben, das dem betreffenden Trennzeichen entspricht. `grouping` wird durch eine in Anführungszeichen eingeschlossene Zeichenkette angegeben, in der jedes Element oktal oder hexadezimal dargestellt werden kann. Zum Beispiel kann mit `\3` oder `\x3` der Wert eines Elements der Zeichenkette auf 3 gesetzt werden.

BEISPIEL

Es folgt ein Beispiel für eine Eingabedatei, mit der die Definitionstabelle für den Codesatz für AMERIKANISCHES ENGLISCH in einer Datei mit dem Namen `usa` und die Informationen zur Formatierung von Zahlen, die keine Währungen angeben, in einer Datei mit dem Namen `num-usa` festgelegt werden.

```
LC_CTYPE usa
isupper 0x41 - 0x5a
islower 0x61 - 0x7a
isdigit 0x30 - 0x39
isspace 0x20 0x9 - 0xd
ispunct 0x21 - 0x2f 0x3a - 0x40 \
        0x5b - 0x60 0x7b - 0x7e
iscntrl 0x0 - 0x1f 0x7f
isblank 0x20
isxdigit 0x30 - 0x39 0x61 - 0x66 \
        0x41 - 0x46
ul      <0x41 0x61> <0x42 0x62> <0x43 0x63> \
        <0x44 0x64> <0x45 0x65> <0x46 0x66> \
        <0x47 0x67> <0x48 0x68> <0x49 0x69> \
        <0x4a 0x6a> <0x4b 0x6b> <0x4c 0x6c> \
        <0x4d 0x6d> <0x4e 0x6e> <0x4f 0x6f> \
        <0x50 0x70> <0x51 0x71> <0x52 0x72> \
        <0x53 0x73> <0x54 0x74> <0x55 0x75> \
        <0x56 0x76> <0x57 0x77> <0x58 0x78> \
        <0x59 0x79> <0x5a 0x7a>
cswidth 1:1,0:0,0:0
LC_NUMERIC num-usa
decimal_point .
thousands_sep ,
grouping      "\3"
```


DATEIEN

`/usr/lib/locale/standort/LC_CTYPE`
Datendateien mit Informationen zur Zeichenklassifikation,
Zeichenumsetzung und Zeichensatzbreite, die von `chrtbl`
erstellt werden

`/usr/lib/locale/standort/LC_NUMERIC`
Datendateien mit Informationen zur Formatierung von Zahlen,
die von `chrtbl` erstellt werden

`/usr/include/ctype.h`
Include-Datei mit Informationen, die von Zeichenklassifikations-
und -umsetzungsroutinen verwendet werden

`/usr/lib/locale/C/chrtbl_C`
Eingabedatei, die zur Erstellung von `LC_CTYPE` und
`LC_NUMERIC` im Standardverzeichnis `locale` verwendet wird

SIEHE AUCH

`environ(5)`.
`ctype(3C)`, `setlocale(3C)` im *Referenzhandbuch für Programmierer*.

HINWEIS

Die Fehlermeldungen von `chrtbl` bedürfen in der Regel keiner Erklärung. Sie weisen auf Fehler in der Kommandozeile oder auf syntaktische Fehler in der Eingabedatei hin.

WARNUNG

Werden die Dateien im Verzeichnis `/usr/lib/locale/C` geändert, wird das Verhalten des Systems unvorhersagbar.

BEZEICHNUNG

ckbinarsys – stellt fest, ob das ferne System binäre Meldungen akzeptiert

ÜBERSICHT

ckbinarsys [-S] -s *fernsystemname* -t *inhaltsart*

BESCHREIBUNG

Mit `rmail` können binäre Daten transportiert werden. Daher muß unter Umständen geprüft werden, ob ein bestimmtes fernes System (beispielsweise der nächste Knoten) binäre Daten bearbeiten kann, die entsprechend dem ausgewählten Protokoll (uux, SMTP etc.) gesendet werden.

`ckbinarsys` entnimmt der Datei `/etc/mail/binarsys` Informationen zu den einzelnen fernen Systemen. `ckbinarsys` gibt seine Ergebnisse über einen entsprechenden Exit-Code zurück. Der Exit-Code 0 bedeutet, daß eine Meldung mit dem angegebenen Inhalt an das angegebene System gesendet werden kann. Ein Exit-Code verschieden von 0 bedeutet, daß das ferne System Meldungen mit binärem Inhalt nicht verarbeiten kann.

Gibt es keine Datei `binarsys`, endet `ckbinarsys` mit einem Exit-Code verschieden von 0.

Argumente sind:

-s *fernsystemname*

Name des fernen Systems, das in der Datei `/etc/mail/binarsys` gesucht werden soll

-t *inhaltsart*

Art des Inhalts der Meldung, die gesendet werden soll. Im Falle von `rmail` kann dieser durch zwei Zeichenketten angegeben werden, nämlich `text` oder `binary`. Dies wird von `mail` festgelegt, und zwar unabhängig von Kopfzeilen innerhalb der Meldung, die die Art des Inhalts angeben (Content-Type). Alle anderen Argumente gelten als Äquivalente von `binary`.

-S

Normalerweise gibt `ckbinarsys` eine Meldung aus, wenn eine binäre Meldung vom fernen System nicht verarbeitet werden kann. Diese ausgegebene Meldung kann von `rmail` als negative Rückmeldung verwendet werden. Wird `-S` spezifiziert, wird keine Meldung ausgegeben.

DATEIEN

`/etc/mail/binarsys`

`/usr/lib/mail/surrcmd/ckbinarsys`

SIEHE AUCH

`mailsurr(4)`, `binarsys(4)`

`mail(1)`, `uux(1)` in den *Kommandos*.

BEZEICHNUNG

ckbupscd – überprüft den Sicherungsplan des Dateisystems

ÜBERSICHT

ckbupscd [-m]

BESCHREIBUNG

ckbupscd durchsucht die Datei /etc/bupsched nach Zeilen, die ein Datum oder eine Uhrzeit enthalten, das bzw. die zur aktuellen Zeit paßt, und gibt die dazugehörenden Dateisystemlisten aus. Ist Option -m angegeben, wird die Anfangsmeldung in der Ausgabe unterdrückt, so daß nur die Dateisystemlisten selbst ausgegeben werden. Die Einträge in der Datei bupsched werden, gesteuert von cron, ausgegeben.

Die Datei bupsched enthält Zeilen mit vier oder mehr Feldern, die durch Leerzeichen oder Tabulatorzeichen voneinander getrennt sind. Die ersten drei Felder, die Zeitplanfelder, geben einen Bereich von Daten und Zeiten an. Die übrigen Felder enthalten eine Liste von Dateisystemnamen, die ausgegeben werden soll, wenn ckbupscd zu einem Zeitpunkt innerhalb des in den Zeitplanfeldern angegebenen Zeitraums ausgeführt wird. Das allgemeine Format lautet:

zeit[,zeit] tag[,tag] monat[,monat] dsysliste

Dabei steht

- zeit* für die Uhrzeit (0 Uhr bis 23 Uhr), wobei eine beliebige Zeit innerhalb der angegebenen Stunde gemeint sein oder ein genauer Zeitpunkt (0:00 Uhr bis 23:59 Uhr) angegeben werden kann.
- tag* für den Wochentag, nämlich sun (Sonntag) bis sat (Samstag), oder den Tag des Monats (1 bis 31).
- monat* für den Monat, für den die Felder "zeit" und "tag" gelten. Gültige Werte sind die Zahlen von 1 bis 12.
- dsysliste* Der Rest der Zeile wird als Dateisystemliste aufgefaßt, die ausgegeben werden soll.

Mehrere Uhrzeit-, Tages- und Monatsangaben können, durch Komma voneinander getrennt, angegeben werden. In diesem Falle werden sie von links nach rechts gelesen und in dieser Reihenfolge ausgewertet.

Ein Stern (*) entspricht immer dem aktuellen Wert für das Feld, in dem er steht.

Eine Zeile, die mit dem Zeichen # beginnt, gilt als Kommentar und wird ignoriert.

Die größte erlaubte Zeilenlänge (einschließlich Fortsetzungszeilen) beträgt 1024 Zeichen.

BEISPIELE

Es folgen Beispiele für Zeilen, wie sie in der Datei `/etc/bupscd` enthalten sein könnten.

```
06:00-09:00 fri 1,2,3,4,5,6,7,8,9,10,11 /applic
```

Der Dateisystemname `/applic` wird ausgegeben, wenn `ckbupscd` in irgendeinem Monat außer im Dezember freitags zwischen 6:00 und 9:00 Uhr ausgeführt wird.

```
00:00-06:00,16:00-23:59 1,2,3,4,5,6,7 1,8 /
```

Eine Meldung wird ausgegeben, daß eine Sicherung des Dateisystems `/` (root) erstellt werden sollte, wenn `ckbupscd` in der ersten August- oder Januarwoche zwischen 16:00 und 6:00 Uhr ausgeführt wird.

DATEIEN

`/etc/bupscd` Spezifikationsdatei mit den Dateisystemen, von denen Sicherungen erstellt werden sollen, und den Zeiten, zu denen dies geschehen soll.

SIEHE AUCH

`cron(1M)`.

`echo(1)`, `sh(1)`, `sysadm(1)` in den *Kommandos*.

HINWEIS

Das Kommando `ckbupscd` meldet jedesmal, wenn es aufgerufen wird, welche Dateisysteme zur Sicherung anstehen. Wenn kurz zuvor Sicherungen durchgeführt wurden, wird dies nicht berücksichtigt.

`ckbupscd` ist in der nächsten SINIX-Version nicht mehr enthalten.

BEZEICHNUNG

ckdate, errdate, helpdate, valdate – fordert zur Eingabe eines Datums auf und überprüft die Eingabe auf ihre Gültigkeit

ÜBERSICHT

ckdate [-Q] [-w *länge*] [-f *format*] [-d *standard*] [-h *hilfe*] [-e *fehler*] [-p *prompt*]
 [-k *pid* [-s *signal*]]

errdate [-w] [-e *fehler*] [-f *format*]

helpdate [-w] [-h *hilfe*] [-f *format*]

valdate [-f *format*] *eingabe*

BESCHREIBUNG

ckdate fordert den Benutzer zu einer Eingabe auf und überprüft die Gültigkeit der Eingabe. Mit diesem Befehl wird ein Prompt definiert, auf das hin ein Datum eingegeben werden muß. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER drückt. Die Eingabe des Benutzers muß dem definierten Datumsformat entsprechen.

Alle Meldungen sind auf eine Länge von 70 Zeichen begrenzt und werden automatisch formatiert. Leerzeichen, Tabulatoren, auch das Neue-Zeile-Zeichen in der Definition werden entfernt. Mit der Option -w wird die automatische Formatierung unterdrückt. Wird am Anfang oder am Ende einer Meldungsdefinition eine Tilde eingefügt, erscheint an dieser Stelle der Text der Standardmeldung. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter "HINWEIS" angegeben.

Drei Menü-Module sind mit dem Kommando ckdate verbunden, nämlich errdate zur Formatierung und Anzeige einer Fehlermeldung, helpdate zur Formatierung und Anzeige einer Hilfmeldung und valdate zur Überprüfung der Benutzereingabe. Diese Module sollten zusammen mit FML-Objekten verwendet werden. In diesem Fall definiert das FML-Objekt das Prompt. Wenn format in den Modulen errdate und helpdate definiert wird, beschreiben die Meldungen das erwartete Format.

Für diesen Befehl gibt es folgende Optionen und Argumente:

- Q Gibt an, daß Quit nicht als gültige Antwort gilt.
- w *länge* Gibt an, daß Prompt-, Hilfe- und Fehlermeldungen auf die Zeilenlänge *länge* formatiert werden.
- f *format* Gibt das Format an, auf das hin die Eingabe überprüft wird. Es folgen die Formate und ihre Definitionen:
 - %b = abgekürzter Monatsname
 - %B = vollständiger Monatsname
 - %d = Tag des Monats (01 - 31)
 - %D = Datum in der Form %m/%t/%j (Standardformat)

- %e = Tag des Monats (1 - 31; einfachen Ziffern geht ein Leerzeichen voraus)
- %h = abgekürzter Monatsname (jan, feb, mar)
- %m = Monatszahl (01 - 12)
- %y = zwei letzten Ziffern der Jahreszahl (z.B. 89)
- %Y = vollständige Jahreszahl in der Form JJJJ (z.B. 1989)
- d *standard* *standard* wird als Standardwert definiert. Der Standardwert braucht den Formatkriterien nicht zu entsprechen.
- h *hilfe* *hilfe* erscheint als Hilfemeldung.
- e *fehler* *fehler* erscheint als Fehlermeldung.
- p *prompt* *prompt* erscheint als Prompt.
- k *pid* Sendet an den Prozeß mit der Nummer *pid* ein Signal, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s *signal* Sendet an den Prozeß mit der Nummer *pid* das Signal *signal*, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definiert wurde. Ist kein Signal angegeben, wird SIGTERM benutzt.
- eingabe* Eingabe, die auf die Übereinstimmung mit den Formatkriterien hin überprüft werden soll.

EXIT-CODES

Die Exit-Codes von ckdate haben folgende Bedeutung:

- 0 = Ausführung erfolgreich
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)
- 4 = Verstümmelte Formatangabe

HINWEIS

Das standardmäßige Prompt von ckdate lautet:

```
Enter the date [?,q]:
(Datum eingeben [?,q]:)
```

Die standardmäßige Fehlermeldung lautet:

```
ERROR - Please enter a date, using the following format:
<format>..
(Fehler - Bitte Datum im folgenden Format eingeben:
<format>..)
```

Die standardmäßige Hilfemeldung lautet:

Please enter a date, using the following format: *<format>*.
(Bitte Datum im folgenden Format eingeben: *<format>*.)

Wird die Option Quit, sofern sie erlaubt ist, gewählt, wird *q* zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul *valdate* erzeugt keine Ausgabe. Es gibt eine Null zurück, wenn *eingabe* den Prüfkriterien entsprach, einen anderen Wert, wenn *eingabe* den Kriterien nicht entsprach.

BEZEICHNUNG

ckgid, errgid, helpgid, valgid – fordert zur Eingabe der Kennung einer Gruppe auf und überprüft deren Gültigkeit

ÜBERSICHT

ckgid [-Q] [-w *länge*] [-m] [-d *standard*] [-h *hilfe*] [-e *fehler*] [-p *prompt*]
 [-k *pid* [-s *signal*]]

errgid [-w] [-e *fehler*]
 helpgid [-w] [-m] [-h *hilfe*]
 valgid *eingabe*

BESCHREIBUNG

ckgid fordert den Benutzer zu einer Eingabe auf und überprüft die Eingabe auf ihre Gültigkeit. Mit diesem Kommando wird eine Prompt-Meldung definiert, auf die hin eine vorhandene Gruppennummer eingegeben werden muß. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt.

Meldungen dürfen höchstens 70 Zeichen lang sein und werden automatisch formatiert. Leerzeichen, Tabulatoren und Neue-Zeile-Zeichen in der Definition werden entfernt. Die Option -w unterdrückt die automatische Formatierung. Beginnt oder endet die Definition einer Meldung mit einer Tilde, wird an dieser Stelle der Text der Standardmeldung eingesetzt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter "HINWEIS" angegeben.

Drei Menü-Module sind mit dem Kommando ckgid verbunden: errgid zur Formatierung und Anzeige einer Fehlermeldung, helpgid zur Formatierung und Anzeige einer Hilfmeldung und valgid zur Überprüfung der Benutzereingabe. Diese Module sollten zusammen mit FML-Objekten benutzt werden. In diesem Fall definiert das FML-Objekt das Prompt.

Zu diesem Kommando gibt es folgende Optionen und Argumente:

- Q Gibt an, daß Quit nicht als gültige Antwort erlaubt ist.
- w *länge* Gibt an, daß Prompt-, Hilfe- und Fehlermeldungen auf eine Zeilenlänge von *länge* formatiert werden.
- m Zeigt eine Liste aller Gruppen an, wenn Hilfe angefordert wird oder der Benutzer einen Fehler macht.
- d *standard* Definiert *standard* als den Standardwert. Der Standardwert wird nicht auf seine Gültigkeit hin überprüft und braucht daher keinen Kriterien zu entsprechen.

- `-h hilfe` Definiert *hilfe* als die Hilfemeldung.
- `-e fehler` Definiert *fehler* als die Fehlermeldung.
- `-p prompt` Definiert *prompt* als die Eingabeaufforderung (Prompt).
- `-k pid` Gibt an, daß an den Prozeß mit der Nummer *pid* ein Signal gesendet werden soll, wenn der Benutzer die Ausführung des Kommandos abbricht.
- `-s signal` Gibt an, daß an den Prozeß mit der Nummer *pid* das Signal *signal* gesendet werden soll, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option `-k` definiert wurde. Wird kein Signal angegeben, wird `SIGTERM` benutzt.
- eingabe* Ist die Eingabe, die anhand von `/etc/group` überprüft wird.

EXIT-CODES

Die Exit-Codes von `ckgid` haben folgende Bedeutung:

- 0 = Ausführung erfolgreich
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)

HINWEIS

Das standardmäßige Prompt für `ckgid` lautet:

```
Enter the name of an existing group [?,q]:
(Name einer vorhandenen Gruppe eingeben [?,q]:)
```

Die Standardfehlermeldung lautet:

```
ERROR - Please enter the name of an existing group.
(FEHLER - Bitte den Namen einer vorhandenen Gruppe eingeben.)
```

Wird die Option `-m` des Kommandos `ckgid` benutzt, wird hier eine Liste der gültigen Gruppen angezeigt.

Die Standardhilmemeldung lautet:

```
Please enter an existing group name.
(Bitte den Namen einer vorhandenen Gruppe eingeben.)
```

Wird die Option `-m` des Kommandos `ckgid` benutzt, wird hier eine Liste der gültigen Gruppen angezeigt.

Ist die Option `Quit` erlaubt und wird sie gewählt, wird `q` zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul `valgid` erzeugt keine Ausgabe. Es gibt eine Null zurück, wenn *eingabe* den Prüfkriterien entsprach, einen anderen Wert, wenn *eingabe* den Kriterien nicht entsprach.

BEZEICHNUNG

`ckint`, `errint`, `helpint`, `valint` – gibt ein Prompt aus, überprüft die Eingabe des Benutzers und gibt eine ganze Zahl zurück

ÜBERSICHT

`ckint [-Q] [-W länge] [-b basis] [-d standard] [-h hilfe] [-e fehler] [-p prompt]
[-k pid [-s signal]]`

`errint [-W] [-b basis] [-e fehler]`
`helpint [-W] [-b basis] [-h hilfe]`
`valint [-b basis] eingabe`

BESCHREIBUNG

`ckint` fordert den Benutzer zu einer Eingabe auf und überprüft dann die Eingabe auf ihre Gültigkeit. Mit diesem Kommando wird ein Prompt definiert, auf das hin eine ganze Zahl eingegeben werden muß. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt.

Meldungen dürfen höchstens 70 Zeichen lang sein und werden automatisch formatiert. Leerzeichen, Tabulatoren und Neue-Zeile-Zeichen in der Definition werden entfernt. Die Option `-W` unterdrückt die automatische Formatierung. Beginnt oder endet die Definition einer Meldung mit einer Tilde, wird an dieser Stelle der Text der Standardmeldung eingesetzt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter "HINWEIS" angegeben.

Drei Menü-Module sind mit dem Kommando `ckint` verbunden, nämlich `errint` zur Formatierung und Anzeige einer Fehlermeldung, `helpint` zur Formatierung und Anzeige einer Hilfmeldung und `valint` zur Überprüfung der Benutzereingabe. Diese Module sollten zusammen mit FML-Objekten benutzt werden. In diesem Fall definiert das FML-Objekt das Prompt. Wird *basis* in den Modulen `errint` und `helpint` definiert, enthalten die Meldungen die erwartete Basis für die Eingabe.

Zu diesem Kommando gibt es folgende Optionen und Argumente:

- `-Q` Gibt an, daß Quit nicht als gültige Antwort erlaubt ist.
- `-W` Gibt an, daß das Prompt, die Hilfe- und Fehlermeldung auf eine Zeilenlänge von *länge* formatiert werden.
- `-b` Definiert die Basis für die Eingabe. Der Wert muß zwischen 2 und 36 liegen. Der Standardwert ist 10.
- `-d` Definiert *standard* als den Standardwert. Der Standardwert wird nicht auf seine Gültigkeit hin überprüft und braucht daher keinen Kriterien zu entsprechen.
- `-h` Definiert *hilfe* als die Hilfmeldung.

ckint(1)

ckint(1)

- e Definiert *fehler* als die Fehlermeldung.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß an den Prozeß mit der Nummer *pid* ein Signal gesendet werden soll, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s Gibt an, daß an den Prozeß mit der Nummer *pid* das Signal *signal* gesendet werden soll, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definiert wurde. Wird kein Signal angegeben, wird SIGTERM benutzt.

eingabe

ist die Eingabe, die anhand von *basis* überprüft wird.

EXIT-CODES

Die Exit-Codes von *ckint* haben folgende Bedeutung:

- 0 = Ausführung erfolgreich
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)

HINWEIS

Das standardmäßige Prompt des Kommandos *ckint* für die Basis 10 lautet:

```
Enter an integer [?,q]:  
(Ganze Zahl eingeben [?,q]:)
```

Die Standardfehlermeldung für die Basis 10 lautet:

```
ERROR - Please enter an integer.  
(FEHLER - Bitte eine ganze Zahl eingeben.)
```

Die Standardhilfemeldung für die Basis 10 lautet:

```
Please enter an integer.  
(Bitte eine ganze Zahl eingeben.)
```

Statt "integer" erscheint in den Meldungen "base *basis* integer", wenn als Basis eine andere Zahl als 10 gesetzt wird.

Ist die Option Quit erlaubt und wird sie gewählt, wird *q* zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul *valint* erzeugt keine Ausgabe. Es gibt eine Null zurück, wenn die *eingabe* den Prüfkriterien entsprach, einen anderen Wert, wenn sie den Kriterien nicht entsprach.

BEZEICHNUNG

ckitem, erritem, helpitem – generiert ein Menü; fordert zur Eingabe einer Menüoption auf und gibt sie zurück

ÜBERSICHT

```
ckitem [-Q] [-W länge] [-uno] [-f datei] [-l titel] [[-i verdeckt] , ...] [-m max]
[-d standard] [-h hilfe] [-e fehler] [-p prompt] [-k pid [-s signal]] [auswahl ...]
```

```
erritem [-W] [-e fehler] [auswahl ...]
```

```
helpitem [-W] [-h hilfe] [auswahl [...]]
```

BESCHREIBUNG

ckitem generiert ein Menü und fordert den Benutzer dazu auf, aus dem Menü eine Option auszuwählen. Dann überprüft das Kommando die Eingabe des Benutzers auf ihre Gültigkeit. Mit diesem Kommando wird ein Prompt definiert, auf das hin eine Menüoption eingegeben werden muß. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt.

Standardmäßig wird das Menü so formatiert, daß jeder Option eine Nummer vorausgeht und daß die Optionen in Spalten auf dem Bildschirm erscheinen. Die Spaltenbreite richtet sich nach der längsten Option. Die Optionen werden alphabetisch angeordnet.

Meldungen dürfen höchstens 70 Zeichen lang sein und werden automatisch formatiert. Leerzeichen, Tabulatoren, auch das Neue-Zeile-Zeichen in der Definition werden entfernt. Die Option -w unterdrückt die automatische Formatierung. Beginnt oder endet die Definition einer Meldung mit einer Tilde, wird an dieser Stelle der Text der Standardmeldung eingesetzt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter "HINWEIS" angegeben.

Zwei Menü-Module sind mit dem Kommando ckitem verbunden, nämlich erritem zur Formatierung und Anzeige einer Fehlermeldung und helpitem zur Formatierung und Anzeige einer Hilfmeldung. Diese Module sollten zusammen mit FML-Objekten benutzt werden. In diesem Fall definiert das FML-Objekt das Prompt. Wenn in diesen Modulen *auswahl* definiert wird, beschreiben die Meldungen die verfügbare Menüoption bzw. die verfügbaren Menüoptionen.

Zu diesem Kommando gibt es folgende Optionen und Argumente:

- Q Gibt an, daß Quit nicht als gültige Antwort erlaubt ist.
- W Gibt an, daß das Prompt, die Hilfe- und Fehlermeldung auf eine Zeilenlänge von *länge* formatiert werden.
- u Gibt an, daß die Menüoptionen als Liste ohne Numerierung angezeigt werden sollen.
- n Gibt an, daß die Menüoptionen nicht in alphabetischer Reihenfolge angezeigt werden sollen.

- o Gibt an, daß nur ein Menü-Token zurückgegeben wird.
- f Definiert eine Datei mit dem Namen *datei*, die eine Liste von Menüoptionen enthält, die angezeigt werden sollen. Das Format dieser Datei lautet `token<tabulator>beschreibung`. Zeilen, die mit dem Zeichen # beginnen, gelten als Kommentar und werden ignoriert.
- l Definiert einen *titel*, der über dem Menü ausgegeben wird.
- i Definiert verdeckte Menüoptionen, die auf dem Menü nicht angezeigt werden. Wenn beispielsweise "alle" als verdeckte Menüoption benutzt wird, so bedeutet dies, daß "alle" zwar eine gültige Option ist, aber nicht auf dem Menü erscheint. Es können beliebig viele verdeckte Optionen definiert werden. Auf verdeckte Optionen sollte der Benutzer entweder über das Prompt oder in einer Hilfmeldung aufmerksam gemacht werden.
- m Gibt an, wieviele Menüoptionen höchstens erlaubt sind.
- d Definiert *standard* als den Standardwert. Der Standardwert wird nicht auf seine Gültigkeit überprüft und braucht daher keinen Kriterien zu entsprechen.
- h Definiert *hilfe* als die Hilfmeldung.
- e Definiert *fehler* als die Fehlermeldung.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß an den Prozeß mit der Nummer *pid* ein Signal gesendet werden soll, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s Gibt an, daß an den Prozeß mit der Nummer *pid* das Signal *signal* gesendet werden soll, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definiert wurde. Wird kein Signal angegeben, wird SIGTERM benutzt.

auswahl

Definiert die Menüoptionen. Die Menüoptionen sollten durch Leerzeichen, Tabulatoren oder Neue-Zeile-Zeichen voneinander getrennt werden.

SIEHE AUCH

`allocmenu(3X)`, `printmenu(3X)`, `setinvis(3X)`, `setitems(3X)`.

EXIT-CODES

Die Exit-Codes von `ckitem` haben folgende Bedeutung:

- 0 = Ausführung erfolgreich
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)
- 4 = Keine Optionen zur Auswahl

HINWEIS

Der Benutzer kann die Nummer der gewünschten Menüoption eingeben, wenn die Optionen numeriert sind, oder er kann so viel von der Menüoption eingeben, daß die Menüoption durch die eingegebene Zeichenkette eindeutig bestimmt wird. Lange Menüs sind auf mehrere Seiten verteilt, wobei jede Seite 10 Menüoptionen enthält.

Wenn Menüoptionen in einer Datei (über die Option `-f`) und auf der Kommandozeile definiert werden, werden sie in der Regel in alphabetischer Reihenfolge angeordnet. Wird jedoch mit der Option `-n` die alphabetische Sortierung unterdrückt, dann werden die in der Datei definierten Optionen zuerst und die auf der Kommandozeile definierten Optionen dahinter angeordnet.

Das standardmäßige Prompt für `ckitem` lautet:

```
Enter selection [?,??,q]:
(Auswahl eingeben [?,??,q]:)
```

Gibt man ein Fragezeichen ein, erscheint eine Hilfmeldung und dann wieder das Prompt. Gibt man zwei Fragezeichen ein, erscheint eine Hilfmeldung, dann wieder der Titel des Menüs, das Menü und das Prompt.

Die Standardfehlermeldung lautet:

```
ERROR - Does not match an available menu selection.
Enter one of the following:
- the number of the menu item you wish to select
- the token associated with the menu item,
- partial string which uniquely identifies the token for the
  menu item
- ?? to reprint the menu
(FEHLER - Entspricht keiner verfügbaren Menüoption.
Sie haben folgende Möglichkeiten:
- Geben Sie die Nummer der gewünschten Menüoption ein.
- Geben Sie das Token zu der Menüoption ein.
- Geben Sie eine Zeichenkette ein, die das Token der Option
  eindeutig bestimmt.
- Geben Sie ?? ein, damit das Menü nochmals angezeigt wird.)
```

Die Standardhilfmeldung lautet:

```
Enter one of the following:
- the number of the menu item you wish to select
- the token associated with the menu item,
- partial string which uniquely identifies the token for the
  menu item
- ?? to reprint the menu
(Sie haben folgende Möglichkeiten:
- Geben Sie die Nummer der gewünschten Menüoption ein.
- Geben Sie das Token zur Menüoption ein.
- Geben Sie eine Zeichenkette ein, die das Token der Option
  eindeutig bestimmt.
- Geben Sie ?? ein, damit das Menü nochmals angezeigt wird.)
```

Ist die Option `Quit` erlaubt und wird sie gewählt, wird `q` zusammen mit dem Exit-Code 3 zurückgegeben.

BEZEICHNUNG

ckkeywd – fordert zur Eingabe eines Schlüsselwortes auf und überprüft es

ÜBERSICHT

ckkeywd [-Q] [-w *länge*] [-d *standard*] [-h *hilfe*] [-e *fehler*] [-p *prompt*]
[-k *pid* [-s *signal*]] [*schlüsselwort* ...]

BESCHREIBUNG

ckkeywd fordert den Benutzer zu einer Eingabe auf und überprüft diese. Mit diesem Kommando wird ein Prompt für eines der Schlüsselwörter aus einer Liste definiert. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt. Die Antwort auf dieses Kommando muß einem der Schlüsselwörter aus einer Liste entsprechen.

Alle Meldungen dürfen maximal 70 Zeichen lang sein und werden automatisch formatiert. Alle in der Definition verwendeten Leerzeichen, Tabulatoren und Neue-Zeile-Zeichen werden entfernt. Mit der Option -w wird die automatische Formattierung ausgeschaltet. Steht am Anfang oder am Ende einer Meldungsdefinition eine Tilde, wird an dieser Stelle der Text der Standardmeldung eingefügt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter "HINWEIS" angegeben.

- Q Gibt an, daß das Verlassen des Kommandos mit Quit nicht zulässig ist.
- w Gibt an, daß das Prompt, die Hilfe- und die Fehlermeldungen auf eine Zeilenlänge von *länge* formatiert werden.
- d Definiert den Standardwert als *standard*. Der Standardwert wird nicht überprüft und muß daher keinerlei Kriterien entsprechen.
- h Definiert die Hilfemeldungen als *hilfe*.
- e Definiert die Fehlermeldung als *fehler*.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß an den Prozeß mit der Nummer *pid* ein Signal gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s Gibt an, daß an den Prozeß mit der Nummer *pid* ein Signal *signal* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definierte wurde. Ist kein Signal angegeben, wird SIGTERM verwendet.

schlüsselwort

Definiert das Schlüsselwort bzw. die Liste der Schlüsselwörter, anhand dessen/derer die Antwort überprüft wird.

EXIT-CODES

Die Exit-Codes von ckkeywd haben folgende Bedeutung:

- 0 = Erfolgreiche Ausführung
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)
- 4 = Keine Schlüsselwörter zur Auswahl

HINWEIS

Das standardmäßige Prompt für ckkeywd lautet:

```
Enter selection [schlüsselwort, [...], ?, q]:  
(Auswahl eingeben [schlüsselwort, [...], ?, q]:)
```

Die Standardfehlermeldung lautet:

```
ERROR - Does not match any of the valid selections.  
Please enter one of the following keywords:  
keyword[,...]  
(FEHLER - Entspricht keiner gültigen Auswahl.  
Eines der folgenden Schlüsselwörter eingeben:  
schlüsselwort[,...])
```

Die Standardhilfemeldung lautet:

```
Please enter one of the following keywords:  
keyword[,...]  
(Bitte eines der folgenden Schlüsselwörter eingeben:  
schlüsselwort[,...])
```

Wird die Option Quit ausgewählt und ist sie zulässig, wird q zusammen mit dem Exit-Code 3 zurückgegeben.

ckpath(1)

ckpath(1)

BEZEICHNUNG

ckpath, errpath, helppath, valpath – zeigt ein Prompt an; überprüft einen Pfadnamen und gibt ihn zurück

ÜBERSICHT

ckpath [-Q] [-w *länge*] [-a|l] [-b|c|g|y] [-n|[o|z]] [-rtwx] [-d *standard*]
[-h *hilfe*] [-e *fehler*] [-p *prompt*] [-k *pid*] [-s *signal*]

errpath [-w] [-a|l] [-b|c|g|y] [-n|[o|z]] [-rtwx] [-e *fehler*]
helppath [-w] [-a|l] [-b|c|g|y] [-n|[o|z]] [-rtwx] [-h *hilfe*]
valpath [-a|l] [-b|c|g|y] [-n|[o|z]] [-rtwx] *eingabe*

BESCHREIBUNG

ckpath gibt ein Prompt an den Benutzer aus und überprüft die Antwort. Mit diesem Kommando wird ein Prompt definiert, auf das mit einem Pfadnamen geantwortet werden soll. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt.

Der Pfadname muß den Kriterien entsprechen, die mit der ersten Gruppe von Optionen festgelegt wurden. Sind keine Kriterien angegeben, muß es sich um den Pfadnamen einer normalen Datei handeln, die noch nicht besteht. Ist weder -a (absolut) noch -l (relativ) angegeben, werden beide als gültig angenommen.

Alle Meldungen dürfen maximal 70 Zeichen lang sein und werden automatisch formatiert. In der Definition verwendete Leerzeichen, Tabulatoren und Neue-Zeile-Zeichen werden entfernt. Mit der Option -w wird die automatische Formatierung ausgeschaltet. Steht am Anfang oder am Ende einer Meldungsdefinition eine Tilde, wird an dieser Stelle der Text der Standardmeldung eingefügt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter HINWEIS angegeben.

Mit dem Kommando ckpath sind drei Menü-Module verbunden, und zwar errpath (formatiert eine Fehlermeldung und zeigt sie an), helppath (formatiert eine Hilfemeldung und zeigt sie an) und valpath (überprüft die Antwort). Diese Module sollten Sie zusammen mit FACE-Objekten benutzen. In diesem Falle definiert das FACE-Objekt das Prompt.

Für dieses Kommando stehen folgende Optionen und Argumente zur Verfügung:

- Q Gibt an, daß Quit keine gültige Antwort ist.
- w Bestimmt, daß das Prompt, die Hilfe- und Fehlermeldungen mit der Länge *länge* formatiert werden.
- a Der Pfadname muß einen absoluten Pfad bezeichnen.
- l Der Pfadname muß einen relativen Pfad bezeichnen.
- b Der Pfadname muß eine blockorientierte Gerätedatei bezeichnen.
- c Der Pfadname muß eine zeichenorientierte Gerätedatei bezeichnen.

ckpath(1)

ckpath(1)

- g Der Pfadname muß eine reguläre Datei bezeichnen.
- y Der Pfadname muß ein Verzeichnis bezeichnen.
- n Der Pfadname darf nicht existieren (muß neu sein).
- o Der Pfadname muß existieren (muß alt sein).
- z Der Pfadname muß länger als 0 Byte sein.
- r Der Pfadname muß lesbar sein.
- t Der Pfadname muß erstellbar sein. Der Pfadname wird angelegt, sofern er noch nicht vorhanden ist.
- w Der Pfadname muß schreibbar sein.
- x Der Pfadname muß ausführbar sein.
- d Definiert den Standardwert als *standard*. Der Standardwert wird nicht überprüft und muß daher keinem Kriterium entsprechen.
- h Definiert die Hilfmeldungen als *hilfe*.
- e Definiert die Fehlermeldungen als *fehler*.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß ein Signal an den Prozeß mit der Nummer *pid* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s Bestimmt, daß an den Prozeß mit der Nummer *pid* das Signal *signal* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definierte wurde. Wird kein Signal angegeben, wird `SIGTERM` verwendet.

eingabe

Eingabe, die anhand der Prüfoptionen überprüft wird.

EXIT-CODES

Die Exit-Codes von `ckpath` haben folgende Bedeutung:

- 0 = Erfolgreiche Ausführung
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)
- 4 = Sich gegenseitig ausschließende Optionen

HINWEIS

Der Text der Standardmeldungen für `ckpath` hängt von den verwendeten Optionen für die einzelnen Kriterien ab. Ein Beispiel für das voreingestellte Prompt von `ckpath` (mit der Option -a):

```
Enter a pathname [?,q]:  
(Pfadnamen eingeben [?,q]:)
```

Ein Beispiel für eine Standard-Fehlermeldung (mit der Option -a) lautet:

```
ERROR - Invalid pathname entered.  
A pathname is a filename, optionally preceded by parent direc-  
tories.  
(FEHLER - Pfadname ungültig.  
Ein Pfadname ist ein Dateiname, dem optional Väterverzeichnis  
vorausgehen.)
```

Ein Beispiel für eine Standard-Hilfemeldung lautet:

```
A pathname is a filename, optionally preceded by parent direc-  
tories. The pathname you enter:  
- must contain 1 to {NAME_MAX} characters  
- must not contain a spaces or special characters  
(Ein Pfadname ist ein Dateiname, dem optional Väterverzeich-  
nisse vorausgehen.  
Der eingegebene Pfadname  
- muß 1 bis {NAME_MAX} Zeichen enthalten  
- darf keine Leerzeichen oder Sonderzeichen enthalten)
```

NAME_MAX ist eine Systemvariable, die in `limits.h` definiert ist.

Wird die Option Quit ausgewählt (und ist sie zulässig), wird `q` zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul `valpath` erzeugt keinerlei Ausgabe. Es gibt eine Null zurück, wenn die *eingabe* den Prüfkriterien entsprach, einen anderen Wert, wenn sie den Prüfkriterien nicht entsprach.

BEZEICHNUNG

ckrange, errrange, helprange, valrange – fordert zur Eingabe einer ganzen Zahl auf und überprüft diese

ÜBERSICHT

ckrange [-Q] [-w *länge*] [-l *untergrenze*] [-u *obergrenze*] [-b *basis*] [-d *standard*]
 [-h *hilfe*] [-e *fehler*] [-p *prompt*] [-k *pid*] [-s *signal*]

errange [-w] [-l *untergrenze*] [-u *obergrenze*] [-e *fehler*]
 helprange [-w] [-l *untergrenze*] [-u *obergrenze*] [-h *hilfe*]
 valrange [-l *untergrenze*] [-u *obergrenze*] [-b *basis*] *eingabe*

BESCHREIBUNG

ckrange fordert den Benutzer zu einer Eingabe auf und überprüft diese. Unter anderem wird damit ein Prompt definiert, auf das mit einer ganzen Zahl aus dem angegebenen Bereich geantwortet werden soll. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt.

Mit diesem Kommando wird auch ein Bereich für die Eingabe definiert. Wird die Unter- oder Obergrenze dieses Bereiches nicht definiert, ist der Bereich nur nach einer Seite begrenzt.

Alle Meldungen dürfen 70 Zeichen lang sein und werden automatisch formatiert. In der Definition verwendete Leerzeichen, Tabulatoren und Neue-Zeile-Zeichen werden entfernt. Mit der Option -w wird die automatische Formatierung ausgeschaltet. Steht am Anfang oder am Ende einer Meldungsdefinition eine Tilde, wird an dieser Stelle der Text der Standardmeldung eingefügt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter HINWEIS angegeben.

Mit dem Kommando ckrange sind drei Menü-Module verbunden, und zwar errange (formatiert eine Fehlermeldung und zeigt sie an), helprange (formatiert eine Hilfemeldung und zeigt sie an) und valrange (überprüft die Antwort). Diese Module sollten Sie zusammen mit FACE-Objekten benutzen. In diesem Falle definiert das FACE-Objekt das Prompt.

Für dieses Kommando stehen folgende Optionen und Argumente zur Verfügung:

- Q Gibt an, daß Quit keine gültige Antwort ist.
- w Bestimmt, daß das Prompt, die Hilfe- und Fehlermeldungen auf die Länge *länge* formatiert werden.
- l Definiert die Untergrenze des Bereichs als *untergrenze*. Der Standardwert ist die größte negative ganze Zahl bzw. Long Integer der Maschine.
- u Definiert die Obergrenze des Bereichs als *obergrenze*. Der Standardwert ist die größte positive ganze Zahl bzw. Long Integer der Maschine.
- b Definiert die Basis der Eingabe. Bereich 2 bis 36, der Standardwert ist 10.

- d Definiert den Standardwert als *standard*. Der Standardwert wird nicht überprüft und muß daher keinen Kriterien entsprechen.
- h Definiert die Hilfmeldungen als *hilfe*.
- e Definiert die Fehlermeldung als *fehler*.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß ein Signal an den Prozeß mit der Nummer *pid* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s Bestimmt, daß an den Prozeß mit der Nummer *pid* ein Signal *signal* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definiert wurde. Wird kein Signal angegeben, wird SIGTERM verwendet.

eingabe

Die Eingabe, die anhand der Ober- und Untergrenze und der Basis überprüft wird.

EXIT-CODES

Die Exit-Codes von *ckrange* haben folgende Bedeutung:

- 0 = Erfolgreiche Ausführung
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)

HINWEIS

Das standardmäßige Prompt für *ckrange* bei Basis 10 lautet:

Enter an integer between *untergrenze* and *obergrenze* [q,?]:
 (Eine ganze Zahl zwischen *untergrenze* und *obergrenze* eingeben [q,?]:)

Die Standard-Fehlermeldung bei Basis 10 lautet:

ERROR - Please enter an integer between
untergrenze and *obergrenze*.
 (FEHLER - Bitte eine ganze Zahl zwischen
untergrenze und *obergrenze* eingeben.)

Die Standard-Hilfmeldung bei Basis 10 lautet:

Please enter an integer between *untergrenze* and *obergrenze*.
 (Bitte eine ganze Zahl zwischen *untergrenze* und *obergrenze* eingeben.)

Die Meldungen ändern sich von "integer" zu "base *basis* integer", wenn die Basis mit einer anderen Zahl als 10 definiert wird.

Wird die Option Quit ausgewählt (und ist sie zulässig), wird q zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul *valrange* erzeugt keinerlei Ausgabe. Es gibt eine Null zurück, wenn die *eingabe* den Prüfkriterien entsprach, einen anderen Wert, wenn sie den Prüfkriterien nicht entsprach.

BEZEICHNUNG

ckstr, errstr, helpstr, valstr – gibt ein Prompt aus; überprüft eine eingegebene Zeichenkette und gibt sie zurück

ÜBERSICHT

ckstr [-Q] [-w *länge*] [[-r *regex*] [...]] [-l *lang*] [-d *standard*] [-h *hilfe*] [-e *fehler*]
[-p *prompt*] [-k *pid*] [-s *signal*]]

errstr [-w] [-e *fehler*]

helpstr [-w] [-h *hilfe*]

valstr *eingabe*

BESCHREIBUNG

ckstr fordert den Benutzer zu einer Eingabe auf und überprüft diese. Mit diesem Kommando wird ein Prompt definiert, auf das mit einer Zeichenkette geantwortet werden soll. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt.

Der zurückgegebene Wert muß dem definierten regulären Ausdruck entsprechen und darf die angegebene Länge nicht überschreiten. Ist kein regulärer Ausdruck angegeben, muß die Eingabe aus einer Zeichenkette bestehen, deren Länge geringer oder gleich dem angegebenen Wert ist und die ohne führende, interne oder nachfolgende Leerzeichen oder Tabulatoren definiert ist. Ist keine Länge definiert, wird dieser Wert nicht überprüft. Mit dem Kommando muß entweder ein regulärer Ausdruck oder ein Längenwert angegeben werden.

Alle Meldungen dürfen 70 Zeichen lang sein und werden automatisch formatiert. In der Definition verwendete Leerzeichen, Tabulatoren und Neue-Zeile-Zeichen werden entfernt. Mit der Option -w wird die automatische Formatierung ausgeschaltet. Steht am Anfang oder am Ende einer Meldungsdefinition eine Tilde, wird an dieser Stelle der Text der Standardmeldung eingefügt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter HINWEIS angegeben.

Mit dem Kommando ckstr sind drei Menü-Module verbunden, und zwar errstr (formatiert eine Fehlermeldung und zeigt sie an), helpstr (formatiert eine Hilfemeldung und zeigt sie an) und valstr (überprüft die Antwort). Diese Module sollten Sie zusammen mit FACE-Objekten benutzen. In diesem Falle definiert das FACE-Objekt das Prompt.

Für dieses Kommando stehen folgende Optionen und Argumente zur Verfügung:

- Q Gibt an, daß Quit keine gültige Antwort ist.
- w Bestimmt, daß das Prompt, die Hilfe- und Fehlermeldungen mit der Länge *länge* formatiert werden.
- r Gibt einen regulären Ausdruck *regex* an, anhand dessen die Eingabe überprüft werden muß. Dieser Ausdruck darf Leerzeichen und Tabulatoren enthalten. Werden mehrere Ausdrücke definiert, muß die Antwort nur einem davon entsprechen.

ckstr(1)

ckstr(1)

- l Gibt die maximale Länge der Eingabe an.
- d Definiert den Standardwert als *standard*. Der Standardwert wird nicht überprüft und muß daher keinem Kriterium entsprechen.
- h Definiert die Hilfmeldungen als *hilfe*.
- e Definiert die Fehlermeldungen als *fehler*.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß ein Signal an den Prozeß mit der Nummer *pid* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s Bestimmt, daß an den Prozeß mit der Nummer *pid* ein Signal *signal* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definiert wurde. Wird kein Signal angegeben, wird SIGTERM verwendet.

eingabe

Die Eingabe muß anhand der Kriterien Formatlänge und/oder regulärer Ausdruck überprüft werden.

EXIT-CODES

Die Exit-Codes von *ckstr* haben folgende Bedeutung:

- 0 = Erfolgreiche Ausführung
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)

HINWEIS

Das standardmäßige -Prompt bei *ckstr* lautet:

```
Enter an appropriate value [?,q]:  
(Geeigneten Wert eingeben [?,q]:)
```

Die Standard-Fehlermeldung hängt davon ab, welche Art von Überprüfung vorgenommen wird. Dem Benutzer wird mitgeteilt, daß entweder der Längen- oder der Mustervergleich fehlgeschlagen ist.

Die Standard-Hilfmeldung hängt ebenfalls vom Typ der Überprüfung ab. Wurde ein regulärer Ausdruck definiert, lautet die Meldung:

```
Please enter a string which matches the following pattern:  
regexp  
(Bitte eine Zeichenkette eingeben, die folgendem Muster  
entspricht:  
regexp)
```

In anderen Meldungen werden die erforderliche Länge und die Definition einer Zeichenkette definiert.

Wird die Option Quit ausgewählt (und ist sie zulässig), wird *q* zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul *valstr* erzeugt keinerlei Ausgabe. Es gibt eine Null zurück, wenn die *eingabe* den Prüfkriterien entsprach, einen anderen Wert, wenn sie den Prüfkriterien nicht entsprach.

BEZEICHNUNG

cktime, errtime, helptime, valtime – gibt ein Prompt aus; überprüft eine Uhrzeitangabe und gibt sie zurück

ÜBERSICHT

cktime [-Q] [-w *länge*] [-f *format*] [-d *standard*] [-h *hilfe*] [-e *fehler*] [-p *prompt*]
 [-k *pid*] [-s *signal*]

errtime [-w] [-e *fehler*] [-f *format*]

helptime [-w] [-h *hilfe*] [-f *format*]

valtime [-f *format*] *eingabe*

BESCHREIBUNG

cktime gibt ein Prompt aus und überprüft die Eingabe. Mit diesem Kommando wird ein Prompt definiert, auf das mit einer Uhrzeitangabe geantwortet werden soll. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt. Die Antwort des Benutzers muß dem definierten Uhrzeitformat entsprechen.

Alle Meldungen dürfen 70 Zeichen lang sein und werden automatisch formatiert. In der Definition verwendete Leerzeichen, Tabulatoren und Neue-Zeile-Zeichen werden entfernt. Mit der Option -w wird die automatische Formatierung ausgeschaltet. Steht am Anfang oder am Ende einer Meldungsdefinition eine Tilde, wird an dieser Stelle der Text der Standardmeldung eingefügt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter HINWEIS angegeben.

Mit dem Kommando cktime sind drei Menü-Module verbunden, und zwar errtime (formatiert eine Fehlermeldung und zeigt sie an), helptime (formatiert eine Hilfmeldung und zeigt sie an) und valtime (überprüft die Eingabe). Sie sollten diese Module zusammen mit FML-Objekten benutzen. Dabei bestimmt das FML-Objekt das Prompt. Wird *format* in den Modulen errtime und helptime definiert, beschreibt die Meldung das erwartete Format.

Für dieses Kommando stehen folgende Optionen und Argumente zur Verfügung:

- Q Gibt an, daß Quit keine gültige Antwort ist.
- w Bestimmt, daß das Prompt, die Hilfe- und Fehlermeldungen mit der Länge *länge* formatiert werden.
- f Gibt das Format an, anhand dessen die Eingabe überprüft wird. Folgende Formate sind möglich:
 - %H = Stunde (00 - 23)
 - %I = Stunde (00 - 12)
 - %M = Minute (00 - 59)
 - %p = Vormittag bzw. Nachmittag
 - %r = Zeitangabe %I:%M:%S %p
 - %R = Zeitangabe %H:%M (Standardformat)
 - %S = Sekunden (00 - 59)
 - %T = Zeitangabe %H:%M:%S

cktime(1)

cktime(1)

- d Definiert den Standardwert als *standard*. Der Standardwert wird nicht überprüft und muß daher keinem Kriterium entsprechen.
- h Definiert die Hilfmeldungen als *hilfe*.
- e Definiert die Fehlermeldung als *fehler*.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß ein Signal an den Prozeß mit der Nummer *pid* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s Bestimmt, daß an den Prozeß mit der Nummer *pid* ein Signal *signal* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definiert wurde. Wird kein Signal angegeben, wird SIGTERM verwendet.

eingabe

Die Eingabe, die anhand der Formatkriterien überprüft wird.

EXIT-CODES

Die Exit-Codes von `cktime` haben folgende Bedeutung:

- 0 = Erfolgreiche Ausführung
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (quit)
- 4 = Verstümmeltes Formatargument

HINWEIS

Das standardmäßige Prompt für `cktime` lautet:

```
Enter the time of day [?,q]:  
(Uhrzeit eingeben [?,q]:)
```

Die Standard-Fehlermeldung lautet:

```
ERROR - Please enter the time of day, using the following format:  
<format>  
(FEHLER - Bitte Uhrzeit im folgenden Format eingeben:  
<format>)
```

Die Standard-Hilfmeldung lautet:

```
Please enter the time of day, using the following format:  
<format>  
(Bitte Uhrzeit im folgenden Format eingeben:  
<format>)
```

Wird die Option Quit ausgewählt (und ist sie zulässig), wird `q` zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul `valtime` erzeugt keinerlei Ausgabe. Es gibt eine Null zurück, wenn die *eingabe* den Formatkriterien entsprach, einen anderen Wert, wenn sie den Formatkriterien nicht entsprach.

BEZEICHNUNG

ckuid, erruid, helpuid, valuid – fordert zur Eingabe einer Benutzerkennung auf und überprüft diese

ÜBERSICHT

ckuid [-Q] [-w *länge*] [-m] [-d *standard*] [-h *hilfe*] [-e *fehler*] [-p *prompt*]
 [-k *pid* [-s *signal*]]

erruid [-w] [-e *fehler*]
 helpuid [-w] [-m] [-h *hilfe*]
 valuid *eingabe*

BESCHREIBUNG

ckuid gibt ein Prompt aus und überprüft die Antwort. Mit diesem Kommando wird ein Prompt definiert, auf das mit einer existierenden Benutzerkennung geantwortet werden soll. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardwert definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt.

Alle Meldungen dürfen 70 Zeichen lang sein und werden automatisch formatiert. In der Definition verwendete Leerzeichen, Tabulatoren und Neuezeile-Zeichen werden entfernt. Mit der Option -w wird die automatische Formatierung ausgeschaltet. Steht am Anfang oder am Ende einer Meldungsdefinition eine Tilde, wird an dieser Stelle der Text der Standardmeldung eingefügt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter HINWEIS angegeben.

Mit dem Kommando ckuid sind drei Menü-Module verbunden, und zwar erruid (formatiert eine Fehlermeldung und zeigt sie an), helpuid (formatiert eine Hilfemeldung und zeigt sie an) und valuid (überprüft die Eingabe). Diese Module sollten Sie zusammen mit FML-Objekten benutzen. Dabei definiert das FML-Objekt das Prompt.

Für dieses Kommando stehen folgende Optionen und Argumente zur Verfügung:

- Q Gibt an, daß Quit keine gültige Antwort ist.
- w Bestimmt, daß das Prompt, die Hilfe- und Fehlermeldungen mit der Länge *länge* formatiert werden.
- m Zeigt eine Liste aller Benutzerkennungen an, wenn Hilfe angefordert wird oder der Benutzer einen Fehler macht.
- d Definiert *standard* als Standardwert. Der Standardwert wird nicht überprüft und muß daher keinem Kriterium entsprechen.
- h Definiert *hilfe* als die Hilfemeldung.
- e Definiert *fehler* als die Fehlermeldung.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß ein Signal an den Prozeß mit der Nummer *pid* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht.

ckuid(1)

ckuid(1)

-s Bestimmt, daß an den Prozeß mit der Nummer *pid* ein Signal *signal* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option **-k** definiert wurde. Wird kein Signal angegeben, wird **SIGTERM** verwendet.

eingabe

Eingabe, die anhand von `/etc/passwd` überprüft werden soll.

EXIT-CODES

Die Exit-Codes von `ckuid` haben folgende Bedeutung:

- 0 = Erfolgreiche Ausführung
- 1 = EOF (Dateiende) bei Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (Quit)

HINWEIS

Das standardmäßige Prompt bei `ckuid` lautet:

```
Enter the login name of an existing user [?,q]:  
(Eine vorhandene Benutzerkennung eingeben [?,q]:)
```

Die Standard-Fehlermeldung lautet:

```
ERROR - Please enter the login name of an existing user.  
Select the help option (?) for a list of valid login names.  
(Last line appears only if the -m option of ckuid is used)  
(FEHLER - Bitte eine vorhandene Benutzerkennung eingeben.  
Hilfoption (?) auswählen, damit eine Liste gültiger Benutzerkennungen angezeigt wird.  
(Die letzte Zeile erscheint nur, wenn die Option -m von ckuid angegeben wurde)
```

Die Standard-Hilfemeldung lautet:

```
Please enter the login name of an existing user.  
(If the -m option of ckuid is used, a list of valid groups is also displayed.)  
(Bitte eine vorhandene Benutzerkennung eingeben.  
(Wenn die Option -m von ckuid angegeben wurde, wird auch eine Liste der gültigen Gruppen angezeigt.)
```

Wird die Option **Quit** ausgewählt (und ist sie zulässig), wird *q* zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul `valuid` erzeugt keinerlei Ausgabe. Es gibt eine Null zurück, wenn die *eingabe* einem Eintrag in `/etc/passwd` entsprach, einen anderen Wert, wenn sie keinem Eintrag entsprach.

BEZEICHNUNG

ckyornd, erryornd, helpyornd, valyornd – fordert zur Eingabe von yes/no auf und überprüft die Antwort

ÜBERSICHT

ckyornd [-Q] [-w *länge*] [-d *standard*] [-h *hilfe*] [-e *fehler*] [-p *prompt*]
 [-k *pid*] [-s *signal*]

erryornd [-w] [-e *fehler*]
 helpyornd [-w] [-h *hilfe*]
 valyornd *eingabe*

BESCHREIBUNG

ckyornd gibt ein Prompt aus und überprüft die Antwort des Benutzers. Mit diesem Kommando wird ein Prompt definiert, auf das mit yes oder no geantwortet werden soll. Weiter werden Texte für Hilfe- und Fehlermeldungen sowie ein Standardtext definiert. Dieser wird als Eingabe verwendet, wenn der Benutzer nach dem Prompt nur die Taste ENTER betätigt.

Alle Meldungen dürfen maximal 70 Zeichen lang sein und werden automatisch formatiert. In der Definition verwendete Leerzeichen, Tabulatoren und Neue-Zeile-Zeichen werden entfernt. Mit der Option -w wird die automatische Formatierung ausgeschaltet. Steht am Anfang oder am Ende einer Meldungsdefinition eine Tilde, wird an dieser Stelle der Text der Standardmeldung eingefügt. Damit wird sowohl der selbst definierte als auch der Standardtext angezeigt.

Werden das Prompt, die Hilfe- oder Fehlermeldung nicht definiert, erscheint die Standardmeldung wie unter HINWEIS angegeben.

Mit dem Kommando ckyornd sind drei Menü-Module verbunden, und zwar erryornd (formatiert eine Fehlermeldung und zeigt sie an), helpyornd (formatiert eine Hilfemeldung und zeigt sie an) und valyornd (überprüft die Eingabe). Diese Module sollten Sie zusammen mit FACE-Objekten benutzen. In diesem Falle definiert das FACE-Objekt das Prompt.

Für dieses Kommando stehen folgende Optionen und Argumente zur Verfügung:

- Q Gibt an, daß Quit keine gültige Antwort ist.
- w Bestimmt, daß das Prompt, die Hilfe- und Fehlermeldungen mit der Länge *länge* formatiert werden.
- d Definiert *standard* als den Standardwert. Der Standardtext wird nicht überprüft und muß daher keinem Kriterium entsprechen.
- h Definiert *hilfe* als die Hilfemeldung.
- e Definiert *fehler* als die Fehlermeldung.
- p Definiert *prompt* als die Eingabeaufforderung.
- k Gibt an, daß ein Signal an den Prozeß mit der Nummer *pid* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht.
- s Bestimmt, daß an den Prozeß mit der Nummer *pid* ein Signal *signal* gesendet wird, wenn der Benutzer die Ausführung des Kommandos abbricht. *pid* ist die Prozeßnummer, die mit der Option -k definiert wurde. Wird kein Signal angegeben, wird SIGTERM verwendet.

eingabe

Die Eingabe, die y, yes, Y, Yes, YES oder n, no, N, No, NO lauten muß.

EXIT-CODES

Die Exit-Codes von ckyorn haben folgende Bedeutung:

- 0 = Erfolgreiche Ausführung
- 1 = EOF (Dateiende) der Eingabe
- 2 = Falsche Verwendung
- 3 = Abbruch durch Benutzer (Quit)

HINWEIS

Das standardmäßige Prompt bei ckyorn lautet:

```
Yes or No [y,n,?,q]:  
(Yes oder No [y,n,?,q]:)
```

Die Standard-Fehlermeldung lautet:

```
ERROR - Please enter yes or no.  
(FEHLER - Bitte yes oder no eingeben.)
```

Die Standard-Hilfemeldung lautet:

```
To respond in the affirmative, enter y, yes, Y, or YES.  
To respond in the negative, enter n, no, N, or NO.  
(Um mit Ja zu antworten, y, yes, Y oder YES eingeben.  
Um mit Nein zu antworten, n, no, N oder NO eingeben.)
```

Wird die Option Quit ausgewählt (und ist sie zulässig), wird q zusammen mit dem Exit-Code 3 zurückgegeben. Das Modul valyorn erzeugt keinerlei Ausgabe. Es gibt eine Null zurück, wenn die *eingabe* y, yes, Y, Yes, YES oder n, no, N, No, NO lautete. Ein anderer Wert wird zurückgegeben, wenn sie keiner dieser Möglichkeiten entsprach.

BEZEICHNUNG

colltbl – Anlegen einer Sortierdatenbank

ÜBERSICHT

colltbl [*datei* | -]

BESCHREIBUNG

Die Eingabe für das Kommando colltbl ist eine Spezifikationsdatei *datei*, die die Sortierfolge für eine bestimmte Sprache angibt und eine Datenbank anlegt, die von strxfrm(3C) und strcoll(3C) gelesen werden kann. strxfrm(3C) wandelt sein erstes Argument um und benutzt das Ergebnis als zweites Argument. Die umgewandelte Zeichenkette muß sich von strcmp(3C), strncmp(3C) oder memcmp(3C) zusammen mit anderen umgewandelten Zeichenketten richtig einordnen lassen. strcoll(3C) wandelt seine Argumente um und führt einen Vergleich durch.

Wird keine Datei angegeben, wird die Standardeingabe gelesen.

Die erzeugte Ausgabedatei enthält die Datenbank mit Angaben zur Sortierfolge, die von Systemkommandos und Routinen verwendet werden können. Den Namen dieser Ausgabedatei bestimmen Sie durch das Schlüsselwort *codeset*, dem Sie einen Wert zuordnen. Dieses Schlüsselwort wird aus der *datei* eingelesen. Bevor diese Datei verwendet werden kann, muß sie vom Systemverwalter oder einem Mitglied der Gruppe bin unter dem Namen LC_COLLATE in das Dateiverzeichnis /usr/lib/locale/*sprache* eingetragen werden. *sprache* entspricht der Sprache, deren Sortierfolge in *datei* beschrieben ist. Für diese Datei müssen Benutzer (user), Gruppe (group) und Andere (Other) Leseberechtigung haben; andere Zugriffsrechte sollen nicht vergeben werden. Damit die Daten zur Sortierfolge in dieser Datei verwendet werden können, muß die Umgebungsvariable LC_COLLATE entsprechend definiert sein (siehe environ(5) oder setlocale(3C)).

Das Kommando colltbl kann Sprachen unterstützen, deren Sortierfolge sich mit den folgenden Aussagen vollständig beschreiben läßt:

- Einordnen von Einzelzeichen innerhalb des Zeichensatzes. Im Schwedischen beispielsweise steht V nach U, vor X und zusammen mit W (beim Sortieren werden V und W als identisch angesehen).
- Einordnen von "Doppelbuchstaben" in der Sortierfolge. Im Spanischen werden ch und ll beispielsweise nach c bzw. l eingeordnet.
- Einordnen eines Einzelzeichens, als ob dieses aus zwei Zeichen bestünde. Im Deutschen z.B. wird das "scharfe s", ß, als ss sortiert. Dies ist ein Sonderfall zur nachfolgenden Aussage.
- Ersetzen einer Ein-Zeichen-Kette durch eine Zeichenkette. In dem obigen Beispiel wird die Zeichenkette ß beim Sortieren durch ss ersetzt.
- Ignorieren bestimmter Zeichen im Codesatz während des Sortierens. Wird z.B. – (Bindestrich) während des Sortierens ignoriert, sind die Zeichenketten re-locate und relocate gleich.
- Sekundärsortierfolge für die Zeichen. In Fällen, in denen zwei Zeichen in der Sortierfolge zusammen eingeordnet werden, d.h. dieselbe "Primärposition" haben, gibt es manchmal eine Sekundärsortierfolge. Diese wird verwendet, wenn zwei Zeichenketten, von den Zeichen mit derselben Primärposition

abgesehen, identisch sind. So haben im Französischen die Buchstaben *e* und *è* dieselbe Primärposition, aber nach der Sekundärsortierfolge steht *e* vor *è*. Daher würde das Wort *lever* vor *lèver*, *lèver* jedoch vor *levitate* sortiert. (Bitte beachten Sie folgendes: Stünde *e* in der Primärsortierfolge vor *è*, dann würde *lèver* nach *levitate* eingeordnet.)

Die Spezifikationsdatei besteht aus drei Arten von Anweisungen:

1. `codeset` *dateiname*

dateiname ist der Name der von colltbl zu erstellenden Ausgabedatei.

2. `order is` *sort_liste*

sort_liste ist eine durch Strichpunkte gegliederte Liste von Symbolen, die die Sortierfolge definiert. Das Sonderzeichen `...` gibt Symbole, die alphabetisch aufeinanderfolgen, in Kurzfassung an. Zum Beispiel stünde

```
order is  a;b;c;d;...;x;y;z
```

für die Liste der Kleinbuchstaben. Selbstverständlich ließe sich dies weiter auf `a;...;z` verkürzen.

Ein Symbol kann bis zu zwei Byte lang sein und auf eine der folgenden Arten dargestellt werden:

- als das Symbol selbst (z.B. `a` für das kleine `a`),
- als Oktalwert (z.B. `\141` oder `0141` für den Buchstaben `a`) oder
- als Hexadezimalwert (z.B. `\x61` oder `0x61` für den Buchstaben `a`).

Auch beliebige Kombinationen aus diesen Darstellungstypen sind möglich.

Mit dem Gegenschrägstrich `\` wird angegeben, daß die Eingabe in der nächsten Zeile fortgesetzt wird. Nach diesem Zeichen dürfen keine weiteren Zeichen folgen.

In Klammern angegebene Zeichen haben dieselbe Primärposition, aber eine andere Sekundärposition. Symbolen in geschweiften Klammern wurde nur die gleiche Primärposition in der Sortierfolge zugewiesen. Beispiel

```
order is  a;b;c;ch;d;(e;è);f;...;z;\
          {1;...;9};A;...;Z
```

In diesem Beispiel haben *e* und *è* dieselbe Primärposition und unterschiedliche Sekundärpositionen, für die Zahlen 1 bis 9 gilt dieselbe Primärsortierfolge, aber keine Sekundärsortierfolge. Für die restlichen Symbole gilt nur die Primärsortierfolge. Beachten Sie, wie Doppelbuchstaben in der Sortierfolge angegeben werden können (*ch* steht zwischen *c* und *d*).

Wird ein Zeichen nicht in die `order is`-Anweisung aufgenommen, steht es außerhalb der Sortierfolge und wird beim Sortieren ignoriert.

3. `substitute` *zeichenkette* with *ers*

Mit der Anweisung `substitute` wird die Zeichenkette *zeichenkette* durch die Zeichenkette *ers* ersetzt. Damit können z.B. Regeln definiert werden, um abgekürzte Monatsnamen nach der numerischen Reihenfolge zu sortieren:

```

substitute "Jan" with "01"
substitute "Feb" with "02"
.
.
.
substitute "Dez" with "12"

```

Eine einfachere Verwendung der Anweisung `substitute` ist die bereits erwähnte Ersetzung eines Zeichens durch zwei Zeichen, z.B. bei `ß` und `ss` im Deutschen.

Die Anweisung `substitute` ist optional. Die Anweisungen `order is` und `codeset` müssen in der Spezifikationsdatei erscheinen.

Alle mit einem `#` in der ersten Spalte markierten Zeilen in der Spezifikationsdatei werden als Kommentare behandelt und nicht interpretiert. Leerzeilen werden ebenfalls ignoriert.

BEISPIEL

Das folgende Beispiel zeigt die Sortierspezifikation, die z.B. für eine Sortierfolge für ein Telefonbuch benötigt wird.

Die Sortierfolge ist durch die folgenden Regeln definiert:

- Groß- und Kleinbuchstaben müssen zusammen sortiert werden, Großbuchstaben haben aber Priorität vor Kleinbuchstaben.
- Alle Sonderzeichen und Interpunktionszeichen sollen ignoriert werden.
- Ziffern sind wie ihre alphabetischen Entsprechungen zu sortieren (z.B. 0 als null, 1 als eins).
- Die Kombinationen `Ch`, `ch`, `CH` sind zwischen `C` und `D` einzuordnen.
- `V` und `W`, `v` und `w` sind zusammen einzuordnen.

Die Spezifikationsdatei für `colltbl` sieht so aus:

```

codeset      telephone

order is     A;a;B;b;C;c;CH;Ch;ch;D;d;E;e;F;f;\
              G;g;H;h;I;i;J;j;K;k;L;l;M;m;N;n;O;o;P;p;\
              Q;q;R;r;S;s;T;t;U;u; {V;W}; {v;w};X;x;Y;y;Z;z

substitute "0" with "null"
substitute "1" with "eins"
substitute "2" with "zwei"
substitute "3" with "drei"
substitute "4" with "vier"
substitute "5" with "fünf"
substitute "6" with "sechs"
substitute "7" with "sieben"
substitute "8" with "acht"
substitute "9" with "neun"

```


DATEIEN

/lib/locale/*sprache*/LC_COLLATE

LC_COLLATE-Datenbank für *sprache*

/usr/lib/locale/C/colltbl_C

Eingabedatei zum Aufbau von LC_COLLATE in der Standardsprache.

SIEHE AUCH

memory(3C), *setlocale(3C)*, *strcoll(3C)*, *string(3C)*, *strxfrm(3C)*, *environ(5)*
im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

comsat, in.comsat - biff-Server

ÜBERSICHT

in.comsat

BESCHREIBUNG

in.comsat ist ein Server-Prozeß, der Meldungen über ankommende Post abhört und Benutzer benachrichtigt, die dies wünschen. Der Prozeß wird bei Bedarf von inetd(1M) aufgerufen. Wenn er einige Minuten lang nicht aktiv ist, wird er beendet.

in.comsat lauscht auf einem Datagram-Anschluß mit der Servicespezifikation biff (siehe services(4)) nach einzeliligen Nachrichten der Form

benutzer@elbriefk-offset

Ist der angegebene Benutzer *benutzer* im System angemeldet und am angeschlossenen Terminal das Eigentümerbit für Ausführung auf Ein gesetzt (mit biff y), wird *offset* als Such-Offset für die entsprechende Postdatei verwendet, und die ersten 7 Zeilen bzw. 560 Zeichen der Nachricht werden auf dem Terminal des Benutzers ausgegeben. Bis auf die Zeilen From, To, Date oder Subject wird der Kopfsatz der Meldung nicht ausgegeben.

DATEIEN

/var/utmp

gibt an, welche Benutzer an welchen Terminals angemeldet sind

SIEHE AUCH

services(4), inetd(1M).

HINWEIS

Beim Filtern von Meldungsköpfen kann es leicht zu Fehlern kommen.

consoletype (1M)

consoletype (1M)

BEZEICHNUNG

`consoletype` – stellt den Terminaltyp der Konsole fest

ÜBERSICHT

`/etc/consoletype`

BESCHREIBUNG

`consoletype` stellt den Terminaltyp der Konsole fest. Das Kommando wird beim Hochfahren des Systems aufgerufen, um die Terminaleigenschaften der Konsole korrekt einstellen zu können.

SIEHE AUCH

`boot`

BEZEICHNUNG

`crash` – prüft Systemabbilder

ÜBERSICHT

`/usr/sbin/crash [ -d dumpdatei ] [ -n namensliste ] [ -w ausgabedatei ]`

BESCHREIBUNG

Das Kommando `crash` dient zum Prüfen des Systemspeicherabbilds eines laufenden oder eines abgestürzten Systems. Dazu werden Steuerstrukturen, Tabellen und andere Informationen formatiert und ausgegeben. Argumente zu `crash` sind *dumpdatei*, *namensliste* und *ausgabedatei*.

Die Datei *dumpdatei* enthält das Systemspeicherabbild. Die Standard*dumpdatei* ist `/dev/mem`.

namensliste enthält die Symboltabelle, die für symbolischen Zugriff auf das zu prüfende Systemspeicherabbild benötigt wird. Die Standard*namensliste* ist `/stand/unix`. Soll ein Systemspeicherabbild einer anderen Maschine geprüft werden, muß ein entsprechendes Kernprogramm von dieser Maschine kopiert werden.

Wenn das Kommando `crash` aufgerufen wird, wird eine Sitzung eingeleitet. Die Ausgabe einer `crash`-Sitzung wird zur *ausgabedatei* geleitet. *ausgabedatei* ist die Standardausgabe.

Die Eingabe während einer `crash`-Sitzung hat folgende Form:

funktion [*argument...*]

Dabei ist *funktion* eine der `crash`-Funktionen, die im Abschnitt "FUNKTIONEN" dieses Handbucheintrags beschrieben werden. *argumente* sind bestimmte Daten, mit denen angegeben wird, welche Elemente des Systemspeicherabbilds ausgegeben werden.

Prozeßbezogene Elemente werden bei laufenden Systemen vom gerade aktiven Prozeß ausgegeben, bei abgestürzten Systemen vom zur Zeit des Absturzes aktiven Prozeß. Wird vom Inhalt einer Tabelle ein Dump ausgegeben, umfaßt die Ausgabe standardmäßig alle aktiven Tabelleneinträge.

Für die `crash`-Funktionen können die folgenden Optionen angegeben werden, wo sie semantisch gültig sind.

- e Zeigt jeden Eintrag in einer Tabelle an.
- f Zeigt die vollständige Struktur an.
- p Interpretiert alle Adreßargumente in der Kommandozeile als physische Adressen. Handelt es sich nicht um physische Adressen, sind die Ergebnisse nicht konsistent.
- s *prozeß* Gibt einen anderen als den üblichen Prozeß zur Überprüfung an.
- w *datei* Leitet die Ausgabe einer Funktion nach *datei* um.

Die Funktionen `mode`, `defproc` und `redirect` entsprechen den Funktionsoptionen `-p`, `-s` und `-w`. Mit der Funktion `mode` kann die Adreßumwandlung für alle später eingegebenen Funktionen auf physisch oder virtuell gesetzt werden; `defproc` gibt den Prozeß an, dessen Daten von den nachfolgenden Funktionen bearbeitet werden sollen; `redirect` schließlich leitet die gesamte Ausgabe um.

Die Ausgabe von `crash`-Funktionen läßt sich folgendermaßen durch eine Pipe zu einem anderen Programm senden:

```
funktion [ argument... ] ! shell_kommando
```

Beispielsweise schreibt

```
mount ! grep rw
```

alle Einträge einer Einhängetabelle mit der Marke `rw` in die Standardausgabe. Die Umleitungsoption (`-w`) kann hierbei nicht verwendet werden.

Je nach Kontext der Funktion wird für numerische Argumente eine bestimmte Basis angenommen. Zähler sind dabei dezimal, Adressen hexadezimal. Tabellenpositionen, die die Funktionslänge überschreiten, werden nicht korrekt interpretiert. `findslot` wandelt eine Adresse in eine Tabellenposition um. Bei allen Argumenten dürfen die voreingestellten Zahlenbasen umdefiniert werden. Die C-Konventionen zum Bezeichnen der Zahlenbasen werden erkannt. Eine Zahl, die normalerweise als Dezimalzahl interpretiert wird, wird als hexadezimal interpretiert, wenn ihr `0x` vorausgeht, als oktal, wenn `0` vorausgeht. Zum Umstellen auf dezimal dient `0d`, zum Umstellen auf binär `0b`.

Als Aliasname für Funktionen kann jede eindeutig identifizierbare Teilzeichenkette des Funktionsnamens verwendet werden. Bisher aus einem Buchstaben bestehende Aliasnamen wie `p` für `proc` bleiben gültig.

Viele Funktionen akzeptieren unterschiedliche Eingabeformen für dasselbe Argument. So kann bei Anforderungen von Tabellendaten die Nummer eines Tabelleneintrags oder ein Bereich angegeben werden. Ein Bereich kann in der Form `a-b` angegeben werden. `a` und `b` sind Dezimalzahlen. Ein Ausdruck besteht aus zwei Operanden und einem Operator. Ein Operand kann eine Adresse, ein Symbol oder eine Zahl sein; der Operator kann `+`, `-`, `*`, `/`, `&` oder `|` sein. Ist der Operand eine Zahl, muß ihm eine Angabe über die Basis vorausgehen, sofern es sich nicht um eine Dezimalzahl handelt (`0` für oktal, `0x` für hexadezimal, `0b` für binär). Die Ausdrücke müssen in Klammern stehen. Andere Funktionen akzeptieren beliebige dieser Argumentformen.

Zwei Abkürzungen für Argumente zu `crash`-Funktionen werden häufig benutzt. Zu beiden können Daten in unterschiedlicher Form eingegeben werden. Die Langform dieser Kürzel lautet:

```
tabelleneintrag = adresse | position | bereich
```

```
start_adr = adresse | symbol | ausdruck
```

FUNKTIONEN

```
? [-w datei]
```

Listet verfügbare Funktionen auf.

```
!kommando
```

Ruft eine Sub-Shell auf und führt das Kommando *kommando* aus.

```
as [-e] [-f] [-w datei] [proz...]
```

Gibt Informationen zu Prozeßsegmenten aus.

`base [-w datei] zahl ...`
 Gibt die *zahl* als Binär-, Oktal-, Dezimal- und Hexadezimalwert aus. Einer Zahl, die nicht dezimal angegeben wird, muß die Angabe der Basis vorangestellt werden: 0x für hexadezimal, 0 für oktal und 0b für binär.

`buffer [-w datei] [-format] pufferpos`
`buffer [-w datei] [-format] [-p] start_adr`
 Aliasname: b.
 Gibt den Inhalt eines Puffers im angegebenen Format aus. Folgende Formatbezeichnungen werden erkannt: -b, Byte; -c, Zeichen; -d, dezimal; -x, hexadezimal; -o, oktal und -i, I-Node. Ist kein Format angegeben, wird das vorhergehende Format verwendet. Das Standardformat zu Beginn einer crash-Sitzung ist hexadezimal.

`bufhdr [-f] [-w datei] [[-p] tabelleneintrag ...]`
 Aliasname: buf.
 Gibt Kopfsätze von Systempuffern aus. Die Option -f erzeugt unterschiedliche Ausgaben, je nachdem, ob es sich um einen lokalen oder einen fernen Puffer handelt.

`callout [-w datei]`
 Aliasname: c.
 Gibt die callout-Tabelle aus.

`class [-w datei] [tabelleneintrag ...]`
 Gibt Daten zu Prozeß-Scheduler-Klassen aus.

`dbfree [-w datei] [klasse ...]`
 Gibt die Köpfe der freien Streams-Datenblöcke aus. Wird eine Klasse angegeben, werden nur deren Datenblockköpfe ausgegeben.

`dblock [-e] [-w datei] [-c klasse ...]`
`dblock [-e] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt die Köpfe der zugewiesenen streams-Datenblöcke aus. Wird die Klassenoption (-c) verwendet, werden nur die Datenblockköpfe der angegebenen Klasse ausgegeben.

`defproc [-w datei] [-c]`
`defproc [-w datei] [position]`
 Legt den Prozeß fest, dessen Daten bearbeitet werden sollen (default process slot argument). Dieses Argument kann die aktuelle Positionsnummer (-c) oder eine beliebige Positionsnummer (slot number) sein. Wird kein Argument eingegeben, wird die zuvor gesetzte Positionsnummer ausgegeben. Zu Beginn einer crash-Sitzung gilt als Prozeßposition der aktuelle Prozeß.

```

dis [-w datei] [-a] start_adr [zählung]
dis [-w datei] [-a] -c [zählung]
    Disassembliert zählung-Anweisungen ab start_adr. Der Standardzählerwert
    ist 1. Die absolute Option (-a) bewirkt eine nicht-symbolische Disassem-
    blierung. Mit der Option -c anstelle von start_adr kann die Disassem-
    blierung bei der Adresse fortgesetzt werden, bei der sie zuvor beendet
    wurde.

dispq [-w datei] [ tabelleneintrag ... ]
    Gibt die Dispatcher-Warteschlange aus.

ds [-w datei] virtuelle_adresse ...
    Gibt das Datensymbol aus, dessen Adresse der eingegebenen Adresse am
    nächsten kommt, aber nicht größer ist.

file [-e] [-w datei] [[-p] tabelleneintrag ...]
    Aliasname: f.
    Gibt die Dateitabelle aus.

findaddr [-w datei] tabelle position
    Gibt die Adresse von position in tabelle aus. Nur Tabellen, auf die die Funk-
    tion size zugreifen kann, stehen für findaddr zur Verfügung.

findslot [-w datei] virtuelle_adresse ...
    Gibt die Tabelle, die Eintragsnummer und den Offset für die eingegebene
    Adresse aus. Für findslot stehen nur die Tabellen zur Verfügung, auf die
    die Funktion size zugreifen kann.

fs [-w datei] [[-p] tabelleneintrag ...]
    Gibt die Dateisystem-Tabelle aus.

gdp [-e] [-f] [-w datei] [[-p] tabelleneintrag ...]
    Gibt die Tabelle "gift descriptor protocol table" aus.

gdt [-e] [-w datei] [ slot[count] ] tabelleneintrag ...]
    Gibt die Tabelle der Globaldeskriptoren aus.

help [-w datei] funktion ...
    Gibt eine Beschreibung der angegebenen Funktion einschließlich Syntax und
    Aliasnamen aus.

idt [-e] [-w datei] [ slot[count] ]
    Gibt die Tabelle der Unterbrechungsdeskriptoren aus.

inode [-e] [-f] [-w datei] [[-p] tabelleneintrag ...]
    Aliasname: i.
    Gibt die I-Node-Tabelle einschließlich Angaben zum Schalter für das
    Dateisystem (file system switch) aus.

kfp [-w datei] [ wert ...]
    Gibt den kernel frame pointer (kpf) für den Anfang eines kernel stack trace
    aus. Wird das Argument wert verwendet, erhält p dessen Wert. Ohne Argu-
    ment wird der aktuelle Wert des kpf ausgegeben.

```

- knstat** [-w *datei*]
Gibt Statistiken zur Speicherzuweisungsfunktion des Kernels aus.
- lck** [-e] [-w *datei*] [[-p] *tabelleneintrag* ...]
Aliasname: l.
Gibt Satzsperrungsdaten aus. Wird die Option -e verwendet oder werden Argumente mit Tabellenadressen angegeben, wird die Satzsperrrenliste ausgegeben. Ohne Argument werden Angaben zu Sperrungen von I-Nodes ausgegeben.
- ldt** [-e] [-w *datei*] [process[slot[count]]]
Gibt die lokale Deskriptortabelle für den angegebenen Prozeß aus bzw. für den aktuellen Prozeß, wenn kein anderer benannt wurde.
- linkblk** [-e] [-w *datei*] [[-p] *tabelleneintrag* ...]
Gibt die Linkblk-Tabelle aus.
- map** [-w *datei*] *abbildungsname* ...
Gibt die Abbildungsstruktur für den angegebenen Abbildungsnamen aus.
- mbfree** [-w *datei*]
Gibt die Kopfsätze der freien Streams-Meldungsblöcke aus.
- mblock** [-e] [-w *datei*] [[-p] *tabelleneintrag* ...]
Gibt die Kopfsätze der zugewiesenen Streams-Meldungsblöcke aus.
- mode** [-w *datei*] [*modus*]
Setzt die Adreßumwandlung für Argumente auf virtuell (v) oder physisch (p). Ohne Argument wird der aktuelle Modus ausgegeben. Zu Beginn einer crash-Sitzung gilt der Modus virtuell.
- mount** [-e] [-w *datei*] [[-p] *tabelleneintrag* ...]
Aliasname: m, vfs.
Gibt Daten zu eingehängten Dateisystemen aus.
- nm** [-w *datei*] *symbol* ...
Gibt Wert und Typ des angegebenen Symbols aus.
- od** [-p] [-w *datei*] [-format] [-modus] [-s *prozeß*] *start_adr* [*zählung*]
Aliasname: rd.
Gibt die Werte von *zählung*, beginnend mit *start_adr*, in einem der folgenden Formate aus: Zeichen (-c), dezimal (-d), hexadezimal (-x), oktal (-o), ASCII (-a) oder hexadezimal/Zeichen (-h), und in einem der folgenden Modi: lang (-l), kurz (-t) oder Byte (-b). Der Standardmodus für Zeichen- und ASCII-Format ist Byte; der Standardmodus für die Formate dezimal, hexadezimal und oktal ist lang. Bei dem Format -h wird sowohl eine hexadezimale als auch eine Zeichendarstellung der Adressen ausgegeben; es muß kein Modus angegeben werden. Fehlt die Angabe für Format oder Modus, wird der vorherige Wert verwendet. Zu Beginn einer crash-Sitzung ist das Format hexadezimal und der Modus lang. Ohne eine Angabe für Zählung wird 1 angenommen.
- panic** Gibt die letzten Systemhinweise, Warn- und Panikmeldungen aus dem begrenzten Ringpuffer im Hauptspeicher aus.

`page [-e] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt Daten seitenweise aus.

`pcb [-w datei] [prozeß]`
 Gibt den Prozeßsteuerblock (TSS) aus. Ohne Argumente wird der aktive TSS für den aktuellen Prozeß ausgegeben.

`prnode [-e] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt Informationen zu den Benutzerdaten von überwachten Prozessen aus.

`proc [-e] [-f] [-w datei] [[-p] tabelleneintrag ... #proznr]`
`proc [-f] [-w datei] [-r]`
 Aliasname: p.
 Gibt die Prozeßtabelle aus. Daten aus Prozeßtabellen können auf zwei Arten lokalisiert werden. Zum einen ist jede Kombination aus Tabelleneinträgen und Prozeßnummern zulässig. Jeder Prozeßnummer muß ein # vorangestellt werden. Zum andern dürfen diese Daten für lauffähige Prozesse auch mit der Option (-r) bestimmt werden. Die Option (-f) erzeugt eine ausführliche Darstellung der Prozeßtabellen und gibt den Bereich für den Prozeß an.

`ptbl [-e] [-w datei] [-s prozeß] [[-p] adr [zählung]]`
 Gibt Daten zu Seitendeskriptortabellen aus.

`pty [-f] [-e] [-w datei] [-s] [-h] [-l]`
 Gibt die aktuell konfigurierten Pseudobildschirme aus. Mit den Optionen -l, -h und -h lassen sich Daten zu den STREAMS Modulen ldterm, ptem bzw. pckt abrufen.

`grun [-w datei]`
 Gibt die Liste der geplanten Streams-Warteschlangen aus.

`queue [-e] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt Streams-Warteschlangen aus.

`quit` Aliasname: q.
 Beendet die crash-Sitzung.

`rcvd [-e] [-f] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt die Tabelle "receive descriptor table" aus.

`rduser [-e] [-f] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt die Tabelle "receive descriptor user table" aus.

`redirect [-w datei] [-c]`
`redirect [-w datei] [neuedatei]`
 Wird diese Funktion mit einem Dateinamen verwendet, wird die Ausgabe einer crash-Sitzung in *neuedatei* umgeleitet. Ohne Argument wird der Name der Datei ausgegeben, zu der die Ausgabe umgeleitet wird. Mit der Option (-c) dagegen wird die zuvor definierte Datei geschlossen und die Ausgabe zur Standardausgabe umgeleitet.

`resource [-e] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt die Advertise-Tabelle aus.

`rt_dptbl [-w datei] [tabelleneintrag ...]`
 Gibt die Tabelle "real-time scheduler parameter table" aus. Siehe `rt_dptbl(4)`.

`rtproc [-w datei]`
 Gibt Daten zu Prozessen in der Echtzeit-Scheduler-Klasse an.

`search [-p] [-w datei] [-m maske] [-s prozeß] muster start_adr länge`
 Gibt die Lang-Wörter im Speicher an, die dem *muster* entsprechen; dabei wird mit der Startadresse *start_adr* für Wörter mit der Länge *länge* begonnen. Die Maske wird über UND (&) mit jedem Wort im Speicher verknüpft und das Ergebnis mit dem Muster verglichen. Die Maske hat den Standardwert 0xffffffff.

`size [-w datei] [-x] [struktur_name ...]`
 Gibt die Größe der angegebenen Struktur aus. Mit der Option (-x) wird die Größe als Hexadezimalwert ausgegeben. Ohne Argument wird eine Liste der Strukturnamen ausgegeben, für die Größenwerte zur Verfügung stehen.

`sndd [-e] [-f] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt die Tabelle "send descriptor table" aus.

`snode [-e] [-f] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt Daten zu geöffneten Gerätedateien aus.

`smount [-e] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt die Server-Einhängetabelle aus.

`stack [-w datei] [prozeß]`
 Aliasname: s.
 Gibt den Dump des Stacks aus. Ohne Argumente wird der Inhalt des Kernel-Stacks für den aktuellen Prozeß ausgegeben. Der Interrupt-Stack und der Stack des aktuellen Prozeßes sind auf einem laufenden System nicht verfügbar.

`stat [-w datei]`
 Gibt statistische Daten zum System aus.

`stream [-e] [-f] [-w datei] [[-p] tabelleneintrag ...]`
 Gibt die Streams-Tabelle aus.

`strstat [-w datei]`
 Gibt statistische Angaben zu Streams aus.

`trace [-w datei] [-r] [prozeß]`
 Aliasname: t.
 Gibt das Stackprotokoll aus. Der Wert *kfp* wird zusammen mit der Option -r angegeben; die Funktion *kfp* gibt den Wert von *kfp* (kernel frame pointer) aus bzw. definiert ihn.

`ts [-w datei] virtuelle_adresse ...`
 Gibt das Textsymbol aus, das der angegebenen Adresse am nächsten liegt.

`tsdptbl [-w datei] [tabelleneintrag ...]`
 Gibt die Parametertabelle für den Timesharing-Scheduler aus. Siehe `ts_dptbl(4)`.

`tsproc [-w datei]`
 Gibt Informationen über Prozesse in der Time-Sharing-Scheduler-Klasse aus.

`tty [-e] [-f] [-l] [-w datei] [-t typ [[-p] tabelleneintrag ...] | [-p] start_adr]`
 Gültige Typen: `kä`, `asy`, `console`, `comports`.
 Gibt die Bildschirmtable aus. Sind keine Argumente angegeben, wird die Bildschirmtable für beide Bildschirmtypen ausgegeben. Mit der Option `-t` wird die Tabelle für den angegebenen Bildschirmtyp ausgegeben. Wird nach der Option `Typ` kein Argument angegeben, werden alle Einträge der Tabelle ausgegeben. Ein einzelner Eintrag in der Bildschirmtable kann über *start_adr* angegeben werden. Die Option `-l` dient zum Ausgeben von Informationen zur Übertragungsprozedur.

`uinode [-e] [-f] [-w datei] [[-p] tabelleneintrag ...]`
 Aliasname: `ui`.
 Gibt die ufs-I-Node-Tabelle aus.

`user [-f] [-w datei] [prozeß]`
 Aliasname: `u`.
 Gibt den U-Block für den angegebenen Prozeß aus.

`var [-w datei]`
 Aliasname: `v`.
 Gibt die einstellbaren Systemparameter aus.

`vfs [-e] [-w datei] [[-p] tabelleneintrag ...]`
 Aliasname: `mount`, `m`.
 Gibt Daten zu eingehängten Dateisystemen aus.

`vfssw [-w datei] [[-p] tabelleneintrag ...]`
 Gibt Angaben zu konfigurierten Dateisystem-Typen aus.

`vnode [-w datei] [[-p] vnode_adr ...]`
 Gibt Daten zu V-Nodes aus.

`vtop [-w datei] [-s prozeß] start_adr ...`
 Gibt die physische Adresse aus, die aus der virtuellen Adresse *start_adr* errechnet wurde.

BEZEICHNUNG

`cron` – Dämon zum Ausführen von Kommandos zu bestimmten Zeiten

ÜBERSICHT

`/etc/cron`

BESCHREIBUNG

Das Kommando `cron` startet einen Prozeß, der an bestimmten Tagen und zu bestimmten Uhrzeiten Kommandos ausführt. Kommandos, die regelmäßig ausgeführt werden müssen, werden entsprechend den Anweisungen in den `crontab`-Dateien im Verzeichnis `/var/spool/cron/crontabs` aufgerufen. Benutzer können über das Kommando `crontab` ihre eigene Datei `crontab` anlegen. Kommandos, die nur einmal ausgeführt werden sollen, können über das Kommando `at` aufgerufen werden.

`cron` durchsucht die `crontab`-Dateien und die Dateien zum Kommando `at` nur bei der Prozeßinitialisierung und bei der Änderung von Dateien über die Kommandos `crontab` oder `at`. Das erfordert weniger Aufwand als die regelmäßige Überprüfung des Systems auf neue oder geänderte Dateien.

Da das Kommando `cron` nie beendet wird, sollte es nur einmal ausgeführt werden. Dies geschieht routinemäßig über `/etc/rc2.d/S75cron`, wenn das System geladen wird. Die Datei `/etc/cron.d/FIFO` dient als Sperrdatei und verhindert, daß das Kommando `cron` mehrfach ausgeführt wird.

Wenn ein Protokoll aller Aktionen von `cron` angelegt werden soll, muß `CRONLOG=YES` (Standardwert) in der Datei `/etc/default/cron` angegeben werden. Wird `CRONLOG=NO` angegeben, wird kein Protokoll angelegt. Ob ein Protokoll angelegt werden soll oder nicht, kann der Benutzer festlegen, da `cron` in der Regel sehr große Protokolldateien erstellt.

DATEIEN

<code>/etc/cron.d</code>	Hauptverzeichnis von <code>cron</code>
<code>/etc/default/cron</code>	Defaultwert für <code>CRONLOG</code>
<code>/etc/cron.d/FIFO</code>	dient als Sperrdatei
<code>/etc/cron.d/log</code>	Abrechnungsinformationen
<code>/etc/cron.d/queuedefs</code>	Datei, aus der sich <code>cron</code> Informationen für jede Queue holt über die Anzahl der Jobs, die Prioritäten (<code>nice</code>) der Jobs beim Laufen und die Wartezeiten vor einem neuen Versuch. Die Queue <code>a</code> enthält "at jobs", Queue <code>b</code> "batch jobs" und Queue <code>c</code> "cron jobs". Falls diese Datei nicht vorhanden ist, dann sind für jede Queue bis zu 100 Jobs erlaubt, die mit einem <code>nice</code> -Wert 2 gestartet werden und die Wartezeit beträgt 60 s.
<code>/var/spool/cron</code>	Spool-Bereich

cron (1M)

cron (1M)

SIEHE AUCH

at(1), crontab(1), sh(1) in den *Kommandos*.

FEHLERMELDUNGEN

Ein Protokoll aller Aktionen von cron wird in `/etc/cron.d/log` aufgezeichnet.

BEZEICHNUNG

custom - installiert spezifische Teile eines SINIX-Pakets

ÜBERSICHT

custom [-ir] [paket] [-m gerät] [-f [datei]]

BESCHREIBUNG

Mit custom kann der Systemverwalter eine angepaßte Installation erstellen, indem er bestimmte Teile eines SINIX-Pakets im Betriebssystem installiert bzw. Teile löscht. Mit dem Kommando kann ein Dialog zur Installation eingeleitet werden oder es wird mit den entsprechenden Optionen von der Kommandozeile aus aufgerufen.

Dateien werden in *paketen* installiert oder gelöscht. Ein *paket* ist eine Sammlung einzelner Dateien, die zu einer Gruppe zusammengefaßt werden können.

Im Dialogmodus fordert custom zum Einlegen des Datenträgers 1 mit der neuen Version des Produkts auf und entnimmt diesem die zur Unterstützung des Programms notwendigen Produktinformationen. Das folgende Menü erleichtert das Hinzufügen oder Entfernen eines Pakets:

1. Install one or more packages
2. Remove one or more packages
3. List the files in a package
4. Install a single file
5. Select a new set to customize
6. Display current disk usage
7. Help

Wenn man eine Menüoption eingibt, wird man zur Eingabe weiterer Informationen aufgefordert. Im folgenden werden die einzelnen Menüoptionen und die dazugehörigen Aktionen beschrieben.

1. Install (Installieren eines oder mehrerer Pakete)

Fordert zur Angabe eines oder mehrerer Paketnamen auf.

Berechnet, welche Installationsdatenträger benötigt werden und fordert dann den Benutzer dazu auf, die Datenträger mit den entsprechenden Nummern einzulegen. Werden mehrere Pakete angegeben, müssen deren Namen in der Kommandozeile durch Leerzeichen getrennt werden.

Bei dieser Option und bei den Optionen "2" und "3" wird eine Liste der verfügbaren Pakete der ausgewählten Gruppe angezeigt. Die einzelnen Zeilen der Anzeige enthalten den Paketnamen, die Angabe, ob das Paket vollständig installiert, nicht installiert oder teilweise installiert ist, die Größe des Pakets in Blöcken zu 512 Byte und eine einzeilige Beschreibung des Paketinhalts.

2. Remove (Entfernen eines oder mehrerer Pakete)

Fordert zur Eingabe eines oder mehrerer Paketnamen auf.

Löscht die Dateien, die zu dem angegebenen Paket gehören. Werden mehrere Pakete angegeben, müssen deren Namen in der Kommandozeile durch Leerzeichen getrennt werden.

Zeigt die verfügbaren Pakete an (siehe Option 1).

3. List files in a package (Liste der Dateien in einem Paket)
Listet alle Dateien in dem angegebenen Paket auf.
Fordert zur Eingabe eines oder mehrerer Paketnamen auf. Geben Sie die Namen des gewünschten Pakets bzw. der gewünschten Pakete ein.
Zeigt die verfügbaren Pakete an (siehe Option 1).
4. Install a single file (Installation einer einzelnen Datei)
Ruft die angegebene Datei von der Installationsdiskette ab.
Als Dateiname muß ein vollständiger Pfadname relativ zum Verzeichnis root eingegeben werden.
5. Select a new set (Neue Gruppe zur Anpassung auswählen)
Ermöglicht es dem Benutzer, von einer anderen Gruppe aus zu arbeiten.
6. Display current disk usage (Aktuelle Auslastung der Festplatte anzeigen)
Zeigt die aktuelle Auslastung der Festplatte an.
7. Help (Hilfe)
Gibt eine Seite mit Anweisungen zur Benutzung von *custom* aus.
 - s Bezeichnung einer Gruppe
 - i Angegebene(s) Paket(e) installieren
 - r Angegebene(s) Paket(e) entfernen
 - l Dateien in den angegebenen Paketen bzw. dem angegebenen Paket auflisten
 - f Angegebene Datei installieren

Mit der Option *-m* wird das Gerät angegeben, in dem sich der Installationsdatenträger befindet. Der Standardwert ist */dev/install*, der das Gerät 0 bezeichnet, wie bei */dev/fd0*. Die Option *-m* wird benötigt, wenn auf einem System */dev/fd0* ein 5,25-Zoll- und */dev/fd1* ein 3,5 Zoll-Diskettenlaufwerk bezeichnet, der Installationsdatenträger aber eine 3,5-Zoll-Diskette ist. Beispiel:

```
custom -m /dev/rfd196ds9
```

Damit wird der Standardwert für das Gerät außer Kraft gesetzt. Statt dessen wird nun das mit der Option *-m* angegebene Gerät benutzt.

Fehlen Eingaben in der Kommandozeile, fordert *custom* zu diesen Eingaben auf.

HINWEIS

Bei der Installation bestimmter XENIX-Anwendungsprogramme können Fehlermeldungen wie "bad gid" (falsche Gruppennummer) oder "bad uid" (falsche Benutzernummer) ausgegeben werden. Diese Meldungen sind darauf zurückzuführen, daß XENIX und SINIX Benutzernummern (UID) und Gruppennummern (GID) unterschiedlich zuordnen. Wenn nötig können die Datei- und/oder Verzeichnisberechtigungen nach der Installation mit *chmod(1)* geändert werden.

custom (1M)

custom (1M)

DATEIEN

/etc/permissions/

SIEHE AUCH

fixperm(1M), df(1M), du(1M), install(1M)

BEZEICHNUNG

dcopy (generisch) – kopiert Dateisysteme zur Optimierung der Zugriffszeit

ÜBERSICHT

dcopy [-F*dateisystem-typ*] [-V] [*aktuelle_optionen*] [-o *spezifische_optionen*]
eingabeds *ausgabeds*

BESCHREIBUNG

dcopy kopiert das Dateisystem *eingabeds* nach *ausgabeds*. *eingabeds* ist die Gerätedatei für das bestehende Dateisystem; *ausgabeds* ist die Gerätedatei, die das Ergebnis der Neuorganisation aufnimmt. Um eine möglichst effiziente Optimierung zu erreichen, sollte es sich bei *eingabeds* um das Gerät im Raw-Modus und bei *ausgabeds* um das blockorientierte Gerät handeln. *eingabeds* und *ausgabeds* sollten ausgehängte Dateisysteme sein.

aktuelle_optionen sind Optionen, die von dem s5-spezifischen Modul von dcopy unterstützt werden. Andere Dateisystem-Typen unterstützen diese Optionen nicht immer. *spezifische_optionen* bezeichnen Unteroptionen, die in einer Liste von durch Komma getrennten Unteroptionen und/oder Paaren von Schlüsselwort/Attribut aufgeführt sind und von dem dateisystemtyp-spezifischen Modul des Kommandos interpretiert werden sollen.

Folgende Optionen gibt es:

- F Gibt den Dateisystem-Typ *dateisystem-typ* an, mit dem gearbeitet werden soll. Der *dateisystem-typ* muß entweder hier angegeben werden oder aus */etc/vfstab* durch Vergleichen von Datei *eingabeds* (Gerät) mit einem Tabelleneintrag abzuleiten sein.
- V Gibt die gesamte Kommandozeile auf dem Bildschirm aus, führt das Kommando aber nicht aus. Beim Erstellen der Kommandozeile werden die vom Benutzer eingegebenen Optionen und Argumente zusammen mit Daten aus */etc/vfstab* verwendet. Diese Option dient zum Überprüfen und Auswerten der Kommandozeile.
- o Gibt für einen Dateisystem-Typ spezifische Optionen an.

HINWEIS

Dieses Kommando wird u.U. nicht von allen Dateisystem-Typen unterstützt.

DATEIEN

/etc/vfstab Liste der Standardparameter für jedes Dateisystem

SIEHE AUCH

vfstab(4).

Die Handbucheinträge für die dateisystemtyp-spezifischen Module von dcopy.

BEZEICHNUNG

dcopy (s5) – kopiert s5-Dateisysteme zur Optimierung der Zugriffszeit

ÜBERSICHT

dcopy [-F s5] [*generische_optionen*] [-sX] [-an] [-d] [-v] [-f*dgröße*[:*igröße*]]
eingabeds *ausgabeds*

BESCHREIBUNG

generische_optionen sind Optionen, die von dem generischen dcopy-Kommando unterstützt werden.

Ohne Optionen kopiert dcopy Dateien aus *eingabeds* und komprimiert die Verzeichnisse, indem es leere Einträge löscht und aufeinanderfolgende Blöcke in einer Datei mit den optimalen Blockzwischenräumen anlegt.

Folgende Optionen stehen zur Verfügung:

- F s5 Gibt s5 als Dateisystem-Typ an. Diese Option muß nicht angegeben werden, wenn der Dateisystem-Typ durch Vergleichen des Geräts *eingabeds* mit einem Eintrag in der Datei aus */etc/vfstab* abzuleiten ist.
- sX Gibt Gerätedaten zum Anlegen einer optionalen Anordnung der Blöcke einer Datei an. X muß die Form *zylindergröße:blockzwischenraumgröße* haben.
- an Schreibt die *n* Tage lang nicht aufgerufenen Dateien hinter die freien Blöcke des Ziel-Dateisystems. Wird *n* nicht angegeben, unterbleibt diese Verschiebung.
- d Läßt die Anordnung von Verzeichniseinträgen unverändert. Standardmäßig werden Unterverzeichnisse an den Anfang von Verzeichnissen gestellt.
- v Meldet, wie viele Dateien verarbeitet wurden und wie groß die Liste der freien Blöcke für Quelle und Ziel waren.
- f *dgröße*[:*igröße*] Gibt die Größe des Dateisystems *ausgabeds* (*dgröße*) und die Größe der I-Node-Liste (*igröße*) in logischen Blöcken an. Wird die Unteroption (oder *:igröße*) nicht angegeben, werden die Werte von *eingabeds* verwendet.

dcopy erfaßt Unterbrechungen und Abbruchsignale und zeigt die Ausführung des Kommandos an. Um dcopy zu beenden, muß ein Abbruchsignal, gefolgt von einer Unterbrechung oder Quit, eingegeben werden.

HINWEIS

Das neue, mit dcopy erstellte Dateisystem sollte vor dem Einhängen mit fsck überprüft werden.

DATEIEN

/etc/mnttab Liste der aktuell eingehängten Dateisysteme

SIEHE AUCH

generische Kommandos dcopy(1M), fsck(1M), mkfs(1M).

BEZEICHNUNG

dd – konvertiert und kopiert eine Datei

ÜBERSICHT

dd [*option=wert*] [*option=wert ...*]

BESCHREIBUNG

dd kopiert die angegebene Eingabedatei in die angegebene Ausgabe und führt dabei ggf. Umwandlungen durch. Standardmäßig wird die Standardeingabe und -ausgabe verwendet. Die Größe der Ein- und Ausgabeblocke kann angegeben werden, um die Vorteile der Ein-/ Ausgabe im Raw-Modus zu nutzen.

Option	Wert
if= <i>datei</i>	Name der Eingabedatei; Voreinstellung ist die Standardeingabe
of= <i>datei</i>	Name der Ausgabedatei; Voreinstellung ist die Standardausgabe
ibs= <i>n</i>	Größe der Eingabeblocke <i>n</i> Byte (Voreinstellung 512)
obs= <i>n</i>	Größe der Ausgabeblocke <i>n</i> Byte (Voreinstellung 512)
bs= <i>n</i>	definiert die Größe der Ein- und Ausgabeblocke und setzt <i>ibs</i> und <i>obs</i> außer Kraft. Wird keine Umwandlung angegeben, bleibt die Größe der Eingabeblocke erhalten, statt daß kurze Blöcke in den Ausgabepuffer gepackt werden (dies ist besonders wirksam, da nicht im Hauptspeicher kopiert werden muß)
cbs= <i>n</i>	Größe des Umwandlungspuffers (Länge des logischen Datensatzes)
files= <i>n</i>	kopiert und verkettet <i>n</i> Eingabedateien vor dem Abschluß (nur sinnvoll, wenn die Eingabe von einem Magnetband- oder ähnlichem Gerät erfolgt)
skip= <i>n</i>	überspringt <i>n</i> Eingabeblocke vor dem Beginn des Kopiervorgangs (sinnvoll bei Magnetbändern, bei denen <i>iseek</i> nicht definiert ist)
iseek= <i>n</i>	sucht vor dem Kopieren <i>n</i> Blöcke ab dem Anfang der Eingabedatei (günstig für Plattendateien, da <i>skip</i> hierfür zu langsam ist)
oseek= <i>n</i>	sucht vor dem Kopieren <i>n</i> Blöcke ab dem Anfang der Ausgabedatei
seek= <i>n</i>	identisch mit <i>oseek</i> , aus Gründen der Abwärtskompatibilität beibehalten
count= <i>n</i>	kopiert nur <i>n</i> Eingabeblocke
conv=ascii	konvertiert EBCDIC in ASCII
ebcdic	konvertiert ASCII in EBCDIC
ibm	leicht abweichende Abbildung von ASCII in EBCDIC
block	konvertiert mit Neuezeile-Zeichen abgeschlossene ASCII-Sätze in Sätze mit fester Länge
unblock	wandelt ASCII-Sätze mit fester Länge in mit Neuezeile-Zeichen abgeschlossene Datensätze um

dd (1M)

dd (1M)

<code>lcase</code>	bildet alphabetische Zeichen als Kleinbuchstaben ab
<code>ucase</code>	bildet alphabetische Zeichen als Großbuchstaben ab
<code>swab</code>	vertauscht Paare von Bytes
<code>noerror</code>	Verarbeitung wird bei Fehlern fortgesetzt (maximal 5 Fehler nacheinander)
<code>sync</code>	füllt jeden Eingabeblock auf <i>ibs</i> auf

Wo Größen angegeben werden sollen, wird eine Byte-Zahl als Eingabe erwartet. Am Ende einer Zahl kann mit den Buchstaben *k*, *b* oder *w* angegeben werden, daß die Zahl mit 1024, 512 bzw. 2 multipliziert werden soll; ein Zahlenpaar kann durch *x* getrennt werden, um eine Multiplikation anzuzeigen.

cbs wird nur verwendet, wenn eine der Umwandlungen *ascii*, *unblock*, *ebcdic*, *ibm* oder *block* angegeben wird. In den beiden ersten Fällen werden *cbs* Zeichen in den Umwandlungspuffer kopiert, die eventuell angegebene Zeichenabbildung wird vorgenommen, nachgestellte Leerzeichen werden entfernt, und ein Neuezeile-Zeichen wird eingefügt, ehe die Zeile ausgegeben wird. In den drei letzten Fällen werden Zeichen in den Umwandlungspuffer gelesen und Leerzeichen eingefügt, um einen Ausgabedatensatz der Größe *cbs* zu erzeugen. Wird *cbs* nicht oder mit Null angegeben, wandeln die Optionen *ascii*, *ebcdic* und *ibm* den Zeichensatz um, ohne die Blockstruktur der Eingabedatei zu ändern. Die Optionen *unblock* und *block* erstellen lediglich eine Kopie der Datei.

Am Ende des Vorgangs gibt *dd* eine Meldung über die Anzahl der vollständigen und Teil-Ein-/Ausgabeblocke aus.

BEISPIEL

Mit diesem Kommando wird ein geblocktes EBCDIC-Magnetband mit zehn EBCDIC-Kartenabbildern zu 80 Byte pro Block in die ASCII-Datei *x* eingelesen:

```
dd if=/dev/rmt/0h of=x ibs=800 obs=8k cbs=80 conv=ascii,lcase
```

Bitte beachten Sie, daß ein Magnetband im Raw-Modus bearbeitet wird. *dd* eignet sich besonders für die Ein-/Ausgabe auf physikalischen Geräten im Raw-Modus, da zum Lesen und Schreiben beliebige Blockgrößen verwendet werden können.

SIEHE AUCH

`cp(1)`

HINWEIS

Mit *dd* dürfen keine Dateien zwischen Dateisystemen mit unterschiedlicher Blockgröße kopiert werden.

Wird zum Kopieren einer Datei ein blockorientiertes Gerät verwendet, werden zusätzliche Nullen in die Datei eingefügt, um den Endblock bis zur Blockgrenze aufzufüllen.

FEHLERMELDUNGEN

f+p sätze ein(aus)

Anzahl der gelesenen (geschriebenen) vollständigen Blöcke oder Teilblöcke.

BEZEICHNUNG

`delsysadm` – Löschen eines `sysadm`-Menüs oder einer `sysadm`-Funktion

ÜBERSICHT

`delsysadm funktion` | `[-r] menü`

BESCHREIBUNG

Mit dem Kommando `delsysadm` wird eine *funktion* oder ein *menü* aus der `sysadm`-Schnittstelle gelöscht und die Struktur des Schnittstellenverzeichnisses auf dem Zielgerät geändert.

funktion | *menü* Der logische Name und die Position des Menüs bzw. der Funktion (Task) innerhalb der Hierarchie der Menüs. Gehen Sie die Hierarchie, beginnend beim wichtigsten Menü `main`, bis zu dem gewünschten Menü bzw. der gewünschten Funktion durch und trennen Sie dabei alle Namen durch Doppelpunkte (siehe BEISPIELE).

Mit der Option `-r` löscht dieses Kommando rekursiv alle Untermenüs und Funktionen für das angegebene Menü. Wird die Option `-r` nicht angegeben, muß das Menü leer sein.

Mit `delsysadm` sollten nur Elemente gelöscht werden, die als "on-line"-Änderungen mit dem Kommando `edsysadm` angefügt wurden. Ein solcher Zusatz ist mit dem Paketetikett `ONLINE` gekennzeichnet. Enthält die Funktion oder das Menü (und dessen Untermenüs und Teilaufgabe) noch andere Etiketten als `ONLINE`, wird angefragt, ob der Löschvorgang fortgesetzt oder abgebrochen werden soll. Unter diesen Umständen soll wahrscheinlich nicht weitergearbeitet werden, und Sie können es dem betreffenden Paket überlassen, einen derartigen Eintrag zu löschen.

Das Kommando schließt dann die Operation ab, oder es gibt eine Fehlermeldung mit einem Fehlercode aus.

BEISPIELE

Zum Löschen der Funktion `nformat` ist folgendes einzugeben:

```
delsysadm main:applications:ndevices:nformat.
```

EXIT-CODES

Die Exit-Codes von `delsysadm` haben folgende Bedeutung:

- 0 = Ausführung erfolgreich abgeschlossen
- 2 = Ungültige Syntax
- 3 = Menü oder Funktion existiert nicht
- 4 = Menü nicht leer
- 5 = Struktur der Schnittstellenmenüs kann nicht aktualisiert werden

HINWEIS

Ein Menü, das ursprünglich ein Platzhaltermenü war (d.h. das nur erscheint, wenn darunter Untermenüs erscheinen), erhält den Platzhalterstatus, wenn es nach der Löschoperation wieder leer ist.

delsysadm(1M)**delsysadm(1M)**

Mit der Option `-r` überprüft `delsysadm` die Untereinträge auf Abhängigkeiten, bevor sie gelöscht werden. Eine Abhängigkeit besteht, wenn das zu löschende Menü einen Eintrag enthält, der von einem Anwendungspaket stammt. Wird eine solche Abhängigkeit festgestellt, wird dem Benutzer eine Liste der Pakete angezeigt, die von dem zu löschenden Menü abhängen, und er wird gefragt, ob die Operation fortgesetzt werden soll. Lautet die Antwort Ja (yes), werden das Menü und alle seine Untermenüs und Funktionen gelöscht (auch die, bei denen Abhängigkeiten festgestellt wurden). Ist die Antwort Nein (no), wird das Menü nicht gelöscht.

`delsysadm` darf nur zum Löschen von Menü- oder Funktionseinträgen verwendet werden, die mit `edsysadm` in die Schnittstelle eingefügt wurden.

SIEHE AUCH

`edsysadm(1M)`, `sysadm(1M)`.

devattr (1M)

devattr (1M)

BEZEICHNUNG

`devattr` – listet die Geräteattribute auf

ÜBERSICHT

`devattr [-v] gerät [attribut [ . . . ]]`

BESCHREIBUNG

`devattr` zeigt die Attribute eines Geräts an. Dafür stehen zwei Formate zur Verfügung. Wird die Option `-v` nicht angegeben, werden nur die Attributwerte angezeigt. Mit der Option `-v` werden die Attribute im Format *attribut=wert* angezeigt. Werden in der Kommandozeile keine Attribute angegeben, werden alle Attribute für das angegebene Gerät nach Attributnamen in alphabetischer Reihenfolge angezeigt. Werden in der Kommandozeile Attribute angegeben, werden nur diese in derselben Reihenfolge wie in der Kommandozeile angezeigt.

Folgende Optionen und Argumente stehen für dieses Kommando zur Verfügung:

<code>-v</code>	Gibt ausführliches Format an. Die Attributwerte werden im Format <i>attribut=wert</i> angezeigt.
<code>gerät</code>	Definiert das Gerät, dessen Attribute angezeigt werden sollen. Dies kann der Pfadname oder der Aliasname des Geräts ein.
<code>attribut</code>	Bestimmt, welches oder welche Attribute angezeigt werden soll(en). Standardmäßig werden alle Attribute für ein Gerät angezeigt. Eine vollständige Liste und die entsprechende Beschreibung der möglichen Attribute finden Sie im Handbucheintrag <code>putdev(1M)</code> .

EXIT-CODES

Die Exit-Codes von `devattr` haben folgende Bedeutung:

- 0 = erfolgreicher Abschluß der Aufgabe
- 1 = Kommandosyntax nicht korrekt, ungültige Option verwendet oder interner Fehler aufgetreten
- 2 = die Gerätetabelle konnte nicht zum Lesen geöffnet werden
- 3 = das angegebene Gerät wurde in der Gerätetabelle nicht gefunden
- 4 = das angegebene Attribut ist für das gewünschte Gerät nicht definiert

DATEIEN

`/etc/device.tab`

SIEHE AUCH

`devattr(3X)`, `listdev(3X)`, `putdev(1M)`.

devfree (1M)

devfree (1M)

BEZEICHNUNG

devfree – gibt exklusiv benutzte Geräte frei

ÜBERSICHT

devfree *schlüssel* [*gerät* . . .]

BESCHREIBUNG

Mit devfree werden exklusiv benutzte Geräte freigegeben. Die exklusive Nutzung wird mit dem Kommando devreserv angefordert.

Wird devfree nur mit dem Argument *schlüssel* aufgerufen, werden alle Geräte freigegeben, die für diesen *schlüssel* reserviert waren. Wird es mit den Argumenten *schlüssel* und *gerät* aufgerufen, gibt devfree die angegebenen Geräte frei, die mit diesem *schlüssel* reserviert wurden.

Folgende Argumente stehen zur Verfügung:

schlüssel

Bezeichnet den eindeutigen Schlüssel, mit dem das Gerät reserviert wurde.

gerät Definiert das Gerät, das mit diesem Kommando von der exklusiven Nutzung freigegeben werden soll. Dies kann der Pfadname oder der Aliasname des Geräts sein.

EXIT-CODES

Die Exit-Codes von devfree haben folgende Bedeutung:

- 0 = erfolgreiche Ausführung der Aufgabe
- 1 = Kommandosyntax nicht korrekt, ungültige Option verwendet oder interner Fehler aufgetreten
- 2 = Gerätetabelle oder Gerätereservierungstabelle konnte nicht zum Lesen geöffnet werden
- 3 = die Freigabe konnte nicht vollständig durchgeführt werden, weil eines oder mehrere der Geräte nicht reserviert oder nicht mit dem angegebenen Schlüssel reserviert war(en)

DATEIEN

/etc/device.tab
/etc/devlckfile

HINWEIS

Die Kommandos devreserv und devfree verwalten die Verfügbarkeit der Geräte in einem System. Sie schränken die Zugriffsmöglichkeiten auf ein Gerät in keiner Weise ein. Sie dienen lediglich als zentrale Liste für Benutzer, die auf Geräte zugreifen wollen. Prozesse, die devreserv nicht verwenden, können gleichzeitig mit einem Prozeß, der ein Gerät reserviert hat, auf dieses Gerät zugreifen.

SIEHE AUCH

devfree(3X), devreserv(1), devreserv(3X), reservdev(3X).

BEZEICHNUNG

`devnm` – Geräte­name

ÜBERSICHT

`/usr/sbin/devnm [name... ]`

BESCHREIBUNG

Das Kommando `devnm` sucht die Gerätedatei zu dem eingehängten Dateisystem, in dem das Argument *name* vorkommt. Es können ein oder mehrere *name(n)* angegeben werden.

Dieses Kommando wird meist von dem Kommando `brc` zum Erstellen eines Eintrags in die Einhängetabelle für das Gerät `root` verwendet.

BEISPIEL

Das Kommando:

```
/usr/sbin/devnm /usr
```

ergibt:

```
/dev/dsk/c1d0s2 /usr
```

wenn `/usr` in `/dev/dsk/c1d0s2` eingehängt ist.

DATEIEN

`/dev/dsk/*`

`/etc/mnttab`

SIEHE AUCH

`brc(1M)`, `mnttab(4)`.

BEZEICHNUNG

`devreserv` – reserviert Geräte zur exklusiven Nutzung

ÜBERSICHT

`devreserv [pid [geräteliste ... ]]`

BESCHREIBUNG

Mit `devreserv` werden Geräte zur exklusiven Nutzung reserviert. Wird das Gerät nicht mehr benötigt, wird es mit `devfree` wieder freigegeben.

`devreserv` reserviert höchstens ein Gerät pro *geräteliste*. Jede Liste wird sequentiell durchsucht, bis das erste verfügbare Gerät festgestellt wird. Wird in keiner Liste ein entsprechendes Gerät gefunden, schlägt die Reservierungsoperation insgesamt fehl.

Wird `devreserv` ohne Argumente aufgerufen, wird eine Liste der gerade reservierten Geräte und der sie reservierenden Prozesse (Prozeßnummern) ausgegeben. Wird zu dem Kommando `devreserv` nur das Argument *pid* angegeben, enthält die Liste die Geräte, die derzeit für die Prozesse mit dieser Nummer reserviert sind.

Folgende Argumente stehen für dieses Kommando zur Verfügung:

pid Bezeichnet die Nummer des Prozesses, für den das Gerät reserviert ist.

geräteliste Definiert die Geräteliste, die `devreserv` nach verfügbaren Geräten durchsucht. Die Liste muß an die Shell als ein einziges Argument übergeben werden.

BEISPIELE

Reservieren einer Diskette und einer Magnetbandkassette:

```
$ key=$$
$ echo "Die aktuelle Prozeßnummer lautet: $key"
Die aktuelle Prozeßnummer lautet: 10658
$ devreserv $key diskett1 ctapel
/dev/dsk/f0t
/dev/rmt/c0s0
```

Auflisten aller gerade reservierten Geräte:

```
$ devreserv
disk1            2423
/dev/dsk/f0t
/dev/rmt/c0s0
```

Auflisten aller Geräte, die einem bestimmten Schlüssel aktuell zugewiesen sind:

```
$ devreserv $key
/dev/dks/f0t
/dev//rmt/c0s0
```

EXIT-CODES

Die Exit-Codes von `devreserv` haben folgende Bedeutung:

- 0 = erfolgreicher Abschluß der Aufgabe
- 1 = inkorrekte Kommandosyntax, ungültige Option verwendet oder interner Fehler aufgetreten
- 2 = Gerätetabelle oder Gerätereservierungstabelle konnte nicht zum Lesen geöffnet werden
- 3 = Anforderung zum Reservieren eines Geräts konnte nicht erfüllt werden

DATEIEN

`/etc/device.tab`
`/etc/devlkfile`

HINWEIS

Mit den Kommandos `devreserv` und `devfree` wird die Verfügbarkeit von Geräten in einem System gesteuert. Die Kommandos dienen den Benutzern als Orientierung und schränken den tatsächlichen Zugriff auf die Geräte in keiner Weise ein. Sie dienen als zentrale Liste für Benutzer, die bestimmte Geräte verwenden wollen. Reservierte Geräte können von Prozessen, die mit den Reservierungsfunktionen arbeiten, erst dann benutzt werden, wenn sie freigegeben sind. Dagegen können Prozesse, die diese Reservierungsfunktionen nicht verwenden, ein reserviertes Gerät benutzen, da ein solcher Prozeß den Reservierungsstatus des Geräts nicht überprüft. Als Argument *pid* muß die Prozeßnummer des steuernden Prozesses angegeben werden. Dieser muß sowohl beim Reservieren als auch beim Freigeben des Geräts aktiv sein. Wird ein Prozeß deaktiviert, werden alle Reservierungen für dieses Prozesses automatisch gelöscht.

SIEHE AUCH

`devfree(1)`, `devfree(3X)`, `devreserv(3X)`, `reservdev(3X)`.

BEZEICHNUNG

df (generisch) – meldet Anzahl der freien Plattenblöcke und Dateien

ÜBERSICHT

df [-F *dateisystem-typ*] [-begklnTV] [*aktuelle_optionen*] [-o *spezifische_optionen*]
[*verzeichnis* | *gerätedatei* | *ressource...*]

BESCHREIBUNG

Mit df werden die für eingehängte oder ausgehängte Dateisysteme, Verzeichnisse oder eingehängte Ressourcen zugewiesenen Teile des generischen Superblocks ausgegeben. *verzeichnis* ist ein gültiger Verzeichnisname. Wird *verzeichnis* angegeben, gibt df einen Bericht über das Gerät aus, das das *verzeichnis* enthält. *gerätedatei* bezeichnet ein Gerät (z.B. /dev/dsk/0s1). *ressource* ist der Name einer NFS-Ressource. Handelt es sich bei den Argumenten zu df um Pfadnamen, erstellt df einen Bericht zu dem Dateisystem, in dem die angegebene Datei steht.

aktuelle_optionen sind Optionen, die von dem s5-spezifischen Modul von df unterstützt werden. Andere Dateisystem-Typen unterstützen sie dagegen nicht immer. *spezifische_optionen* bezeichnen Unteroptionen, die in einer Liste von durch Komma getrennten Unteroptionen und/oder Paaren von Schlüsselwort/Attribut aufgeführt sind und von dem *dateisystem-typ*-spezifischen Modul des Kommandos interpretiert werden sollen.

Folgende generische Optionens stehen zur Verfügung:

- F Gibt den Dateisystem-Typ *dateisystem-typ* an, mit dem gearbeitet werden soll. Dies ist nur notwendig, wenn das Dateisystem ausgehängt ist. *dateisystem-typ* muß hier angegeben werden, oder er muß sich durch Vergleich von *einhängpunkt*, *gerätedatei* oder *ressource* mit einem Tabelleneintrag aus /etc/vfstab bestimmen lassen.
- b Gibt nur den freien Speicherplatz in Kilobyte aus.
- e Gibt nur die Anzahl der freien Dateien aus.
- g Gibt die gesamte statvfs-Struktur aus. Wird nur für eingehängte Dateisysteme verwendet. Darf nicht mit *aktuelle_optionen* oder der Option -o verwendet werden. Diese Option setzt die Optionen -b, -e, -k, -n und -t außer Kraft.
- k Gibt die zugeordneten Kilobyte aus. Diese Option muß allein aufgerufen werden, da sie ein anderes Ausgabeformat hat als die anderen Optionen.
- l Gibt nur Berichte über lokale Dateisysteme aus. Wird nur für eingehängte Dateisysteme verwendet. Darf nicht zusammen mit *aktuelle_optionen* oder mit der Option -o verwendet werden.
- n Gibt nur den *dateisystem-typ*-Namen aus. Ohne Argumente gibt diese Option eine Liste der eingehängten Dateisystem-Typen aus. Wird nur für eingehängte Dateisysteme verwendet. Darf nicht zusammen mit *aktuelle_optionen* oder Option -o verwendet werden.
- t Gibt vollständige Listen mit Summen an. Diese Option setzt die Optionen -b, -e und -n außer Kraft.

- V Gibt die gesamte Kommandozeile auf dem Bildschirm aus, führt das Kommando aber nicht aus. Beim Erstellen der Kommandozeile werden die vom Benutzer eingegebenen Optionen und Argumente zusammen mit den aus `/etc/mnttab` oder `/etc/vfstab` ermittelten Daten verwendet. Mit dieser Option wird die Kommandozeile überprüft und ausgewertet.
- o Gibt dateisystemtyp-spezifische Optionen an.
- v Meldet den Anteil der belegten Blöcke in Prozent sowie die Anzahl der belegten und freien Blöcke.

Werden keine Argumente oder Optionen angegeben, wird der freie Speicherplatz in allen lokal und fern eingehängten Dateisystemen ausgegeben.

HINWEIS

Die Option `-F` dient der Überprüfung ausgehängter Dateisysteme.

Dieses Kommando wird eventuell nicht von allen Dateisystem-Typen unterstützt.

DATEIEN

<code>/dev/dsk/*</code>	
<code>/etc/mnttab</code>	Einhängetabelle
<code>/etc/vfstab</code>	Liste der Standardparameter für jedes Dateisystem

SIEHE AUCH

`mount(1M)`, `mnttab(4)`, `vfstab(4)`.

`statvfs(2)` im *Referenzhandbuch für Programmierer*.

Die Handbucheinträge für dateisystemtyp-spezifische Module von `df`.

df(1M)

(s5)

df(1M)

BEZEICHNUNG

`df (s5)` – meldet die Anzahl der freien Plattenblöcke und I-Nodes für `s5`-Dateisysteme

ÜBERSICHT

`df [-F s5] [generische_optionen] [-f] [verzeichnis | gerätedatei...]`

BESCHREIBUNG

generische_optionen sind Optionen, die von dem generischen Kommando `df` unterstützt werden.

Mit dem Kommando `df` wird die Anzahl der freien Blöcke und der freien I-Nodes in `s5`-Dateisystemen oder -Verzeichnissen ausgegeben. Dazu werden die Zähler in den Superblöcken überprüft. Der Name der *gerätedatei* (z.B. `.dev/dsk/0s2`) oder der Name des Einhängpunkt-Verzeichnisses *verzeichnis* (z.B. `/usr`) muß angegeben werden. Wird *verzeichnis* angegeben, enthält der Bericht Daten zu dem Gerät, auf dem sich das Verzeichnis befindet.

Folgende Optionen stehen zur Verfügung:

- `-F s5` Gibt `s5` als Dateisystem-Typ an.
- `-f` Die Blöcke in der Liste der freien Blöcke werden gezählt, die Zahl aus dem Superblock wird nicht übernommen.

HINWEIS

Die Option `-f` darf zusammen mit den Optionen `-t`, `-b` und `-e` verwendet werden. Die Option `-k` setzt die Option `-f` außer Kraft.

DATEIEN

`/dev/dsk/*`

SIEHE AUCH

Das generische Kommando `df(1M)`.

BEZEICHNUNG

df (ufs) – meldet freien Plattenplatz in ufs-Dateisystemen

ÜBERSICHT

df [-F ufs] [*generische_optionen*] [-o i] [*verzeichnis* | *gerätedatei*]

BESCHREIBUNG

generische_optionen sind Optionen, die von dem generischen Kommando df unterstützt werden.

Mit df wird angezeigt, wieviel Speicherplatz von ufs-Dateisystemen belegt ist, wieviel verfügbar ist und wieviel von der Gesamtkapazität des Dateisystems belegt ist.

Beachten Sie bitte, daß die Summe des als belegt und frei angegebenen Platzes geringer ist als der gesamte Platz im Dateisystem; der Grund hierfür ist, daß das System einen Bruchteil des Speicherplatzes für das Dateisystem reserviert, damit die Dateisystem-Zuordnungsroutinen reibungslos funktionieren können. In der Regel beträgt der reservierte Anteil etwa 10%. Dieser Wert kann mit tuneufs(1M) geändert werden. Ist der gesamte Platz des Dateisystems mit Ausnahme dieser Reserve belegt, darf nur der Systemverwalter vorhandenen Dateien Datenblöcke und neue Dateien zuweisen. Wird das Dateisystem auf diese Art überbelegt, kann df melden, daß es zu mehr als 100% ausgelastet ist.

Folgende Optionen stehen zur Verfügung.

-F ufs

Gibt ufs als Dateisystem-Typ an.

-o Gibt ufs-dateisystemspezifische Optionen an. Möglich ist:

i Meldet die Anzahl der belegten und der freien I-Nodes. Darf nicht zusammen mit *generische_optionen* verwendet werden.

HINWEIS

df berechnet die Ergebnisse für eingehängte und ausgehängte Dateisysteme unterschiedlich. Bei ausgehängten Systemen wird die oben beschriebene Reserve von 10% berücksichtigt. Daher kann sich der Wert, den das generische Kommando ausgibt, von dem mit diesem Modul ermittelten Wert für verfügbaren Plattenplatz unterscheiden.

Die Optionen -b und -e setzen die Option -t außer Kraft.

DATEIEN

/etc/mnttab

Liste der aktuell eingehängten Dateisysteme.

SIEHE AUCH

Das generische Kommando df(1M), du(1M), quot(1M), tuneufs(1M), mnttab(4)

BEZEICHNUNG

`dfmounts` – zeigt Angaben zu eingehängten Ressourcen an

ÜBERSICHT

`dfmounts [-F dateisystem-typ] [-h] [-o spezifische_optionen] [einschränkung ...]`

BESCHREIBUNG

`dfmounts` gibt die lokalen Ressourcen, die über ein verteiltes Dateisystem *dateisystem-typ* gemeinsam benutzt werden, sowie eine Liste der Clients an, bei denen die Ressource eingehängt ist. Ist *einschränkung* nicht angegeben, zeigt `dfmounts` die fernen Ressourcen an, die in dem lokalen System eingehängt sind. *spezifische_optionen* sowie die Verfügbarkeit und die Semantik von *einschränkung* sind für bestimmte Typen von verteilten Dateisystemen spezifisch.

Wird das Kommando `dfmounts` ohne Argumente abgesetzt, werden alle zu diesem Zeitpunkt im lokalen System eingehängten fernen Ressourcen, unabhängig vom Dateisystem-Typ, angezeigt.

Die Ausgabe von `dfmounts` besteht aus einer optionalen Kopfzeile (kann mit der Option `-h` unterdrückt werden) und einer Liste mit Zeilen, die durch Leerzeichen oder Tabulatoren getrennte Felder enthalten. Zu jeder Ressource werden folgende Felder ausgegeben:

ressource server pfadname client

Im einzelnen:

<i>ressource</i>	Gibt den Namen der Ressource an, den das Kommando <code>mount(1M)</code> benötigt.
<i>server</i>	Gibt das System an, von dem aus die Ressource eingehängt wurde.
<i>pfadname</i>	Gibt den Pfadnamen an, den das Kommando <code>share(1M)</code> benötigt.
<i>client</i>	Listet die Systeme (durch Komma getrennt) auf, die die Ressource eingehängt haben. Clients werden je nach Dateisystem-Typ als <i>domäne.</i> , <i>domäne.system</i> oder <i>system</i> aufgeführt.

Ein Feld kann den Wert Null enthalten. Jedes Nullfeld ist mit einem Bindestrich (–) gekennzeichnet, sofern die übrigen Felder in dieser Zeile nicht ebenfalls Null sind. In diesem Falle ist die Markierung nicht erforderlich.

Felder mit Leerzeichen oder Tabulatoren werden in Anführungszeichen (" ") gesetzt.

DATEIEN

`/etc/dfs/fstypes`

SIEHE AUCH

`dfshares(1M)`, `mount(1M)`, `share(1M)`, `unshare(1M)`.

BEZEICHNUNG

`dfmounts` – zeigt Angaben zu eingehängten NFS-Ressourcen an

ÜBERSICHT

`dfmounts [-F nfs] [-h] [server ...]`

BESCHREIBUNG

Mit `dfmounts` werden die lokalen Ressourcen, die über ein Network File System gemeinsam benutzt werden, und eine Liste der Clients, bei denen die Ressource eingehängt ist, angezeigt. Die Option `-F` muß nicht angegeben werden, wenn NFS der einzige in der Datei `/etc/dfs/fstypes` aufgeführte Dateisystem-Typ ist.

Mit der Option `server` werden Daten zu den Ressourcen angezeigt, die von den einzelnen Servern aus eingehängt wurden. Dabei kann `server` jedes System im Netzwerk sein. Wird kein Server angegeben, wird als `server` das lokale System angenommen.

`dfmounts` ohne Optionen zeigt alle im lokalen System eingehängten fernen Ressourcen an, unabhängig vom Dateisystem-Typ.

Die Ausgabe von `dfmounts` besteht aus einer optionalen Kopfzeile (kann mit der Option `-h` unterdrückt werden), gefolgt von einer Liste von Zeilen, die durch Leerzeichen oder Tabulatoren getrennte Felder enthalten. Zu jeder Ressource werden folgende Felder ausgegeben:

ressource server pfadname client ...

Im einzelnen:

<i>ressource</i>	Gibt den Namen der Ressource an, den das Kommando <code>mount(1M)</code> benötigt.
<i>server</i>	Gibt das System an, von dem aus die Ressource eingehängt wurde.
<i>pfadname</i>	Gibt den Pfadnamen an, den das Kommando <code>share(1M)</code> benötigt.
<i>client</i>	Listet die Systeme (durch Komma getrennt) auf, die die Ressource eingehängt haben.

DATEIEN

`/etc/dfs/fstypes`

SIEHE AUCH

`mount(1M)`, `share(1M)`, `unshare(1M)`.

BEZEICHNUNG

dfshares – listet verfügbare Ressourcen auf fernen oder lokalen Systemen auf

ÜBERSICHT

dfshares [-F *dateisystem-typ*] [-h] [-o *spezifische_optionen*] [*server ...*]

BESCHREIBUNG

dfshares liefert Informationen zu Ressourcen, die dem Host über ein verteiltes Dateisystem vom Typ *dateisystem-typ* zur Verfügung stehen. *Spezifische_optionen* und die Semantik von *server* sind für bestimmte verteilte Dateisysteme spezifisch.

Wird dfshares ohne Argumente eingegeben, werden alle aktuell auf dem lokalen System gemeinsam benutzten Ressourcen angezeigt, unabhängig vom Dateisystem-Typ.

Die Ausgabe von dfshares besteht aus einer optionalen Kopfzeile (kann mit der Option -h unterdrückt werden) und einer Liste mit Zeilen, die durch Leerzeichen oder Tabulatoren getrennte Felder enthalten. Zu jeder Ressource werden folgende Felder ausgegeben:

ressource server zugriff transport beschreibung ...

Im einzelnen:

<i>ressource</i>	Gibt den Namen der Ressource an, den das Kommando mount(1M) benötigt.
<i>server</i>	Gibt den Namen des Systems an, das die Ressource zur Verfügung stellt.
<i>zugriff</i>	Gibt die Zugriffsberechtigung für die Clientensysteme an; entweder <i>ro</i> (Nur-Lesezugriff) oder <i>rw</i> (Lese-/Schreibzugriff). Kann dfshares die Zugriffsberechtigungen nicht feststellen, wird ein Bindestrich (-) angezeigt.
<i>transport</i>	Gibt den Transport-Provider an, über die die <i>ressource</i> gemeinsam benutzt wird.
<i>beschreibung</i>	Beschreibt die Ressource.

Ein Feld kann den Wert Null enthalten. Jedes Nullfeld ist mit einem Bindestrich (-) gekennzeichnet, sofern die übrigen Felder in dieser Zeile nicht ebenfalls Null sind. In diesem Falle ist die Markierung nicht erforderlich.

DATEIEN

/etc/dfs/fstypes

SIEHE AUCH

dfmounts(1M), mount(1M), share(1M), unshare(1M).

BEZEICHNUNG

dfshares – gibt eine Liste der verfügbaren NFS-Ressourcen auf fernen Systemen aus

ÜBERSICHT

dfshares [-F nfs] [-h] [server ...]

BESCHREIBUNG

dfshares gibt Informationen zu Ressourcen aus, die dem Host über das Network File System zur Verfügung stehen. Die Option -F muß nicht angegeben werden, wenn NFS in der Datei /etc/dfs/fstypes als erster Dateisystem-Typ aufgeführt ist.

Die Abfrage kann auf Ressourcen beschränkt werden, die von einem oder mehreren Server(n) bereitgestellt werden.

Mit der Option *server* werden Angaben zu den von jedem Server gemeinsam benutzten Ressourcen angezeigt; *server* kann ein beliebiges System im Netz sein. Wird kein Server angegeben, wird das lokale System als *server* angenommen.

Ohne Argumente zeigt dfshares alle gemeinsam benutzten Ressourcen im lokalen System an, unabhängig vom Dateisystem-Typ.

Die Ausgabe von dfshares besteht aus einer optionalen Kopfzeile (kann mit der Option -h unterdrückt werden) und einer Liste mit Zeilen, die durch Leerzeichen oder Tabulatoren getrennte Felder enthalten. Zu jeder Ressource werden folgende Felder ausgegeben:

ressource server zugriff transport

Im einzelnen:

<i>ressource</i>	Gibt den Namen der Ressource an, den das Kommando mount(1M) benötigt.
<i>server</i>	Gibt das System an, das die Ressource zur Verfügung stellt.
<i>zugriff</i>	Gibt die Zugriffsberechtigungen für die Clientensysteme an; da dfshares diese Angabe für eine NFS-Ressource jedoch nicht ermitteln kann, wird in diesem Feld ein Bindestrich (-) ausgegeben.
<i>transport</i>	Gibt den Transport-Provider an, über die die <i>ressource</i> gemeinsam benutzt wird; da dfshares diese Information für eine NFS-Ressource jedoch nicht ermitteln kann, wird in diesem Feld ein Bindestrich (-) ausgegeben.

DATEIEN

/etc/dfs/fstypes

SIEHE AUCH

share(1M), unshare(1M), mount(1M).

BEZEICHNUNG

diskadd – Dienstprogramm zum Konfigurieren von Festplatten

ÜBERSICHT

/etc/diskadd [*platte*]

BESCHREIBUNG

Die erste und optional die zweite Systemplatte werden bei der Installation des Systems konfiguriert. Weitere Platten können Sie bei der Installation oder mit `diskadd` konfigurieren. `diskadd` ist ein interaktives Kommando, das den Benutzer zur Eingabe von Daten zur Plattenkonfiguration auffordert.

Das optionale Argument *platte* spezifiziert die neu zu konfigurierende Platte. Ohne Argument sucht `diskadd` nach neu eingebauten oder noch nicht konfigurierten Platten und bietet diese nacheinander zum Konfigurieren an.

Beim Aufruf mit dem Argument *platte* müssen die Gerätedateien dieser Platte vorhanden sein. Für die Plattenlaufwerke am Interphase Storer Controller werden diese Dateien bei der Installation eingerichtet. Für SCSI-Platten am NCR Host Adapter (ADP32 Controller) werden sie während des Ladens des Systems eingerichtet. Falls Sie neue Plattenlaufwerke eingebaut haben, werden die Gerätedateien automatisch eingerichtet, wenn `diskadd` ohne das optionale Argument aufgerufen wird.

Das Argument *platte* hat, je nach Typ der zu konfigurierenden Platte, unterschiedliche Formate:

Für Platten am Interphase Storer Controller: `cx dys0`

wobei *x* die Nummer des Controllers ist (0, 1 oder 2)

und *y* die Nummer des Laufwerks an diesem Controller ist (0 oder 1).

Für SCSI Platten am NCR Host Adapter: `shdxtys0`

wobei *x* die Nummer des Host Adapters ist (0, 1 oder 2)

und *y* die Nummer des Targets an diesem Adapter ist (0 bis 6).

Bei der Konfiguration einer Platte führt `diskadd` die folgenden Operationen aus: Zunächst wird die Platte mit `fdisk(1M)` partitioniert. Dadurch wird sie in logische Plattenlaufwerke oder Partitionen unterteilt. Anschließend wird die durch `fdisk` festgelegte aktive UNIX Partition mit `disksetup(1M)` bearbeitet. Dies umfaßt in Kurzfassung: Erstellen und Schreiben der `pdinfo`- und VTOC-Information, Aufteilen der Partition in verschiedene Slices, Initiierung der `mkfs`-Aufrufe für diese Slices, Einhängen der neuen Dateisysteme und Aktualisieren von Informationsdateien.

Durch das Inhaltsverzeichnis des Datenträgers (VTOC: volume table of contents) wird die UNIX-Partition in Slices unterteilt. Slices dienen der Speicherung eines Dateisystems oder stellen ein im Raw-Modus betriebenes Gerät dar (z.B. ein Swap- oder Dump-Gerät). Bei der Ausführung des Kommandos `mkfs(1M)` wird für die benötigten Dateisysteme ein bestimmter Dateisystemtyp pro Slice angelegt. Wurde automatisches Einhängen angegeben, werden im Dateisystem / (`root`) Verzeichnisse angelegt, die die neuen Dateisysteme aufnehmen; sie werden eingehängt und `/etc/vfstab` wird so aktualisiert, daß sie künftig automatisch eingehängt werden, wenn das System geladen wird.

diskadd(1M)

(MX300)

diskadd(1M)

Wurden neue Platten konfiguriert, dann wird durch `prtvtoc` das Inhaltsverzeichnis der Datenträger in der Datei `/etc/partitions` aktualisiert.

Werden auf dem neuen Laufwerk Swap-/Paging-Bereiche hinzugefügt, so aktualisiert `disksetup` automatisch die Datei `/etc/swaptab`, damit beim nächsten Laden des Systems dieser Bereich durch `swap(1M)` bekannt gemacht wird.

DATEIEN

`/dev/dsk/cxdys0`
`/dev/dsk/shdxtys0`
`/dev/rdisk/cxdys0`
`/dev/rdisk/shdxtys0`
`/etc/partitions`
`/etc/swaptab`
`/etc/vfstab`

SIEHE AUCH

`fdisksetup(1M)`, `fdisk(1M)`, `mkdir(1M)`, `mkfs(1M)`, `prtvtoc(1M)`, `swap(1M)`.

BEZEICHNUNG

diskadd – Dienstprogramm zum Konfigurieren von Festplatten

ÜBERSICHT

/etc/diskadd [*platte*]

BESCHREIBUNG

Die erste und optional die zweite Systemplatte werden bei der Installation des Systems konfiguriert. Weitere Platten können Sie bei der Installation oder mit `diskadd` konfigurieren. `diskadd` ist ein interaktives Kommando, das den Benutzer zur Eingabe von Daten zur Plattenkonfiguration auffordert.

Das optionale Argument *platte* spezifiziert die neu zu konfigurierende Platte. Ohne Argument sucht `diskadd` nach neu eingebauten oder noch nicht konfigurierten Platten und bietet diese nacheinander zum Konfigurieren an.

Beim Aufruf mit dem Argument *platte* müssen die Gerätedateien dieser Platte vorhanden sein. Für die Plattenlaufwerke am SPA und am Interphase Storager Controller werden diese Dateien bei der Installation eingerichtet. Für SCSI-Platten am NCR Host Adapter (ADP32 Controller) werden sie während des Ladens des Systems eingerichtet. Falls Sie neue Plattenlaufwerke eingebaut haben, werden die Gerätedateien automatisch eingerichtet, wenn `diskadd` ohne das optionale Argument aufgerufen wird.

Das Argument *platte* hat, je nach Typ der zu konfigurierenden Platte, unterschiedliche Formate:

Für Platten am SPA Controller: `cx dys0`

wobei *x* die Nummer des Controllers ist (0, 1, ... oder 7)

und *y* die Nummer des Laufwerks an diesem Controller ist (0, 1, ... oder 7).

Für Platten am Interphase Storager Controller: `cx dys0`

wobei *x* die Nummer des Controllers ist (8, 9, ... oder 13)

und *y* die Nummer des Laufwerks an diesem Controller ist (0 oder 1).

Für SCSI Platten am NCR Host Adapter: `shd xtys0`

wobei *x* die Nummer des Host Adapters ist (0, 1 oder 2)

und *y* die Nummer des Targets an diesem Adapter ist (0 bis 6).

Bei der Konfiguration einer Platte führt `diskadd(1M)` die folgenden Operationen aus: Zunächst wird die Platte mit `fdisk(1M)` partitioniert. Dadurch wird sie in logische Plattenlaufwerke oder Partitionen unterteilt. Anschließend wird die durch `fdisk` festgelegte aktive UNIX Partition mit `disksetup(1M)` bearbeitet. Dies umfaßt in Kurzfassung: Erstellen und Schreiben der `pdinfo`- und VTOC-Information, Aufteilen der Partition in verschiedene Slices, Initiierung der `mkfs`-Aufrufe für diese Slices, Einhängen der neuen Dateisysteme und Aktualisieren von Informationsdateien.

Durch das Inhaltsverzeichnis des Datenträgers (VTOC, volume table of contents) wird die UNIX-Partition in Slices unterteilt. Slices dienen der Speicherung eines Dateisystems oder stellen ein im Raw-Modus betriebenes Gerät dar (z.B. ein Swap- oder Dump-Gerät). Bei der Ausführung des Kommandos `mkfs(1M)` wird für die benötigten Dateisysteme ein bestimmter Dateisystemtyp pro Slice angelegt. Wurde automatisches Einhängen angegeben, werden im Dateisystem / (`root`) Verzeichnisse angelegt, die die neuen Dateisysteme aufnehmen; sie werden eingehängt und

/etc/vfstab wird so aktualisiert, daß sie künftig automatisch eingehängt werden, wenn das System geladen wird.

Wurden neue Platten konfiguriert, dann wird durch prtvtoc das Inhaltsverzeichnis der Datenträger in der Datei /etc/partitions aktualisiert.

Werden auf dem neuen Laufwerk Swap-/Paging-Bereiche hinzugefügt, so aktualisiert disksetup automatisch die Datei /etc/swaptab, damit beim nächsten Laden des Systems dieser Bereich durch swap(1M) bekannt gemacht wird.

DATEIEN

/dev/dsk/cxdys0
/dev/dsk/shdxtys0
/dev/rdisk/cxdys0
/dev/rdisk/shdxtys0
/etc/partitions
/etc/swaptab
/etc/vfstab

SIEHE AUCH

fdisksetup(1M), fdisk(1M), mkdir(1M), mkfs(1M), prtvtoc(1M), swap(1M).

BEZEICHNUNG

diskadd – Dienstprogramm zum Konfigurieren von Festplatten

ÜBERSICHT

/etc/diskadd [*platte*]

BESCHREIBUNG

Die erste und optional die zweite Systemplatte werden bei der Installation des Systems konfiguriert. Weitere Platten können Sie bei der Installation oder mit `diskadd` konfigurieren. `diskadd` ist ein interaktives Kommando, das den Benutzer zur Eingabe von Daten zur Plattenkonfiguration auffordert.

Das optionale Argument *platte* spezifiziert die neu zu konfigurierende Platte. Ohne Argument sucht `diskadd` nach neu eingebauten oder noch nicht konfigurierten Platten und bietet diese nacheinander zum Konfigurieren an.

Beim Aufruf mit dem Argument *platte* müssen die Gerätedateien dieser Platte vorhanden sein. Für die Plattenlaufwerke am ESDI Controller und am AHA1542B Host Adapter werden diese Dateien bei der Installation eingerichtet. Falls Sie neue Plattenlaufwerke eingebaut haben, werden die Gerätedateien automatisch eingerichtet, wenn `diskadd` ohne das optionale Argument aufgerufen wird.

Das Argument *platte* hat, je nach Typ der zu konfigurierenden Platte, unterschiedliche Formate:

Für Platten am ESDI Controller: `cx dys0`

wobei

x die Nummer des Controllers ist (0)

y die Nummer des Laufwerks an diesem Controller ist (0 oder 1).

Für SCSI Platten am AHA1542B Host Adapter: `cx dys0`

wobei

x die Nummer des Controllers ist

(8 für den ersten oder 9 für den zweiten Controller),

y die Nummer des Targets an diesem Adapter ist

(bei *x*=8: 0 oder 1, bei *x*=9: 1 oder 2, bzw. je nach gewählter Targetnummer)

Bei der Konfiguration einer Platte führt `diskadd` die folgenden Operationen aus: Zunächst wird die Platte mit `fdisk(1M)` partitioniert. Dadurch wird sie in logische Plattenlaufwerke oder Partitionen unterteilt. Anschließend wird die durch `fdisk` festgelegte aktive UNIX Partition mit `disksetup(1M)` bearbeitet. Dies umfaßt in Kurzfassung: Erstellen und Schreiben der `pdinfo`- und VTOC-Information, Aufteilen der Partition in verschiedene Slices, Initiierung der `mkfs`-Aufrufe für diese Slices, Einhängen der neuen Dateisysteme und Aktualisieren von Informationsdateien.

Durch das Inhaltsverzeichnis des Datenträgers (VTOC: volume table of contents) wird die UNIX-Partition in Slices unterteilt. Slices dienen der Speicherung eines Dateisystems oder stellen ein im Raw-Modus betriebenes Gerät dar (z.B. ein Swap- oder Dump-Gerät). Bei der Ausführung des Kommandos `mkfs(1M)` wird für die benötigten Dateisysteme ein bestimmter Dateisystemtyp pro Slice angelegt. Wurde automatisches Einhängen angegeben, werden im Dateisystem / (`root`) Verzeichnisse angelegt, die die neuen Dateisysteme aufnehmen; sie werden eingehängt und `/etc/vfstab` wird so aktualisiert, daß sie künftig automatisch eingehängt werden, wenn das System geladen wird.

Wurden neue Platten konfiguriert, dann wird durch `prtvtoc` das Inhaltsverzeichnis der Datenträger in der Datei `/etc/partitions` aktualisiert.

Werden auf dem neuen Laufwerk Swap-/Paging-Bereiche hinzugefügt, so aktualisiert `disksetup` automatisch die Datei `/etc/swaptab`, damit beim nächsten Laden des Systems dieser Bereich durch `swap(1M)` bekannt gemacht wird.

DATEIEN

- `/dev/dsk/cxdys0`
- `/dev/rdisk/cxdys0`
- `/etc/partitions`
- `/etc/swaptab`
- `/etc/vfstab`

SIEHE AUCH

`fdisksetup(1M)`, `fdisk(1M)`, `mkdir(1M)`, `mkfs(1M)`, `prtvtoc(1M)`, `swap(1M)`.

BEZEICHNUNG

disksetup – Dienstprogramm zur Aufteilung der aktiven UNIX Partition einer Festplatte

ÜBERSICHT

```
/etc/disksetup -I -B [-d standardwertedatei] -b ladedatei raw_gerät
(Installiert die Primärplatte)
/etc/disksetup -I [-d standardwertedatei] raw_gerät
(Installiert zusätzliche Platten)
/etc/disksetup -b ladedatei raw_gerät
(schreibt Boot-Code auf Platte)
```

BESCHREIBUNG

-I installiert *raw_gerät*: Erstellen und Schreiben von *pdinfo* und VTOC.
 -B zeigt an, daß das im Raw-Modus betriebene Gerät das Systemladegerät (Boot-Device) werden soll.

-d *standardwertedatei*
 dient zum Übergeben einer Standardaufteilung für das im Raw-Modus betriebene Gerät. Anhand der Daten aus der Standardwertedatei werden die Standardslices für die UNIX-Partition definiert. Die Aufteilung in der Standardwertedatei wird in einem der folgenden Abschnitte erklärt.

-b *ladedatei*
 disksetup schreibt den in *ladedatei* gefundenen Lade-code (Boot-Code) in den Zylinder 0 des angegebenen Plattenlaufwerks. Das Format der Ladedatei kann entweder ELF oder COFF sein. Es werden nur die benötigten Abschnitte/Segmente geladen. Die mit dem System gelieferte Ladedatei ist */etc/boot*.

raw_gerät

Dieses Argument ist erforderlich und gibt das im Raw-Modus betriebene Gerät für das Plattenlaufwerk an, auf das zugegriffen werden soll. Man muß Slice 0 des Gerätes angeben, damit das gesamte Gerät erfaßt wird: */dev/rdisk/crdys0* (Interphase Storager, ESDI Controller bzw. AHA1542B Host Adapter) oder */dev/rdisk/shdxtys0* (NCR Host Adapter, ADP32 Controller)

disksetup führt die grundlegenden Aufgaben aus, die zur Installation des Primär-Laufwerks oder weiterer Laufwerke erforderlich sind. Zum Konfigurieren von Platten müssen folgende Aufgaben ausgeführt werden: Unterstützung des Benutzers beim Anlegen der Slices (durch eine Reihe von Standardwerten oder durch Abfragen), Schreiben von *pdinfo* und des Datenträgerinhaltsverzeichnisses (VTOC), Absetzen der benötigten *mkfs*-Aufrufe, Anlegen von Einhängepunkten, Einhängen von Dateisystemen und Aktualisieren der Dateien */etc/vfstab* und */etc/swaptab*.

Steht zur Definition der Slices keine Datei mit Standardwerten zur Verfügung, fragt disksetup den Benutzer zunächst, welche Slices er anlegen möchte und welche Größe sie haben sollen. Dann muß der Benutzer seine Auswahl bestätigen. Diese Schritte können wiederholt werden, wenn der Benutzer seine Wahl revidieren möchte. Ist eine Datei mit Standardwerten vorhanden, wird die darauf basierende Standardaufteilung der Slices angelegt. Wählt der Benutzer die Standardaufteilung

aus, wird ein Datenträgerinhaltsverzeichnis (VTOC), das dieser Aufteilung entspricht, auf die Platte geschrieben. Entscheidet sich der Benutzer gegen die Standardaufteilung, kann er die Größe der in der Standardwertdatei definierten Slices angeben.

Bei einer Standardwertdatei, wie sie bei der Installation verwendet wird, ergibt sich folgende Aufteilung:

slice #	slice name	FStype	slice size
(Slice	Name der Slice	Dateisystemtyp	Slicegröße)
1	/	ufs	20M
2	/dev/swap	-	2m
3	/usr	ufs	65M
4	/home	ufs	120W
10	/stand	bfs	5M
11	/var	ufs	50W
12	/opt	ufs	40W
optional			
13	/tmp	ufs	20W

Die Slicenummer gibt den Eintrag im VTOC an, unter dem die Slice zu finden ist. Der Name der Slice ist ein Einhängpunkt, wenn es sich bei der Slice um ein Dateisystem handelt, oder ein beschreibender Name, wenn kein Dateisystem angelegt wird. *FStype* gibt den Dateisystem-Typ für die Slice an; dabei wird mit *s5*, *ufs* und *bfs* bestimmt, welcher Typ von *mkfs* gewählt werden soll, während ein Bindestrich oder *swap* bedeutet, daß für diesen Slice kein *mkfs* aufgerufen werden soll. Bei der Angabe *swap* wird diese Slice in die Datei */etc/swaptab* eingetragen. Diese Slice wird dann beim nächsten Laden des Systems als *swap*-Bereich benutzt.

Die Slicegröße ist ein ganzzahliger Wert, gefolgt von einem Kennbuchstaben für die Größe. Der Kennbuchstabe *M* bedeutet Megabyte (MB), *35M* bedeutet also eine Slicegröße von 35 MB. Der Kennbuchstabe *m* gibt den Faktor an, mit dem die Hauptspeichergröße multipliziert wird. Bei einer Speichergröße von 4 MB bedeutet *2m* also eine Slicegröße von 8 MB. Der Kennbuchstabe *W* steht für gewichtetes Verhältnis. Um ein gewichtetes Verhältnis für *xW* zu errechnen, wird *x* durch die Summe der *W*-Anforderungen dividiert und dieser Wert mit dem restlichen Plattenplatz (nach Abhandlung von Anforderungen des Typs *M* und *m*) multipliziert; daraus ergibt sich die Slicegröße.

disksetup (1M)

disksetup (1M)

DATEIEN

/dev/dsk/cxdys0
/dev/dsk/shdxtys0
/dev/rdisk/cxdys0
/dev/rdisk/shdxtys0
/etc/partitions
/etc/swaptab
/etc/vfstab

SIEHE AUCH

fdisk(1M), mkdir(1M), mkfs(1M), mount(1M), scsimkdev(1M), swap(1M).

BEZEICHNUNG

`diskusg` – erstellt Abrechnungsdaten für beanspruchten Plattenspeicherplatz nach Benutzernummer

ÜBERSICHT

`/usr/lib/acct/diskusg [optionen] [dateien]`

BESCHREIBUNG

`diskusg` erstellt Zwischendaten zur Plattenabrechnung aus den Daten in den angegebenen *dateien*, anderenfalls aus den Daten in der Standardeingabe. `diskusg` gibt eine Zeile pro Benutzer in der Standardausgabe aus, und zwar im folgenden Format:

benutzernummer benutzername #blöcke

Dabei ist:

benutzernummer

die Benutzernummer für den Benutzer,

benutzername

die Benutzerkennung für den Benutzer,

#blöcke

die Gesamtzahl der diesem Benutzer zugewiesenen Plattenblöcke.

Normalerweise liest `diskusg` zur Plattenabrechnung nur die I-Nodes von Dateisystemen. In diesem Falle werden mit *dateien* die Gerätedatei-Namen dieser Geräte angegeben.

`diskusg` erkennt die folgenden Optionen:

- `-s` Die Eingabedaten haben bereits das `diskusg`-Ausgabeformat. `diskusg` faßt alle für einen Benutzer vorhandenen Zeilen in einer Zeile zusammen.
- `-v` Ausführlich. Gibt in der Standardfehlerausgabe eine Liste aller Dateien aus, die keinem Benutzer in Rechnung gestellt werden.
- `-i dateiliste` Ignoriert die Daten in den Dateisystemen, deren Dateisystemname in *dateiliste* steht. *dateiliste* ist eine Liste mit Dateisystemnamen, die durch Komma getrennt oder in Anführungszeichen eingeschlossen sind. `diskusg` vergleicht jeden Namen in dieser Liste mit dem Dateisystemnamen, der in der Datenträgernummer gespeichert ist (siehe `labelit(1M)`).
- `-p datei` verwendet *datei* als den Namen der Paßwortdatei, um Benutzerkennungen für die Anmeldung anzulegen. Standardmäßig wird `/etc/passwd` verwendet.
- `-u datei` legt eine *datei* mit Datensätzen an, in der Dateien aufgelistet werden, die keinem Benutzer in Rechnung gestellt werden. Die Datensätze bestehen aus dem Namen der Gerätedatei, der I-Node-Nummer und der Benutzernummer.

Die Ausgabe von `diskusg` bildet normalerweise die Eingabe für das Kommando `acctdisk` (siehe `acct(1M)`), mit dem Gesamtabrechnungssätze erstellt werden, die mit anderen Abrechnungssätzen gemischt werden können. `diskusg` wird normalerweise in `dodisk` aktiviert (siehe `acctsh(1M)`).

diskusg (1M)

diskusg (1M)

BEISPIELE

Im folgenden Beispiel werden die Plattenabrechnungsdaten für root auf /dev/dsk/c1d0s0 pro Tag erstellt:

```
diskusg /dev/dsk/c1d0s0 | acctdisk > disktaacct
```

DATEIEN

/etc/passwd dient zur Umwandlung von Benutzernummer in Benutzerkennung

SIEHE AUCH

acct(1M), acctsh(1M), acct(4)

HINWEIS

diskusg gilt nur für 95-Dateisysteme. acctdusg (siehe acct(1M)) gilt für alle Dateisysteme, ist jedoch langsamer als diskusg.

BEZEICHNUNG

dispadmin – Verwaltung des Prozeß-Schedulers

ÜBERSICHT

```
dispadmin -l
dispadmin -c klasse -g [-r maß]
dispadmin -c klasse -s datei
```

BESCHREIBUNG

Mit dem Kommando `dispadmin` werden Parameter des Prozeß-Schedulers angezeigt oder geändert, während das System arbeitet.

Mit der Option `-l` wird eine Liste der aktuell im System konfigurierten Scheduler-Klassen ausgegeben.

Mit der Option `-c` wird die Klasse angegeben, deren Parameter angezeigt oder geändert werden sollen. Gültige Werte für *klasse* sind `RT` für die Klasse Real-Time (Echtzeit) und `TS` für die Klasse Time-Sharing.

Mit der Option `-g` werden die Parameter für die angegebene Klasse abgerufen und in die Standardausgabe geschrieben. Die Parameter für die Klasse Real-Time werden unter `rt_dptbl(4)` beschrieben. Die Parameter für die Klasse Time-Sharing werden im Handbucheintrag `ts_dptbl(4)` beschrieben.

Wenn die Option `-g` verwendet wird, kann auch die Option `-r` angegeben werden, mit der die Maßeinheit für die Zeitangaben definiert werden. Ist keine Maßeinheit angegeben, wird die Zeit in Millisekunden berechnet. Ist *maß* angegeben, muß es eine positive Ganzzahl zwischen 1 und 1000000000 sein, und die verwendete Maßeinheit für die Zeit wird als reziproker Wert von *maß* in Sekunden ausgedrückt. So ergibt beispielsweise *maß* = 10 eine Zeitangabe, die in Zehntelsekunden ausgedrückt wird; *maß* = 1000000 eine Zeitangabe in Mikrosekunden. Läßt sich die Zeitangabe in der angegebenen Maßeinheit nicht als Ganzzahl ausdrücken, wird sie auf das nächste ganzzahlige Vielfache der Maßeinheit aufgerundet.

Mit der Option `-s` werden Scheduler-Parameter für die angegebene Klasse anhand der Werte in *datei* definiert. Diese Werte setzen die aktuellen Werte im Hauptspeicher außer Kraft und steuern nun die Zeitplanung der Prozesse in der angegebenen Klasse. Die Werte in *datei* müssen das Format haben, das über die Option `-g` ausgegeben wird. Darüber hinaus müssen die Werte eine Tabelle ergeben, die genauso groß ist (dieselbe Anzahl von Prioritätsstufen hat) wie die überschriebene. Zur Verwendung der Option `-s` muß der Benutzer privilegiert sein.

Die Optionen `-g` und `-s` schließen sich gegenseitig aus: Eine Tabelle kann nicht abgerufen und gleichzeitig überschrieben werden.

`dispadmin` führt eine Plausibilitätsprüfung der Werte in *datei* durch und prüft, ob die erforderlichen Grenzwerte eingehalten werden. Bei dieser Prüfung wird jedoch nicht versucht, die Wirkung der neuen Werte auf die Systemleistung zu analysieren. Ungeeignete Werte können die Systemleistung entscheidend beeinträchtigen. Näheres hierzu finden Sie im *Leitfaden für Systemverwalter*.

BEISPIELE

Mit dem folgenden Kommando werden die aktuellen Scheduler-Parameter für die Klasse Real-Time aus dem Hauptspeicher abgerufen und in die Standardausgabe geschrieben. Die Maßeinheit für die Zeit ist Mikrosekunden.

```
dispadmin -c RT -g -r 1000000
```

Mit dem folgenden Kommando werden die aktuellen Scheduler-Parameter für die Klasse Real-Time mit den in `rt.config` angegebenen Werten überschrieben.

```
dispadmin -c RT -s rt.config
```

Mit dem folgenden Kommando werden die aktuellen Scheduler-Parameter für die Klasse Time-Sharing aus dem Hauptspeicher abgerufen und in die Standardausgabe geschrieben. Die Maßeinheit ist Nanosekunden.

```
dispadmin -c TS -g -r 1000000000
```

Mit dem folgenden Kommando werden die aktuellen Scheduler-Parameter für die Klasse Time-Sharing mit den in `ts.config` angegebenen Werten überschrieben.

```
dispadmin -c TS -s ts.config
```

FEHLERMELDUNGEN

`dispadmin` gibt eine entsprechende Fehlermeldung aus, wenn die aktuellen Scheduler-Parameter nicht überschrieben werden können, weil die erforderlichen Zugriffsberechtigungen fehlen oder bezüglich der angegebenen Eingabedatei ein Problem besteht.

SIEHE AUCH

`priocntl(1)`, `priocntl(2)`, `rt_dptbl(4)`, `ts_dptbl(4)`

dispgid (1)

dispgid (1)

BEZEICHNUNG

dispgid – zeigt eine Liste aller gültigen Gruppennamen an

ÜBERSICHT

dispgid

BESCHREIBUNG

Mit dispgid wird eine Liste aller Gruppennamen im System angezeigt (eine Gruppe pro Zeile).

EXIT-CODES

Die Exit-Codes von dispgid haben folgende Bedeutung:

0 = erfolgreiche Ausführung

1 = Gruppendatei kann nicht gelesen werden

dispuid (1)

dispuid (1)

BEZEICHNUNG

dispuid – zeigt eine Liste aller gültigen Benutzernamen an

ÜBERSICHT

dispuid

BESCHREIBUNG

Mit dispuid wird eine Liste aller Benutzernamen im System angezeigt (eine Zeile pro Name).

EXIT-CODES

Die Exit-Codes von dispuid haben folgende Bedeutung:

0 = Erfolgreiche Ausführung

1 = Paßwortdatei kann nicht gelesen werden.

BEZEICHNUNG

du – erstellt eine Übersicht über die Plattenauslastung

ÜBERSICHT

du [-sar] [name ...]

BESCHREIBUNG

Mit dem Kommando du wird die Anzahl der Blöcke aller Dateien und, rekursiv, aller Unterverzeichnisse in den angegebenen Verzeichnissen und Dateien gemeldet. Die Blockzählung schließt auch die Zeigerblöcke der Datei ein. Werden keine *namen* angegeben, wird das aktuelle Verzeichnis verwendet.

Folgende Optionen stehen zur Verfügung:

-s Bewirkt, daß nur die Gesamtsumme (für jeden der angegebenen *namen*) ausgegeben wird.

-a Erzeugt eine Ausgabezeile pro Datei.

Wird weder -s noch -a angegeben, wird nur eine Ausgabezeile pro Verzeichnis erstellt.

-r Bewirkt, daß du Meldungen ausgibt, wenn Verzeichnisse nicht gelesen, Dateien nicht geöffnet werden können usw. Standardmäßig werden keine Meldungen ausgegeben.

Eine Datei mit zwei oder mehr Verweisen (links) wird nur einmal gezählt.

HINWEIS

Wenn die Option -a nicht angegeben wird, werden Werte, die kein Verzeichnis sind und als Argument eingegeben wurden, nicht aufgeführt.

Bestehen Verweise zwischen Dateien in unterschiedlichen Verzeichnissen und stehen die Verzeichnisse in unterschiedlichen Zweigen der Dateisystemhierarchie, zählt du diese Dateien mehrmals.

Für Dateien, die Lücken enthalten, kann keine korrekte Blockzählung vorgenommen werden.

SIEHE AUCH

Das Kapitel "Verwalten von Dateisystemen" im *Leitfaden für Systemverwalter*.

BEZEICHNUNG

dumping – Speicherabzug des Systems erlauben

ÜBERSICHT

dumping -d [-n kern]

dumping -e [-n kern]

BESCHREIBUNG

Mit dumping können Sie die Erlaubnis erteilen, daß gewollt ein Speicherabzug des Systems erzeugt wird. Tritt eine Systemverklemmung auf und nimmt Ihr Rechner keine Eingabe mehr an, dann müssen Sie folgendermaßen vorgehen, um den Speicherabzug zu erzeugen:

- Sie führen einen Hard-Reset aus und warten, bis die Maschine wieder betriebsbereit ist.
- Sie rufen `dumping -e` auf. `dumping` aktiviert die Tastenkombination CTRL-ALT-SHIFT-PgUp, mit der Sie einen gewollten Systemabsturz herbeiführen können.
- Sie reproduzieren den Fehler, der zur Systemverklemmung führte.
- Sie drücken die Tastenkombination CTRL-ALT-SHIFT-PgUp. Die Taste PgUp befindet sich im rechten Ziffernblock der Tastatur (Ziffer 9).
- Automatisch wird PANIC (user sysdump request) durchgeführt. Dabei wird ein Speicherabzug in die Swap-Partition bzw. Dump-Partition geschrieben.
- Der Speicherabzug wird beim nächsten Hochfahren des Systems automatisch auf die Festplatte geschrieben.

Das Programm `dumping` benötigt Leserechte auf den Systemkern sowie Lese- und Schreibrechte auf den Hauptspeicher `/dev/kmem`.

`dumping` kennt folgende Optionen:

`-n kern`

kern ist der Name des Kerns, den `dumping` verwenden soll. Ohne diese Angabe benutzt `dumping` den Standardsystemkern `/stand/unix`.

`-d` Deaktivieren der Tastenkombination, die einen gewollten Systemabsturz herbeiführt. Bei der Standardeinstellung des Systemkerns ist die Tastenkombination deaktiviert.

`-e` Aktivieren der Tastenkombination, die einen gewollten Systemabsturz herbeiführt.

Sie müssen genau eine der Optionen `-d` oder `-e` angeben.

BEZEICHNUNG

edquota – editiert Benutzerquoten

ÜBERSICHT

```
edquota [ -p proto_benutzer ] benutzername...
edquota -t
```

BESCHREIBUNG

edquota ist ein Quoteneditor. In der Kommandozeile können ein oder mehrere Benutzer angegeben werden. Für jeden Benutzer wird eine temporäre Datei angelegt, in der im ASCII-Format die aktuellen Plattenquoten für diesen Benutzer pro eingehängtem UFS-Dateisystem, für das eine Datei *quotas* existiert, aufgeführt sind. Anschließend wird für diese Datei ein Editor aktiviert. Wenn für ein Dateisystem keine Datei *quotas* besteht, gilt ein Null-Eintrag. Nun können Quoten geändert, neue hinzugefügt werden usw. Beim Verlassen des Editors liest edquota die temporäre Datei und ändert die binären Quotendateien entsprechend ab.

Als Editor wird *vi*(1) aufgerufen, sofern die Umgebungsvariable *EDITOR* nichts anderes festlegt.

Nur ein Benutzer mit entsprechenden Privilegien darf die Quoten editieren. Damit in einem Dateisystem Quoten definiert werden können, muß das Root-Verzeichnis des Dateisystems eine *root* gehörende Datei namens *quotas* enthalten. Näheres finden Sie unter *quotaon*(1M).

proto_benutzer und *benutzername* können numerisch sein, wenn sie einer Benutzer-nummer entsprechen. Es können auch nicht zugewiesene Benutzernummern angegeben werden; für nicht zugewiesene Benutzernamen gilt dies jedoch nicht. Auf diese Weise können für Benutzer, denen erst später eine Benutzernummer zugewiesen wird, Standardquoten vergeben werden.

Folgende Optionen stehen zur Verfügung:

- p Dupliziert die Quoten des angegebenen *proto_benutzers* für jeden angegebenen *benutzernamen*. Dies ist das übliche Vorgehen zum Initialisieren von Quoten für Benutzergruppen.
- t Editiert die über die Software festgelegten Zeitlimits für die einzelnen Dateisysteme. Ist das Zeitlimit Null, gelten die Standardzeitlimits aus */usr/include/sys/fs/ufs_quota.h*. Erkennt werden die Zeiteinheiten *sec(onds)* (Sekunde/n), *min(utes)* (Minute/n), *hour(s)* (Stunde/n), *day(s)* (Tag/e), *week(s)* (Woche/n) und *month(s)* (Monat/e). Die Zeitlimits werden in der größten möglichen Einheit ausgegeben, so daß der Wert größer oder gleich eins ist.

DATEIEN

<i>quotas</i>	Quotendatei im Dateisystem <i>root</i>
<i>/etc/mnttab</i>	Tabelle der eingehängten Dateisysteme

SIEHE AUCH

quota(1M), *quotacheck*(1M), *quotaon*(1M), *repquota*(1M), *vi*(1)

BEZEICHNUNG

edsysadm – Editierwerkzeug für sysadm-Schnittstelle

ÜBERSICHT

edsysadm

BESCHREIBUNG

edsysadm ist ein Werkzeug, das im Dialog mit dem Benutzer Menü- oder Aufgabendefinitionen in der sysadm-Schnittstelle einfügt oder ändert. Damit können an einem bestimmten Rechner Änderungen on-line vorgenommen werden, oder es können Änderungen als Teil eines Software-Pakets eingefügt werden. Das Kommando legt die Verwaltungsdateien an, die für die gewünschten Änderungen der Schnittstelle erforderlich sind, und positioniert sie an der entsprechenden Stelle als On-line-Änderungen oder sichert sie so, daß sie in ein Software-Paket aufgenommen werden können.

edsysadm umfaßt mehrere Bildschirme. Zuerst wird nach dem Typ von Menüoption, der geändert werden soll (menu (Menü) oder task (Aufgabe)), anschließend nach der gewünschten Maßnahme (add (einfügen) oder change (ändern)) gefragt. Wird add gewählt, erscheint ein leeres Menü bzw. eine leere Aufgabendefinition, wie nachstehend beschrieben, das bzw. die auszufüllen ist. Wird change gewählt, wird eine Reihe von Bildschirmen angezeigt, damit Sie die Definition, die Sie ändern wollen, leichter finden. Als letzter Bildschirm erscheint das Menü bzw. die Aufgabendefinition mit den aktuellen Werten, die nun editiert werden können.

Folgende Punkte stehen für Menüdefinitionen zur Verfügung:

Menu Name	(Menüname) Der Name des neuen Menüs, der in der linken Bildschirmspalte erscheinen soll. Dieses Feld darf maximal 16 alphanumerische Zeichen enthalten.
Menu Description	(Menübeschreibung) Eine Beschreibung des neuen Menüs, die in der rechten Bildschirmspalte erscheinen soll. Dieses Feld darf maximal 58 Zeichen enthalten; alle alphanumerischen Zeichen außer @, ^ (Zirkumflex), ~ (Tilde), ' (Akut), ' (Gravis) und " (Anführungszeichen) dürfen verwendet werden.
Menu Location	(Menüposition) Die Position des Menüs in der Menühierarchie, ausgedrückt als Menü-Pfadname. Der Pfadname muß mit dem Hauptmenü beginnen und alle weiteren Menüs, die zwischen dem Hauptmenü und dem gewünschten Menü liegen, in der richtigen Reihenfolge angeben. Die Menünamen müssen durch Doppelpunkte getrennt sein. Die Position eines Menüeintrags, der beispielsweise in das Menü Applications eingefügt werden soll, ist main:applications. Der Menüname darf in dieser Positionsangabe nicht angegeben werden. Der vollständige Pfadname für diesen Menüeintrag besteht aus der Menüposition und dem bei der ersten Eingabeaufforderung eingegebenen Menünamen.

	Es handelt sich um ein verschiebbares Feld, das maximal 50 alphanumerische Zeichen auf einmal anzeigt.
Menu Help File Name	(Name der Menü-Hilfedatei) Pfadname für die Hilfedatei zu diesem Menüeintrag. Befindet sich diese Datei in dem Verzeichnis, in dem <code>edsysadm</code> aufgerufen wurde, muß nicht der gesamte Pfadname eingegeben werden. Wurde eine Hilfedatei angegeben, die nicht existiert, wird ein Editor aktiviert (wie mit <code>\$EDITOR</code> definiert), damit sie erstellt werden kann. Die neue Datei wird im aktuellen Verzeichnis erstellt und mit <code>Help</code> (Hilfe) bezeichnet.
Für die Aufgabendefinition stehen die folgenden Punkte zur Verfügung:	
Task Name	(Name der Aufgabe) Der Name der neuen Aufgabe, wie sie in der linken Bildschirmspalte erscheinen soll. Dieses Feld darf maximal 16 alphanumerische Zeichen enthalten.
Task Description	(Beschreibung der Aufgabe) Eine Beschreibung der neuen Aufgabe, wie sie in der rechten Bildschirmspalte erscheinen soll. Dieses Feld darf maximal 58 Zeichen enthalten; alle alphanumerischen Zeichen außer <code>@</code> , <code>^</code> (Zirkumflex), <code>~</code> (Tilde), <code>'</code> (Akut), <code>`</code> (Gravis) und <code>"</code> (Anführungszeichen) dürfen verwendet werden.
Task Location	(Position der Aufgabe) Die Position der Aufgabe in der Menühierarchie, ausgedrückt als Pfadname. Der Pfadname muß mit dem Hauptmenü beginnen und alle weiteren Menüs, die zwischen dem Hauptmenü und der gewünschten Aufgabe liegen, in der richtigen Reihenfolge angeben. Die Menünamen müssen durch Doppelpunkte getrennt sein. Die Position einer Aufgabe, die beispielsweise in das Menü Applications eingefügt werden soll, ist <code>main:applications</code> . Der Aufgabenname darf in dieser Positionsangabe nicht angegeben werden. Der vollständige Pfadname für diese Aufgabe besteht aus der Aufgabenposition und dem bei der ersten Eingabeaufforderung eingegebenen Aufgabennamen.
	Es handelt sich um ein verschiebbares Feld, das maximal 50 Zeichen auf einmal anzeigt.
Task Help File Name	(Name der Aufgaben-Hilfedatei) Pfadname für die Hilfedatei zu diesem Aufgabeneintrag. Befindet sich diese Datei in dem Verzeichnis, in dem <code>edsysadm</code> aufgerufen wurde, muß nicht der gesamte Pfadname eingegeben werden. Wurde eine Hilfedatei angegeben, die nicht existiert, wird ein Editor aktiviert (wie mit <code>\$EDITOR</code> definiert), damit sie erstellt werden kann.

Task Action	Die neue Datei wird im aktuellen Verzeichnis erstellt und mit <code>Help</code> (Hilfe) bezeichnet. (Aktion zur Aufgabe) Der Name der FACE-Maske oder der ausführbaren Datei, die aktiviert wird, wenn diese Aufgabe ausgewählt wird. Es handelt sich um ein verschiebbares Feld, das maximal 58 alphanumerische Zeichen auf einmal anzeigt. Der Pfadname kann bezogen auf das aktuelle Verzeichnis oder absolut angegeben werden.
Task Files	(Aufgabendateien) Die FACE-Objekte oder andere ausführbare Dateien, die die oben beschriebene Aktion zur Aufgabe unterstützen und von dieser Aktion aus aufgerufen werden können. In dieser Liste darf der Name der Hilfedatei oder die Aktion zur Aufgabe nicht angegeben werden. Pfadnamen können auf das aktuelle Verzeichnis bezogen oder absolut angegeben werden. Ein Punkt (.) steht für "alle Dateien im aktuellen Verzeichnis" und schließt auch die Dateien in Unterverzeichnissen ein. Es handelt sich um ein verschiebbares Feld, das bis zu 50 alphanumerische Zeichen auf einmal anzeigt.

Sobald das Menü bzw. die Aufgabe definiert wurde, erscheinen Bildschirme, mit denen das Menü bzw. die Aufgabe installiert oder zur Verwendung in einem Software-Paket gesichert werden kann. Die Erstellung des Pakets oder die On-line-Installation wird überprüft und zum Abschluß eine Meldung ausgegeben.

HINWEIS

Zum Erstellen oder Ändern von Paketen legt dieses Kommando automatisch eine Datei mit Daten zu den Menüs und eine Datei `prototype` (Prototyp) im aktuellen Verzeichnis (dem Verzeichnis, von dem aus das Kommando ausgeführt wird) an. Die Datei mit den Menüdaten wird während der Installation von Paketen zum Ändern von Menüs in der Menüstruktur benutzt. Eine `prototype`-Datei ist eine Installationsdatei mit einem Inhaltsverzeichnis für das Paket. Die von `edsysadm` erstellte Datei `prototype` listet die unter "Aktion zur Aufgabe" definierten Dateien auf und ordnet ihnen die Installationsklasse "admin" zu. Der Inhalt dieser `prototype`-Datei muß in die `prototype`-Datei des Pakets eingefügt werden.

Bei der On-line-Installation legt `edsysadm` automatisch eine Datei mit Menüdaten an und ändert die Struktur der Schnittstellenmenüs direkt bzw. fügt sie direkt ein.

Die Hilfedatei muß dem Format entsprechen, das im *Leitfaden für Programmierer, System-Schnittstellen und Werkzeuge für die Anwendungsprogrammierung* im Kapitel "Änderung der sysadm-Schnittstelle" beschrieben ist.

SIEHE AUCH

`delsysadm(1M)`, `pkgmk(1)`, `prototype(4)`, `sysadm(1M)`

BEZEICHNUNG

edvtoc – Dienstprogramm zur Modifikation des VTOC einer Platte

ÜBERSICHT

/etc/edvtoc -f vtoc_datei raw_gerät

BESCHREIBUNG

Mit edvtoc kann der Benutzer das VTOC einer Platte (VTOC: volume table of contents, Inhaltsverzeichnis eines Datenträgers) ändern. Er muß dazu die folgenden drei Schritte ausführen: Zuerst muß er prtvtoc -f vtoc_datei aufrufen (siehe prtvtoc(1M)). prtvtoc schreibt das aktuelle VTOC in die Datei vtoc_datei. Anschließend muß er diese Datei editieren und die gewünschten Änderungen am VTOC eingeben. Schließlich muß er edvtoc -f vtoc_datei aufrufen, damit die Änderungen übernommen werden.

edvtoc führt folgendes aus: Lesen/Interpretieren der Datei vtoc_datei, begrenzte Plausibilitätsprüfung des neuen VTOC, Anzeigen des neuen VTOC und Schreiben des VTOC auf Platte, sofern der Benutzer dies wünscht.

-f vtoc_datei

schreibt ein neues VTOC auf die aktive UNIX Partition. Das VTOC wird aus vtoc_datei aufgebaut. Der Benutzer muß vtoc_datei vorher mit prtvtoc erstellt und mit einem Editor geändert haben. Jede Zeile der Datei vtoc_datei beschreibt eine Slice anhand von fünf Angaben: Slice-nummer (#slice), Tag der Slice (tag), Flags der Slice (flags), Startsektor der Slice (start), Größe der Slice in Sektoren (size).

raw_gerät

Im Raw-Modus betriebenes Gerät für das gewünschte Plattenlaufwerk. Damit das gesamte Gerät angesprochen wird, muß das Gerät auf Slice 0 angegeben werden:

/dev/rdisk/cxdys0 bei ESDI-Platten am Interphase Storager,
/dev/rdisk/shdxtys0 bei SCSI-Platten am NCR Host Adapter
(ADP32 Controller).

DATEIEN

/dev/rdisk/cxdys0
/dev/rdisk/shdxtys0
/etc/vfstab
/etc/partitions

SIEHE AUCH

prtvtoc(1M)

HINWEIS

edvtoc zerstört i.a. den Platteninhalt und damit die Benutzerdaten. Zusätzlich zum VTOC müssen bei einer Änderung der Plattenaufteilung noch Dateien wie /etc/vfstab und /etc/partitions auf den neuesten Stand gebracht werden. Zur Plattenverwaltung ist deshalb diskadd besser geeignet.

BEZEICHNUNG

edvtoc – Dienstprogramm zur Modifikation des VTOC einer Platte

ÜBERSICHT

/etc/edvtoc -f vtoc_datei raw_gerät

BESCHREIBUNG

Mit edvtoc kann der Benutzer das VTOC einer Platte (VTOC: volume table of contents, Inhaltsverzeichnis eines Datenträgers) ändern. Er muß dazu die folgenden drei Schritte ausführen: Zuerst muß er prtvtoc -f vtoc_datei aufrufen (siehe prtvtoc(1M)). prtvtoc schreibt das aktuelle VTOC in die Datei vtoc_datei. Anschließend muß er diese Datei editieren und die gewünschten Änderungen am VTOC eingeben. Schließlich muß er edvtoc -f vtoc_datei aufrufen, damit die Änderungen übernommen werden.

edvtoc führt folgendes aus: Lesen/Interpretieren der Datei vtoc_datei, begrenzte Plausibilitätsprüfung des neuen VTOC, Anzeigen des neuen VTOC und Schreiben des VTOC auf Platte, sofern der Benutzer dies wünscht.

-f vtoc_datei

schreibt ein neues VTOC auf die aktive UNIX Partition. Das VTOC wird aus vtoc_datei aufgebaut. Der Benutzer muß vtoc_datei vorher mit prtvtoc erstellt und mit einem Editor geändert haben. Jede Zeile der Datei vtoc_datei beschreibt eine Slice anhand von fünf Angaben: Slice-nummer (#slice), Tag der Slice (tag), Flags der Slice (flags), Startsektor der Slice (start), Größe der Slice in Sektoren (size).

raw_gerät

Im Raw-Modus betriebenes Gerät für das gewünschte Plattenlaufwerk. Damit das gesamte Gerät angesprochen wird, muß das Gerät auf Slice 0 angegeben werden:

/dev/rdisk/cxdys0 bei ESDI-Platten am ESDI Controller und
bei SCSI-Platten am AHA1542B Host Adapter

DATEIEN

/dev/rdisk/cxdys0
/etc/vfstab
/etc/partitions

SIEHE AUCH

prtvtoc(1M)

HINWEIS

edvtoc zerstört i.a. den Platteninhalt und damit die Benutzerdaten. Zusätzlich zum VTOC müssen bei einer Änderung der Plattenaufteilung noch Dateien wie /etc/vfstab und /etc/partitions auf den neuesten Stand gebracht werden. Zur Plattenverwaltung ist deshalb diskadd besser geeignet.

BEZEICHNUNG

fconv – Konversion von UFS-Gerätedateien

ÜBERSICHT

fconv [-zahl] *gerätedatei*

BESCHREIBUNG

fconv konvertiert das UFS-Dateisystem auf *gerätedatei* vom SINIX-L V5.40-Format ins SINIX-L V5.41-Format oder umgekehrt. *gerätedatei* bezeichnet die zeichenorientierte Slice (d.h. im Raw-Modus), auf der das Dateisystem liegt. Das Dateisystem muß ausgehängt sein.

Für SINIX-L V5.40 war die maximale Anzahl von Fragmenten pro UFS-Block gleich 16. Für SINIX-L V5.41 mußte diese Größe auf 8 korrigiert werden. Als Konsequenz kann ein V5.40-UFS in V5.41 zwar eingehängt, aber nicht beschrieben werden. Entsprechendes gilt für ein V5.41-UFS, das in V5.40 eingehängt wird. fconv konvertiert das Dateisystem in die geeignete Form.

Wenn die Option -zahl nicht angegeben wird, zeigt fconv die aktuelle Anzahl von Fragmenten für das angegebene UFS-Dateisystem an. Daran kann man erkennen, ob es sich um ein SINIX-L V5.40 oder ein SINIX-L V5.41 UFS handelt.

Parameter:

-zahl zahl darf nur die Werte 8 oder 16 annehmen. Für zahl gleich 8 wird das angegebene UFS-Dateisystem ins SINIX-L V5.41-Format konvertiert, für zahl gleich 16 ins SINIX-L V5.40-Format.

HINWEIS

Bei der Verwendung von fconv ist größte Vorsicht geboten. Das zu konvertierende Dateisystem darf auf keinen Fall eingehängt sein!

BEZEICHNUNG

fdisk – Dienstprogramm zur Aufteilung einer Festplatte in Partitionen

ÜBERSICHT

fdisk *zeichenorientiertes_gesät*

BESCHREIBUNG

fdisk teilt ein physikalisches Plattenlaufwerk in ein oder mehrere logische Laufwerke (logische Platten, Partitionen) auf. Dabei erlaubt fdisk nicht nur die anfängliche Aufteilung einer Platte in logische Laufwerke, sondern informiert auch über die momentane Aufteilung, kann diese ändern und kann die aktive UNIX Partition auswählen.

fdisk arbeitet interaktiv. Im Falle einer noch nicht aufgeteilten Platte wird dem Benutzer empfohlen, diese zu 100% für das UNIX System zu verwenden. Beantwortet der Benutzer diesen Vorschlag negativ oder trägt die Platte bereits eine Aufteilung in Partitionen, so wird die gewünschte Aufteilung der Platte in Partitionen im Dialog ermittelt.

Die Basis für diesen Dialog bildet eine Liste der bereits existierenden Partitionen auf der Platte und ein Menü mit mehreren möglichen Operationen. Der Bildschirm hat dabei etwa folgendes Aussehen:

Total hard disk size is x cylinders.

Partition	Status	Type	Start	End	Length
=====	=====	=====	=====	=====	=====
<number>	active	UNIX System	<cyl.>	<cyl.>	<cyls>
	<or NULL>	EXT-DOS			
		DOS-DATA			
		DOS			
		Other			

SELECT ONE OF THE FOLLOWING:

1. Create a partition
2. Change Active (Boot from) partition
3. Delete a partition
4. Exit (Update disk configuration and exit)
5. Cancel (Exiting without updating disk configuration)

Beim Anlegen einer neuen Partition wird unter anderem die Größe dieser Partition spezifiziert. Dies kann entweder durch eine Prozent-Angabe oder durch die Angabe des ersten Zylinders und der Anzahl der Zylinder erfolgen, die die Partition belegen soll.

Nur eine der Partitionen in der Tabelle kann den Status "active" haben. Für die Systemplatte bedeutet dies, daß beim nächsten Reboot das System von dieser Partition geladen wird. Für alle Plattenlaufwerke gilt, daß die Slice 0 (/dev/dsk/cxdys0 oder /dev/dsk/shdxtys0) genau diese Partition abdeckt.

Die weitere Aufteilung einer Platte in Slices erfolgt mit Hilfe von disksetup(1M).

fdisk (1M)

(MX300, MX500)

fdisk (1M)

DATEIEN

/dev/dsk/cxdys0

(Slice 0 auf Platte *y* am Controller *x*)

/dev/dsk/shdxtys0

(Slice 0 am Target *y* am Host Adapter *x*)

/dev/rdsk/cxdys0

(zeichenorientiertes Gerät, raw device)

/dev/rdsk/shdxtys0

(zeichenorientiertes Gerät, raw device)

SIEHE AUCH

disksetup(1M), diskadd(1M)

HINWEIS

Mit `fdisk` werden zwar keine Daten in den Dateisystemen einer Platte zerstört, eine Modifikation der Partitionstabelle kann jedoch zu Inkonsistenzen und Verlust von Daten führen, wenn diese unter UNIX nicht mehr erreichbar sind.

BEZEICHNUNG

fdisk – Dienstprogramm zur Aufteilung einer Festplatte in Partitionen

ÜBERSICHT

fdisk *zeichenorientiertes_gesät*

BESCHREIBUNG

fdisk teilt ein physikalisches Plattenlaufwerk in ein oder mehrere logische Laufwerke (logische Platten, Partitionen) auf. Dabei erlaubt fdisk nicht nur die anfängliche Aufteilung einer Platte in logische Laufwerke, sondern informiert auch über die momentane Aufteilung, kann diese ändern und kann die aktive UNIX Partition auswählen.

fdisk arbeitet interaktiv. Im Falle einer noch nicht aufgeteilten Platte wird dem Benutzer empfohlen, diese zu 100% für das UNIX System zu verwenden. Beantwortet der Benutzer diesen Vorschlag negativ oder trägt die Platte bereits eine Aufteilung in Partitionen, so wird die gewünschte Aufteilung der Platte in Partitionen im Dialog ermittelt.

Die Basis für diesen Dialog bildet eine Liste der bereits existierenden Partitionen auf der Platte und ein Menü mit mehreren möglichen Operationen. Der Bildschirm hat dabei etwa folgendes Aussehen:

Total hard disk size is x cylinders.

Partition	Status	Type	Start	End	Length
=====	=====	=====	=====	=====	=====
<number>	active	UNIX System	<cyl.>	<cyl.>	<cyls>
	<or NULL>	EXT-DOS			
		DOS-DATA			
		DOS			
		Other			

SELECT ONE OF THE FOLLOWING:

1. Create a partition
2. Change Active (Boot from) partition
3. Delete a partition
4. Exit (Update disk configuration and exit)
5. Cancel (Exiting without updating disk configuration)

Beim Anlegen einer neuen Partition wird unter anderem die Größe dieser Partition spezifiziert. Dies kann entweder durch eine Prozent-Angabe oder durch die Angabe des ersten Zylinders und der Anzahl der Zylinder erfolgen, die die Partition belegen soll.

Nur eine der Partitionen in der Tabelle kann den Status "active" haben. Für die Systemplatte bedeutet dies, daß beim nächsten Reboot das System von dieser Partition geladen wird. Für alle Plattenlaufwerke gilt, daß die Slice 0 (/dev/dsk/cxdys0) genau diese Partition abdeckt.

Die weitere Aufteilung einer Platte in Slices erfolgt mit Hilfe von disksetup(1M).

fdisk (1M)

(WX200)

fdisk (1M)

DATEIEN

/dev/dsk/cxdys0

/dev/rdisk/cxdys0

SIEHE AUCH

disksetup(1M), diskadd(1M)

HINWEIS

Mit `fdisk` werden zwar keine Daten in den Dateisystemen einer Platte zerstört, eine Modifikation der Partitionstabelle kann jedoch zu Inkonsistenzen und Verlust von Daten führen, wenn diese unter UNIX nicht mehr erreichbar sind.

BEZEICHNUNG

fdp – legt ein vollständiges Dateisystemarchiv an oder stellt die Dateien in einem solchen wieder her

ÜBERSICHT

fdp -B [-dovAENS] [-c *zahl*] *sichauftragsnr* *udpname* *udpger* *udpknenn* *beschreibung*
fdp -RC [-dovAENS] [-c *zahl*] *udpname* *udpger* *wddpname* *wdger* *wdauftragsnr*
beschreibung

BESCHREIBUNG

Das Kommando **fdp** wird von anderen Shell-Kommandos als Sohnprozeß aufgerufen. Der Kommandoname **fdp** wird entweder aus der Datei *bkhist.tab* oder aus dem Kommando (mit Option) **bkreg -m** gelesen. Die Optionen **-B**, **-R** und **-C** werden von den Shell-Kommandos **backup** und **restore** an **fdp** übergeben. Die anderen Optionen werden von der Datei *bkhist.tab* oder von dem Kommando (mit Option) **bkreg -p** aus übergeben. Die Argumente werden von verschiedenen Positionen im Backup-Service aus an **fdp** gesendet.

fdp -B wird als Sohnprozeß vom Kommando **backdaemon** aufgerufen, um eine Sicherung der Datenpartition *udpger* (Ursprungsdatenpartition) anzulegen. Alle Blöcke in der Datenpartition werden archiviert. Die daraus resultierende Sicherung wird in dem in *dd(1)* beschriebenen Format angelegt. Die Sicherung wird im Sicherungsprotokoll */etc/bkup/bkhist.tab* aufgezeichnet.

fdp -RC wird als Sohnprozeß vom Kommando **rsoper** aufgerufen, um die gesamte Datenpartition aus einem von **fdp -B** angelegten Archiv wiederherzustellen. Dabei wird davon ausgegangen, daß das Datenpartitionsarchiv das in *dd(1)* beschriebene Format aufweist.

Die Argumente zu **fdp** sind folgendermaßen definiert:

sichauftragsnr

ist die Auftragsnummer, die von **backup** zugeordnet wird. Die *sichauftragsnr* wird benutzt, wenn Einträge in die Protokolldatei aufgenommen werden.

udpname

ist der Name der Datenpartition, von der eine Sicherung erstellt werden soll. Wird von **fdp** nicht benutzt, aber von **backup** zur Verfügung gestellt, damit die Kompatibilität mit anderen Archivierungsverfahren im Hinblick auf die Kommandozeile gewährleistet ist.

udpger

ist der Name des blockorientierten Geräts, auf dem sich die Datenpartition befindet.

udpknenn

ist der Name des Datenträgers auf dem Dateisystem (siehe *labelit(1M)*). Wird von **fdp** nicht benutzt, aber von **backup** zur Verfügung gestellt, damit die Kompatibilität mit anderen Archivierungsverfahren im Hinblick auf die Kommandozeile gewährleistet ist.

beschreibung

ist eine Beschreibung für ein Zielgerät in der Form:

ggruppe:gname:gchar:gkenn

ggruppe gibt eine Gerätegruppe an (siehe *devgroup.tab(4)*).
gname gibt einen bestimmten Gerätenamen an (siehe *device.tab(4)*).
gchar gibt Charakteristika des Geräts an. Wenn *gchar* angegeben wird, werden die Standardwerte für das angegebene Gerät und die angegebene Gruppe außer Kraft gesetzt (unter *device.tab(4)* finden Sie eine ausführliche Beschreibung von Gerätecharakteristika).
gkenn gibt die Namen der Datenträger für die Medien an, die zum Lesen und Erstellen des Archivs benutzt werden sollen.

wdauftragsnr

ist die Auftragsnummer, die von *restore* zugewiesen wird.

wdger

ist, wenn der Wert nicht Null lautet, die Partition, auf die die Daten bei der Wiederherstellung geschrieben werden sollen, wenn sie nicht auf *udpger* geschrieben werden sollen.

wddpname

wird nicht benutzt, steht aber aus Gründen der Kompatibilität mit anderen Verfahren zur Verfügung.

Optionen

Einige Optionen sind nur von Bedeutung, wenn *fdp -B* aufgerufen wird; wenn *fdp -R* aufgerufen wird, werden sie akzeptiert, aber ignoriert, da das Kommando mit seinen Optionen automatisch von *restore* aufgerufen wird. Diese Optionen sind mit einem Stern (*) gekennzeichnet.

- c*zahl* Archiviert oder stellt nur die ersten *zahl* Datenblöcke von jeweils 512 Byte in der Datenpartition wieder her.
- d** Verhindert, daß die Archivierung im Sicherungsprotokoll aufgezeichnet wird.
- o* Ermöglicht es dem Benutzer, Anforderungen, ein Medium einzulegen, außer Kraft zu setzen (siehe *getvol(1M)* und die Beschreibung der Option *-o*).
- v** Überprüft das Archiv, wenn es geschrieben wird. Während das Archiv geschrieben wird, wird eine Prüfsumme errechnet. Jedes Medium wird anschließend noch einmal gelesen, und die Prüfsumme wird noch einmal errechnet, um zu überprüfen, ob jeder Datenblock lesbar und fehlerfrei ist. Schlägt eine dieser Prüfungen fehl, gilt das Medium als nicht lesbar. Wurde die Option *-A* angegeben, schlägt die Archivierung fehl; andernfalls wird der Bediener dazu aufgefordert, das fehlerhafte Medium zu ersetzen.
- A* Aktiviert den automatischen Modus, d.h. der Benutzer wird nicht dazu aufgefordert, Medien einzulegen oder zu entfernen.
- E** Schätzt, wieviel Speicherplatz das Archiv auf dem Medium beanspruchen wird und gibt diese Schätzung aus. Dann wird die Sicherung ausgeführt.
- N** Schätzt, wieviel Speicherplatz das Archiv auf dem Medium beanspruchen wird und gibt diese Schätzung aus. Anschließend wird keine Sicherung ausgeführt.

- S Zeigt einen Punkt (.) für jeweils 100 Datenblöcke zu 512 Byte an, die vom Medium auf dem Zielgerät gelesen oder auf dieses Medium geschrieben wurden.

Dialog mit dem Benutzer

Die Verbindung zwischen Archivierungsverfahren und `backup` ist komplizierter als eine einfache Verzweigung (`fork/exec`) oder Pipe. Das Kommando `backup` regelt entweder direkt oder über das Kommando `bkoper` den Dialog mit dem Benutzer. Daher liest `fdp` weder von der Standardeingabe, noch schreibt es auf die Standardausgabe oder Standardfehlerausgabe. Um Berichte (Schätzungen, Dateinamen, Perioden, Statusangaben etc.) an `backup` zu übermitteln, muß eine Verfabrensbibliothek benutzt werden (siehe `libbrmeth(3)`).

EXIT-CODES

Die Exit-Codes von `fdp` haben folgende Bedeutung:

- 0 = Aufgabe wurde erfolgreich ausgeführt
- 1 = ein oder mehrere Parameter zu `fdp` sind ungültig
- 2 = ein Fehler trat auf, so daß `fdp` nicht alle Teile der Aufgabe vollständig ausführen konnte

DATEIEN

<code>/etc/bkup/bkexcept.tab</code>	listet die Dateien auf, die von der Teilsicherung des Dateisystems ausgeschlossen werden sollen
<code>/etc/bkup/bkhist.tab</code>	listet die Kennsätze aller Datenträger auf, die für Sicherungsoperationen benutzt wurden
<code>/etc/bkup/rsstatus.tab</code>	verfolgt den Status aller Sicherungsanforderungen von Benutzern
<code>/etc/bkup/bklog</code>	protokolliert Fehler, die von den Sicherungsverfahren und dem Kommando <code>backup</code> generiert wurden
<code>/etc/bkup/rslog</code>	protokolliert Fehler, die von den Wiederherstellungsverfahren und dem Kommando <code>restore</code> generiert wurden
<code>\$TMP/filelist\$\$</code>	speichert temporär das Inhaltsverzeichnis eines Sicherungsarchivs

SIEHE AUCH

`backup(1M)`, `device.tab(4)`, `fdp(1)`, `ffile(1)`, `fimage(1)`, `getvol(1M)`, `incfile(1)`, `labelit(1M)`, `libbrmeth(3)`, `prvtoc(1M)`, `rsoper(1M)`,

BEZEICHNUNG

ff (generisch) – listet Dateinamen und statistische Angaben zu einem Dateisystem auf

ÜBERSICHT

ff [-F *dateisystem-typ*] [-V] [*aktuelle_optionen*] [-o *spezifische_optionen*] *gerätedatei* ...

BESCHREIBUNG

ff liest die Dateien und Verzeichnisse der *gerätedatei*. I-Node-Daten werden für Dateien gespeichert, die den Auswahlkriterien entsprechen, d.h. I-Node-Nummer und/oder I-Node-Alter. Die Ausgabe besteht aus dem Pfadnamen und anderen Informationen zur Datei. Die Ausgabefelder werden mit Tabulatoren positioniert. Die Ausgabe wird in der Reihenfolge der I-Nodes generiert. Die von ff erzeugte Standardzeile lautet:

pfadname i-nummer

aktuelle_optionen sind Optionen, die vom s5-spezifischen Modul von ff unterstützt werden. Andere Dateisystem-Typen unterstützen diese Optionen nicht in jedem Fall. Die *spezifischen_optionen* stehen für Unteroptionen, die als durch Komma getrennte Liste von Unteroptionen und/oder als Paare von Schlüsselwort/Attribut angegeben werden und vom *dateisystemtyp*-spezifischen Modul des Kommandos interpretiert werden.

Die Optionen lauten:

- F Gibt den *dateisystem-typ* an, auf den sich das Kommando beziehen soll. Der *dateisystem-typ* sollte entweder hier angegeben werden oder durch einen Vergleich von *gerätedatei* mit einem Tabelleneintrag in */etc/vfstab* hervorgehen.
- V Gibt die Kommandozeile vollständig aus, führt das Kommando aber nicht aus. Die Kommandozeile wird aus den Optionen und Argumenten, die der Benutzer angibt, und den zusätzlichen Informationen aus */etc/vfstab* generiert. Diese Option dient dazu, die Kommandozeile zu überprüfen und auszuwerten.
- o Gibt *dateisystemtyp*-spezifische Optionen an.

HINWEIS

Dieses Kommando wird nicht für alle Dateisystem-Typen unterstützt.

DATEIEN

/etc/vfstab Liste der Standardparameter für jedes Dateisystem

SIEHE AUCH

ncheck(1M), vfstab(4).

find(1) in den *Kommandos*.

Handbucheinträge zu den DStyp-spezifischen Modulen von ff.

BEZEICHNUNG

ff (s5) – zeigt I-Listeninformationen an

ÜBERSICHT

ff [-F s5] [*generische_optionen*] [-I] [-l] [-ppr_äfix] [-s] [-u] [-an] [-mn] [-cn]
[-ndatei] [-ii-liste] gerätedatei...

BESCHREIBUNG

generische_optionen sind Optionen, die vom generischen Kommando ff unterstützt werden.

ff liest die I-Liste und die Verzeichnisse der *gerätedatei*, sofern es sich um ein s5-Dateisystem handelt. I-Node-Daten werden zu den Dateien gesichert, die den Auswahlkriterien entsprechen. Die Ausgabe besteht aus dem Pfadnamen für jeden gesicherten I-Node sowie weiteren Informationen zu Dateien, die über die Optionen unten angefordert werden. Die Ausgabefelder werden mit Tabulatoren positioniert. Die Ausgabe wird in der Reihenfolge der I-Nodes erzeugt; die Felder sind durch Tabulatoren getrennt. Die von ff erzeugte Standardzeile lautet:

pfadname i-nummer

Dem Pfadnamen geht ein . (Punkt) voraus, es sei denn, die Option -p wird angegeben.

Mit diesem Kommando können folgende Informationen angefordert werden werden:

pfadname i-nummer gröÙe benutzernummer

Das Argument *n* in der Beschreibung der folgenden *optionen* entspricht einer ganzen Zahl (mit oder ohne Vorzeichen). Dabei bedeutet + *n* mehr als *n*, - *n* weniger als *n* und *n* genau *n*. Ein Tag wird im 24-Stunden-Format angegeben.

Die Optionen lauten:

- F s5 Gibt den Dateisystem-Typ s5 an.
- I Gibt an, daß die I-Node-Nummer nach dem Pfadnamen nicht ausgegeben werden soll.
- l Generiert eine zusätzliche Liste aller Pfadnamen für mehrfach verbundene (link) Dateien.
- ppr_äfix Das angegebene *präfix* wird zu jedem generierten Pfadnamen hinzugefügt. Der Standardwert lautet . (Punkt).
- s Gibt an, daß nach jedem Pfadnamen die Dateigröße in Byte ausgegeben wird.
- u Gibt an, daß nach jedem Pfadnamen die Benutzerkennung des Eigentümers ausgegeben wird.
- an Auswählen, wenn auf den I-Node innerhalb *n* Tagen zugegriffen wurde.
- mn Auswählen, wenn der I-Node innerhalb von *n* Tagen modifiziert wurde.

ff(1M)

(s5)

ff(1M)

- `-cn` Auswählen, wenn der I-Node innerhalb von *n* Tagen geändert wurde.
- `-ndatei` Auswählen, wenn der I-Node später als im Argument *datei* angegeben modifiziert wurde.
- `-ii-liste` Generiert Namen nur für die I-Nodes, die in *i-liste* angegeben wurden. *i-liste* ist eine Liste von Zahlen, die durch Komma, aber nicht durch Leerzeichen getrennt sind.

HINWEIS

Wird die Option `-l` nicht angegeben, wird für einen mehrfach verbundenen (link) I-Node nur einer der möglichen Pfadnamen generiert. Wird `-l` angegeben, erscheinen in der Ausgabe alle Namen für alle verbundenen Dateien im Dateisystem. Auf die generierten Namen werden jedoch keine Auswahlkriterien angewendet.

SIEHE AUCH

generisches Kommando `ff(1M)`, `ncheck(1M)`.
`find(1)` in den *Kommandos*.

BEZEICHNUNG

ff (ufs) – listet Dateinamen und statistische Angaben für ein ufs-Dateisystem auf

ÜBERSICHT

ff [-F ufs] [*generische_optionen*] [-o a,m,s] *gerätedatei* ...

BESCHREIBUNG

generische_optionen sind Optionen, die von dem generischen Kommando ff unterstützt werden.

ff liest die I-Liste und die Verzeichnisse der *gerätedatei*, sofern es sich um ein Dateisystem handelt. I-Node-Daten, die den Auswahlkriterien entsprechen, werden zu den Dateien gespeichert. Die Ausgabe besteht aus dem Pfadnamen zu jedem gespeicherten I-Node sowie weiteren Informationen zu Dateien, die mit den Optionen unten angefordert werden. Die Ausgabefelder werden mit Tabulatoren positioniert. Die Ausgabe wird in der Reihenfolge der I-Nodes generiert; die Felder sind durch Tabulatoren voneinander getrennt. Die von ff erzeugte Standardzeile lautet:

pfadname i-nummer

Die Optionen sind:

-F ufs

Gibt ufs als Dateisystem-Typ an.

-o Gibt spezifische Optionen zum ufs-Dateisystem an. Folgende Optionen stehen zur Verfügung:

a Gibt die Verzeichniseinträge '.' und '..' aus.

m Gibt Modusinformationen aus.

s Gibt nur Gerätedateien und Dateien mit dem Modus Set-User-Id (s-Bit für Benutzer setzen) aus.

HINWEIS

Wird die Option -l nicht angegeben, wird für einen mehrfach verbundenen (link) I-Node nur einer der möglichen Pfadnamen generiert. Wird -l angegeben, umfaßt die Ausgabe alle Namen jeder verbundenen Datei im Dateisystem. Auf die generierten Namen werden jedoch keine Auswahlkriterien angewendet.

SIEHE AUCH

find(1), generisches Kommando ff(1M), ncheck(1M).

BEZEICHNUNG

`ffile` – erstellt ein vollständiges Dateisystemarchiv oder stellt die Dateien darin wieder her

ÜBERSICHT

`ffile -B [-dlmortvAENSV] sichauftrnr udsname udsg udskenn beschr`

`ffile -RC [-dlmortvAENSV] udsname udsg wdsname wger wauftrnr beschr`

`ffile -RF [-dlmortvAENSV] udsname udsg beschr wauftrnr:uid:datum:typ:name
[:wname][:i-node]] ...`

BESCHREIBUNG

Das Kommando `ffile` wird von anderen Shell-Kommandos als Sohnprozeß aufgerufen. Der Kommandoname `ffile` wird entweder aus der Datei `bkhist.tab` oder aus dem Kommando (mit Option) `bkreg -m` gelesen. Die Optionen `-B`, `-R`, `-F` und `-C` werden von den Shell-Kommandos `backup`, `restore` und `urestore` an `ffile` gesendet. Die anderen Optionen werden von `bkhist.tab` oder dem Kommando (mit Option) `bkreg -p` übergeben. Die Argumente werden von verschiedenen Stellen im Backup-Service an `ffile` übergeben.

`ffile -B` wird als Sohnprozeß von `bkdaemon` aufgerufen, um eine Sicherung des Dateisystems *udsname* (ursprünglicher Dateisystemname) auszuführen. Alle Dateien in *udsname* werden archiviert. Die daraus resultierende Sicherung wird in dem Format angelegt, das in `cpio(4)` beschrieben ist. Die Sicherung wird im Sicherungsprotokoll `/usr/oam/bkrs/tables/bkhist.tab` aufgezeichnet.

`ffile -RC` und `RF` werden als Sohnprozesse von `rsoper` aufgerufen, um Dateien aus einem vollständigen Dateisystemarchiv wiederherzustellen, das mit `ffile -B` angelegt wurde. Dabei wird davon ausgegangen, daß das Dateisystemarchiv das in `cpio(4)` beschriebene Format hat.

Wenn die Option `-RC` ausgewählt wird, wird das gesamte Dateisystem wiederhergestellt.

Wenn die Option `-RF` angegeben wird, werden nur ausgewählte Objekte aus dem Archiv wiederhergestellt. Jedes 7-Tupel, bestehend aus *wauftrnr:uid:datum:typ:name:wname:i-node*, gibt ein Objekt an, das aus dem Dateisystemarchiv wiederhergestellt werden soll. Die 7-Tupel Objekte werden an `ffile` von `rsstatus.tab` übergeben.

Die Argumente zu `ffile` sind folgendermaßen definiert:

sichauftrnr ist die von `backup` zugeordnete Auftragsnummer für die Sicherung. Bei dem Verfahren wird die *sichauftrnr* benutzt, wenn Protokolle und Inhaltsverzeichnis-Einträge erstellt werden.

udsname ist der Name des (ursprünglichen) Dateisystems, das gesichert werden soll.

udsg ist der Name des blockorientierten Geräts, auf dem das Dateisystem gespeichert ist.

udskenn ist der Datenträgerkennsatz des Dateisystems (siehe `labelit(1M)`).

<i>beschr</i>	ist eine Beschreibung des Zielgeräts in der Form: <i>zgruppe:zname:zchar:zkenns</i> <i>zgruppe</i> gibt die Zielgerätegruppe an (siehe <i>devgroup.tab(4)</i>). <i>zname</i> gibt den Namen des Zielgeräts an (siehe <i>device.tab(4)</i>). <i>zchars</i> gibt die Charakteristika des Geräts an. Wenn <i>zchar</i> angegeben wird, werden die Standardwerte für das angegebene Gerät und die angegebene Gruppe außer Kraft gesetzt (eine nähere Beschreibung von Gerätecharakteristika enthält <i>device.tab(4)</i>). <i>zkenns</i> gibt die Datenträgernamen für die Medien an, auf die das Archiv geschrieben bzw. von denen es gelesen werden soll.
<i>wsdname</i>	gibt, wenn der Wert nicht Null beträgt, den Namen des Dateisystems an, auf das anstelle von <i>udsname</i> das wiederherzustellende Dateisystem geschrieben werden soll. Mindestens eine der Variablen <i>wsdname</i> oder <i>wger</i> muß Null sein.
<i>wger</i>	gibt, wenn der Wert nicht Null beträgt, die Partition an, auf die anstelle von <i>udsgger</i> das wiederherzustellende Dateisystem geschrieben werden soll. Mindestens eine der Variablen <i>wsdname</i> oder <i>wger</i> muß Null sein.
<i>wauftrnr</i>	ist die Nummer des Wiederherstellungsauftrags, die von <i>restore</i> oder <i>urestore</i> zugewiesen wird.
<i>uid</i>	ist die reale Benutzernummer des Benutzers, der die Wiederherstellung des Objekts angefordert hat. Bei dieser Nummer muß es sich um die reale Benutzernummer des Superusers oder um die reale Benutzernummer des Benutzers handeln, der zu der Zeit, als das Archiv erstellt wurde, Eigentümer des Objekts war.
<i>datum</i>	ist das Datum der letzten Änderung ("last modification time"), das für ein wiederherzustellendes Objekt akzeptabel ist. Das Objekt wird aus dem Archiv mit unmittelbar vorausgehendem Datum wiederhergestellt. <i>datum</i> ist die hexadezimale Darstellung der Werte Zeit und Datum, die im Systemaufruf <i>time</i> angegeben werden (siehe <i>time(2)</i>).
<i>typ</i>	ist entweder F oder D, womit angegeben wird, daß es sich bei dem Objekt um eine Datei (file) bzw. um ein Verzeichnis (directory) handelt.
<i>name</i>	ist der Name, den das Objekt im Dateisystemarchiv hatte.
<i>wname</i>	ist der Name, unter dem das Objekt wiederhergestellt werden soll. Dieser Name kann sich von dem Namen unterscheiden, den das Objekt im Dateisystemarchiv hatte. Wird diese Variable nicht angegeben, wird das Objekt als <i>name</i> wiederhergestellt.
<i>i-node</i>	ist die I-Node-Nummer des Objektes, unter der es im Dateisystemarchiv gespeichert wurde. <i>i-node</i> wird von <i>ffile -R</i> nicht benutzt. Der Wert wird nur aus Gründen der Kompatibilität mit anderen Wiederherstellungsverfahren angegeben.

Optionen

Einige Optionen sind nur beim Kommando `ffile -B` von Bedeutung; beim Kommando `ffile -R` werden sie akzeptiert, aber ignoriert, da das Kommando automatisch von `restore` aufgerufen wird und die Optionen automatisch angegeben werden. Diese Optionen sind durch einen Stern (*) gekennzeichnet.

- d* Verhindert, daß die Archivierung im Sicherungsprotokoll aufgezeichnet wird.
- l* Erstellt ein Sicherungsprotokoll in Langform, das ein Inhaltsverzeichnis für das Archiv mit einschließt. Dieses besteht aus den Daten, die zur Generierung der einzelnen Dateien im Archiv benutzt werden (wie beim Kommando `ls -l`).
- m* Hängt das Ursprungsdateisystem mit Nur-Lese-Berechtigung ein, bevor der Sicherungsvorgang beginnt, und hängt es nach dem Sicherungsvorgang mit den ursprünglichen Berechtigungen nochmals ein. Kann nicht mit den Dateisystemen `root` oder `/usr` benutzt werden.
- o Ermöglicht es dem Benutzer, Anforderungen, Datenträger einzulegen, außer Kraft zu setzen [siehe `getvol(1M)` und die Beschreibung der Option `-o`].
- r* Schließt fern eingehängte Ressourcen bei der Archivierung mit ein.
- t* Erstellt ein Inhaltsverzeichnis für die Sicherung auf zusätzlichen Medien, nicht im Sicherungsprotokoll.
- v* Überprüft das Archiv bei der Erstellung. Während es geschrieben wird, wird eine Prüfsumme errechnet. Wenn ein Medium vollständig beschrieben ist, wird es nochmals gelesen und die Prüfsumme errechnet. Damit wird geprüft, ob jeder Datenblock lesbar und korrekt ist. Schlägt die Überprüfung fehl, gilt das Medium als nicht lesbar. Wurde die Option `-A` angegeben, schlägt die Archivierung fehl; andernfalls wird der Bediener aufgefordert, das fehlerhafte Medium zu ersetzen.
- A Aktiviert den automatischen Modus, d.h. der Benutzer wird nicht dazu aufgefordert, Medien einzulegen oder zu entfernen.
- E* Gibt eine Schätzung aus, wieviel Speicherplatz auf den Medien für das Archiv benötigt wird, dann wird die Sicherung durchgeführt.
- N* Gibt eine Schätzung aus, wieviel Speicherplatz auf den Medien für das Archiv benötigt wird, es wird jedoch keine Sicherung durchgeführt.
- S Zeigt für jeweils 100 Datenblöcke zu 512 Byte, die in das Archiv auf dem Zielgerät geschrieben oder von diesem gelesen wurden, einen Punkt (.) an.
- V Zeigt den Namen jeder Datei an, die in das Archiv auf dem Zielgerät geschrieben oder aus diesem gelesen wurde.

Dialog mit dem Benutzer

Die Verbindung zwischen Archivierungsmethode und `backup` ist komplexer als eine einfache Verzweigung (`fork/exec`) oder Pipe. Das Kommando `backup` führt den Dialog mit dem Benutzer, entweder direkt oder über `bkoper`. Daher liest `ffile` weder von der Standardeingabe, noch schreibt es auf die Standardausgabe

oder die Standardfehlerausgabe. Eine Verfahrensbibliothek muß für Berichte (Schätzungen, Dateinamen, Perioden, Statusangaben etc.) an `backup` benutzt werden (siehe `libbrmeth(3)`).

EXIT-CODES

Die Exit-Codes von `ffile` haben folgende Bedeutung:

- 0 = Aufgabe wurde erfolgreich ausgeführt
- 1 = ein oder mehrere Parameter zu `ffile` sind ungültig
- 2 = ein Fehler trat auf, wodurch `ffile` nicht alle Teile der Aufgabe ausführen konnte

DATEIEN

<code>/usr/oam/bkrs/tables/bkexcept.tab</code>	listet die Dateien auf, die bei einer Teilsicherung eines Dateisystems nicht berücksichtigt werden sollen
<code>/usr/oam/bkrs/tables/bkhist.tab</code>	listet die Kennsätze aller Datenträger auf, die für Sicherungsoperationen verwendet wurden
<code>/usr/oam/bkrs/tables/rsstatus.tab</code>	verfolgt den Status aller Wiederherstellungsanforderungen von Benutzern
<code>/usr/oam/bkrs/logs/bklog</code>	protokolliert die Fehler, die bei den Sicherungsverfahren und vom Kommando <code>backup</code> generiert wurden
<code>/usr/oam/bkrs/logs/rslog</code>	protokolliert die Fehler, die bei den Wiederherstellungsverfahren und vom Kommando <code>restore</code> gemeldet wurden
<code>\$TMP/filelist\$\$</code>	speichert vorübergehend ein Inhaltsverzeichnis für ein Sicherungsarchiv

SIEHE AUCH

`backup(1M)`, `bkoper(1M)`, `cpio(1)`, `cpio(4)`, `device.tab(4)`, `fdp(1)`, `ffile(1)`, `fimage(1)`, `getvol(1M)`, `incfile(1)`, `labelit(1M)`, `libbrmeth(3)`, `ls(1)`, `restore(1M)`, `rsoper(1M)`, `time(2)`, `urestore(1)`

BEZEICHNUNG

`fimage` – erstellt ein Image-Archiv eines Dateisystems oder stellt Dateien aus einem solchen wieder her

ÜBERSICHT

```
fimage -B [-dlmotuvAENS] sichauftrnr udsname udsger udskenn beschr
fimage -RC [-dlmotuvAENS] udsname udsger udsname wger wauftrnr beschr
fimage -RF [-dlmotuvAENS] udsname udsger beschr wauftrnr:uid:datum:typ:name
[:wname][:i-node]] ...
```

BESCHREIBUNG

Das Kommando `fimage` wird von anderen Shell-Kommandos als Sohnprozeß aufgerufen. Der Kommandoname `fimage` wird entweder aus der Datei `bkhist.tab` oder aus dem Kommando (mit Option) `bkreg -m` gelesen. Die Optionen `-B`, `-R`, `-F` und `-C` werden von den unten beschriebenen Shell-Kommandos `backup`, `restore` und `urestore` an `fimage` übergeben. Die anderen Optionen werden von der Datei `bkhist.tab` oder dem Kommando (mit Option) `bkreg -p` übergeben. Die Argumente werden von verschiedenen Stellen im Backup-Service an `fimage` gesendet. `fimage` liest weder von der Standardeingabe, noch schreibt es auf die Standardausgabe oder auf die Standardfehlerausgabe.

`fimage -B` wird als Sohnprozeß von `bkdaemon` aufgerufen, um eine Image-Sicherung des Dateisystems *udsname* (ursprüngliches Dateisystem) anzulegen. Alle Dateien in *udsname* werden archiviert. Die resultierende Sicherung wird in dem Format angelegt, das in `volcopy(1M)` beschrieben ist. Die Sicherung wird im Sicherungsprotokoll `/etc/bkup/bkhist.tab` aufgezeichnet.

`fimage -RC` und `-RF` werden als Sohnprozesse vom Kommando `rsoper` aufgerufen, um Dateien aus einem mit `fimage -B` angelegten Image-Archiv wiederherzustellen. Dabei wird davon ausgegangen, daß das Dateisystemarchiv das in `volcopy` beschriebene Format hat.

Wenn die Option `-RC` ausgewählt wird, wird das gesamte Dateisystem wiederhergestellt.

Wenn die Option `-RF` angegeben wird, werden nur ausgewählte Objekte aus dem Archiv wiederhergestellt. Jedes 7-Tupel, bestehend aus *wauftrnr:uid:datum:typ:name:wname:i-node*, gibt ein Objekt an, das aus dem Dateisystemarchiv wiederhergestellt werden soll. Die 7-Tupel Objekte werden an `fimage` von der Datei `rsstatus.tab` übergeben.

Die Argumente zu `fimage` sind folgendermaßen definiert:

sichauftrnr ist die von `backup` zugeordnete Sicherungsauftragsnummer. Bei dem Verfahren wird die *sichauftrnr* benutzt, wenn das Protokoll und die Inhaltsverzeichnis-Einträge erstellt werden.

udsname ist der Name des Dateisystems, von dem eine Sicherung erstellt werden soll.

udsger ist der Name des blockorientierten Geräts, auf dem das Dateisystem gespeichert ist.

<i>udskenn</i>	ist der Datenträgername (Kennsatz) des Dateisystems (siehe <i>labelit(1M)</i>).
<i>beschr</i>	ist eine Beschreibung des Zielgeräts in der Form: <i>zgruppe:zname:zchar:zkenns</i> <i>zgruppe</i> gibt die Zielgerätegruppe an (siehe <i>devgroup.tab(4)</i>). <i>zname</i> gibt den Namen des Zielgeräts an (siehe <i>device.tab(4)</i>). <i>zchars</i> gibt die Charakteristika des Zielgeräts an. Wenn <i>zchar</i> angegeben wird, werden die Standardwerte für die angegebene Gruppe und das angegebene Gerät außer Kraft gesetzt (<i>device.tab(4)</i> enthält eine nähere Erläuterung von Gerätemerkmalen). <i>zkenns</i> gibt die Datenträgernamen für die Medien an, auf die das Archiv geschrieben bzw. von denen es gelesen werden soll.
<i>wdsname</i>	gibt, wenn der Wert nicht Null beträgt, den Namen des Dateisystems an, auf das anstelle von <i>udskenn</i> das wiederherzustellende Dateisystem geschrieben werden soll. Mindestens eine der Variablen <i>wdsname</i> oder <i>wger</i> muß Null sein.
<i>wger</i>	gibt, wenn der Wert nicht Null beträgt, die Partition an, auf die anstelle von <i>udsker</i> das wiederherzustellende Dateisystem geschrieben werden soll. Mindestens eine der Variablen <i>wdsname</i> oder <i>wger</i> muß Null sein.
<i>wauftrnr</i>	ist die Nummer des Wiederherstellungsauftrags, die von <i>restore</i> oder <i>urestore</i> zugewiesen wird.
<i>uid</i>	ist die reale Benutzernummer des Benutzers, der die Wiederherstellung des Objekts angefordert hat. Bei dieser Nummer muß es sich um die reale Benutzernummer eines privilegierten Benutzers oder um die reale Benutzernummer des Benutzers handeln, der zu der Zeit, als das Archiv erstellt wurde, Eigentümer des Objekts war.
<i>datum</i>	ist das Datum der letzten Änderung ("last modification time"), das für ein wiederherzustellendes Objekt akzeptabel ist. Das Objekt wird aus dem Archiv mit unmittelbar vorausgehendem Datum wiederhergestellt. <i>datum</i> ist die hexadezimale Darstellung der Werte Zeit und Datum, die im Systemaufruf <i>time</i> angegeben werden (siehe <i>time(2)</i>).
<i>typ</i>	ist entweder F oder D, womit angegeben wird, daß es sich bei dem Objekt um eine Datei (file) bzw. um ein Verzeichnis (directory) handelt.
<i>name</i>	ist der Name, den das Objekt im Dateisystemarchiv hatte.
<i>wname</i>	ist der Name, unter dem das Objekt wiederhergestellt werden soll. Dieser Name kann sich von dem Namen unterscheiden, den das Objekt im Dateisystemarchiv hatte. Wird diese Variable nicht angegeben, wird das Objekt als <i>name</i> wiederhergestellt.
<i>i-node</i>	ist die I-Node-Nummer des Objektes, unter der es im Dateisystemarchiv gespeichert wurde. <i>i-node</i> wird von <i>ffile -R</i> nicht benutzt. Der Wert wird nur aus Gründen der Kompatibilität mit anderen Wiederherstellungsverfahren angegeben.

Optionen

Einige Optionen sind nur beim Kommando `fimage -B` von Bedeutung; beim Kommando `fimage -R` werden sie akzeptiert, aber ignoriert, da das Kommando automatisch von `restore` aufgerufen wird und die Optionen automatisch angegeben werden. Diese Optionen sind durch einen Stern (*) gekennzeichnet.

- d* Verhindert, daß die Archivierung im Sicherungsprotokoll aufgezeichnet wird.
- l* Erstellt ein Sicherungsprotokoll in Langform, das ein Inhaltsverzeichnis für das Archiv mit einschließt. Dieses besteht aus den Daten, die zur Generierung einer Auflistung der einzelnen Dateien im Archiv benutzt werden (wie beim Kommando `ls -l`).
- m* Hängt das Ursprungsdateisystem mit Nur-Lese-Berechtigung ein, bevor der Sicherungsvorgang beginnt, und hängt es nach dem Sicherungsvorgang mit den ursprünglichen Berechtigungen nochmals ein. Kann nicht mit den Dateisystemen `root` oder `/usr` benutzt werden.
- o Ermöglicht es dem Benutzer, Anforderungen, Datenträger einzulegen, außer Kraft zu setzen [siehe `getvol(1M)` und die Beschreibung der Option `-o`].
- t* Erstellt ein Inhaltsverzeichnis für die Sicherung auf einem zusätzlichen Medium, nicht im Sicherungsprotokoll.
- u* Hängt das Ursprungsdateisystem mit Nur-Lese-Berechtigung aus, bevor der Sicherungsvorgang beginnt, und hängt es nach dem Sicherungsvorgang mit den ursprünglichen Berechtigungen wieder ein. Kann nicht mit den Dateisystemen `root` oder `usr` benutzt werden. Die Option `-u` setzt die Option `-m` außer Kraft.
- v* Überprüft das Archiv bei der Erstellung. Während es geschrieben wird, wird eine Prüfsumme errechnet. Wenn ein Medium vollständig beschrieben ist, wird es nochmals gelesen und die Prüfsumme errechnet. Damit wird geprüft, ob jeder Datenblock lesbar und korrekt ist. Schlägt die Überprüfung fehl, gilt das Medium als nicht lesbar. Wurde die Option `-A` angegeben, schlägt die Archivierung fehl; andernfalls wird der Bediener aufgefordert, das fehlerhafte Medium zu ersetzen.
- A Aktiviert den automatischen Modus, d.h. der Benutzer wird nicht dazu aufgefordert, Medien einzulegen oder zu entfernen.
- E* Gibt eine Schätzung aus, wieviel Speicherplatz auf den Medien für das Archiv benötigt wird, dann wird die Sicherung durchgeführt.
- N* Gibt eine Schätzung aus, wieviel Speicherplatz auf den Medien für das Archiv benötigt wird, es wird jedoch keine Sicherung durchgeführt.
- S Zeigt für jeweils 100 Datenblöcke zu 512 Byte, die in das Archiv auf dem Zielgerät geschrieben oder von diesem gelesen wurden, einen Punkt (.) an.

Dialog mit dem Benutzer

Die Verbindung zwischen Archivierungsmethode und `backup` ist komplexer als eine einfache Verzweigung (`fork/exec`) oder Pipe. Das Kommando `backup` führt den Dialog mit dem Benutzer, entweder direkt oder über `bkoper`. Daher liest `fimage` weder von der Standardeingabe, noch schreibt es auf die Standardausgabe oder die Standardfehlerausgabe. Eine Verfahrensbibliothek muß für Berichte (Schätzungen, Dateinamen, Perioden, Statusangaben etc.) an `backup` benutzt werden (siehe `libbrmeth(3)`).

EXIT-CODES

Die Exit-Codes von `fimage` haben folgende Bedeutung:

- 0 = Aufgabe wurde erfolgreich ausgeführt
- 1 = ein oder mehrere Parameter zu `fimage` sind ungültig
- 2 = ein Fehler trat auf, wodurch `fimage` nicht alle Teile der Aufgabe ausführen konnte

DATEIEN

<code>/etc/bkup/bkhist.tab</code>	listet die Kennsätze aller Datenträger auf, die für Sicherungsoperationen verwendet wurden
<code>/etc/bkup/rsstatus.tab</code>	verfolgt den Status aller Wiederherstellungsanforderungen von Benutzern
<code>/etc/bkup/bklog</code>	protokolliert die Fehler, die bei den Sicherungsverfahren und vom Kommando <code>backup</code> generiert wurden
<code>/etc/bkup/rslog</code>	protokolliert die Fehler, die bei den Wiederherstellungsverfahren und vom Kommando <code>restore</code> generiert wurden
<code>\$TMP/filelist\$\$</code>	speichert vorübergehend ein Inhaltsverzeichnis für ein Sicherungsarchiv

SIEHE AUCH

`backup(1M)`, `bkoper(1M)`, `device.tab(4)`, `fdp(1)`, `ffile(1)`, `fimage(1)`, `getvol(1M)`, `incfile(1)`, `labelit(1M)`, `libbrmeth(3)`, `ls(1)`, `restore(1M)`, `rsoper(1M)`, `time(2)`, `urestore(1)`, `volcopy(1M)`

BEZEICHNUNG

fingerd, in.fingerd – Server für Informationen über ferne Benutzer

ÜBERSICHT

in.fingerd

BESCHREIBUNG

in.fingerd implementiert den Serveranteil des Name/Finger-Protokolls, das in RFC 742 beschrieben wird. Das Name/Finger-Protokoll bietet eine ferne Schnittstelle zu Programmen, die Informationen zum Systemstatus und zu einzelnen Benutzern anzeigen. Das Protokoll legt das Format des Austauschs zwischen Server und Client nur in geringem Maße fest. Der Client sendet eine einzelne Kommandozeile zum Finger-Server, der eine ausdrückbare Antwort schickt.

in.fingerd wartet auf eine Verbindung über TCP-Anschluß 79. Ist die Verbindung hergestellt, wird eine einzelne Kommandozeile gelesen, die mit dem Zeichen RETURN-LINE-FEED endet, welche an in.finger(1) übergeben wird. in.fingerd beendet die Verbindung, sobald die Ausgabe abgeschlossen ist.

Wenn die Zeile den Wert Null hat, wenn also nur das Zeichen RETURN-LINE-FEED gesendet wird, gibt in.finger einen Standardbericht zurück, in dem alle Benutzer aufgelistet sind, die zu dem gegebenen Zeitpunkt am System angemeldet sind.

Wenn ein Benutzername angegeben wird, z.B. erichRETURN-LINE-FEED, enthält die Antwort ausführlichere Informationen ausschließlich zu diesem Benutzer, unabhängig davon, ob er gerade angemeldet ist oder nicht. Zulässige Namen in der Kommandozeile sind der Login-Name und Benutzername. Ist ein Name nicht eindeutig, werden alle Ableitungen zurückgegeben.

DATEIEN

/var/utmp	angemeldete Benutzer
/etc/passwd	Benutzernamen
/var/adm/lastlog	Zeit der letzten Anmeldung
\$HOME/.plan	Pläne
\$HOME/.project	Projekte

SIEHE AUCH

finger(1)

Harrenstien, Ken, *NAME/FINGER*, RFC 742, Network Information Center, SRI International, Menlo Park, Kalif., Dezember 1977

HINWEIS

Versucht man, eine direkte Verbindung zum Server über ein Benutzerprogramm auf der Basis von TIP oder einem ähnlich anfälligen TELNET-Protokoll herzustellen, so kann dies dazu führen, daß Server und Client sich über Optionen zu verständigen versuchen, die für den Server bedeutungslos sind. Dadurch wird die Interpretation der Kommandozeile unmöglich. in.fingerd muß so angelegt werden, daß IACs herausgefiltert werden oder daß Optionskommandos sogar eine negative Rückmeldung (IAC unmöglich) erzeugen.

BEZEICHNUNG

fixperm - XENIX-Dateiberechtigungen und -Eigentümerrechte korrigieren oder einrichten

ÜBERSICHT

fixperm [-acDfgilnSsw [-d *paket*] [-u *paket*]] *spez_datei*

BESCHREIBUNG

Zu jeder Zeile in der Spezifikationsdatei *spez_datei* wird der aufgelistete Pfadname von fixperm so umgewandelt, daß er einer Spezifikation entspricht. fixperm wird in der Regel von einem privilegierten Benutzer verwendet, um ein XENIX-System bei der Installation zu konfigurieren. Das Kommando steht für alle erhältlichen XENIX-Pakete zur Verfügung, die auf einem SINIX-System installiert werden können. Nicht privilegierte Benutzer können fixperm nur mit den Optionen -D, -f, -l oder -n benutzen.

Folgende Optionen stehen zur Verfügung:

- a Alle Dateien in der Berechtigungsdatei müssen vorhanden sein. Das bedeutet, daß Dateien, die als optional gekennzeichnet sind (Typkennzeichen in Großbuchstaben), vorhanden sein müssen.
- c Leere Dateien und fehlende Verzeichnisse werden angelegt.
- D Listet Verzeichnisse nur auf der Standardausgabe auf. Ändert die Zieldateien nicht.
- d *paket* Verarbeitet die Eingabezeilen, die mit der Zeichenkette beginnen, die das Paket kennzeichnet (siehe oben). Beispielsweise werden bei -dBASE nur die Elemente verarbeitet, von denen angegeben ist, daß sie zu den Basic Utilities gehören. Standardmäßig werden alle Zeilen verarbeitet.
- f Listet Dateien nur auf der Standardausgabe auf. Modifiziert die Zieldateien nicht.
- g Listet alle Geräte auf der Standardausgabe auf. Die Zieldateien werden nicht modifiziert (analog zu -l, -f und -D).
- i Überprüft, ob die ausgewählten Pakete installiert sind. Die Rückgabewerte lauten
 - 0: Paket vollständig installiert
 - 4: Paket nicht installiert
 - 5: Paket teilweise installiert

Wurde das äquivalente Paket als SINIX-Paket installiert, findet -i es nicht.
- l Listet Dateien und Verzeichnisse auf der Standardausgabe auf. Modifiziert die Zieldateien nicht.
- n Meldet nur Fehler. Modifiziert die Zieldateien nicht.

- S Gibt eine Meldung aus, wenn die Dateien nicht im Format x.out vorliegen.
- s Modifiziert zusätzlich zu der restlichen Berechtigungsliste die Gerätedatei.
- u *paket* Bewirkt eine ähnliche Aktion wie die Option -d, verarbeitet aber Elemente, die nicht zum gegebenen Paket gehören.
- v (*meldung*) Gibt eine Meldung aus, wenn ausführbare Dateien:
 - 1) "word-swapped" (Wort-Swapping),
 - 2) "not fixed-stack" (kein Feststapel),
 - 3) "not separate I and D" (keine Trennung zwischen I und D), oder
 - 4) "not stripped" (nicht bereinigt) sind.
- w Listet die Position (Datenträgernummer) der angegebenen Dateien oder Verzeichnisse auf.

Format der Spezifikationsdatei

Jede nicht leere Zeile in der Spezifikationsdatei enthält entweder einen Kommentar oder eine Elementspezifikation. Ein Kommentar ist ein beliebiger Text, der mit dem Zeichen # beginnt und bis zum Zeilenende geht. Pro Zeile gibt es eine Elementspezifikation. Benutzer- und Gruppennummer müssen für jeden Benutzer und jede Gruppe, die in der Datei erwähnt werden, am Anfang der Spezifikationsdatei angegeben werden.

Eine Elementspezifikation besteht aus einem Paketkennzeichen, einer Berechtigungsspezifikation, Eigentümer- und Gruppenspezifikationen, der Anzahl der Verbindungen (links), die zu der Datei bestehen, dem Dateinamen und einer optionalen Datenträgernummer.

Das Paketkennzeichen ist eine beliebige Zeichenkette, die dem Namen eines Pakets innerhalb eines Programmsatzes entspricht. Ein Paket ist ein Satz von Dateien.

Eine Berechtigungsspezifikation folgt auf das Paketkennzeichen. Die Berechtigungsspezifikation besteht aus dem Dateityp, gefolgt von einer numerischen Berechtigungsspezifikation. Die Elementspezifikation entspricht einem der folgenden Zeichen:

Zeichen	Beschreibung
x	ausführbare Datei
a	Archivdatei
e	leere Datei (anlegen, wenn die Option -c angegeben wird)
b	blockorientiertes Gerät
c	zeichenorientiertes Gerät
d	Verzeichnis
f	Textdatei
p	benannte Pipe

fixperm (1M)

fixperm (1M)

Ist die Elementspezifikation ein Großbuchstabe, ist die zugehörige Datei optional, und `fixperm` gibt keine Fehlermeldung zurück, wenn die Datei nicht existiert.

Die numerische Berechtigung entspricht dem bei `chmod(1)` beschriebenen Schema. Die Eigentümer- und Gruppenberechtigungen stehen in der dritten Spalte und sind durch Schrägstrich getrennt, z.B. "bin/bin". Die vierte Spalte gibt die Anzahl der Verbindungen (links) an. Bestehen Verbindungen zu der Datei, enthält die nächste Zeile den Namen der verbundenen Datei ohne weitere Informationen. Die fünfte Spalte enthält einen Pfadnamen, der relativ angegeben werden muß, d.h. es darf ihm kein Schrägstrich "/" vorausgehen. Die sechste Spalte wird nur für Gerätedateien, Geräteklassennummern, Gerätenummern oder Datenträgernummern benutzt.

BEISPIELE

Mit den folgenden beiden Zeilen wird eine Verteilung durchgeführt und `tar (1)` aufgerufen, so daß nur die Dateien in `meinpaket` auf `/dev/beispiel` archiviert werden:

```
/etc/fixperm -f /etc/perm/meinpaket> list  
tar cff /dev/beispiel list
```

Diese Kommandozeile meldet Paketfehler:

```
/etc/fixperm -nd meinpaket
```

HINWEIS

`fixperm` wird in der Regel nur von einem Shell-Skript bei der Installation ausgeführt.

`fixperm` sollte nur von dem Verzeichnis aus ausgeführt werden, zu dem die Zieldateien gehören.

SIEHE AUCH

`custom(1M)`.

BEZEICHNUNG

fromsmtp – Empfang von RFC822-Post von SMTP

ÜBERSICHT

fromsmtp [-d] [-h *host*] [-s *absender*] to ...

BESCHREIBUNG

fromsmtp liest eine RFC822-Meldung von der Standardeingabe, konvertiert die Meldung, so daß sie für das Postsystem akzeptabel ist und sendet das Ergebnis per Pipe an *rmail*. Die Argumente *to* werden als Argumente an *rmail* übergeben. fromsmtp wird normalerweise von *smtpd* aufgerufen, um ankommende Post zuzustellen.

Die Option *-d* kann zur Fehlersuche bei fromsmtp verwendet werden. Durch diese Option wird ein Echo der Kommandozeile für *rmail* auf der Standardausgabe ausgegeben, ebenso wie die Ergebnisse der Meldung nach der Konvertierung. Die Meldung wird nicht an *rmail* übergeben, wenn diese Option benutzt wird.

Die Option *-h host* dient dazu, einen Host- oder Netzwerknamen an den Anfang des Absenderpfadnamens in der Zeile *From* am Beginn der Meldung zu stellen. Diese Option ist nützlich, wenn angegeben werden muß, aus welchem von mehreren möglichen Netzwerken eine Meldung stammt, so daß man zum Beispiel eine Antwort zurücksenden kann.

Über die Option *-s absender* wird ein Standardabsendername angegeben für den Fall, daß fromsmtp den Namen des Absenders nicht der gelesenen Meldung entnehmen kann. Wird diese Option nicht angegeben, wird der Standardabsendername *unknown* (unbekannt) benutzt.

DATEIEN

/usr/bin/rmail

an diese Stelle wird konvertierte Post über Pipe geschickt

SIEHE AUCH

rmail(1M), *smtpd*(1M)

RFC822 – Standard für das Format von ARPA Internet Meldungen

BEZEICHNUNG

fsba – Analyse der Blockgröße von s5-Dateisystemen

ÜBERSICHT

/usr/sbin/fsba [-b *zielblockgröße*] *dateisystem1* [*dateisystem2* ...]

BESCHREIBUNG

Das Kommando *fsba* stellt fest, wieviel Speicherplatz nötig ist, um die Daten eines vorhandenen Dateisystems in einem neuen Dateisystem mit der angegebenen logischen Blockgröße zu speichern. Jedes auf der Kommandozeile aufgelistete *dateisystem* bezieht sich auf ein vorhandenes Dateisystem und sollte als Gerätenamen (z.B. /dev/rdisk/c1d0s2) angegeben werden.

Die *zielblockgröße* gibt die Größe der logischen Blöcke des neuen Dateisystems in Byte an. Gültige Blockgrößen sind 512, 1024 und 2048. Der Standardwert ist 1024.

Das Kommando *fsba* gibt aus, wieviele Plattensektoren zu 512 Byte zur Datenspeicherung im alten (vorhandenen) Dateisystem reserviert sind und wieviele erforderlich wären, um diese Daten im neuen Dateisystem mit der angegebenen logischen Blockgröße zu speichern. Außerdem werden auch die Anzahl der zugeordneten und der freien I-Nodes für das vorhandene Dateisystem ausgegeben.

Wenn die Anzahl der freien Sektoren für das neue Dateisystem negativ ist, passen die Daten nur dann in das neue Dateisystem, wenn das neue Dateisystem größer ist als das vorhandene. Das neue Dateisystem muß mindestens so groß werden wie die Anzahl der Sektoren, die von *fsba* als dem neuen Dateisystem zugeordnet aufgelistet werden. Die Größe des neuen Dateisystems ist begrenzt durch die Größe der Partition, die für das neue Dateisystem benutzt wird.

Beachten Sie bitte, daß Sie eine *zielblockgröße* angeben können, die kleiner ist als die logische Blockgröße des vorhandenen Dateisystems. In diesem Fall erforderte das neue Dateisystem weniger Sektoren zur Speicherung der Daten.

SIEHE AUCH

mkfs(1M).

BEZEICHNUNG

fsck – prüft und korrigiert Dateisysteme

ÜBERSICHT

```
fsck [-F dateisystem-typ] [-V] [-m] [gerätedatei ...]
fsck [-F dateisystem-typ] [-V] [aktuelle_optionen] [-o spezifische_optionen]
      [gerätedatei ...]
```

BESCHREIBUNG

fsck überprüft Inkonsistenzen bei Dateisystemen und korrigiert sie im Dialog mit dem Benutzer. Ist das Dateisystem nicht konsistent, wird die Zustimmung des Benutzers eingeholt, bevor eine Korrektur vorgenommen wird. Bitte beachten Sie, daß bei einigen Korrekturmaßnahmen Daten verlorengehen können. Umfang und Bedeutung des Datenverlusts können anhand der Diagnosemeldungen bestimmt werden. Standardmäßig wartet das System bei jeder Korrektur, bis der Benutzer yes (ja) oder no (nein) eingibt. Hat der Benutzer keine Schreibberechtigung, gilt die Antwort no als Standard.

Das Dateisystem muß ausgehängt sein, wenn fsck benutzt wird. Ist dies nicht möglich, muß sich das System unbedingt im Ruhezustand befinden und unmittelbar nach der Korrektur neu geladen werden, sofern es sich um ein wichtiges Dateisystem, z.B. root, handelt.

aktuelle_optionen sind Optionen, die von dem ufs-spezifischen Modul von fsck unterstützt werden. Andere *dateisystem-typen* unterstützen diese Optionen nicht immer. *spezifische_optionen* bezeichnen Unteroptionen, die in einer Liste von durch Komma getrennten Unteroptionen und/oder Paaren von Schlüsselwort/Attribut aufgeführt sind und vom *dateisystem-typ*-spezifischen Modul des Kommandos interpretiert werden sollen.

gerätedatei steht für eine block- oder zeichenorientierte Gerätedatei (z.B. /dev/rdisk/cld0s8). Vorzugsweise ist ein zeichenorientiertes Gerät zu verwenden. fsck kann bei einer eingehängten blockorientierten Gerätedatei nicht ausgeführt werden. Wird *gerätedatei* nicht angegeben, wird /etc/vfstab durchsucht. fsck wird dann für alle zeichenorientierten Geräte im Feld fsckdev der Datei /etc/vfstab durchgeführt, für die das Feld fsckpass einen numerischen Eintrag enthält.

Folgende Optionen stehen zur Verfügung:

- F Gibt den *dateisystem-typ* an, mit dem gearbeitet werden soll. Der *dateisystem-typ* muß entweder hier angegeben werden oder sich durch Vergleich der *gerätedatei* mit einem Eintrag in der Tabelle in /etc/vfstab ableiten lassen. Wird *dateisystem-typ* nicht angegeben bzw. existiert *dateisystem-typ* in /etc/vfstab nicht, wird standardmäßig s5 angenommen.
- V Gibt die gesamte Kommandozeile auf dem Bildschirm aus, führt das Kommando jedoch nicht aus. Die Kommandozeile wird mit den vom Benutzer eingegebenen Optionen und Argumenten sowie den aus /etc/vfstab abgeleiteten Daten erstellt. Diese Option dient zur Überprüfung und Auswertung der Kommandozeile.

- m Prüft das Dateisystem, korrigiert es aber nicht. Mit dieser Option wird überprüft, ob das Dateisystem zum Einhängen geeignet ist.
- o Gibt *dateisystem-typ*-spezifische Optionen an.

HINWEIS

Diese Kommando wird eventuell nicht für alle *dateisystem-Typen* unterstützt.

DATEIEN

/etc/vfstab Liste der Standardparameter für jedes Dateisystem

SIEHE AUCH

checkfsys(1M), mkfs(1M), vfstab(4).

Die Handbucheinträge für die *dateisystemtyp*-spezifischen Module von *fsck*.

fsck(1M)

(bfs)

fsck(1M)

BEZEICHNUNG

fsck(bfs) - prüft und korrigiert bfs-Dateisysteme

ÜBERSICHT

fsck [-F bfs] [generische_optionen] [gerätedatei...]

fsck [-F bfs] [generische_optionen] [-y | -n] [gerätedatei...]

BESCHREIBUNG

generische_optionen sind Optionen, die das generische fsck-Kommando unterstützt.

fsck prüft, ob eine Komprimierung im Gange war, aber nicht abgeschlossen wurde, möglicherweise aufgrund eines Systemabsturzes. In diesem Falle beendet fsck die Komprimierung der Datei (siehe fs_bfs(4)).

Folgende Optionen stehen zur Verfügung:

-F bfs Gibt den Dateisystem-Typ bfs an.

-y Nimmt yes (Ja) als Antwort auf alle von fsck gestellten Fragen an.

-n Nimmt no (Nein) als Antwort auf alle von fsck gestellten Fragen an.

SIEHE AUCH

checkfsys(1M), generisches Kommando fsck(1M), mkfs(1M), fs_bfs(4)
Kapitel 4 des *Leitfadens für Systemverwalter*.

BEZEICHNUNG

fsck (s5) – prüft und korrigiert s5-Dateisysteme

ÜBERSICHT

fsck [-F s5] [*generische_optionen*] [*gerätedatei...*]

fsck [-F s5] [*generische_optionen*] [-y] [-n] [-p] [-sX] [-sX] [-t*datei*] [-l] [-q]
[-D] [-f] [*gerätedatei...*]

BESCHREIBUNG

generische_optionen sind Optionen, die vom generischen fsck-Kommando unterstützt werden.

Folgende Optionen stehen zur Verfügung:

- F s5 Gibt den Dateisystem-Typ s5 an.
- y Nimmt die Antwort yes (Ja) auf alle von fsck gestellten Fragen an.
- n Nimmt die Antwort no (Nein) auf alle von fsck gestellten Fragen an; das Dateisystem darf nicht zum Schreiben geöffnet werden.
- p Korrigiert Inkonsistenzen, die sich automatisch bereinigen lassen, d.h. Inkonsistenzen, die als nicht schwerwiegend angesehen werden und ohne Bestätigung durch den Systemverwalter korrigiert werden können. Beispiele für solche Inkonsistenzen sind I-Nodes ohne Verweis, fehlerhafte Zähler in den Superblöcken sowie fehlende Blöcke in der Liste der freien Blöcke.
- sX Ignoriert die Liste der freien Blöcke und erstellt (unbedingt) eine neue durch Überschreiben des Superblocks des Dateisystems. Das Dateisystem muß während dieses Vorgangs ausgehängt sein; ist dies nicht möglich, ist dafür zu sorgen, daß sich das System im Ruhezustand befindet und anschließend sofort neu geladen wird. Diese Vorsichtsmaßnahme ist notwendig, damit die alte, fehlerhafte Kopie des Superblocks im Hauptspeicher nicht weiterhin benutzt oder in das Dateisystem geschrieben wird.

Mit der Unteroption -sX kann eine optimal organisierte Liste der freien Blöcke erstellt werden.

Wird X nicht angegeben, werden die Werte verwendet, die beim Anlegen des Dateisystems galten. Das Format von X lautet *zylindergröße:blockzwischenraumgröße*.
- SX Bedingter Wiederaufbau der Liste der freien Blöcke. Diese Unteroption hat die gleiche Funktion wie -sX, allerdings wird die Liste der freien Blöcke nur dann neu aufgebaut, wenn innerhalb des Dateisystems keine Diskrepanzen festgestellt werden. Bei der Option S wird für alle von fsck gestellten Fragen die Antwort no (Nein) erzwungen. Diese Unteroption dient dazu, eine Reorganisation der Liste der freien Blöcke in unbeschädigten Dateisystemen zu erzwingen.
- t*datei* Findet fsck nicht genügend Speicherplatz für seine Tabellen, wird eine temporäre Datei verwendet. Ist die Option t angegeben, wird die angegebene *datei* bei Bedarf als temporäre Datei verwendet. Ohne die Option t fordert fsck den Benutzer auf, den Namen der temporären

Datei einzugeben. Die gewählte Datei darf nicht zu dem zu prüfenden Dateisystem gehören; wenn es sich nicht um eine Gerätedatei handelt und wenn sie noch nicht bestand, wird sie beim Abschluß von `fsck` gelöscht.

- l erkennt beschädigte Dateien an ihrem logischen Namen.
- q `fsck` ohne Meldungen. `fifos` ohne Referenz werden ohne Meldung entfernt. Auf Anforderung von `fsck` werden Zähler im Superblock automatisch korrigiert, und die Liste der freien Blöcke bleibt erhalten.
- D Verzeichnisse werden auf fehlerhafte Blöcke überprüft. Nützlich nach Systemabsturz.
- f Schnelle Prüfung der Blöcke, Blockgrößen und der Liste der freien Blöcke. Diese wird bei Bedarf neu erstellt. Folgende Inkonsistenzen werden überprüft:

1. Blöcke, die von mehreren I-Nodes oder der Liste der freien Blöcke beansprucht werden.
2. Blöcke, die von einem I-Node oder der Liste der freien Blöcke beansprucht werden und außerhalb des Bereichs des Dateisystems liegen.
3. Falsche Verweiszähler.
4. Größenprüfungen:
 - Falsche Anzahl von Blöcken.
 - Verzeichnisgröße nicht im 16-Byte-Format.
5. Fehlerhaftes I-Node-Format.
6. Blöcke, die nirgends aufgeführt sind.
7. Verzeichnisprüfungen:
 - Datei, die auf nicht zugewiesenen I-Node zeigt.
 - I-Node-Nummer außerhalb des zugewiesenen Bereichs.
8. Prüfungen des Superblocks:
 - Mehr als 65536 I-Nodes.
 - Mehr Blöcke für I-Nodes, als das Dateisystem enthält.
9. Fehlerhaftes Format der Liste der freien Blöcke.
10. Summe der freien Blöcke und /oder der freien I-Nodes fehlerhaft.

Verwaiste Dateien und Verzeichnisse (d.h. zugewiesen, aber ohne Referenz) werden mit Zustimmung des Benutzers neu verbunden, indem sie in das Verzeichnis `lost+found` geschrieben werden, sofern die Dateien nicht leer sind. Der Benutzer wird benachrichtigt, ob die Datei oder das Verzeichnis leer ist. Leere Dateien oder Verzeichnisse werden entfernt, sofern die Unteroption `n` nicht angegeben ist. `fsck` erzwingt die erneute Einbindung nicht leerer Verzeichnisse. Der zugewiesene Name entspricht der I-Node-Nummer.

HINWEIS

In nahezu allen Fällen läßt sich das zeichenorientierte Gerät schneller überprüfen.

Die I-Node-Nummern für `.` und `..` in den einzelnen Verzeichnissen unterliegen nicht der Gültigkeitsprüfung.

fsck(1M)

(s5)

fsck(1M)

SIEHE AUCH

checkfsys(1M), crash(1M), generisches Kommando fsck(1M), mkfs(1M),
ncheck(1M), fs(4).

BEZEICHNUNG

fsck (ufs) – Konsistenzprüfung des Dateisystems und Korrektur im Dialog mit dem Benutzer

ÜBERSICHT

```
fsck [ -F ufs ] [generische_optionen] [gerätedatei ...]
fsck [ -F ufs ] [generische_optionen] [ (-y|-Y) | (-n|-N) ] [ -o p,b=#,w ]
[ gerätedatei .... ]
```

BESCHREIBUNG

generische_optionen sind Optionen, die das generische fsck-Kommando unterstützt. *aktuelle_optionen* sind Optionen, die das ufs-spezifische Modul des Kommandos fsck unterstützt.

fsck prüft Dateisysteme auf Inkonsistenzen und korrigiert diese im Dialog mit dem Benutzer. In diesem Falle wird die Zustimmung des Benutzers eingeholt, bevor Korrekturmaßnahmen durchgeführt werden. Andere als die oben genannten Inkonsistenzen können zu Datenverlusten führen. Der Umfang und die Schwere der Verluste läßt sich anhand der Diagnosemeldungen bestimmen.

fsck korrigiert weniger schwerwiegende Inkonsistenzen wie I-Nodes ohne Referenz, zu hohe Verweiszähler in I-Nodes, fehlende Blöcke in der Liste der freien Blöcke, Blöcke, die in der Liste der freien Blöcke und gleichzeitig in Dateien erscheinen oder falsche Zähler im Superblock automatisch. Für jede korrigierte Inkonsistenz wird eine Meldung ausgegeben, die die Art der Korrektur und das Dateisystem, auf dem sie ausgeführt wird, anzeigt. Nach der Korrektur eines Dateisystems gibt fsck die Anzahl der Dateien in diesem Dateisystem, die Anzahl der belegten und der freien Blöcke und die entsprechenden Angaben in Prozent aus.

Standardmäßig wartet das Kommando vor jeder Korrektur auf die Antwort des Benutzers, die yes (Ja) oder no (Nein) lauten muß. Hat der Benutzer keine Schreibberechtigung für das Dateisystem, geht fsck standardmäßig von -n (keine Korrekturen) aus.

Folgende Inkonsistenzen werden überprüft:

1. Blöcke, die von mehreren I-Nodes oder der Liste der freien Blöcke beansprucht werden.
2. Blöcke, die von einem I-Node oder der Liste der freien Blöcke beansprucht werden und außerhalb des Bereichs des Dateisystems liegen.
3. Falsche Verweiszähler.
4. Falsche Verzeichnisgrößen.
5. Fehlerhaftes Format der I-Nodes.
6. Blöcke, die nirgends aufgeführt sind.
7. Verzeichnisprüfungen, Dateien, die auf einen nicht zugewiesenen I-Node zeigen, I-Node-Nummer außerhalb des zugewiesenen Bereichs, Fehlen von '.' und '..' als erste Einträge in den Verzeichnissen.
8. Prüfungen des Superblocks: Mehr Blöcke für I-Nodes, als das Dateisystem enthält.

9. Fehlerhaftes Format der Liste der freien Blöcke.
10. Summe der freien Blöcke und/oder der freien I-Nodes fehlerhaft.

Verwaiste Dateien und Verzeichnisse (d.h. zugewiesen, aber ohne Referenz) werden mit Zustimmung des Benutzers neu verbunden, indem sie in das Verzeichnis `lost+found` geschrieben werden. Der zugewiesene Name entspricht der I-Node-Nummer. Besteht das Verzeichnis `lost+found` noch nicht, wird es nun angelegt. Reicht der Platz nicht aus, wird es vergrößert.

Zur Bezeichnung eines Dateisystems kann der Name des block- oder zeichenorientierten Geräts, auf dem es steht, oder der Name des Einhängepunkts angegeben werden.

Folgende Optionen sind möglich:

- F `ufs`
Gibt `ufs` als Dateisystem-Typ an.
- y | -Y
Nimmt an, daß alle von `fsck` gestellten Fragen mit `yes` (Ja) beantwortet werden.
- n | -N
Geht davon aus, daß alle von `fsck` gestellten Fragen mit `no` (Nein) beantwortet werden; das Dateisystem darf nicht zum Schreiben geöffnet werden.
- o
Gibt für das `ufs`-Dateisystem spezifische Unteroptionen an. Diese Unteroptionen können eine Kombination aus folgenden Möglichkeiten sein:
 - p
Kein Dialog mit dem Benutzer. Abbrechen, wenn ein Problem festgestellt wird, das einen Eingriff des Benutzers erfordert.
 - b=#
Der angegebene Block soll als Superblock für das Dateisystem verwendet werden. Block 32 ist immer ein alternativer Superblock.

HINWEIS

In nahezu allen Fällen läßt sich das zeichenorientierte Gerät schneller überprüfen.

SIEHE AUCH

`checkfsys(1M)`, `crash(1M)`, generisches `fsck(1M)`, `mkfs(1M)`, `ufs(4)`.

BEZEICHNUNG

fsdb (generisch) – Debugger für die Fehlersuche in Dateisystemen

ÜBERSICHT

fsdb [-F *dateisystem-typ*] [-V] [*aktuelle_optionen*] [-o *spezifische_optionen*] *gerätedatei*

BESCHREIBUNG

fsdb ist ein Debugger für die manuelle Fehlersuche in Dateisystemen. Hiermit können Dateisysteme nach einem Systemabsturz wiederhergestellt werden. *gerätedatei* ist eine Gerätedatei, mit der das auf Fehler zu prüfende Dateisystem bezeichnet wird. fsdb ist nur für sehr erfahrene Benutzer bestimmt. *dateisystem-typ* ist der Dateisystem-Typ, für den die Fehlersuche durchgeführt werden soll. Da die einzelnen *dateisystem-typen* unterschiedliche Strukturen und somit unterschiedliche Fehlersuch-Funktionen haben, sind die eingehenderen Beschreibungen der jeweiligen Fehlersuch-Funktionen in den Handbucheinträgen für die *dateisystem-typ*-spezifischen fsdb-Kommandos nachzuschlagen.

aktuelle_optionen sind die Optionen, die das s5-spezifische Modul von fsdb unterstützt. Andere *dateisystem-typen* unterstützen diese Optionen nicht immer. *spezifische_optionen* bezeichnen Unteroptionen in einer durch Komma getrennten Liste von Unteroptionen und/oder Paaren von Schlüsselwort/Attribut, die von dem *dateisystem-typ*-spezifischen Modul des Kommandos interpretiert werden.

Folgende Optionen stehen zur Verfügung:

- F Gibt den zu bearbeitenden *dateisystem-typ* an. Der *dateisystem-typ* muß entweder hier angegeben werden oder anhand eines Vergleichs der *gerätedatei* mit einem Eintrag in der Tabelle in */etc/vfstab* abzuleiten sein.
- V Gibt die Kommandozeile vollständig auf dem Bildschirm aus, ohne das Kommando auszuführen. Die Kommandozeile setzt sich aus den vom Benutzer eingegebenen Optionen und Argumenten und ergänzenden, aus */etc/vfstab* abgeleiteten Daten zusammen. Diese Option dient zur Überprüfung und Auswertung der Kommandozeile.
- o Gibt *dateisystem-typ*-spezifische Optionen an.

HINWEIS

Dieses Kommando wird eventuell nicht für alle *dateisystem-typen* unterstützt.

DATEIEN

/etc/vfstab Liste der Standardparameter für die einzelnen Dateisysteme

SIEHE AUCH

mkfs(1M), vfstab(4).

Die Handbucheinträge zu den *dateisystem-typ*-spezifischen Modulen von fsdb.

BEZEICHNUNG

fsdb (s5) – Debugger für die Fehlersuche in s5-Dateisystemen

ÜBERSICHT

fsdb [-F s5] [*generische_optionen*] [-z *i-nummer*] *gerätedatei* [-]

BESCHREIBUNG

generische_optionen sind Optionen, die von dem generischen Kommando fsdb unterstützt werden.

fsdb kann zur Korrektur eines beschädigten s5-Dateisystems nach einem Systemabsturz verwendet werden. *gerätedatei* ist eine Gerätedatei, mit der das zu prüfende Dateisystem angegeben wird. Es enthält Umwandlungsfunktionen, um Block- und *i-nummern* in die entsprechenden Plattenadressen umzuwandeln. Ferner enthält es mnemonische Offsets zum Zugriff auf unterschiedliche Teile eines I-Node. Dadurch wird die Korrektur von Einträgen im Steuerblock oder der Weg durch den Dateisystembaum wesentlich vereinfacht.

fsdb enthält mehrere Routinen zur Überprüfung von I-Node- und Blockadressen. Diese können bei Bedarf durch Aufrufen von fsdb mit dem optionalen Argument - oder durch Verwendung des Symbols 0 deaktiviert werden. fsdb liest die Einträge "i-size" (i-größe) und "f-size" (f-größe) aus dem Superblock des Dateisystems als Grundlage für diese Prüfungen.

Folgende Optionen stehen zur Verfügung:

- F s5 Gibt s5 als Dateisystem-Typ an.
- z *i-nummer* Löscht den Inhalt des durch die *i-nummer* gekennzeichneten *i-node*; ohne Dialog mit dem Benutzer.

Zahlen sind standardmäßig Dezimalzahlen. Oktalzahlen muß eine Null vorausgehen. Während aller Zuweisungsoperationen werden Zahlen auf Rundungsfehler (truncation error) überprüft.

fsdb liest immer einen Block auf einmal und arbeitet daher mit Ein-/Ausgabe im Raw-Modus ebenso gut wie mit blockorientierter Ein-/Ausgabe. Mit einer bestimmten Routine werden häufig verwendete Datenblöcke in Puffern zurückgehalten, um die Anzahl der Systemaufrufe zum Lesen gering zu halten. Alle Zuweisungsoperationen bewirken eine sofortige Aktualisierung des entsprechenden Blocks.

Folgende Symbole erkennt fsdb:

- # absolute Adresse
- i Umwandlung von *i-nummer* zu I-Node-Adresse
- b Umwandlung in Blockadresse
- d Verzeichnisposition-Offset
- +, - Adreßarithmetik
- q Verlassen (quit)
- >, < Sichern, Wiederherstellen einer Adresse
- = numerische Zuweisung

= +	Zuweisung mit Erhöhung
= -	Zuweisung mit Erniedrigung
= "	Zeichenkettenzuweisung
O	Fehlerprüf-Flipflop
p	allgemeine Druckeinrichtungen
f	Dateidruckeinrichtung
B	Byte-Modus
W	Wortmodus
D	Doppelwortmodus
!	Sub-Shell aufrufen

Die Druckeinrichtungen erstellen formatierte Ausgaben in verschiedenen Formen. Die aktuelle Adresse wird vor dem Ausdrucken auf eine entsprechende Wortgrenze ausgerichtet. Sie bewegt sich mit dem Druckvorgang weiter und bleibt bei der Adresse des letzten ausgedruckten Elements stehen. Die Ausgabe kann jederzeit durch das Drücken der Taste DELETE beendet werden. Erscheint nach dem Symbol *p* eine Zahl, werden entsprechend viele Einträge gedruckt. Eine Prüfung dient zum Feststellen eines Blockgrenzenüberlaufs, da logisch aufeinanderfolgende Blöcke physisch im allgemeinen nicht aufeinanderfolgen. Wird der Zähler Null verwendet, werden alle Einträge bis zum Ende des aktuellen Blocks gedruckt. Folgende Druckoptionen stehen zur Verfügung:

i	Ausgabe als I-Nodes
d	Ausgabe als Verzeichnisse
o	Ausgabe als Oktalwörter
e	Ausgabe als Dezimalwörter
c	Ausgabe als Zeichen
b	Ausgabe als Oktal-Bytes

Das Symbol *f* dient zum Drucken von Datenblöcken, die zu dem aktuellen I-Node gehören. Folgt danach eine Zahl, wird der entsprechende Block der Datei gedruckt. Die Numerierung der Blöcke beginnt mit Null. Der Buchstabe für die gewünschte Druckoption folgt nach der Blocknummer, sofern vorhanden, oder dem Symbol *f*. Diese Druckfunktion funktioniert gleichermaßen bei kleinen und großen Dateien. Sie prüft, ob Gerätedateien vorhanden sind und ob die zum Auffinden der Daten verwendeten Blockzeiger nicht Null sind.

Punkte, Tabulatorzeichen und Leerzeichen können als Funktionsbegrenzer angegeben werden, sind aber nicht notwendig. Eine Zeile, die nur ein Zeilenende-Zeichen enthält, erhöht die aktuelle Adresse um die Größe des zuletzt gedruckten Datentyps. D.h. die Adresse wird auf das nächste Byte, Wort, Doppelwort, den nächsten Verzeichniseintrag oder I-Node gesetzt. Dadurch kann der Benutzer einen Bereich eines Dateisystems durchsuchen. Das Format der Ausgabe ist dem Datentyp angepaßt. Byte, Wörter und Doppelwörter werden mit der oktalen Adresse, gefolgt vom Wert in Oktal- und Dezimalschreibweise, angezeigt. An die Adresse wird *.B* oder *.D* angehängt, wenn es sich um Byte- bzw. Doppelwort-Werte handelt. Verzeichnisse werden als Verzeichnisposition-Offset, gefolgt von der dezimalen *i-nummer* und den Zeichen für den Eintragsnamen ausgegeben. I-Nodes werden mit gekennzeichneten Feldern, in denen die einzelnen Elemente beschrieben werden, ausgegeben.

Die folgenden mnemonischen Symbole werden für die Prüfung der I-Nodes verwendet und beziehen sich auf den aktuellen I-Node:

md	Modus
ln	Verweiszähler
uid	Benutzernummer
gid	Gruppennummer
sz	Dateigröße
a #	Datenblocknummern (0 – 12)
at	Zugriffszeit
mt	Änderungszeit
maj	Geräteklassennummer
min	Gerätenummer

BEISPIELE

386i	gibt die <i>i-nummer</i> 386 im I-Node-Format aus. Dieser wird nun zum aktuellen I-Node.
ln=4	ändert den Verweiszähler für den aktuellen I-Node in 4.
ln+=1	erhöht den Verweiszähler um 1.
fc	gibt Block 0 der zum aktuellen I-Node gehörenden Datei im ASCII-Format aus.
2i.fd	gibt die ersten 32 Verzeichniseinträge für den root-I-Node dieses Dateisystems aus.
d5i.fc	ändert den aktuellen I-Node in den I-Node, der zum fünften Verzeichniseintrag (vom vorhergehenden Kommando aus) gehört (Numerierung ab Null). Der erste logische Block der Datei wird dann im ASCII-Format ausgegeben.
512B.p0o	gibt den Superblock dieses Dateisystems im Oktalformat aus.
2i.a0b.d7=3	ändert die <i>i-nummer</i> für die siebte Verzeichnisposition im root-Verzeichnis in 3. Dieses Beispiel zeigt auch, daß mehrere Operationen in einer Kommandozeile zusammengefaßt werden können.
d7.rm="name"	ändert das Namensfeld in der Verzeichnisposition in die angegebene Zeichenkette. Ist das erste Zeichen ein Buchstabe, brauchen bei <i>rm</i> keine Anführungszeichen angegeben zu werden.
a2b.p0d	gibt den dritten Block des aktuellen I-Nodes in Form von Verzeichniseinträgen aus.

SIEHE AUCH

fsck(1M), generisches fsdb(1M).
 dir(4), fs(4) im Referenzhandbuch für Programmierer

fsdb (1M)

(UFS)

fsdb (1M)

BEZEICHNUNG

fsdb (ufs) – Debugger für die Fehlersuche in *ufs*-Dateisystemen

ÜBERSICHT

fsdb [-F *ufs*] [*generische_optionen*] [-z *i-nummer*] *gerätedatei*

BESCHREIBUNG

generische_optionen sind Optionen, die das generische Kommando *fsdb* unterstützt.

Folgende Optionen stehen zur Verfügung:

-F *ufs*

Gibt *ufs* als Dateisystem-Typ an.

-z *i-nummer*

Löscht den Inhalt der mit *i-nummer* gekennzeichneten I-Node; ohne Dialog mit dem Benutzer.

SIEHE AUCH

fsck(1M), generisches Kommando *fsdb*(1M), *dir*(4), *ufs*(4)

BEZEICHNUNG

`fstyp` (generisch) – bestimmt den Dateisystem-Typ

ÜBERSICHT

`fstyp [-v] gerätedatei`

BESCHREIBUNG

Mit `fstyp` kann der Benutzer anhand heuristischer Programme den Dateisystem-Typ ausgehängter Dateisysteme bestimmen.

Für jeden zu prüfenden Dateisystem-Typ wird ein eigenes `fstyp`-Modul ausgeführt; jedes dieser Module verwendet die entsprechenden heuristischen Programme, um festzustellen, ob die angegebene *gerätedatei* dem Typ entspricht, den das Modul sucht. Ist dies der Fall, gibt das Programm in der Standardausgabe den üblichen Dateisystembezeichner für diesen Typ aus und schließt mit dem Return-Code 0 ab; ist keines der Module erfolgreich, wird die Fehlermeldung `unknown_fstyp (no matches)` (unbekannter Dateisystem-Typ (keine Entsprechung)) zurückgegeben, und der Exit-Status ist 1. Sind mehrere Module erfolgreich, erscheint die Fehlermeldung `unknown_fstyp (multiple matches)` (unbekannter Dateisystem-Typ (mehrere Entsprechungen)), und der Exit-Status ist 2.

Die folgende Option steht zur Verfügung:

- `-v` Generiert ausführliche Ausgabe. Hierbei handelt es sich gewöhnlich um Informationen zum Superblock des Dateisystems, die je nach *dateisystem-typ* variieren.

HINWEIS

Die Verwendung heuristischer Programme bedeutet, daß das Ergebnis von `fstyp` nicht zwangsläufig korrekt ist.

BEZEICHNUNG

ftpd, in.ftpd – Server für Dateiübertragungsprotokolle

ÜBERSICHT

in.ftpd [-dl] [-tzeitlimit] host .socket

BESCHREIBUNG

in.ftpd ist der Serverprozeß für das Internet File Transfer Protocol (FTP). Der Server wird von dem Internet-Dämon inetd(1M) immer dann aufgerufen, wenn eine Verbindung zum FTP-Service (siehe services(4)) hergestellt wird. Dabei stehen die verfügbare Verbindung als Deskriptor 0, sowie Host und Socket-Schnittstelle, von denen die Verbindung ausgeht (in Hexadezimal- bzw. Dezimalschreibweise), als Argument zur Verfügung.

Bei inaktiven Verbindungen tritt nach 90 Sekunden eine Zeitlimitüberschreitung auf.

Folgende Optionen stehen zur Verfügung:

-tzeitlimit

Setzt das Zeitlimit bei Inaktivität auf den Wert *zeitlimit* in Sekunden. Ist diese Option nicht angegeben, dann bricht der FTP-Server eine inaktive Sitzung nach einer Wartezeit von 15 Minuten ab.

Anforderungen

Der FTP-Server unterstützt die folgenden FTP-Anforderungen; es wird nicht nach Groß- und Kleinschreibung unterschieden.

Anforderung Beschreibung

ABOR	vorheriges Kommando abbrechen
ACCT	Konto angeben (ignoriert)
ALLO	Speicher zuweisen (willkürlich)
APPE	an eine Datei anhängen
CDUP	ins Vater-Verzeichnis des aktuellen Arbeitsverzeichnisses wechseln
CWD	Arbeitsverzeichnis wechseln
DELE	eine Datei löschen
HELP	Hilfeinformationen anzeigen
LIST	Listendateien in einem Verzeichnis anzeigen (ls -lg)
MKD	Verzeichnis anlegen
MODE	<i>modus</i> für Datenübertragung definieren
NLST	Namensliste der Dateien in einem Verzeichnis anzeigen (ls)
NOOP	keine Operation
PASS	Paßwort angeben
PASV	Server-zu-Server-Übertragung vorbereiten

PORT	Datenverbindungsanschluß definieren
PWD	aktuelles Arbeitsverzeichnis ausgeben
QUIT	Sitzung beenden
RETR	Datei abrufen
RMD	Verzeichnis entfernen
RNFR	alten Dateinamen zum Umbenennen angeben
RNTO	neuen Dateinamen zum Umbenennen angeben
STOR	Datei speichern
STOU	Datei mit einem eindeutigen Namen speichern
STRU	<i>struktur</i> für Datenübertragung angeben
TYPE	<i>typ</i> für Datenübertragung angeben
USER	Benutzernamen angeben
XCUP	zum Vater-Verzeichnis des aktuellen Arbeitsverzeichnisses wechseln
XCWD	Arbeitsverzeichnis wechseln
XMKD	Verzeichnis anlegen
XPWD	aktuelles Arbeitsverzeichnis ausgeben
XRMD	Verzeichnis entfernen

Die übrigen FTP-Anforderungen, die in RFC 959 angegeben sind, werden erkannt, aber nicht implementiert.

Der FTP-Server bricht eine aktive Dateiübertragung nur dann ab, wenn dem Kommando ABOR ein Telnet Interrupt Process (IP)-Signal und ein Telnet Sync-Signal im Telnet-Datenstrom des Kommandos vorausgehen, wie in RFC 959 beschrieben.

in.ftpd interpretiert Dateinamen entsprechend den Konventionen von *sh(1)*. Daher können Benutzer die folgenden Metazeichen verwenden: * ? [] { } ~

in.ftpd überprüft Benutzer nach drei Regeln.

- 1) Der Benutzername muß in der Paßwortdatenbank */etc/passwd* enthalten sein, und das Paßwort darf nicht Null sein. Ist dies der Fall, muß der Client ein Paßwort angeben, ehe Dateioperationen durchgeführt werden können.
- 2) Erscheint der Benutzername in der Datei */etc/ftpusers*, wird der ftp-Zugriff verweigert.
- 3) Der ftp-Zugriff wird nur dann gewährt, wenn die Shell des Benutzers (aus */etc/passwd*) in der Datei */etc/shells* aufgeführt ist oder wenn die Benutzer-Shell zur folgenden Gruppe gehört:

```

/bin/sh
/bin/ksh
/bin/csh
/usr/bin/sh
/usr/bin/ksh
/usr/bin/csh

```

- 4) Ist der Benutzername `anonym` oder `ftp`, muß die Paßwortdatei eine anonyme FTP-Kennung (`user ftp`) enthalten. In diesem Fall darf sich der Benutzer mit einem beliebigen Paßwort anmelden (dieser wird laut Konvention als Name des Hosts des Client angegeben).

Im letzten Fall ergreift `in.ftpd` besondere Maßnahmen, um die Zugriffsrechte des Client einzuschränken. Der Server führt ein Kommando `chroot(2)` für das Home-Verzeichnis des `ftp`-Benutzers aus. Um die Systemsicherheit zu wahren, muß der Teilbaum `ftp` mit besonderer Umsicht aufgebaut werden; dazu werden folgende Regeln empfohlen:

home-verzeichnis

Macht `ftp` zum Eigentümer des Home-Verzeichnisses; niemand anders hat Schreibberechtigung dafür.

home-verzeichnis/usr/bin

Macht den privilegierten Benutzer zum Eigentümer dieses Verzeichnisses, für das dann niemand anders Schreibberechtigung hat. Das Programm `ls(1)`, das die Listenkommandos unterstützt, muß vorhanden sein. Es muß das Zugriffsrecht `111` haben.

home-verzeichnis/etc

Macht den privilegierten Benutzer zum Eigentümer dieses Verzeichnisses, für das niemand sonst Schreibberechtigung hat. Kopien der Dateien `passwd(4)`, `group(4)` und `netconfig` müssen vorhanden sein, damit das Kommando `ls` ordnungsgemäß ausgeführt werden kann. Diese Dateien müssen das Zugriffsrecht `444` haben.

home-verzeichnis/pub

Gibt diesem Verzeichnis das Zugriffsrecht `777` und macht `ftp` zum Eigentümer. Benutzer sollten Dateien, die über die anonyme Kennung verfügbar sein sollen, in dieses Verzeichnis schreiben.

home-verzeichnis/dev

Macht den privilegierten Benutzer zum Eigentümer dieses Verzeichnisses, für das niemand sonst Schreibberechtigung hat. Geben Sie in diesem Verzeichnis folgendes ein:

```

FTP="'grep ^ftp: /etc/passwd | cut -d: -f6'"
MAJORMINOR="'ls -l /dev/tcp | awk '{ gsub(/./, \"\"); print $5, $6}'"
mknod $FTP/dev/tcp c $MAJORMINOR
chmod 666 $FTP/dev/tcp

```

SIEHE AUCH

`ftp(1)`, `getsockopt(3N)`, `passwd(4)`, `services(4)`.

Postel, Jon und Joyce Reynolds, *File Transfer Protocol (FTP)*, RFC 959, Network Information Center, SRI International, Menlo Park, Kalif., Oktober 1985.

HINWEIS

Die anonyme Kennung birgt Gefahren in sich und sollte daher nach Möglichkeit nicht angelegt werden.

Der Server muß als privilegierter Benutzer arbeiten, um Socket-Schnittstellen mit privilegierten Anschlußnummern erstellen zu können. Er behält die effektive Benutzernummer des angemeldeten Benutzers bei und greift nur auf die Privilegierung zurück, um Adressen an Socket-Schnittstellen zu binden. Mögliche Sicherheitslücken wurden eingehend untersucht, können aber möglicherweise noch immer bestehen.

`/etc/ftpusers` enthält eine Liste der Benutzer, die nicht auf das System zugreifen dürfen; die Datei gibt einen Benutzernamen pro Zeile an.

BEZEICHNUNG

`fuser` – identifiziert Prozesse, die eine Datei oder Dateistruktur verwenden

ÜBERSICHT

```
/usr/sbin/fuser [-c|f]ku] dateien | ressourcen [[-] [c|f]ku]
dateien | ressourcen ]...
```

BESCHREIBUNG

`fuser` gibt die Prozeßnummern von Prozessen aus, die die als Argumente angegebenen *dateien* oder fernen *ressourcen* verwenden. Nach jeder Prozeßnummer ist ein Buchstabencode angegeben, der folgendermaßen interpretiert wird:

Verwendet der Prozeß die Datei als

- 1) sein aktuelles Verzeichnis, lautet der Code *c*,
- 2) Root-Verzeichnis, lautet der Code *r*,
- 3) offene Datei, lautet der Code *o*,
- 4) Textdatei, lautet der Code *t*.

Bei blockorientierten Gerätedateien mit eingehängten Dateisystemen werden alle Prozesse aufgelistet, die eine Datei auf diesem Gerät benutzen. Bei den Namen ferner Ressourcen werden alle Prozesse, die eine mit der fernen Ressource zusammenhängende Datei verwenden (Remote File Sharing), gemeldet. Bei allen sonstigen Dateitypen (Textdateien, ausführbare Dateien, Verzeichnisse, Geräte usw.) werden nur die Prozesse gemeldet, die diese bestimmte Datei verwenden.

Die folgenden Optionen können mit `fuser` verwendet werden:

- c Kann mit Dateien benutzt werden, die Einhängpunkte für Dateisysteme sind. Mit dieser Option erstreckt sich der Bericht auf Einhängpunkte und alle Dateien innerhalb der eingehängten Dateisysteme.
- f Hierbei bezieht sich der Bericht nur auf die genannte Datei, nicht auf Dateien innerhalb eines eingehängten Dateisystems.
- u Die Benutzerkennung in Klammern, wird auch nach der Prozeßnummer angegeben.
- k Das Signal SIGKILL wird an jeden Prozeß gesendet. Da diese Option bei jedem Prozeß zum Abbruch führt, erscheinen die Abbruchmeldungen möglicherweise nicht sofort (siehe `kill(2)`).

Werden mehrere Dateigruppen angegeben, können die Optionen für jede weitere Dateigruppe erneut angegeben werden. Mit einem Bindestrich werden die aktuellen Optionen außer Kraft gesetzt.

Die Prozeßnummern werden in der Standardausgabe, durch Leerzeichen getrennt, in einer Zeile ausgegeben, die mit einem Zeilenende-Zeichen abgeschlossen wird. Die gesamte sonstige Ausgabe wird in die Standardfehlerausgabe geschrieben.

Alle Benutzer, die zum Lesen von `/dev/kmem` und `/dev/mem` berechtigt sind, können `fuser` verwenden. Nur ein privilegierter Benutzer kann Prozesse eines anderen Benutzers beenden.

DATEIEN

/stand/unix für System-Namensliste
/dev/kmem für Systemabbild
/dev/mem ebenfalls für Systemabbild

HINWEIS

Das Systemabbild, mit dem `fuser` arbeitet, bezieht sich nur auf einen bestimmten Augenblick. Daher werden Prozesse, die erst auf eine Datei zugreifen, wenn `fuser` bereits läuft, eventuell nicht registriert. Und Prozesse, die als Benutzer einer Datei erfaßt werden, können ihre Arbeit mit dieser Datei bereits beendet haben, während `fuser` noch läuft. Aufgrund dieser Faktoren ist von der Verwendung der Option `-k` abzusehen.

SIEHE AUCH

`mount(1M)`.
`ps(1)` in den *Kommandos*.
`kill(2)`, `signal(2)` im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

fwtmp, wtmpfix – bearbeitet Abrechnungssätze für Verbindungen

ÜBERSICHT

```
/usr/lib/acct/fwtmp [-ic]
/usr/lib/acct/wtmpfix [dateien]
```

BESCHREIBUNG

fwtmp liest die Standardeingabe, schreibt in die Standardausgabe und wandelt die binären Datensätze des Typs in /var/adm/wtmp in formatierte ASCII-Datensätze um. Die ASCII-Version ist nützlich, wenn fehlerhafte Sätze editiert werden müssen.

Mit dem Argument `-ic` wird angezeigt, daß die Eingabe im ASCII-Format vorliegt und die Ausgabe im Binärformat erfolgen soll.

wtmpfix prüft die Standardeingabe oder angegebene Dateien im Format utmp.h, korrigiert die Zeit-/Datumsstempel, um die Einträge konsistent zu machen, und schreibt in die Standardausgabe. Ein Bindestrich – anstelle von *dateien* bezeichnet die Standardeingabe. Werden keine Zeit-/Datumskorrekturen durchgeführt, kann acctcon bestimmte Sätze mit Datumsänderung nicht bearbeiten.

Bei jeder Datumseinstellung werden zwei Datumsänderungssätze in /var/adm/wtmp geschrieben. Der erste enthält das alte Datum und wird mit der Zeichenkette "old time" im Feld line und der Marke OLD_TIME im Feld type der utmp-Struktur gekennzeichnet. Der zweite Satz gibt das neue Datum an und ist mit der Zeichenkette new time im Feld line und der Marke NEW_TIME im Feld type gekennzeichnet. wtmpfix verwendet diese Datensätze zum Synchronisieren aller Zeitstempel der Datei.

wtmpfix korrigiert nicht nur die Zeit-/Datumsstempel, sondern prüft auch das Feld name auf Gültigkeit, um sicherzugehen, daß es nur alphanumerische Zeichen und Leerzeichen enthält. Wird ein als ungültig aufgefaßter Name festgestellt, wird der Login-Name in INVALID geändert und eine Diagnosemeldung in die Standardfehlerausgabe geschrieben. Auf diese Weise begrenzt wtmpfix das Risiko, daß Verbindungsabrechnungssätze nicht von acctcon verarbeitet werden können.

DATEIEN

```
/var/adm/wtmp
/usr/include/utmp.h
```

SIEHE AUCH

acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), runacct(1M), acct(4), utmp(4)
acctcom(1), ed(1) in den *Kommandos*
acct(2) im *Referenzhandbuch für Programmierer*

BEZEICHNUNG

gencc – legt ein Front-End zu dem Kommando cc an

ÜBERSICHT

gencc

BESCHREIBUNG

gencc legt ein Front-End zu dem Kommando cc an. Es führt einen Dialog mit dem Benutzer. Da alle festverdrahteten Pfadnamen aus dem C-Übersetzungssystem (CCS) gelöscht wurden, können Teile des CCS ohne erneutes Übersetzen zu neuen Positionen verschoben werden. Diese neuen Positionen können mit der Option -Y zum Kommando cc angegeben werden. Es ist jedoch nicht sinnvoll, bei jedem Aufruf dieses Kommandos die entsprechenden -Y-Optionen anzugeben. Darüber hinaus sollte das Verschieben von Teilen des CCS durch den Systemverwalter für die Benutzer unsichtbar geschehen.

Das Front-End zu dem Kommando cc, das mit gencc generiert wird, ist ein Shell-Skript von einer Zeile, das das Kommando cc mit Angabe der richtigen -Y-Optionen aufruft. Ferner übergibt das Front-End zum Kommando cc alle vom Benutzer eingegebenen Optionen an das Kommando cc.

gencc fordert zur Eingabe der Position aller Werkzeuge und Verzeichnisse auf, die mit einer -Y-Option zum Kommando cc angegeben werden können. Wird keine Position angegeben, wird angenommen, daß dieser Teil des CCS nicht verschoben wurde. Wenn alle Positionen eingegeben sind, legt gencc das Front-End zu dem Kommando cc an.

gencc legt dieses Front-End im aktuellen Arbeitsverzeichnis an und gibt der Datei denselben Namen wie dem Kommando cc. Daher darf gencc nicht in dem Verzeichnis aufgerufen werden, in dem das eigentliche Kommando cc steht. Wenn ein Systemverwalter das C-Übersetzungssystem neu verteilt hat, muß er ferner das eigentliche Kommando cc an eine Position stellen, die normalerweise nicht über den Benutzerpfad erreichbar ist, z.B. /usr/lib. Dadurch kann er verhindern, daß Benutzer das Kommando cc versehentlich ohne das Front-End aufrufen.

ACHTUNG!

gencc gibt keine Warnung aus, falls ein Werkzeug oder ein Verzeichnis an der angegebenen Position nicht existiert. Auch verschiebt gencc selbst keine Dateien an neue Positionen. Das Kommando gencc ist veraltet.

DATEIEN

./cc Front-End zu cc

SIEHE AUCH

cc(1).

BEZEICHNUNG

getdev – listet Geräte nach bestimmten Kriterien auf

ÜBERSICHT

getdev [-ae] [*kriterien*] [*gerät* [. . .]]

BESCHREIBUNG

getdev erstellt eine Liste von Geräten, die bestimmten Kriterien entsprechen. Diese bestehen aus einer Liste von Attributen (die in Ausdrücken angegeben werden) und einer Liste von Geräten. Werden keine Kriterien angegeben, werden alle Geräte in der Liste aufgeführt.

Sofern nicht die Option -a angegeben wurde, müssen alle Geräte mindestens eines der Kriterien erfüllen. Wurde sie angegeben, werden nur die Geräte aufgeführt, die allen Kriterien einer Liste entsprechen.

Geräte, die in der Kommandozeile definiert wurden und den Kriterien entsprechen, werden in der ausgegebenen Liste aufgeführt. Wird jedoch die Option -e verwendet, werden die in der Eingabeliste aufgeführten Geräte aus der Ausgabeliste ausgeschlossen.

Typen von Kriterienausdrücken

Es gibt vier mögliche Typen von Ausdrücken, mit denen die Kriterien im Argument *kriterien* angegeben werden können:

<i>attribut</i> = <i>wert</i>	Gibt alle Geräte an, für die ein Attribut <i>attribut</i> definiert ist, das <i>wert</i> entspricht.
<i>attribut</i> != <i>wert</i>	Gibt alle Geräte an, für die ein Attribut <i>attribut</i> definiert ist, das nicht gleich <i>wert</i> ist.
<i>attribut</i> :*	Gibt alle Geräte an, für die das Attribut <i>attribut</i> definiert ist.
<i>attribut</i> ! : *	Gibt alle Geräte an, für die das Attribut <i>attribute</i> nicht definiert ist.

Eine vollständige Liste der verfügbaren Attribute und der zugehörigen Beschreibung dazu finden Sie im Handbucheintrag zu putdev(1M).

Optionen und Argumente

Folgende Optionen und Argumente stehen für dieses Kommando zur Verfügung:

-a	Gibt an, daß das Gerät allen Kriterien entsprechen muß, wenn es in der mit diesem Kommando erstellten Liste aufgeführt werden soll. Sind keine Kriterien definiert, hat die Option keine Wirkung.
-e	Bestimmt, daß die Geräte in der Liste, die in der Kommandozeile angegeben wird, von der Ausgabeliste dieses Kommandos ausgeschlossen sein sollen. Ohne die Option -e werden die genannten Geräte in die erstellte Liste eingeschlossen. Sind keine Geräte definiert, hat die Option keine Wirkung.
<i>kriterien</i>	Definiert die Kriterien, die ein Gerät erfüllen muß, das in die Ausgabeliste aufgenommen werden soll. Diese Kriterien sind als Ausdrücke anzugeben.

gerät Definiert Geräte, die in der Ausgabeliste enthalten sein sollen. Hierzu wird der Pfadname oder der Aliasname des Geräts angegeben.

EXIT-CODES

Die Exit-Codes von `getdev` haben folgende Bedeutung:

- 0 = Aufgabe erfolgreich abgeschlossen
- 1 = falsche Kommandosyntax, ungültige Option oder interner Fehler
- 2 = Gerätetabelle konnte nicht zum Lesen geöffnet werden

DATEIEN

`/etc/device.tab`

SIEHE AUCH

`devattr(1)`, `getdgrp(1)`, `putdev(1)`, `putdgrp(1)`, `getdev(3X)`.

BEZEICHNUNG

`getdgrp` – listet Gerätegruppen auf, deren Geräte bestimmten Kriterien entsprechen

ÜBERSICHT

`getdgrp [-ael] [kriterien ] [ggruppe ... ]`

BESCHREIBUNG

`getdgrp` erstellt eine Liste von Gerätegruppen mit Geräten, die den angegebenen Kriterien entsprechen. Die Kriterien werden in Form von Ausdrücken angegeben.

kriterien kann aus einem Ausdruck oder einer Liste von Ausdrücken bestehen, dem bzw. der ein Gerät entsprechen muß, wenn es in der von `getdgrp` erstellten Liste erscheinen soll. Sind keine Kriterien angegeben, werden alle Gerätegruppen in der Liste ausgegeben.

Die Geräte müssen mindestens eines der Kriterien in der Liste erfüllen. Mit der Option `-a` kann jedoch bestimmt werden, daß eine logische UND-Operation ausgeführt wird. Dann werden nur die Gruppen aufgeführt, deren Geräte allen Kriterien der Liste entsprechen.

Mit *ggruppe* wird eine Reihe von Gerätegruppen definiert, die in der Liste erscheinen sollen. Diese Gerätegruppen müssen definiert sein und Geräte enthalten, die den Kriterien entsprechen. Wird jedoch die Option `-e` verwendet, werden die Gerätegruppen dieser Liste ausgeschlossen. Wenn die Option `-e` angegeben ist und auch Kriterien definiert sind, enthält die Ausgabeliste Gerätegruppen mit Geräten, die den Kriterien entsprechen und nicht in der Liste in der Kommandozeile enthalten sind.

Typen der Kriterienausdrücke

Es gibt vier mögliche Typen von Ausdrücken:

<i>attribut=wert</i>	Gibt alle Gerätegruppen mit einem Mitglied an, für das ein Attribut <i>attribut</i> definiert ist, das gleich <i>wert</i> ist.
<i>attribut!=wert</i>	Gibt alle Gerätegruppen mit einem Mitglied an, für das ein Attribut <i>attribut</i> definiert ist, das ungleich <i>wert</i> ist.
<i>attribut:*</i>	Gibt alle Gerätegruppen an, bei denen für ein Mitglied das Attribut <i>attribut</i> definiert ist.
<i>attribut!:*</i>	Gibt alle Gerätegruppen mit einem Mitglied an, für das das Attribut <i>attribut</i> nicht definiert ist.

Eine vollständige Liste der verfügbaren Attribute und die entsprechende Beschreibung finden Sie in dem Handbucheintrag `putdev(1M)`.

Optionen und Argumente

Für dieses Kommando stehen folgende Optionen und Argumente zur Verfügung:

<code>-a</code>	Bestimmt, daß ein Gerät allen Kriterien entsprechen muß, damit die Gerätegruppe, zu der es gehört, in die angegebene Liste aufgenommen werden kann. Sind keine Kriterien definiert, hat diese Option keine Wirkung.
-----------------	---

getdgrp(1M)

getdgrp(1M)

<code>-e</code>	Bestimmt, daß die Gerätegruppen aus der Liste, die in der Kommandozeile angegeben wurde, aus der Ausgabeliste ausgeschlossen sein soll. (Ohne die Option <code>-e</code> dürfen nur die genannten Gerätegruppen in die Ausgabeliste eingeschlossen werden.) Die Option hat keine Wirkung, wenn keine Gerätegruppen definiert sind.
<code>-l</code>	Bestimmt, daß alle Gerätegruppen (entsprechend der Option <code>-e</code> und der Liste <i>ggruppe</i>) aufgeführt werden sollen, auch wenn sie keine entsprechenden Geräte enthalten. Diese Option hat keine Wirkung, wenn in der Kommandozeile <i>kriterien</i> angegeben sind.
<i>kriterien</i>	Definiert <i>kriterien</i> , die ein Gerät erfüllen muß, damit die Gerätegruppe, zu der es gehört, in die Ausgabeliste aufgenommen werden kann.
<i>ggruppe</i>	Definiert Gerätegruppen, die in die Ausgabeliste aufgenommen oder von ihr ausgeschlossen werden sollen.

EXIT-CODES

Die Exit-Codes von `getdgrp` haben folgende Bedeutung:

- 0 = Aufgabe erfolgreich ausgeführt
- 1 = falsche Kommandosyntax, ungültige Option oder interner Fehler
- 2 = Geräte- oder Gerätegruppentabelle konnte nicht zum Lesen geöffnet werden

DATEIEN

`/etc/device.tab`
`/etc/dgroup.tab`

SIEHE AUCH

`devattr(1)`, `getdev(1)`, `putdev(1)`, `putdgrp(1)`, `getdgrp(3X)`.

gettable(1M)

gettable(1M)

BEZEICHNUNG

`gettable` – ruft Host-Tabelle im DoD-Internet-Format von einem Host ab

ÜBERSICHT

`gettable host`

BESCHREIBUNG

`gettable` ist ein einfaches Programm zum Abrufen der DoD-Internet-Host-Tabelle von einem Hostnamen-Server. Die Tabelle wird auf dem angegebenen *host* abgefragt. Sobald die Tabelle gefunden ist, wird sie in die Datei `hosts.txt` geschrieben.

`gettable` stellt eine TCP-Verbindung zu dem in der Service-Spezifikation für Hostname angegebenen Anschluß her. Anschließend werden alle Namen abgefragt, und das Ergebnis wird in die Ausgabedatei geschrieben.

Es empfiehlt sich, `gettable` zusammen mit dem Programm `htable(1M)` einzusetzen, das das Format der DoD-Internet-Host-Tabelle in das Format der Suchroutinen für die Netzerbibliothek umwandelt.

SIEHE AUCH

`htable(1M)`

Harrenstien, Ken, Mary Stahl und Elizabeth Feinler, *HOSTNAME Server*, RFC 953, Network Information Center, SRI International, Menlo Park, Kalif., Oktober 1985.

HINWEIS

Es ist auch möglich, nur Teile der Datenbank anzufordern.

BEZEICHNUNG

getty – startet lokale Login-Sitzung

ÜBERSICHT

/sbin/getty [-h] [-t *wartezeit*] *leitung* [*name*]

/sbin/getty -c *datei*

BESCHREIBUNG

getty gibt eine Login-Aufforderung aus, wartet auf die Eingabe eines Login-Namens und ruft dann das Kommando login auf. Mittels der Optionen und Argumente in der Kommandozeile versucht getty, das System an die Übertragungsgeschwindigkeit des Terminals anzupassen. getty darf nur von einem privilegierten Benutzer ausgeführt werden, der unter der Benutzerkennung root arbeitet.

Ohne optionale Argumente verwendet getty die Angaben, die im ersten Eintrag der Datei /etc/gettydefs definiert sind. getty gibt die dort angegebene Login-Aufforderung aus, bevor der Benutzername zeichenweise gelesen wird. Den Empfang eines Nullzeichens oder eines Rahmenfehlers (framing error) führt getty darauf zurück, daß der Benutzer die Taste BREAK gedrückt hat. Dies veranlaßt getty, den gettydefs-Eintrag zu verwenden, der als Folgeeintrag angegeben ist.

leitung

Name einer Bildschirmleitung in /dev, an die sich getty anschließen soll. getty verwendet diese Zeichenkette als den Namen einer Datei im Verzeichnis /dev, die zum Lesen und Schreiben geöffnet werden soll.

-h Ist die Option -h nicht gesetzt, wird ein Hangup-Signal erzwungen, bevor getty die Übertragungsgeschwindigkeit einstellt.

-t *wartezeit*

getty wird abgebrochen, wenn die Leitung aufgebaut werden konnte und in *wartezeit* Sekunden keine Eingabe erfolgte.

name

Name eines Eintrags in der Datei /etc/gettydefs, der Terminaleinstellungen, eine Übertragungsgeschwindigkeit und eine Login-Aufforderung definiert. Dieser Definition entnimmt getty, mit welcher Geschwindigkeit das System zuerst arbeiten soll, welche Bildschirmwerte eingestellt werden sollen und mit welchen Einstellungen weitergearbeitet werden soll, falls der Benutzer durch Drücken der Taste BREAK angibt, daß die zuerst verwendete Einstellung nicht geeignet ist.

-c *datei*

Die Option -c listet den Inhalt von *datei* auf und führt eine Gültigkeitsprüfung durch. Die Einträge in *datei* müssen das gleiche Format haben wie die Einträge in /etc/gettydefs.

getty (1M)

getty (1M)

DATEIEN

/etc/gettydefs

SIEHE AUCH

ct(1C), sttydefs(1M), tty(7), ttymon(1M).
login(1) in den *Kommandos*.
ioctl(2) im *Referenzhandbuch für Programmierer*.

HINWEIS

Terminals vom Typ 9780? besitzen keine BREAK-Taste.

BEZEICHNUNG

`getvol` – überprüft die Zugriffsmöglichkeiten auf Geräte

ÜBERSICHT

```
getvol -n [-l kenn] gerät
getvol [-f|-F] [-wo] [-l kenn|-x kenn] gerät
```

BESCHREIBUNG

`getvol` überprüft, ob auf das angegebene Gerät zugegriffen werden kann und ob ein geeigneter Datenträger eingelegt wurde. Das Kommando führt mit dem Benutzer einen Dialog, wobei es Anweisungen gibt, Fehler beschreibt und die erforderlichen Kennsätze ausgibt.

Folgende Optionen und Argumente stehen für dieses Kommando zur Verfügung:

- n Führt das Kommando ohne Dialog mit dem Benutzer aus. Es wird vorausgesetzt, daß der Datenträger beim Absetzen des Kommandos eingelegt ist.
- l *kenn* Bestimmt, daß der Kennsatz *kenn* auf dem eingelegten Datenträger vorhanden sein muß (kann durch die Option -o außer Kraft gesetzt werden).
- f Formatiert den Datenträger nach dem Einlegen und verwendet dazu das für dieses Gerät in der Gerätetabelle definierte Kommando zur Formatierung.
- F Formatiert den Datenträger nach dem Einlegen und schreibt ein Dateisystem auf das Gerät. Auch hier wird das Formatkommando verwendet, das in der Gerätetabelle für dieses Gerät definiert ist.
- w Ermöglicht es dem Verwalter, einen neuen Kennsatz auf das Gerät zu schreiben. Der Benutzer wird aufgefordert, den Text für den Kennsatz einzugeben. Diese Option wird unwirksam, wenn die Option -n aktiviert ist.
- o Ermöglicht es dem Verwalter, eine Kennsatzprüfung außer Kraft zu setzen.
- x *kenn* Bestimmt, daß der Kennsatz *kenn* auf dem Gerät vorhanden sein muß. Diese Option ist anstelle der Option -l anzugeben, wenn der Kennsatz nur visuell überprüft werden kann. Wird diese Option verwendet, wird eine Meldung ausgegeben, in der der Verwalter aufgefordert wird, zu überprüfen, ob es sich bei dem Kennsatz tatsächlich um *kenn* handelt.
- gerät* Nennt das Gerät, das auf Zugriffsmöglichkeiten überprüft werden soll.

EXIT-CODES

Die Exit-Codes von `getvol` haben folgende Bedeutung:

- 0 = Aufgabe erfolgreich ausgeführt
- 1 = falsche Kommandosyntax, ungültige Option oder interner Fehler
- 3 = Gerätetabelle konnte nicht zum Lesen geöffnet werden

getvol(1M)

getvol(1M)

HINWEIS

Dieses Kommando verwendet bei der Überprüfung des Datenträgerkennsatzes die Gerätetabelle, um die Gerätemerkmale festzustellen.

DATEIEN

/etc/device.tab

SIEHE AUCH

getvol(3X).

groupadd (1M)

groupadd (1M)

BEZEICHNUNG

groupadd – Hinzufügen (Anlegen) einer neuen Gruppendefinition auf dem System

ÜBERSICHT

groupadd [-g *gruppennr* [-o]] *gruppe*

BESCHREIBUNG

Mit dem Kommando `groupadd` wird eine neue Gruppendefinition auf dem System hinzugefügt, indem ein entsprechender Eintrag in die Datei `/etc/group` aufgenommen wird.

Folgende Optionen stehen zur Verfügung:

-g *gruppennr*

Die Gruppennummer für die neue Gruppe. Diese Gruppennummer muß eine positive, ganze Dezimalzahl sein, die kleiner ist als der in der Include-Datei `<param.h>` definierte Wert `MAXUID`. Als Gruppennummer wird standardmäßig die nächsthöhere verfügbare (eindeutige) Zahl nach der höchsten aktuell zugewiesenen Zahl gewählt. Sind z.B. die Zahlen 100, 105 und 200 als Gruppennummern zugewiesen, ist die nächste standardmäßige Gruppennummer 201. Gruppennummern von 0 bis 99 sind reserviert.

-o Bei dieser Option darf die *gruppennr* dupliziert werden und braucht dann nicht mehr eindeutig zu sein.

gruppe Eine Zeichenkette druckbarer Zeichen, die den Namen der neuen Gruppe angibt. Sie darf keinen Doppelpunkt (:) und kein Neue-Zeile-Zeichen (\n) enthalten.

DATEIEN

`/etc/group`

SIEHE AUCH

`groupdel(1M)`, `groupmod(1M)`, `logins(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`, `users(1)`.

EXIT-CODES

Die Exit-Codes von `groupadd` haben folgende Bedeutung:

- 0 = erfolgreiche Ausführung
- 2 = ungültige Kommandosyntax; es wird eine Meldung ausgegeben, wie die Syntax von `groupadd` lautet
- 3 = es wurde ein ungültiges Argument zu einer Option eingegeben
- 4 = *gruppennr* ist nicht eindeutig (wenn die Option -o nicht verwendet wird)
- 9 = *gruppe* ist nicht eindeutig
- 10 = Datei `/etc/group` kann nicht aktualisiert werden

groupdel(1M)

groupdel(1M)

BEZEICHNUNG

`groupdel` – löscht eine Gruppendefinition aus dem System

ÜBERSICHT

`groupdel` *gruppe*

BESCHREIBUNG

Mit dem Kommando `groupdel` wird eine Gruppendefinition aus dem System gelöscht, d.h. der entsprechende Eintrag in der Datei `/etc/group`.

Die folgende Option steht zur Verfügung:

gruppe Eine Zeichenkette druckbarer Zeichen, die die zu löschende Gruppe angibt.

DATEIEN

`/etc/group`

SIEHE AUCH

`groupadd(1M)`, `groupmod(1M)`, `logins(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`, `users(1)`.

EXIT-CODES

Die Exit-Codes von `groupdel` haben folgende Bedeutung:

- 0 = erfolgreiche Ausführung
- 2 = ungültige Kommandosyntax; es wird eine Meldung ausgegeben, wie die Syntax der Kommandos `groupdel` lautet
- 6 = *gruppe* ist nicht vorhanden
- 10 = die Datei `/etc/group` kann nicht aktualisiert werden

BEZEICHNUNG

`groupmod` – ändert eine Gruppendefinition im System

ÜBERSICHT

`groupmod [-g gruppennr [-o]] [-n name] gruppe`

BESCHREIBUNG

Das Kommando `groupmod` ändert die Definition der angegebenen Gruppe durch Ändern des entsprechenden Eintrags in der Datei `/etc/group`.

Folgende Optionen stehen zur Verfügung:

`-g gruppennr`

Die Gruppennummer für die neue Gruppe. Diese Gruppennummer muß eine positive, ganze Dezimalzahl sein, die kleiner ist als der in der Include-Datei `<param.h>` definierte Wert `MAXUID`. Als Gruppennummer wird standardmäßig die nächsthöhere verfügbare (eindeutige) Zahl nach der höchsten aktuell zugewiesenen Zahl gewählt. Gruppennummern von 0 bis 99 sind reserviert. Wird nur der Gruppenname geändert, dann bleibt die Gruppennummer unverändert.

`-o` Bei dieser Option darf die *gruppennr* dupliziert werden und braucht dann nicht mehr eindeutig zu sein.

`-n name`

Eine Zeichenkette druckbarer Zeichen, die einen neuen Namen für die Gruppe angeben. Sie darf keinen Doppelpunkt (:) und kein Neue-Zeilen-Zeichen (\n) enthalten.

gruppe Der aktuelle Name der Gruppe, die geändert werden soll.

DATEIEN

`/etc/group`

SIEHE AUCH

`groupadd(1M)`, `groupdel(1M)`, `logins(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`, `users(1)`.

EXIT-CODES

Die Exit-Codes von `groupmod` haben folgende Bedeutung:

- 0 = erfolgreiche Ausführung
- 2 = ungültige Kommandosyntax; es wird eine Meldung ausgegeben, wie die Syntax des Kommandos `groupmod` lautet
- 3 = es wurde ein ungültiges Argument zu einer Option eingegeben
- 4 = *gruppennr* ist nicht eindeutig (und die Option `-o` ist nicht aktiv)
- 6 = *gruppe* existiert nicht
- 9 = *name* ist als Gruppenname bereits vorhanden
- 10 = die Datei `/etc/group` kann nicht aktualisiert werden

htable (1M)

htable (1M)

BEZEICHNUNG

htable – wandelt Host-Tabelle im DoD-Internet-Format um

ÜBERSICHT

htable *dateiname*

BESCHREIBUNG

htable wandelt eine Host-Tabelle in dem von RFC 952 angegebenen Format in das Format um, das die Netzwerk-Bibliotheksroutinen verwenden. Mit htable werden drei Dateien erstellt: `hosts`, `networks` und `gateways`. Die Datei `hosts` wird von den `gethostent(3N)`-Routinen zum Zuordnen (Mapping) von Host-Namen zu Adressen verwendet. Die Datei `networks` dient den `getnetent(3N)`-Routinen zum Zuordnen von Netzwerknamen zu Nummern. Mit der Datei `gateways` identifiziert der Routing-Dämon passive Internet-Gateways; eine Erläuterung hierzu finden Sie unter `routed(1M)`.

Steht eine Datei `localhosts`, `localnetworks` oder `localgateways` im aktuellen Verzeichnis, wird ihr Inhalt der Ausgabedatei vorangestellt und nicht interpretiert. Dadurch können Rechner lokale Aliasnamen und Einträge, die in der Master-Datenbank normalerweise nicht vorkommen, beibehalten.

Es empfiehlt sich, das Kommando `htable` zusammen mit dem Programm `gettable(1M)` zu verwenden, das die Host-Tabelle im DoD-Internet-Format von einem Host abruft.

DATEIEN

`localhosts`
`localnetworks`
`localgateways`

SIEHE AUCH

`gethostent(3N)`, `getnetent(3N)`, `gettable(1M)`, `routed(1M)`.

Harrenstien, Ken, Mary Stahl und Elizabeth Feinler, *DoD Internet Host Table Specification*, RFC 952, Network Information Center, SRI International, Menlo Park, Kalif., Oktober 1985.

HINWEIS

Die Datei `gateways` wird nicht ordnungsgemäß berechnet.

id (1M)

id (1M)

BEZEICHNUNG

`id` – gibt Benutzernamen und -nummer sowie Gruppennamen und -nummer aus

ÜBERSICHT

`id [ -a ]`

BESCHREIBUNG

`id` zeigt den Namen und die Nummer (ID) des aufrufenden Prozesses sowie den Namen und die Nummer (ID) der Gruppe an. Stimmen reale und effektive IDs nicht überein, werden beide ausgegeben.

Mit der Option `-a` werden alle Gruppen gemeldet, zu denen der aufrufende Prozeß gehört, ihre Nummern und der Name des Benutzers. Stimmen reale und effektive IDs nicht überein, werden beide ausgegeben.

Die Option `-a` meldet alle Gruppen, zu denen der aufrufende Benutzer gehört.

SIEHE AUCH

`getuid(2)` im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

idbuild – Aufbau eines neuen SINIX-Systemkerns

ÜBERSICHT

/etc/conf/bin/idbuild

BESCHREIBUNG

Dieses Skript baut anhand der aktuellen Systemkonfiguration in /etc/conf/ einen neuen SINIX-Systemkern auf. Ein Kern wird nach der Installation eines Gerätetreibers oder nach Änderungen an den einstellbaren Systemparametern neu konfiguriert. Die Prozedur verwendet die Shell-Variable `ROOT` aus der Benutzerumgebung als Beginn des Suchpfads. Die Shell-Variable `ROOT` muß immer auf Null oder "/" gesetzt sein, es sei denn, daß der Kern in einem anderen Quellbaum als `root` erstellt wird. `idbuild` schließt bei erfolgreicher Ausführung mit dem Exit-Code Null ab, bei nicht erfolgreicher Ausführung mit einem anderen Wert.

Beim Aufbau eines neuen SINIX-Systemkerns werden neue Systemkonfigurationsdateien erstellt. Anschließend werden die Objektmodule für Kern und Gerätetreiber in dem Objektbaum /etc/conf/pack.d gebunden. Hierzu ruft `idbuild` die folgenden Kommandos auf:

/etc/conf/bin/idconfig Zum Aufbau der Kernkonfigurationsdateien.

/etc/conf/bin/idmkunix Zum Verarbeiten der Konfigurationsdateien und Binden des neuen SINIX-Systemkerns.

Die Systemkonfigurationsdateien werden durch Verarbeiten der Master- und Systemdateien erstellt, die die Spezifikationen für Gerätetreiber und einstellbare Parameter enthalten. Die Masterdaten stehen in den Dateien /etc/conf/cf.d/mdevice und /etc/conf/cf.d/mtune. Die Dateien /etc/conf/cf.d/stune und die in /etc/conf/sdevice.d/* angegebenen Dateien enthalten die Systemdaten. Für den Kern sind in den durch /etc/conf/sfsys.d/* und /etc/conf/mfsys.d/* angegebenen Dateien auch dateisystemtyp-spezifische Daten angegeben.

Sobald ein neuer SINIX-Systemkern konfiguriert ist, wird in /etc/.new_unix eine Lockdatei angelegt. Sie bewirkt, daß der neue Kern beim nächsten Herunterfahren des Systems (d.h. beim nächsten Aktivieren des Status *init 0*) den Systemkern /stand/unix ersetzt. Beim nächsten Hochfahren des Systems wird der neue Kern verwendet.

FEHLERMELDUNGEN

Da `idbuild` zum Neukonfigurieren des Systems und Binden andere Systemkommandos aufruft, meldet es alle von diesen Kommandos festgestellten Fehler und bereinigt dann die temporären Dateien, die im Zuge der Operationen erstellt wurden. Im allgemeinen zeigt der Exit-Code 1 an, daß `idbuild` einen Fehler festgestellt hat.

Es können Fehler der folgenden Kategorien auftreten:

- Masterdateifehler
- Systemdateifehler
- Fehler in den Dateien mit den einstellbaren Parametern
- Compiler- und Binderfehler.

Alle Fehlermeldungen sind so ausgelegt, daß sie keiner weiteren Erläuterung bedürfen.

SIEHE AUCH

idinstall(1M), idtune(1M).
mdevice(4), mfsys(4), mtune(4), sdevice(4), sfsys(4), stune(4) im *Referenzhandbuch für Programmierer*.

HINWEIS

Ein Kern, der größer als 4MB ist, kann nicht ordnungsgemäß gebootet werden.

BEZEICHNUNG

`idcheck` – gibt ausgewählte Daten zur Systemkonfiguration aus

ÜBERSICHT

`/etc/conf/bin/idcheck`

BESCHREIBUNG

Dieses Kommando gibt ausgewählte Daten zur Systemkonfiguration aus. Bei Installationsprozeduren für zusätzliche Softwarepakete von Gerätetreibern (Driver Software Package - DSP) kann damit überprüft werden, ob ein bestimmter Gerätetreiber bereits installiert wurde oder ob ein bestimmter Interrupt-Vektor, eine bestimmte E/A-Adresse oder ein anderer auswählbarer Parameter tatsächlich zur Verfügung steht. Folgende Formen des Kommandos sind möglich:

```
idcheck -p gerätename [-i verzeichnis] [-r]
idcheck -v vektor [-i verzeichnis] [-r]
idcheck -d dma-kanal [-i verzeichnis] [-r]
idcheck -a -l untergrenze -u obergrenze [-i verzeichnis] [-r]
idcheck -c -l untergrenze -u obergrenze [-i verzeichnis] [-r]
```

Dieses Kommando durchsucht das System- und das Master-Modul und gibt folgende Werte zurück:

100, wenn ein Fehler festgestellt wird.

0, wenn kein Konflikt auftritt.

eine positive Zahl über 0 und unter 100, wenn ein Konflikt besteht.

Folgende Optionen können in der Kommandozeile angegeben werden:

- `-r` Meldet den Gerätenamen eines konkurrierenden Geräts auf der Standardausgabe.
- `-p gerätename` Mit dieser Option wird geprüft, ob vier verschiedene Komponenten des Softwarepakets vorhanden sind. Der Exit-Code ergibt sich aus den addierten Return-Codes der vier Prüfungen.
 Der Exit-Code wird um 1 erhöht, wenn das Verzeichnis des Softwarepakets unter `/etc/conf/pack.d` existiert.
 Der Exit-Code wird um 2 erhöht, wenn das Master-Modul installiert ist.
 Der Exit-Code wird um 4 erhöht, wenn das System-Modul installiert ist.
 Der Exit-Code wird um 8 erhöht, wenn der Kernel mit dem System-Modul gebunden wurde.
 Der Exit-Code wird um 16 erhöht, wenn die Datei `Driver.o` Teil des Softwarepakets ist (statt einer `stubs.c`-Datei).
- `-v vektor` Gibt das Feld 'typ' des Geräts zurück, das den angegebenen *vektor* benutzt (d.h. ein anderes Softwarepaket benutzt den *vektor* bereits).

- d *dma-kanal* Gibt 1 zurück, wenn der angegebene *dma-kanal* belegt ist.
- a Mit dieser Option wird geprüft, ob der durch *untergrenze* und *obergrenze* begrenzte E/A-Adreßbereich mit einem anderen Softwarepaket konkurriert (*untergrenze* und *obergrenze* werden mit der Option -l bzw. -u definiert). Der Exit-Code addiert sich aus zwei verschiedenen Return-Codes.
Der Exit-Code erhöht sich um 1, wenn der E/A-Adreßbereich sich mit dem eines anderen Geräts überschneidet.
Der Exit-Code erhöht sich um 2, wenn sich der E/A-Adreßbereich mit einem anderen Gerät überschneidet und für dieses Gerät im Feld *typ* des Master-Moduls die Option 'O' angegeben ist. Aufgrund dieser Option ist es zulässig, daß ein Treiber den E/A-Adreßbereich eines anderen Treibers überlagert.
- c Gibt 1 zurück, wenn der durch *untergrenze* und *obergrenze* begrenzte CMA-Bereich (Controller Memory Address-Bereich) mit einem anderen Softwarepaket konkurriert (*untergrenze* und *obergrenze* werden mit den Optionen -l und -u gesetzt).
- l *untergrenze* Die Untergrenze des Adreßbereichs im Hexadezimalformat. 0x muß der Angabe nicht vorangestellt werden.
- u *obergrenze* Obergrenze des Adreßbereichs im Hexadezimalformat. 0x muß der Angabe nicht vorangestellt werden.
- i *verzeichnis* Gibt das Verzeichnis an, in dem die ID-Dateien *sdevice* und *mdevice* stehen. Das Standardverzeichnis ist */etc/conf/cf.d*.

FEHLERMELDUNGEN

Es sind keine Fehlermeldungen und keine Prüfungen für gültige Argumente zu Optionen vorhanden. *idcheck* interpretiert diese Argumente nach den Regeln von *scanf(3)* und fragt die Dateien *sdevice* und *mdevice* ab. Wird z.B. ein Buchstabe anstelle einer Ziffer verwendet, wandelt *scanf(3)* den Buchstaben in 0 um. *idcheck* verwendet diesen Wert dann in der Abfrage.

HINWEIS

Speziell für die MX300 gilt:
DMA-Kanal, E/A-Adreßbereich und CMA-Bereich werden unterschiedlich interpretiert (vgl. *mdevice(4)*, *sdevice(4)*).

SIEHE AUCH

idinstall(1m).

mdevice(4), *sdevice(4)* im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

`idconfig` – erstellt eine neue Kernelkonfiguration

ÜBERSICHT

`/etc/conf/bin/idconfig [option...]`

BESCHREIBUNG

Das Kommando `idconfig` verwendet als Eingabe eine Reihe von Dateien, die die Konfiguration des aufzubauenden SINIX-Systems enthalten. Ausgabedateien, die von `idmkunix` verwendet werden, werden erzeugt.

Folgende Eingabedateien erwartet `idconfig`:

- `mdevice` – Master-Datei für Geräte-Spezifikationen
- `sdevice` – System-Datei für Geräte-Spezifikationen
- `mtune` – Master-Datei zur Spezifikation von einstellbaren Parametern
- `stune` – System-Datei zur Spezifikation von einstellbaren Parametern
- `mfsys` – Master-Datei für Dateisystem-Typen
- `sfsys` – System-Datei für Dateisystem-Typen
- `sassign` – Gerätezuweisungsdatei

Folgende Ausgabedateien erstellt `idconfig`:

- `conf.c` – Kerneldatenstrukturen und Funktionsdefinitionen
- `config.h` – Kernelparameter und Gerätedefinitionen
- `vector.c` – Definitionen für Interrupt-Vektoren
- `direct` – Liste aller am Aufbau beteiligter Treiberkomponenten
- `fsconf.c` – Konfigurationsdaten für Dateisystemtyp

Folgende Optionen können in der Kommandozeile angegeben werden:

- `-o verzeichnis` Ausgabedateien werden im angegebenen Verzeichnis angelegt und nicht in `/etc/conf/cf.d`.
- `-i verzeichnis` Eingabedateien, die normalerweise in `/etc/conf/cf.d` stehen, befinden sich im angegebenen Verzeichnis.
- `-r verzeichnis` Das angegebene Verzeichnis wird anstelle von `/etc/conf` als `root`-Verzeichnis verwendet.
- `-d datei` Zur Eingabe wird statt `sdevice` diese *datei* verwendet.
- `-t datei` Die Eingabe erfolgt von dieser *datei* statt von `stune`.
- `-T datei` Die Eingabe erfolgt von dieser *datei* statt von `mtune`.
- `-a datei` Die Eingabe erfolgt von dieser *datei* statt von `sassign`.
- `-c datei` Die Ausgabe von `conf.c` wird in *datei* umgeleitet.
- `-h datei` Die Ausgabe von `config.h` wird in *datei* umgeleitet.
- `-v datei` Die Ausgabe von `vector.c` wird in *datei* umgeleitet.
- `-p datei` Die Ausgabe von `direct` wird in *datei* umgeleitet.

`-D`, `-m`, `-s` Diese Optionen werden nicht mehr unterstützt.

`-#` Ausgeben von Daten zur Fehlersuche.

Diese Version von SINIX unterstützt mehrere Geräteklassennummern (major numbers) für Treiber. `idconfig` generiert über Definitionen zusätzliche Konstanten in der Datei `config.h`, auf die in der Datei `space.c` verwiesen wird, so daß der Treiber sie zum Anlegen geeigneter Datenstrukturen verwenden kann. Diese Konstanten geben an, wie viele Geräteklassennummern dem Gerät zugewiesen wurden und welche Werte sie haben. Folgende Konstanten gibt es:

`PRFX_CMAJOR_X`

`PRFX_BMAJOR_X`

Dabei steht `PRFX` für einen Gerätepräfix. Bei einem SCSI-Gerät z.B. wäre dies SCSI. Das `X` steht für den Listenindex, beginnend mit 0.

Darüber hinaus werden in der Konfigurationsdatei `conf.c`, in der die Tabellen `bdevsw[]` und `cdevsw[]` initialisiert sind, Einträge für jede dieser Geräteklassennummern hinzugefügt. Damit werden dieselben Treibereinsprungspunkte für jede dieser Geräteklassennummern wiederholt.

FEHLERMELDUNGEN

Der Exit-Code Null steht für erfolgreiche Ausführung. Wurde ein Fehler festgestellt, schließt `idconfig` mit einem anderen Wert ab und gibt eine Fehlermeldung aus. Alle Fehlermeldungen sind ohne weitere Erläuterung verständlich.

SIEHE AUCH

`idmkunix(1M)`, `idbuild(1M)`, `idinstall(1M)`, `mdevice(4)`, `mtune(4)`, `sdevice(4)`, `stune(4)`.

BEZEICHNUNG

iddefaults – Standardwerte für einstellbare Parameter vorgeben

ÜBERSICHT

/etc/conf/bin/iddefaults

BESCHREIBUNG

iddefaults stellt die Standardwerte für bestimmte einstellbare (tunable) Parameter ein. Es handelt sich dabei um Parameter, deren Einstellung dem Hauptspeicherausbau angepaßt sein sollte.

Ein Aufruf von iddefaults ist nur dann sinnvoll, nachdem der Hauptspeicher des Rechners erweitert oder verringert worden ist.

Für die MX300 nimmt iddefaults folgende Einstellungen vor:

Parameter	Hauptspeicher			
	4MB	8-12MB	16-32MB	48-64MB
NCALL	300	600	1000	1500
NPROC	200	300	600	900
MAXUP	30	30	50	70
FLCKREC	300	600	1000	1500
NBUF	100	100	200	400
NHBUF	64	64	256	512
BUFHWM	200	800	2000	4000
NINODE	300	300	600	1000
UFSNINODE	300	500	1500	3000
NRNODE	300	300	600	1000
STRTHRESH	0x20000	0x200000	0x300000	0x400000
MSGMAP	100	100	200	300
MSGMNB	4096	8192	16384	32768
MSGMNI	50	100	150	200
MSGTQL	40	80	160	240
MSGSEG	1024	1024	2048	4096
SEMAP	50	100	150	200
SEMMNI	50	100	150	200
SEMMNS	60	100	200	300
SEMMNU	60	100	200	300
SHMMNI	100	100	150	200
SHMSEG	8	16	32	48
SPTMAP	100	150	250	400
NDQUOT	200	200	300	400
PGOVERFLOW	16	16	32	48
NOTPGOVERFLOW	16	16	32	48
SYSSEGSZ	1024	1024	2048	4096

Der Parameter SEGMAPSZ wird auf 1/4 des Hauptspeichers gesetzt. Der Parameter SHMMAX wird auf das Doppelte des Hauptspeichers, maximal aber auf 33554432 gesetzt.

Bitte beachten Sie, daß `iddefaults` nur Standardwerte einstellt. Einzelne Parameter können natürlich bei Bedarf durch `idtune(1M)` im Rahmen der durch `mtune(4)` vorgegebenen Bereiche wieder verändert werden. Die einstellbaren Parameter sind im Anhang B des *"Leitfadens für Systemverwalter"* beschrieben.

Damit die neuen Parametereinstellungen wirksam werden, muß nach dem Aufruf von `iddefaults` ein neuer SINIX-Kern mit `idbuild(1M)` erzeugt und dann gebootet werden.

SIEHE AUCH

`idtune(1M)`, `mtune(4)`

BEZEICHNUNG

iddefaults – Standardwerte für einstellbare Parameter vorgeben

ÜBERSICHT

/etc/conf/bin/iddefaults

BESCHREIBUNG

iddefaults stellt die Standardwerte für bestimmte einstellbare (tunable) Parameter ein. Es handelt sich dabei um Parameter, deren Einstellung dem Hauptspeicherausbau angepaßt sein sollte.

Ein Aufruf von iddefaults ist nur dann sinnvoll, nachdem der Hauptspeicher des Rechners erweitert oder verringert worden ist.

Für die MX500 nimmt iddefaults folgende Einstellungen vor:

Parameter	Hauptspeicher		
	16-32MB	48-64MB	96-128MB
NPROC	800	1200	1820
MAXUP	60	70	90
NHBUF	128	256	512
FLCKREC	1200	1600	2000
SPTMAP	150	300	450
PIOMAP	100	100	100
NBUF	200	400	600
BUFHWM	2000	4000	6000
NINODE	1000	1500	2000
UFSNINODE	1000	1500	2000
NDQUOT	200	400	600
NRNODE	1000	1500	2000
STRTHRESH	0x300000	0x400000	0x500000
MSGMAP	160	240	320
MSGMNB	8192	16384	32768
MSGMNI	150	200	250
MSGTQL	160	240	320
MSGSEG	5120	5120	5120
SEMAP	200	250	300
SEMMNI	200	250	300
SEMMNS	400	400	400
SEMMNU	200	250	300
SHMMAX	33554432	33554432	33554432
SHMMNI	160	240	320
SHMSEG	32	48	64
SYSSEGSZ	2048	4096	8192
PGOVERFLOW	32	48	64
NOTPGOVERFLOW	32	48	64

Der Parameter SEGMAPSZ wird auf 1/4 des Hauptspeichers gesetzt.

Bitte beachten Sie, das `iddefaults` nur Standardwerte einstellt. Einzelne Parameter können natürlich bei Bedarf durch `idtune(1M)` im Rahmen der durch `mtune(4)` vorgegebenen Bereiche wieder verändert werden. Die einstellbaren Parameter sind im Anhang B des *"Leitfadens für Systemverwalter"* beschrieben.

Damit die neuen Parametereinstellungen wirksam werden, muß nach dem Aufruf von `iddefaults` ein neuer SINIX-Kern mit `idbuild(1M)` erzeugt und dann gebootet werden.

SIEHE AUCH

`idtune(1M)`, `mtune(4)`

BEZEICHNUNG

iddefaults – Standardwerte für einstellbare Parameter vorgeben

ÜBERSICHT

/etc/conf/bin/iddefaults

BESCHREIBUNG

iddefaults stellt die Standardwerte für bestimmte einstellbare (tunable) Parameter ein. Es handelt sich dabei um Parameter, deren Einstellung dem Hauptspeicherausbau angepaßt sein sollte.

Ein Aufruf von iddefaults ist nur dann sinnvoll, nachdem der Hauptspeicher des Rechners erweitert oder verringert worden ist.

Für die WX200 nimmt iddefaults folgende Einstellungen vor:

Parameter	Hauptspeicher		
	4MB	8MB	16-32MB
NCALL	100	180	250
NPROC	200	300	600
MAXUP	50	60	60
FLCKREC	300	600	1000
NBUF	100	100	200
NHBUF	64	64	256
BUFHWM	200	800	2000
NINODE	300	300	600
UFSNINODE	300	500	1500
NRNODE	300	300	600
STRTHRESH	0x20000	0x200000	0x300000
MSGMAP	100	100	200
MSGMNB	4096	8192	16384
MSGMNI	50	100	150
MSGTQL	40	80	160
MSGSEG	1024	1024	2048
SEMAP	50	100	150
SEMMNI	50	100	150
SEMMNS	60	100	200
SEMMNU	60	100	200
SHMMNI	100	100	150
SHMSEG	8	16	32
SPTMAP	100	150	250
NDQUOT	200	200	300
PGOVERFLOW	16	16	32
NOTPGOVERFLOW	16	16	32
SYSSEGSZ	1024	1024	2048

Der Parameter SEGMAPSZ wird auf 1/4 des Hauptspeichers gesetzt. Der Parameter SHMMAX wird auf das Doppelte des Hauptspeichers, maximal aber auf 33554432 gesetzt.

Bitte beachten Sie, daß `iddefaults` nur Standardwerte einstellt. Einzelne Parameter können natürlich bei Bedarf durch `idtune(1M)` im Rahmen der durch `mtune(4)` vorgegebenen Bereiche wieder verändert werden. Die einstellbaren Parameter sind im Anhang B des *"Leitfadens für Systemverwalter"* beschrieben.

Damit die neuen Parametereinstellungen wirksam werden, muß nach dem Aufruf von `iddefaults` ein neuer SINIX-Kern mit `idbuild(1M)` erzeugt und dann gebootet werden.

SIEHE AUCH

`idtune(1M)`, `mtune(4)`

BEZEICHNUNG

idinstall – Hinzufügen, Löschen, Aktualisieren oder Abrufen von Daten zur Gerätetreiberkonfiguration

ÜBERSICHT

/etc/conf/bin/idinstall [-adug] [-e] [-msoptnirhcl] *gerätename*

BESCHREIBUNG

Das Kommando `idinstall` wird von einem Install- oder Remove-Skript des Softwarepakets eines Gerätetreibers (Driver Software Package - DSP) aufgerufen, um Daten zur Gerätetreiberkonfiguration hinzuzufügen (-a), zu löschen (-d), zu aktualisieren (-u) oder aufzurufen (-g). `idinstall` erwartet Dateien mit Treiberkomponenten im aktuellen Verzeichnis. Beim Installieren oder Aktualisieren von Komponenten werden sie auf Dateien im Verzeichnis `/etc/conf` kopiert oder an solche Dateien angehängt und anschließend aus dem aktuellen Verzeichnis gelöscht, sofern nicht die Option `-k` verwendet wird. Folgende Optionen stehen für das Kommando zur Verfügung:

Zur Angabe von Aktionen:

- a DSP-Komponenten hinzufügen
- d DSP-Komponenten entfernen
- u DSP-Komponenten aktualisieren
- g DSP-Komponenten, außer der Master-Komponente, auf die Standardausgabe ausgeben

Zur Angabe von Komponenten: (*)

- m Master-Komponente
- s System-Komponente
- o Komponente Driver.o
- p Komponente Space.c
- t Komponente Stubs.c
- n Komponente Node (Geräte-datei)
- i Komponente Inittab
- r Komponente Device Initialization (rc)
- h Komponente Device Shutdown (sd)
- c Komponente Mfsys: Konfigurationsdaten zum Dateisystemtyp-Typ (Master)
- l Komponente Sfsys: Lokale Daten für Dateisystem-Typ (System)

(*) Wird keine Komponente angegeben, dann sind standardmäßig alle Komponenten angesprochen; eine Ausnahme ist die Option `-g`, bei der eine einzelne Komponente ausdrücklich angegeben werden muß.

Verschiedenes:

- e Prüfung auf freien Plattenplatz ausschalten
- k Dateien beim Hinzufügen oder Aktualisieren nicht aus dem aktuellen Verzeichnis löschen.

Im einfachsten Fall, wenn das Softwarepaket eines neuen Gerätetreibers *gerätename* installiert werden soll, lautet der Kommandoaufruf `idinstall -a gerätename`. Hier erfordert und installiert das Kommando die `Driver.o`-Einträge, Master-Einträge und System-Einträge und optional die Komponenten `Space.c`, `Stubs.c`, `Node`, `Init`, `Rc`, `Shutdown`, `Mfsys` und `Sfsys`, wenn sich diese im aktuellen Verzeichnis befinden.

Die Dateien `Driver.o`, `space.c` und `stubs.c` werden in einem Unterverzeichnis von `/etc/conf/pack.d` angelegt, das den Namen *gerätename* erhält. Die übrigen Komponenten werden in den entsprechenden Unterverzeichnissen von `/etc/conf` als Datei *gerätename* gespeichert. Die Komponente `Node` beispielsweise wird als `/etc/conf/node.d/gerätename` installiert.

Mit `idinstall -m` wird eine Schnittstelle zu dem Kommando `idmaster` hergestellt, mit dem über eine Master-Datei aus dem lokalen Verzeichnis Einträge in der Datei `mdevice` hinzugefügt, gelöscht und aktualisiert werden. Damit steht für die Programmierung von Treibern eine konsistente Schnittstelle zum Installieren beliebiger DSP-Komponenten zur Verfügung.

Wie oben erwähnt, verwendet man im allgemeinen das Kommando nur in der Form `idinstall -a gerätename`. Andere Optionen von `idinstall` ermöglichen die Installation eines aktualisierten Softwarepakets (d.h. eines Pakets, das eine Komponente des vorhandenen Gerätetreibers ersetzt) und unterstützen die Installation mehrerer Controller-Karten gleichen Typs.

Wird beim Aufruf von `idinstall` die Option `-u` (Aktualisieren) angegeben, dann werden die Dateien des alten Softwarepakets durch die des neuen überschrieben. Außerdem wird das Kommando `idmaster` mit der Aktualisierungsoption aufgerufen, wenn ein Master-Modul Teil des neuen Softwarepakets ist.

`idinstall` überprüft auch, ob genügend Plattenplatz frei ist, um die Neukonfiguration zu starten. Dazu wird das Kommando `idspace` aufgerufen. `idinstall` schlägt fehl, wenn nicht genügend Plattenplatz vorhanden ist, und wird mit einem Return-Code ungleich Null abgebrochen. Mit der Option `-e` wird diese Prüfung umgangen.

Unterstützt ein Treiber sowohl block- als auch zeichenorientierte Ein-/Ausgabe, weist `idinstall` Geräteklassennummern für beide E/A-Typen zu. Diese Geräteklassennummern müssen nicht identisch sein. SCSI-Entwickler, die dafür identische Geräteklassennummern brauchen, müssen in der Master-Datei an das dritte Feld ein neues Feld 'v' anfügen.

`idinstall` legt einen Datensatz für das letzte installierte Gerät in der Datei `/etc/.last_dev_add` an und sichert alle bei der letzten Löschoperation entfernten Dateien in dem Verzeichnis `/etc/.last_dev_del`. Diese Dateien lassen sich mit `/etc/conf/bin/idmkenv` wiederherstellen, wenn festgestellt wird, daß eine Neukonfiguration wegen Stromausfalls oder eines unerwarteten Wiederladens (Reboot) des Systems abgebrochen wurde.

idinstall(1M)

idinstall(1M)

SIEHE AUCH

idspace(1M), idcheck(1M).

mdevice(4), sdevice(4) im *Referenzhandbuch für Programmierer*.

FEHLERMELDUNGEN

Der Exit-Code Null bedeutet erfolgreiche Ausführung. Wurde ein Fehler festgestellt, schließt `idinstall` mit einem anderen Wert ab und gibt eine Fehlermeldung aus. Die Fehlermeldungen bedürfen keiner weiteren Erklärung. Folgende Fehlermeldungen werden mit `idinstall` normalerweise ausgegeben:

Device package already exists.

(Gerätepaket besteht bereits.)

Cannot make the driver package directory.

(Verzeichnis für Treiberpaket kann nicht erstellt werden.)

Cannot remove driver package directory.

(Verzeichnis des Treiberpakets kann nicht gelöscht werden.)

Local directory does not contain a Driver object (Driver.o) file.

(Lokales Verzeichnis enthält keine Treiberobjektdatei (Driver.o).)

Local directory does not contain a Master file.

(Lokales Verzeichnis enthält keine Master-Datei.)

Local directory does not contain a System file.

(Lokales Verzeichnis enthält keine System-Datei.)

Cannot remove driver entry.

(Treibereintrag kann nicht gelöscht werden.)

BEZEICHNUNG

`idmkernel` – liest Dateien, die Spezifikationen enthalten

ÜBERSICHT

`/etc/conf/bin/idmkernel`

BESCHREIBUNG

Dieses Kommando liest Dateien mit Spezifikationen zu `/etc/inittab`-Einträgen aus `/etc/conf/init.d` und erstellt eine neue `inittab`-Datei in `/etc/conf/cf.d`. Bei erfolgreicher Ausführung wird eine 0 ausgegeben, bei Fehlern eine positive Zahl.

Die Dateien in `/etc/conf/init.d` sind Kopien der Init-Module in Softwarepaketen von Gerätetreibern (DSP). Pro DSP darf höchstens eine Init-Datei vorhanden sein. In jeder Datei ist für jeden zu installierenden `inittab`-Eintrag eine Zeile angelegt. In einer Datei können mehrere Zeilen, d.h. mehrere `inittab`-Einträge enthalten sein. Ein `inittab`-Eintrag hat folgendes Format (das Feld *nr* wird häufig als *marke* oder *tag* bezeichnet):

nr:btrart:aktion:prozeß

Der Eintrag im Init-Modul muß eines der folgenden Formate haben:

aktion:prozeß

btrart:aktion:prozeß

nr:btrart:aktion:prozeß

Trifft `idmkernel` auf einen Eintrag des ersten Typs, wird ein Feld mit einer gültigen *nr* angelegt, ebenso ein Feld mit *btrart* 2 (das die Betriebsart 2 bezeichnet). Bei einem Eintrag des zweiten Typs wird lediglich das Feld *nr* vorangestellt. Ein Eintrag des dritten Typs wird unverändert in die neue `inittab` übernommen.

Da zusätzliche `inittab`-Einträge in den meisten Fällen im Feld *btrart* den Wert 2 angeben, kann fast immer ein Eintrag des ersten Typs verwendet werden. Ein Eintrag des zweiten Typs kann verwendet werden, wenn eine andere Betriebsart als 2 gewünscht wird. Bei Treiber-Softwarepaketen sollte das Feld *nr* möglichst nicht angegeben werden, wie im dritten Eintragstyp, da andere zusätzliche Anwendungen oder DSPs den angegebenen *nr*-Wert eventuell bereits verwendet haben. In dem Programm `/etc/init` kommt es zu schweren Fehlern, wenn mehrere `inittab`-Einträge im Feld *nr* denselben Wert aufweisen.

`idmkernel` verlangt für jeden Eintrag ein gültiges *aktions*-Schlüsselwort und bestimmt damit, welches der genannten Formate für den Eintrag verwendet wird. Folgende gültige Werte für *aktion* stehen zur Verfügung:

off
respawn
ondemand
once
wait
boot
bootwait
powerfail
powerwait


```
initdefault
sysinit
```

Das Kommando `idmkninit` wird automatisch aufgerufen, wenn beim nächsten Neuladen des Systems nach einer Neukonfiguration des Systemkerns Betriebsart 2 aktiviert wird, um die richtige `/etc/inittab` für den aktuellen `/stand/unix`-Kernel einzurichten. Das Kommando `idmkninit` kann auf Benutzerebene abgesetzt werden, um Änderungen an `inittab` zu überprüfen, bevor ein DSP aufgebaut wird. Auch für Installations-Skripts, die zwar den Kernel nicht neu konfigurieren, aber `inittab`-Einträge anlegen müssen, ist dieses Kommando nützlich. In einem solchen Fall muß die mit `idmkninit` erstellte `inittab` nach `/etc/inittab` kopiert werden, und das Kommando `telinit q` muß ausgeführt werden, um den neuen Eintrag in Kraft zu setzen.

Folgende Optionen stehen für die Kommandozeile zur Verfügung:

- o *verzeichnis* `inittab` wird in dem angegebenen Verzeichnis und nicht in `/etc/conf/cf.d` erstellt.
- i *verzeichnis* Die Datei `init.base`, die Kennnummern enthält und normalerweise in `/etc/conf/cf.d` steht, ist in dem angegebenen Verzeichnis zu finden.
- e *verzeichnis* Die Init-Module, die gewöhnlich in `/etc/conf/init.d` stehen, befinden sich jetzt in dem angegebenen Verzeichnis.
- # Ausgeben von Daten zur Fehlersuche.

FEHLERMELDUNGEN

Bei erfolgreicher Ausführung lautet der Exit-Code Null. Wurde ein Fehler festgestellt, endet `idmkninit` mit einem anderen Wert, und eine Fehlermeldung wird ausgegeben. Alle Fehlermeldungen sind ohne weitere Erklärung verständlich.

SIEHE AUCH

`idbuild(1)`, `idinstall(1M)`, `idmknod(1M)`, `init(1M)`.

`inittab(4)` im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

`idmknod` – löscht Knoten und liest Spezifikationen zum Erzeugen von Knoten

ÜBERSICHT

`/etc/conf/bin/idmknod [-o dtv] [-i dtv] [-e dtv] [-s]`

BESCHREIBUNG

Dieses Kommando erfüllt folgende Funktionen:

Entfernen der Knoten für nicht benötigte Geräte (d.h. solche Geräte, bei denen Feld 3 des `mdevice`-Eintrags kein `r` enthält) aus `/dev`. Normale Dateien werden nicht gelöscht. Enthält das Verzeichnis `/dev` Unterverzeichnisse, werden diese ebenfalls durchsucht und dort gefundene Knoten für nicht benötigte Geräte ebenfalls gelöscht. Entstehen durch das Löschen von Knoten leere Unterverzeichnisse, werden diese anschließend ebenfalls entfernt.

Lesen der Spezifikationen für Knoten aus den Dateien in `/etc/conf/node.d` und Installation dieser Knoten in `/dev`. Wenn die Knotenspezifikation einen Pfad mit Unterverzeichnissen angibt, werden diese automatisch angelegt.

Rückgabe von 0 bei erfolgreicher Ausführung und einer positiven Zahl bei einem Fehler.

Das Kommando `idmknod` wird automatisch ausgeführt, wenn beim nächsten Systemstart nach einer Neukonfiguration des Kerns Betriebsart 2 oder 3 angegeben wird, um die richtige Darstellung der Geräteknoten im Verzeichnis `/dev` für den aktuellen `/unix`-Kern einzurichten. Das Kommando `idmknod` kann auf Benutzerebene aufgerufen werden, um Änderungen am Verzeichnis `/dev` zu überprüfen, bevor ein Softwarepaket für Gerätetreiber (DSP) aufgebaut wird. Das Kommando ist auch bei Installations-Skripts nützlich, die zwar den Kern nicht neu konfigurieren, aber Einträge in `/dev` anlegen müssen.

Die Dateien in `/etc/conf/node.d` sind Kopien der Node-Komponenten. Die Dateien werden beim Installieren von Softwarepaketen für Gerätetreiber (DPS) erzeugt (siehe `idinstall(1M)`). Pro DSP existiert höchstens eine solche Datei. Für jeden zu installierenden Knoten enthält diese Datei eine Zeile, die folgendes Format hat:

Name des Geräteeintrags (Feld 1) in der Datei `mdevice`. Der `mdevice`-Eintrag besteht aus einer Zeile, die das DSP gemäß seiner Master-Komponente nach `mdevice` schreibt (siehe `idinstall(1M)`). Dieses Feld darf 1 bis 8 Zeichen lang sein. Das erste Zeichen muß ein Buchstabe sein. Die übrigen können Buchstaben, Ziffern oder Unterstriche sein.

Name des Knotens, der in `/dev` eingefügt werden soll. Das erste Zeichen muß ein Buchstabe sein, die übrigen Zeichen können Buchstaben, Ziffern oder Unterstriche sein. Dieses Feld kann einen Pfad, bezogen auf `/dev`, enthalten. `idmknod` legt dann bei Bedarf die entsprechenden Unterverzeichnisse an.

Das Zeichen `b` oder `c`. `b` gibt an, daß es sich bei dem Knoten um ein 'block'-orientiertes Gerät handelt, und `c` steht für ein 'zeichen'-orientiertes Gerät.

Bei Geräten mit mehreren Geräteklassennummern wird anhand des folgenden Schemas angegeben, welche Geräteknoten zu welcher Geräteklasse gehören. Das dritte Feld ist erweitert, so daß darin ein Geräteklassen-Offset folgendermaßen angegeben werden kann:

[b/c]: *gkl_off*. Dabei bezieht sich [b/c] auf block- oder zeichenorientierte Geräteklassen und *gkl_off* auf eine Offsetnummer innerhalb des Bereichs der Geräteklassennummern in der Datei *mdevice*. Die Angabe "C:2" zum Beispiel bezieht sich auf den Offset 2 für eine zeichenorientierte Geräteklasse. Bei einem Geräteklassenbereich von "15-18" wäre dies die zeichenorientierte Geräteklasse 17.

Gerätenummer. Dies muß ein Wert zwischen 0 und 255 sein. Enthält dieses Feld eine nicht numerische Angabe, wird angenommen, daß es sich um eine Anforderung für einen Streams-Klon-Geräteknoten handelt, und *idmknod* setzt die Gerätenummer auf den Wert der Geräteklassennummer des angegebenen Geräts.

UID. Gibt die UID des Benutzers an, dem das Gerät gehört. Dieses Feld ist optional.

GID. Gibt die GID der Gruppe an, zu der das Gerät gehört. Dieses Feld ist optional.

Zugriffsrechte. Gibt eine Maske für die Zugriffsrechte an, mit denen das Gerät versehen wird. Dieses Feld ist optional.

Hier einige Beispiele für Einträge in der Knotendatei:

```
asy    tty00    c    1
        erstellt /dev/tty00 für Gerät asy mit Gerätenummer 1.

qt     rmt/c0s0    c    4
        erstellt /dev/rmt/c0s0 für Gerät qt mit Gerätenummer 4.

clone  net/nau/clone    c    nau
        erstellt /dev/net/nau/clone für Gerät clone. Die
        Gerätenummer erhält den Wert der Geräteklassennummer für
        das Gerät nau.

scsi   tty1 C:0    5
        erstellt tty1 für Gerät scsi mit Gerätenummer 1 und
        Geräteklassennummer Offset 0.
```

`idmknod` stellt folgende Optionen zur Verfügung:

- o *dtv* Die Knoten werden nicht in `/dev`, sondern in dem angegebenen Verzeichnis installiert.
- i *dtv* Die Datei `mdevice`, die normalerweise in `/etc/conf/cf.d` steht, befindet sich in dem angegebenen Verzeichnis.
- e *dtv* Die Node-Dateien, die normalerweise in `/etc/conf/node.d` stehen, befinden sich in dem angegebenen Verzeichnis.
- s Unterdrückt das Löschen von Knoten (fügt nur neue Knoten hinzu).

FEHLERMELDUNGEN

Der Exit-Code Null steht für erfolgreiche Ausführung. Wenn aufgrund eines Syntax- oder Formatfehlers in einem Knoteneintrag ein Fehler festgestellt wird, wird ein entsprechender Hinweis in die Standardausgabe geschrieben und das Kommando weiter ausgeführt. Tritt ein schwerer Fehler auf, dann endet `idmknod` mit einem Exit-Code ungleich Null. Es wird eine Fehlermeldung ausgegeben. Alle Fehlermeldungen sind selbsterklärend. Ein schwerer Fehler tritt z.B. auf, wenn eine benötigte Datei nicht zu finden ist.

SIEHE AUCH

`idinstall(1M)`, `idmkinit(1M)`.

`mdevice(4)`, `sdevice(4)` im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

idmkunix – baut einen neuen SINIX-Systemkern auf

ÜBERSICHT

/etc/conf/bin/idmkunix

BESCHREIBUNG

Das Kommando `idmkunix` erstellt einen ladbaren Kernel für das Betriebssystem SINIX in dem Verzeichnis `/etc/conf/cf.d`. Als Eingabe dienen die Dump-Dateien für die Kernelkomponenten und die Gerätetreiber-Objektdateien in Unterverzeichnissen von `/etc/conf/pack.d`, ferner die von `idconfig` erstellten Geräte- und Parameterdefinitionsdateien. Hier in Kürze die benötigten Eingabedateien:

<code>/etc/conf/cf.d/conf.c</code>	Kerneldatenstrukturen und Funktionsdefinitionen
<code>/etc/conf/cf.d/config.h</code>	Kernelparameter- und Gerätedefinitionen
<code>/etc/conf/cf.d/vector.c</code>	Interrupt-Vektordefinitionen
<code>/etc/conf/cf.d/direct</code>	Liste aller am Aufbau beteiligten Treiberkomponenten
<code>/etc/conf/cf.d/fsconf.c</code>	Konfigurationsdaten zum Dateisystem-Typ
<code>/etc/conf/cf.d/vuifile</code>	Speicherverwaltungsdefinitionen für den Kernel
<code>/etc/conf/pack.d/*/Driver.o</code>	Objektdateien für Kernelkomponenten
<code>/etc/conf/pack.d/*/space.c</code>	Speicherzuweisungsdateien für Kernelkomponenten
<code>/etc/conf/pack.d/*/stubs.c</code>	stubs-Dateien für Kernelkomponenten

Folgende Optionen stehen für die Kommandozeile zur Verfügung:

- `-o verzeichnis` Die Datei `unix` soll in dem angegebenen Verzeichnis statt in `/etc/conf/cf.d` angelegt werden.
- `-i verzeichnis` Eingabedateien, die normalerweise in `/etc/conf/cf.d` stehen, befinden sich in dem angegebenen Verzeichnis.
- `-r verzeichnis` Das angegebene Verzeichnis wird anstelle von `/etc/conf` als `root`-Verzeichnis verwendet.
- `-c, cc, -l, ld` Diese Optionen werden nicht mehr unterstützt.
- `-#` Gibt Daten zur Fehlersuche aus.

FEHLERMELDUNGEN

Der Exit-Code Null steht für erfolgreiche Ausführung. Wurde ein Fehler gefunden, schließt `idmkunix` mit einem anderen Wert ab und gibt eine Fehlermeldung aus. Alle Fehlermeldungen sind ohne Erklärung verständlich.

SIEHE AUCH

`idbuild(1M)`, `idconfig(1M)`, `idinstall(1M)`, `mdevice(4)`, `mtune(4)`, `sdevice(4)`, `stune(4)`.

BEZEICHNUNG

idspace – überprüft freien Speicherplatz

ÜBERSICHT

```
/etc/conf/bin/idspace [ -i i-nodes ] [ -r blöcke ] [ -u blöcke ] [ -t blöcke ]
```

BESCHREIBUNG

Dieses Kommando überprüft freien Speicherplatz in den Dateisystemen `/`, `/usr` und `/tmp`, um festzustellen, ob noch genügend Plattenblöcke und I-Nodes vorhanden sind. `idspace` führt folgende Standardtests durch:

`idspace` prüft, ob das Dateisystem `root (/)` um 400 freie Blöcke größer ist als die aktuelle Datei `/stand/unix`. Dies bedeutet, daß ein Gerätetreiber, der `/stand/unix` hinzugefügt werden soll, im Verzeichnis `root` aufgebaut und dort gespeichert werden kann. Weiter wird überprüft, ob im Verzeichnis `root` 100 freie I-Nodes vorhanden sind.

`idspace` stellt fest, ob ein Dateisystem `/usr` existiert. Ist es vorhanden, wird geprüft, ob darin 400 freie Blöcke und 100 freie I-Nodes zur Verfügung stehen. Existiert das Dateisystem nicht, gibt `idspace` keine Meldung aus. Denn dann werden Dateien, die beim Neukonfigurieren in `/usr` erstellt werden sollten, stattdessen im Dateisystem `root` angelegt. Dessen freier Speicherplatz ist aber mit dem oben beschriebenen Test bereits geprüft worden.

`idspace` stellt fest, ob ein Dateisystem `/tmp` vorhanden ist. Ist dies der Fall, wird geprüft, ob darin 400 freie Blöcke und 100 I-Nodes verfügbar sind. Existiert es nicht, gibt `idspace` keine Meldung aus. Denn dann werden Dateien, die beim Neukonfigurieren in `/tmp` erstellt werden sollten, stattdessen im Dateisystem `root` angelegt. Dessen freier Speicherplatz ist aber mit dem oben beschriebenen Test bereits geprüft worden.

`idspace` stellt folgende Optionen zur Verfügung:

- `-i i-nodes` `idspace` prüft, ob in den Dateisystemen noch mindestens *i-nodes* I-Nodes verfügbar sind. Diese Option setzt damit den Standardtest außer Kraft, mit dem das Vorhandensein von 100 freien I-Nodes festgestellt wird.
- `-r blöcke` `idspace` prüft, ob im Dateisystem `root (/)` zusätzlich zu `/stand/unix` noch mindestens *blöcke* Blöcke frei sind. Diese Option setzt den Standardtest außer Kraft, mit dem das zusätzliche Vorhandensein von 400 freien Blöcke geprüft wird. Wenn Sie diese Option verwenden, dann testet `idspace` die Dateisysteme `/usr` und `/tmp` nur dann, wenn Sie es ausdrücklich angeben.
- `-u blöcke` `idspace` prüft, ob im Dateisystem `/usr` noch mindestens *blöcke* Blöcke frei sind. Diese Option setzt den Standardtest außer Kraft, mit dem das Vorhandensein von 400 freien Blöcke geprüft wird. Wenn Sie diese Option verwenden, dann testet `idspace` die Dateisysteme `root (/)` und `/tmp` nur dann, wenn Sie es ausdrücklich angeben. Ist `/usr` kein separates Dateisystem, dann wird ein Fehler gemeldet.

idspace(1M)

idspace(1M)

-t *blöcke* idspace prüft, ob im Dateisystem `/tmp` noch mindestens *blöcke* Blöcke frei sind. Diese Option setzt den Standardtest außer Kraft, mit dem das Vorhandensein von 400 freien Blöcke geprüft wird. Wenn Sie diese Option verwenden, dann testet idspace die Dateisysteme `root (/)` und `/usr` nur dann, wenn Sie es ausdrücklich angeben. Ist `/tmp` kein separates Dateisystem, dann wird ein Fehler gemeldet.

FEHLERMELDUNGEN

Der Exit-Code Null steht für erfolgreiche Ausführung. Reicht in einem Dateisystem der Plattenplatz nicht aus oder wird aufgrund eines Syntax- oder Formatfehlers ein Fehler festgestellt, gibt idspace eine entsprechende Meldung aus. Alle Fehlermeldungen sind ohne weitere Erklärung verständlich. Die Exit-Codes von idspace haben folgende Bedeutung:

- 0 erfolgreiche Ausführung
- 1 Fehler in der Kommandosyntax, oder die benötigte Datei existiert nicht
- 2 Speicherplatz oder I-Nodes im Dateisystem reichen nicht aus
- 3 angefordertes Dateisystem existiert nicht (nur bei den Optionen `-u` und `-t`)

SIEHE AUCH

`idbuild(1M)`, `idinstall(1M)`.

BEZEICHNUNG

id_tune – versucht, den Wert eines einstellbaren Parameters zu definieren

ÜBERSICHT

/etc/conf/bin/id_tune [-f | -m] *name wert*

BESCHREIBUNG

Dieses Skript versucht, den Wert eines einstellbaren Parameters zu definieren. Der zu ändernde einstellbare Parameter wird mit *name* bezeichnet. Der gewünschte Wert für den einstellbaren Parameter ist *wert*.

Hat dieser Parameter bereits einen Wert in der Datei *stune*, wird der Benutzer mit der folgenden Meldung aufgefordert, die Änderung zu bestätigen:

```
Tunable Parameter name is currently set to old_value.
Is it OK to change it to value? (y/n)
(Einstellbarer Parameter name hat zur Zeit den Wert alter_wert.
Soll dieser in wert geändert werden? (y/n))
```

Gibt der Benutzer *y* an, wird die Änderung ausgeführt. Ansonsten wird der einstellbare Parameter nicht geändert, und die folgende Meldung wird ausgegeben:

```
name left at old_value.
(name bleibt bei alter_wert.)
```

Wird jedoch die Option *-f* (force - erzwingen) verwendet, wird die Änderung in jedem Falle vorgenommen, und es werden keinerlei Meldungen ausgegeben.

Wird die Option *-m* (Minimum) verwendet und es ist ein Wert vorhanden, der größer als der gewünschte ist, wird die Änderung nicht ausgeführt und keine Meldung ausgegeben.

Werden einstellbare Systemparameter als Teil eines Gerätetreibers oder eines zusätzlichen Anwendungspakets geändert, ist es eventuell nicht wünschenswert, die obige Frage auszugeben. Das Installations-Skript des Zusatzpakets kann den bestehenden Wert mit Hilfe der Optionen *-f* oder *-m* überschreiben. Es ist jedoch sorgfältig darauf zu achten, daß kein einstellbarer Parameter außer Kraft gesetzt wird, der durch einen Benutzer oder ein anderes Zusatzpaket vorher definiert wurde.

Damit die Parameteränderung wirksam werden kann, muß der Systemkern neu aufgebaut und das System neu geladen werden.

EXIT-CODES

Der Exit-Code ist ungleich Null, wenn ein Fehler festgestellt wird.

SIEHE AUCH

idbuild(1).

mtune(4), stune(4) im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

ifconfig – konfiguriert Schnittstellenparameter im Netzwerk

ÜBERSICHT

```
ifconfig schnittstelle [ adrefamilie ] [ adresse [ zieladresse ] ] [ parameter ]
      [ netmask maske ] [ broadcast adresse ] [ metric n ]
```

```
ifconfig schnittstelle [ protokollfamilie ]
```

BESCHREIBUNG

ifconfig dient dazu, einer Netzwerkschnittstelle eine Adresse zuzuweisen und/oder Parameter der Netzwerkschnittstelle zu konfigurieren. ifconfig muß beim Start des Systems ausgeführt werden, um die Netzwerkadresse aller auf dem Rechner vorhandenen Schnittstellen zu definieren; zu einem späteren Zeitpunkt kann es auch benutzt werden, um eine Schnittstellenadresse oder andere Betriebsparameter zu ändern. Ohne Optionen zeigt ifconfig die aktuelle Konfiguration für eine Netzwerkschnittstelle an. Wird eine Protokollfamilie angegeben, enthält die Meldung von ifconfig nur Angaben zu dieser Familie. Nur der Superuser darf die Konfiguration einer Netzwerkschnittstelle ändern.

Der Parameter *schnittstelle* ist eine Zeichenkette mit der Form *name einheit*, z.B. *emdl*. Der Schnittstellenname *-a* ist reserviert und bewirkt, daß die übrigen Argumente nacheinander auf alle Adressen aller Schnittstellen angewandt werden.

Da eine Schnittstelle Übertragungen mit unterschiedlichen Protokollen empfangen kann, die u.U. jeweils eigene Namenskonventionen erfordern, werden die Parameter und Adressen nach den Regeln einer Adrefamilie interpretiert, die mit dem Parameter *adrefamilie* angegeben wird. Zur Zeit werden die Adrefamilien *ether* und *inet* unterstützt. Ist keine Adrefamilie angegeben, wird *inet* angenommen.

Für die DARPA Internet-Familie (*inet*) ist die Adresse entweder ein Hostname in der Hostnamen-Datenbank (siehe *hosts(4)*) oder eine DARPA Internet-Adresse in der standardmäßigen Punktschreibweise von Internet. Normalerweise besteht eine in Punktschreibweise angegebene Internet-Adresse aus der Netzwerknummer des Systems und der eindeutigen Host-Nummer des Rechners. Eine typische Internet-Adresse ist 192.9.200.44, wobei 192.9.200 die Netzwerknummer und 44 die Host-Nummer des Rechners ist.

Bei der Adrefamilie *ether* handelt es sich um Ethernet-Adressen, die folgendermaßen dargestellt werden: *x:x:x:x:x*. Dabei ist *x* eine Hexadezimalzahl zwischen 0 und ff. Diese Adrefamilie darf nur der Superuser verwenden.

Ist zusätzlich zu dem Parameter *adresse* der Parameter *zieladresse* vorhanden, wird damit die Adresse des Gegenstücks am anderen Ende einer Punkt-zu-Punkt-Verbindung angegeben.

OPTIONEN

Die folgenden *parameter* können mit *ifconfig* definiert werden:

- up Schnittstelle als aktiv (up) kennzeichnen. Damit kann eine Schnittstelle nach einem *ifconfig down* aktiviert werden. Dies geschieht automatisch bei der Eingabe der ersten Adresse einer Schnittstelle. Wurde eine zuvor als inaktiv gekennzeichnete Schnittstelle zurückgesetzt, wird die Hardware neu initialisiert.
- down Schnittstelle als inaktiv (down) kennzeichnen. Ist eine Schnittstelle als inaktiv gekennzeichnet, versucht das System nicht, Nachrichten über diese Schnittstelle zu übertragen. Wenn möglich wird sie auch für den Empfang deaktiviert (disable reception). Mit dieser Maßnahme werden Routen, die diese Schnittstelle benutzen, nicht automatisch deaktiviert.
- trailers (nur *inet*) Verwendung einer IP-Trailer-Einbettung auf Verweisebene beim Senden aktivieren. Unterstützt eine Netzwerkschnittstelle die IP-Trailer-Einbettung, kapselt das System abgehende Meldungen so, daß die Anzahl der vom Empfänger auszuführenden Kopiervorgänge von Speicher zu Speicher auf ein Minimum beschränkt wird. Diese Funktion ist rechnerabhängig und daher nicht empfehlenswert. Bei Netzwerken, die das Address Resolution Protocol (Adreßauflösungsprotokoll) unterstützen (siehe *arp(7)*) - derzeit ist dies nur Ethernet mit 10 MB/s, zeigt diese Option an, daß das System andere Systeme auffordern soll, beim Senden an diesen Host die IP-Trailer-Einbettung zu verwenden. IP-Trailer-Einbettung wird auch eingesetzt, wenn andere Hosts dies fordern und an sie gesendet wird.
- trailers Deaktivieren der IP-Trailer-Einbettung auf Verweisebene.
- arp Aktivieren des Address Resolution Protocol beim Zuordnen von Adressen auf Netzwerkebene zu Adressen auf Verbindungsebene (Standard). Dies ist gegenwärtig für die Zuordnung zwischen DARPA Internet-Adressen und 10MB/s-Ethernet-Adressen möglich.
- arp Deaktivieren des Address Resolution Protocol.
- metric *n* Einstellen des metrischen Werts für die Route auf *n*, Standardwert 0. Der metrische Wert für die Route wird vom Routing-Protokoll verwendet (*routed(1M)*). Je höher der metrische Wert ist, um so ungünstiger ist eine Route; metrische Werte gelten als zusätzliche Knoten auf dem Weg zum Ziel-Netzwerk oder -Host.
- netmask *maske* (nur *inet*) Angabe, wie viele Elemente der Adresse zum Unterteilen von Netzwerken in Subnetze reserviert werden sollen. Die Maske umfaßt den Netzwerkteil der lokalen Adresse und den Subnetzteil, der dem Host-Feld der Adresse entnommen wird. Die Maske kann als einzelne Hexadezimalzahl, der 0x vorangestellt ist, als Internet-Adresse in Punktschreibweise oder als in der Netzwerktabelle *net-works(4)* enthaltener Pseudo-Netzwerkname angegeben werden. Die Maske enthält für die zu Netz und Subnetz gehörenden Positionen der 32-Bit-Adresse Einsen, für die zum Host gehörenden Positionen

Nullen. Sie muß mindestens den standardmäßigen Netzwerkteil enthalten, und das Subnetzfeld muß an den Netzwerkteil angrenzen. Wird beim Wert für `netmask` ein Pluszeichen + angegeben, wird die Netzwerknummer in der Datei `/etc/netmasks` gesucht.

`broadcast` *adresse*

(nur `inet`) Angabe der Adresse, mit der Meldungen an alle Benutzer gegenüber dem Netzwerk dargestellt werden. Die standardmäßige broadcast-Adresse enthält im Host-Teil nur Nullen. Wird der Wert mit einem Pluszeichen + angegeben, wird die broadcast-Adresse auf einen Standardwert zurückgesetzt, der für die (eventuell neue) Internet-Adresse und als Netzmaske geeignet ist. Bitte beachten Sie, daß die Argumente zu `ifconfig` von links nach rechts interpretiert werden, und daher ist es möglich, daß bei

```
ifconfig -a netmask + broadcast +
```

und

```
ifconfig -a broadcast + netmask +
```

unterschiedliche Werte für die broadcast-Adressen der Schnittstellen zugewiesen werden.

BEISPIELE

Wenn der Arbeitsplatzrechner nicht an ein Ethernet angeschlossen ist, muß die Schnittstelle `emdl` folgendermaßen als inaktiv gekennzeichnet werden:

```
ifconfig emdl down
```

Zum Ausgeben von Adressierdaten für die einzelnen Schnittstellen dient

```
ifconfig -a
```

Um die broadcast-Adresse der Schnittstelle zurückzusetzen, nachdem alle Netzmasken ordnungsgemäß definiert wurden, verwendet man

```
ifconfig -a broadcast +
```

DATEIEN

```
/dev/nit  
/etc/netmasks
```

SIEHE AUCH

`netstat(1M)`, `netmasks(4)`.

FEHLERMELUNGEN

Es werden Meldungen ausgegeben, wenn die angegebene Schnittstelle nicht vorhanden ist, die angeforderte Adresse unbekannt ist oder ein Benutzer ohne die entsprechenden Zugriffsberechtigungen versucht hat, die Schnittstellenkonfiguration zu ändern.

BEZEICHNUNG

incfile – Anlegen, Wiederherstellen eines Teilarchivs von Dateisystemen

ÜBERSICHT

```
incfile -B [-dilmortvxAENSV] siaufnr udsname udsgcr widsgcr wiaufnr beschr
incfile -T siaufnr ivzdname beschr
incfile -RC [-dilmortvxAENSV] udsname udsgcr widsgcr wiaufnr beschr
incfile -RF [-dilmortvxAENSV] udsname udsgcr beschr wiaufnr:uid:datum:typ:name
[:[rename]:[inode]] ...
```

BESCHREIBUNG

incfile wird als Sohnprozeß von anderen Shell-Kommandos aufgerufen. Der Kommandoname incfile wird entweder aus der Datei bkhist.tab oder dem Kommando (mit Option) bkreg -m gelesen. Die Optionen -B, -T, -R, -F und -C werden dem Kommando incfile von den nachstehend beschriebenen Shell-Kommandos backup, restore und urestore(1) übergeben. Die mit einem Minuszeichen versehenen Optionen werden von der Datei bkhist.tab oder dem Kommando (mit Option) bkreg -p übergeben. Die Argumente werden von verschiedenen Positionen im Backup-Service an incfile gesendet.

incfile -B wird als Sohnprozeß von dem Kommando bkdaemon aufgerufen, um eine Teilsicherung des Dateisystems *udsname* (des ursprünglichen Dateisystems) durchzuführen. Alle Dateien in *udsname*, die seit der letzten vollständigen Sicherung geändert wurden oder bei denen der I-Node geändert wurde, werden archiviert. Die entsprechende Sicherungskopie wird im cpio-Dateiformat erstellt. Für die Sicherungsoperation wird ein Datensatz in das Sicherungsprotokoll /etc/bkup/bkhist.tab geschrieben.

siaufnr

ist die von backup zugewiesene Auftragsnummer. Die Sicherungsmethode verwendet *siaufnr* beim Erstellen des Protokolls und der Inhaltsverzeichnis-Einträge.

udsname

der Name des Dateisystems, von dem eine Sicherungskopie erstellt werden soll.

udsgcr

der Name des blockorientierten SINIX-Geräts, auf dem sich das Dateisystem befindet.

udset

der Name des Datenträgers im Dateisystem (siehe labelit(1M)).

beschr

ist die Beschreibung eines Ziel-Geräts in der Form:

ggruppe:gname:gmerk:geti

ggruppe

gibt eine Gerätegruppe an (siehe devgroup.tab(4)).

gname gibt einen bestimmten Gerätenamen an (siehe device.tab(4)).

gmerk gibt zu dem Gerät gehörende Merkmale an. Ist *gmerk* angegeben, werden dadurch die Standardwerte für das angegebene Gerät und die angegebene Gruppe außer Kraft gesetzt. (Eine Beschreibung der Gerätemerkmale finden Sie unter device.tab(4).)

geti gibt die Datenträgernamen für die Datenträger an, die zum Lesen des Archivs bzw. zum Schreiben in das Archiv verwendet werden sollen.

incfile -T wird als Sohnprozeß von dem Kommando *backup* aufgerufen, um ein Inhaltsverzeichnis für die in *beschr* genannten Datenträger zu archivieren.

ivzdname

der Name der Datei, die das Inhaltsverzeichnis enthält.

incfile -RC und *incfile -RF* werden von dem Kommando *rsoper* als Sohnprozesse aufgerufen, um Dateien aus einem von *incfile -B* erstellten Teilsicherungsarchiv für ein Dateisystem zu extrahieren. Es wird vorausgesetzt, daß das Dateisystemarchiv im *cpio*-Format vorliegt.

Wird die Option *-RC* ausgewählt, werden alle Dateien im Archiv wiederhergestellt.

widsname

der Name des wiederherzustellenden Dateisystems anstelle von *udsname*, wenn der Wert nicht Null ist.

wiger die wiederherzustellene Partition anstelle von *udsgcr*, wenn der Wert nicht Null ist.

Mindestens einer der Werte *widsname* und *wiger* muß Null sein.

Ist die Option *-RF* angegeben, werden nur ausgewählte Objekte aus dem Archiv wiederhergestellt. Jedes 7-Tupel, das aus den Elementen *wiaufnr:uid:datum:typ:name:i-node* besteht, gibt ein Objekt an, das aus dem Dateisystemarchiv wiederhergestellt werden soll. Die 7teiligen Objekte für *incfile* stammen aus der Datei *rsstatus.tab*.

wiaufnr

die von *restore* oder *urestore* zugewiesene Auftragsnummer zum Wiederherstellen.

uid die reale Benutzernummer des Benutzers, der das Objekt wiederherstellen will. Diese Nummer muß mit der Benutzernummer übereinstimmen, die der Eigentümer bei der Erstellung des Archivs hatte, oder es muß die Benutzernummer des Superusers sein.

datum das neueste "Datum der letzten Änderung", das für ein wiederherstellbares Objekt gültig ist. Das Objekt wird aus dem Archiv wiederhergestellt, das direkt vor diesem Datum erstellt wurde. *datum* ist eine hexadezimale Darstellung des Datums und der Uhrzeit und wird mit dem Systemaufruf *time* ermittelt.

typ F oder D, d.h. das Objekt ist entweder eine Datei (file) oder ein Verzeichnis (directory).

name der Name des Objekts im Dateisystemarchiv.

winame

der Name, mit dem das Objekt wiederhergestellt werden soll. Dies kann ein anderer Name sein als der, den das Objekt in dem Dateisystemarchiv hatte. Fehlt diese Angabe, wird das Objekt als *name* wiederhergestellt.

i-node die I-Node-Nummer des Objekts, als es im Dateisystemarchiv gespeichert wurde. *i-node* wird nicht von `incfile -R` verwendet und wird nur angegeben, um die Kompatibilität der Kommandozeile mit anderen Wiederherstellungsmethoden zu gewährleisten.

Optionen

Manche Optionen sind nur für das Kommando `incfile -B` von Bedeutung; beim Aufruf von `incfile -R` werden sie akzeptiert, aber ignoriert, da `restore` das Kommando automatisch aufruft und die Optionen angibt. Diese Optionen sind mit einem Stern (*) markiert.

- d* Verhindert, daß das Archiv im Sicherungsprotokoll aufgezeichnet wird.
- i* Schließt Dateien von der Sicherungsoperation aus, bei denen sich lediglich der I-Node geändert hat.
- l* Erstellt eine ausführliche Version des Sicherungsprotokolls einschließlich eines Inhaltsverzeichnisses für das Archiv. Hierzu gehören die Daten, anhand derer eine Liste aller Dateien im Archiv erstellt wird, wie dies mit dem Kommando `ls -l` geschieht.
- m* Hängt das ursprüngliche Dateisystem vor der Sicherungsoperation mit Schreibschutz und nach dem Sichern mit den ursprünglichen Berechtigungen ein. Darf nicht mit den Dateisystemen `root` oder `/usr` verwendet werden.
- o Ermöglicht es dem Benutzer, Anforderungen zum Einlegen von Datenträgern außer Kraft zu setzen (siehe `getvol(1M)`, Option -o).
- r* Nimmt fern eingehängte Ressourcen in das Archiv auf.
- t* Erstellt ein Inhaltsverzeichnis für die Sicherung auf zusätzlichen Datenträgern anstatt im Sicherungsprotokoll.
- v* Überprüft das Archiv, während es erstellt wird. Beim Schreiben des Archivs wird eine Prüfsumme berechnet; sobald ein Datenträger vollständig beschrieben ist, wird er nochmals gelesen, und die Prüfsumme wird nochmals berechnet, um sicherzugehen, daß jeder Block lesbar und korrekt ist. Schlägt eine der beiden Prüfungen fehl, wird der Datenträger als nicht lesbar betrachtet. Wurde -A angegeben, schlägt die Archivierung fehl; anderenfalls wird der Bediener aufgefordert, den defekten Datenträger auszuwechseln.
- x* Ignoriert die Ausnahmeliste; sichert alle geänderten oder bearbeiteten Dateien.
- A Aktiviert einen Automatikmodus, d.h. der Benutzer wird nicht zum Einlegen bzw. Entfernen von Datenträgern aufgefordert.
- E* Gibt eine Schätzung des Datenträgerbedarfs für das Archiv aus und führt dann die Sicherungsoperation aus.
- N* Gibt eine Schätzung des Datenträgerbedarfs für das Archiv aus, führt die Sicherungsoperation jedoch nicht durch.

- S Zeigt alle 100 Blöcke (zu 512 Byte), die aus dem Archiv auf dem Zielgerät gelesen bzw. in das Archiv geschrieben werden, einen Punkt (.) an.
- V Zeigt den Namen jeder Datei an, in die geschrieben bzw. die aus dem Archiv auf dem Zielgerät extrahiert wurde.

Dialog mit dem Benutzer

Die Verbindung zwischen einer Archivierungsmethode und dem Kommando `backup` läßt sich nicht einfach als `fork/exec` oder `pipe` ausdrücken. Das Kommando `backup` ist sowohl für den direkten wie auch für den über das Kommando `bkoper` ablaufenden Dialog mit dem Benutzer zuständig. Daher liest das Kommando `incfile` weder aus der Standardeingabe, noch schreibt es in die Standardausgabe oder die Standardfehlerausgabe. Um Berichte (Schätzungen, Dateinamen, Punkte, Status usw.) an das Kommando `backup` zu übermitteln, muß eine Verfabrensbibliothek verwendet werden (siehe `libbrmeth(3)`).

EXIT-CODES

Die Exit-Codee von `incfile` haben folgende Bedeutung:

- 0 = erfolgreicher Abschluß der Aufgabe
- 1 = ein oder mehrere Parameter zu `incfile` sind ungültig
- 2 = es ist ein Fehler aufgetreten, so daß `incfile` nicht alle Teile dieser Aufgabe abschließen konnte

DATEIEN

<code>/etc/bkup/bkexcept.tab</code>	listet die Dateien auf, die von einer Teilsicherung des Dateisystems ausgeschlossen werden sollen
<code>/etc/bkup/bkhist.tab</code>	Listet die Etiketten aller Datenträger auf, die für Sicherungsoperationen verwendet wurden
<code>/etc/bkup/rsstatus.tab</code>	verfolgt den Status aller von Benutzern eingegebenen Aufforderungen zum Wiederherstellen
<code>/etc/bkup/bklog</code>	listet die Fehler auf, die von den Sicherungsmethoden und dem Kommando <code>backup</code> verursacht wurden
<code>/etc/bkup/rslog</code>	protokolliert die Fehler, die von den Methoden zum Wiederherstellen und dem Kommando <code>restore</code> verursacht wurden
<code>\$TMP/filelist\$\$</code>	speichert temporär ein Inhaltsverzeichnis für ein Sicherungsarchiv

SIEHE AUCH

`backup(1M)`, `bkoper(1M)` `cpio(1)`, `cpio(4)`, `device.tab(4)`, `fdp(1)`, `ff(1M)`, `ffile(1)`, `fimage(1)`, `getvol(1M)`, `incfile(1)`, `labelit(1M)`, `libbrmeth(3)`, `ls(1)`, `restore(1M)`, `rsoper(1M)`, `time(2)`

BEZEICHNUNG

inetd – Dämon für Internet-Dienste

ÜBERSICHT

inetd [-d] [-s] [*konfigurationsdatei*]

BESCHREIBUNG

inetd, der Dämon für Internet-Dienste, wird normalerweise beim Laden von der Service Access Facility (SAF) aktiviert. Nach dem Start liest inetd seine Konfigurationsdaten aus der *konfigurationsdatei*, standardmäßig ist dies */etc/inetd.conf*. Nähere Angaben zum Format dieser Datei finden Sie unter *inetd.conf(4)*. Der Dämon horcht auf Verbindungen auf den Internet-Adressen der Dienste, die in der Konfigurationsdatei angegeben sind. Wird eine Verbindung gefunden, wird der Server-Dämon aufgerufen, der in der Konfigurationsdatei für den angeforderten Dienst angegeben ist. Sobald ein Server-Prozeß besteht, horcht inetd auf der Socket-Schnittstelle.

Mit der Option *-s* kann inetd "eigenständig", d.h. außerhalb der Service Access Facility (SAF), arbeiten.

Anstelle mehrerer Dämonprozesse, die konkurrierend mit nur wenigen Anforderungen arbeiten, verringert inetd die Belastung des Systems, indem er Internet-Server nur nach Bedarf aktiviert.

inetd selbst stellt eine Reihe einfacher Dienste auf TCP-Basis zur Verfügung. Dazu gehören: *echo*, *discard*, *chargen* (Zeichengenerator), *daytime* (benutzerlesbare Zeit) und *time* (maschinenlesbare Zeit in der Form Anzahl der Sekunden seit Mitternacht des 1. Januars 1900). Ausführliche Informationen zu diesen Diensten finden Sie im entsprechenden RFC (Request For Comment)-Dokument des Network Information Centers.

inetd liest seine Konfigurationsdatei jedesmal neu, wenn er ein Hangup-Signal SIGHUP empfängt. Zwischen den einzelnen Lesevorgängen können neue Dienste aktiviert und bestehende gelöscht oder geändert werden.

SIEHE AUCH

comsat(1M), *ftpd(1M)*, *rexecd(1M)*, *rlogind(1M)*, *rshd(1M)*, *telnetd(1M)*, *tftpd(1M)*, *inetd.conf(4)*.

Postel, Jon, "Echo Protocol," RFC 862, Network Information Center, SRI International, Menlo Park, Calif., Mai 1983.

Postel, Jon, "Discard Protocol," RFC 863, Network Information Center, SRI International, Menlo Park, Calif., Mai 1983.

Postel, Jon, "Character Generator Protocol," RFC 864, Network Information Center, SRI International, Menlo Park, Calif., Mai 1983.

Postel, Jon, "Daytime Protocol," RFC 867, Network Information Center, SRI International, Menlo Park, Calif., Mai 1983.

Postel, Jon, and Ken Harrenstien, "Time Protocol," RFC 868, Network Information Center, SRI International, Menlo Park, Calif., Mai 1983.

BEZEICHNUNG

infocmp – vergleicht *terminfo*-Beschreibungen oder gibt sie aus

ÜBERSICHT

```
infocmp [-d] [-c] [-n] [-I] [-L] [-C] [-r] [-u] [-s d|i|l|c] [-v] [-V]
        [-l] [-w länge] [-A verzeichnis] [-B verzeichnis] [termname ...]
```

BESCHREIBUNG

Mit infocmp kann ein binärer terminfo-Eintrag mit anderen terminfo-Einträgen verglichen werden, eine terminfo-Beschreibung kann geändert werden, um das Feld use= terminfo zu nutzen, oder es kann eine terminfo-Beschreibung aus der binären Datei (*term*) in verschiedenen Formaten ausgegeben werden. In allen Fällen werden zuerst die booleschen Felder, danach die numerischen Felder und anschließend die Felder mit Zeichenketten ausgegeben.

Standardoptionen

Sind keine Optionen und kein oder nur ein *termname* angegeben, wird die Option -I vorausgesetzt. Ist mehr als ein *termname* angegeben, wird die Option -d vorausgesetzt.

Vergleichsoptionen [-d] [-c] [-n]

infocmp vergleicht die terminfo-Beschreibung des ersten Terminals *termname* mit allen Beschreibungen, die in den Einträgen für die *termnamen* anderer Terminals enthalten sind. Ist ein Merkmal nur für eines der Terminals definiert, hängt der zurückgegebene Wert vom Typ des Merkmals ab: F für boolesche Variablen, -1 für ganzzahlige Variablen und NULL für Zeichenkettenvariablen.

- d erstellt eine Liste aller Merkmale, in denen sich zwei Einträge unterscheiden. Diese Option ist nützlich, um Unterschiede zwischen zwei Einträgen aufzuzeigen, die von verschiedenen Personen für dasselbe oder ähnliche Terminals erstellt wurden.
- c erstellt eine Liste aller Merkmale, die bei zwei Einträgen gleich sind. Nicht angegebene Merkmale werden ignoriert. Mit dieser Option lässt sich prüfen, ob es sinnvoll ist, die Option -u zu benutzen.
- n erstellt eine Liste der Merkmale, die keiner der beiden Einträge enthält. Werden keine *termnamen* angegeben, wird die Umgebungsvariable TERM für beide *termnamen* verwendet. Damit lässt sich schnell überprüfen, ob bei einer Beschreibung etwas ausgelassen wurde.

Optionen für Quellenlisten [-I] [-L] [-C] [-r]

Mit den Optionen -I, -L und -C wird für jedes genannte Terminal eine Quellliste erstellt.

- I verwendet die terminfo-Namen
- L verwendet den langen C-Variablennamen aus <term.h>
- C verwendet die termcap-Namen
- r wird -C verwendet, werden alle Merkmale im termcap-Format ausgegeben.

Werden keine *termnamen* angegeben, wird die Umgebungsvariable TERM als Terminalname verwendet.

Die mit der Option `-C` erstellte Quelle kann direkt als `termcap`-Eintrag verwendet werden, aber unter Umständen werden nicht alle der als Parameter angegebenen Zeichenketten ins `termcap`-Format umgewandelt. `infocmp` versucht, die meisten der als Parameter verwendeten Daten umzuwandeln, die nicht umgewandelten Teile werden in der Ausgabe markiert und auf Kommentar gesetzt. Diese Elemente müssen von Hand editiert werden.

Alle Auffülldaten für Zeichenketten werden zusammengefaßt und an den Anfang der Zeichenkette gestellt, wo `termcap` sie erwartet. Das obligatorische Auffüllen (Auffülldaten mit angehängtem `'/'`) wird aufgehoben, es kann nun wahlweise erfolgen.

Alle `termcap`-Variablen, die nicht mehr von `terminfo` unterstützt werden, sich aber aus anderen `terminfo`-Variablen ableiten lassen, werden ausgegeben. Nicht alle `terminfo`-Merkmale werden umgewandelt; ausgegeben werden normalerweise nur die Variablen, die Teil von `termcap` waren. Die Option `-r` hebt diese Einschränkung auf und ermöglicht, daß alle Merkmale ins `termcap`-Format ausgegeben werden.

Da die Auffülldaten am Anfang des Merkmals gesammelt werden, werden nicht alle Merkmale ausgegeben. Obligatorisches Auffüllen wird nicht unterstützt. Da `termcap`-Zeichenketten weniger flexibel sind, läßt sich eine `terminfo`-Zeichensteuerung nicht immer in ein entsprechendes `termcap`-Format umwandeln. Wird die `termcap`-Datei später in das `terminfo`-Format zurück konvertiert, ist das Ergebnis nicht unbedingt die ursprüngliche `terminfo`-Quelle.

Hier einige der gängigen `terminfo`-Parametersequenzen, die `termcap`-Entsprechungen und einige Terminaltypen, in denen diese Sequenzen gewöhnlich vorkommen:

<code>terminfo</code>	<code>termcap</code>	Repräsentative Terminals
<code>%p1%c</code>	<code>%.</code>	adm
<code>%p1%d</code>	<code>%d</code>	hp, ANSI standard, vt100
<code>%p1%'x'%'%+c</code>	<code>%+x</code>	concept
<code>%i</code>	<code>%i</code>	ANSI standard, vt100
<code>%p1%?'x'%'>%t%p1%'y'%'%;</code>	<code>%>xy</code>	concept
<code>%p2 wird vor %p1 ausgegeben</code>	<code>%r</code>	hp

Use= Option [-u]

`-u` erstellt eine `terminfo`-Quellenbeschreibung des ersten Terminals *term-name*, die sich auf die Summe der Beschreibungen bezieht, die sich aus den Einträgen für die anderen Terminals *terminnamen* ergibt. Dazu werden die Unterschiede zwischen dem ersten *terminnamen* und den anderen *terminnamen* analysiert und eine Beschreibung erstellt, die für die übrigen Terminals `use=-`Felder enthält. So ist es möglich, generische `terminfo`-Einträge in die Beschreibung eines Terminals nachträglich einzupassen. Sind zwei ähnliche Terminals vorhanden, die jedoch zu unterschiedlichen Zeiten oder von verschiedenen Personen codiert wurden, so daß für beide vollständige Beschreibungen vorliegen, läßt sich mit `infocmp` feststellen, wie eine Beschreibung so geändert werden kann, daß sie sich auf die andere bezieht.

Ein Merkmal wird mit dem Zeichen @ ausgegeben, wenn es im ersten *termnames* nicht mehr vorkommt, andere *termname*-Einträge aber einen Wert dafür enthalten. Der Wert für ein Merkmal wird ausgegeben, wenn der Wert im ersten *termnames* in keinem der übrigen *termname*-Einträge vorhanden ist oder wenn der erste der anderen *termname*-Einträge, der dieses Merkmal aufweist, einen anderen Wert dafür enthält.

Die Reihenfolge der anderen *termname*-Einträge ist von Bedeutung. Da der *terminfo*-Übersetzer *tie* die Merkmale von links nach rechts liest, kann es bei zwei *use=-*-Einträgen, die unterschiedliche Angaben für dieselben Merkmale enthalten, zu unterschiedlichen Ergebnissen kommen.

Dies hängt von der Reihenfolge ab, in der die Angaben eingegeben wurden. *infocmp* markiert solche Diskrepanzen zwischen den anderen *termname*-Einträgen.

Wird dagegen ein Merkmal nach einem *use=-*-Eintrag angegeben, der dasselbe Merkmal enthält, wird die zweite Angabe ignoriert. Wenn eine Beschreibung mit *infocmp* erneut erstellt wird, ist dies gleichzeitig eine gute Möglichkeit zu überprüfen, ob alle Angaben in der ursprünglichen Quellenbeschreibung richtig eingegeben wurden.

Ein weiterer Fehler, der zwar nicht zu fehlerhaft übersetzten Dateien führt, den Übersetzungsvorgang jedoch verlangsamt, besteht darin, daß zusätzliche, überflüssige *use=-*-Felder angelegt werden. *infocmp* markiert jedes zweite nicht benötigte *termname-use=-*-Feld.

Weitere Optionen [-s d | i | l | c] [-v] [-V] [-1] [-w *länge*]

-s sortiert die Felder innerhalb der einzelnen Typen nach den folgenden Argumenten:

- d Die Felder in der Reihenfolge belassen, wie sie in der *terminfo*-Datenbank gespeichert sind.
- i Sortieren nach dem *terminfo*-Namen.
- l Sortieren nach dem langen C-Variablennamen.
- c Sortieren nach dem *termcap*-Namen.

Wird die Option -s nicht angegeben, werden die ausgegebenen Felder innerhalb der einzelnen Typen alphabetisch nach dem *terminfo*-Namen sortiert. Ausgenommen hiervon sind die Optionen -C und -L, bei denen nach dem *termcap*-Namen bzw. dem langen C-Variablennamen sortiert wird.

-v Gibt Ablaufverfolgungsdaten in der Standardfehlerausgabe aus, während das Programm läuft.

-V Gibt die aktuelle Programmversion in der Standardfehlerausgabe aus und verläßt das Programm.

infocmp(1M)

infocmp(1M)

- l Bewirkt, daß jedes Feld in einer eigenen Zeile ausgegeben wird. Ansonsten werden in einer Zeile mehrere Felder ausgegeben. Eine Zeile kann bis zu 60 Zeichen lang sein.
- w Ändert die Ausgabe auf *länge* Zeichen.

Wechseln der Datenbanken [-A *verzeichnis*] [-B *verzeichnis*]

Die Speicherposition der übersetzten terminfo-Datenbank ergibt sich aus der Umgebungsvariablen `TERMINFO`. Ist die Variable nicht definiert oder wird das Terminal an dieser Stelle nicht gefunden, wird die terminfo-Systemdatenbank verwendet, die gewöhnlich in `/usr/share/lib/terminfo` steht. Mit den Optionen `-A` und `-B` kann diese Position außer Kraft gesetzt werden. Die Option `-A` setzt `TERMINFO` für den ersten *termnamen*, und die Option `-B` setzt `TERMINFO` für die anderen *termnamen*. Dadurch ist es möglich, die Beschreibungen für ein Terminal gleichen Namens, die in zwei verschiedenen Datenbanken stehen, zu vergleichen. Dies ist nützlich, um von verschiedenen Personen erstellte Beschreibungen für dasselbe Terminal zu vergleichen.

DATEIEN

`/usr/share/lib/terminfo/?/*` übersetzte Terminal-Beschreibungsdatenbank.

SIEHE AUCH

`curses(3X)`, `captainfo(1M)`, `terminfo(4)`, `tic(1M)`.

BEZEICHNUNG

`init`, `telinit` – Initialisierung der Prozeß-Steuerung

ÜBERSICHT

`/sbin/init [ 0123456SsQqabc ]`

`/sbin/telinit [ 0123456SsQqabc ]`

BESCHREIBUNG**init**

`init` ist ein Systemprozeß. Seine Hauptaufgabe ist die Erzeugung von Prozessen anhand von Informationen, die in der Datei `/etc/inittab` stehen (siehe `inittab(4)`).

Zu jedem beliebigen Zeitpunkt befindet sich das System in einem von acht möglichen Betriebsstufen (Betriebsstatus, run level). Eine Betriebsstufe ist eine Software-Konfiguration des Systems, in der nur eine ausgewählte Gruppe von Prozessen ablaufen darf. Die von `init` für jede dieser Betriebsstufen erzeugten Prozesse sind in `/etc/inittab` definiert. Entsprechend kann sich `init` in einer von acht Betriebsstufen, 0 - 6 und S oder s, befinden (die Betriebsstufen S und s sind identisch). Die Betriebsstufe ändert sich, wenn ein privilegierter Benutzer `/sbin/init` aufruft. Dieses vom Benutzer erzeugte Programm `init` sendet Signale an das ursprüngliche, vom Betriebssystem beim Laden erzeugte Programm `init`, die ihm übermitteln, in welche Betriebsstufe es wechseln soll.

`init` kennt die folgenden Argumente:

- 0 schaltet das System ab, so daß die Stromversorgung unterbrochen werden kann. Wenn möglich, soll das System die Stromversorgung selbst unterbrechen.
- 1 setzt das System in den Systemverwaltermodus. Alle Dateisysteme sind eingehängt. Nur eine kleine Gruppe wichtiger Prozesse des Kerns ist noch aktiv. Dieser Modus wird für Verwaltungsaufgaben, z.B. zur Installation zusätzlicher Dienstprogrammpakete, verwendet. Auf alle Dateien kann zugegriffen werden, und am System sind keine Benutzer angemeldet.
- 2 setzt das System in den Mehrbenutzermodus. Alle Terminalprozesse und Dämonen für die Mehrbenutzerumgebung werden erzeugt. Dieser Status wird im allgemeinen als Mehrbenutzerstatus bezeichnet.
- 3 startet die NFS-Prozesse und -Dämonen. Hängt ferne Ressourcen ein und macht sie bekannt. Betriebsstufe 3 erweitert den Mehrbenutzermodus und wird als NFS-Status bezeichnet.
- 4 kann als alternative Konfiguration für die Mehrbenutzerumgebung definiert werden. Diese Operation ist für den Systembetrieb nicht notwendig und wird daher gewöhnlich nicht benutzt.
- 5 Stoppt das Betriebssystem SINIX und wechselt zum Firmware-Monitor.

- 6 Stoppt das Betriebssystem SINIX und lädt das System neu mit dem Status, der im Eintrag `initdefault` in `/etc/inittab` festgelegt ist.
- a,b,c verarbeitet nur die `/etc/inittab`-Einträge, die für Betriebsstufen a, b oder c definiert sind. Diese Betriebsstufen sind zum Arbeiten mit bestimmten Kommandos definiert und ändern die aktuelle Betriebsstufe nicht, sie heißen deshalb auch Pseudo-Betriebsstufen.
- Q,q überprüft `/etc/inittab` nochmals.
- S,s aktiviert den Einbenutzermodus. Dadurch wird das Terminal, das dieses Kommando ausführte, zur Systemkonsole. Dies ist die einzige Betriebsstufe, für die keine ordnungsgemäß formatierte Datei `/etc/inittab` erforderlich ist. Diese Betriebsstufe ist auch die standardmäßig vorgegebene Betriebsstufe in `/etc/inittab`. Besteht diese Datei nicht, steht `init` standardmäßig als einzige zulässige Betriebsstufe der Einbenutzer-Modus zur Verfügung. Wird das System mit den Optionen `S` oder `s` gestartet, werden keine Dateisysteme für Benutzerdateien eingehängt, und nur wichtige Prozesse des Systemkerns werden ausgeführt. Wird das System auf die Optionen `S` oder `s` heruntergesetzt, bleiben alle eingehängten Dateisysteme eingehängt, und alle von `init` gestarteten Prozesse, die nur im Mehrbenutzer-Modus laufen dürfen, werden abgebrochen. Darüber hinaus werden alle Prozesse mit einem `utmp`-Eintrag abgebrochen. Die letzte Bedingung sorgt dafür, daß alle von SAC gestarteten Anschlußmonitore sowie alle von diesen gestarteten Dienste abgebrochen werden. Andere Prozesse, die nicht direkt von `init` gestartet wurden, bleiben aktiv, so zum Beispiel `cron`.

Beim Laden eines SINIX-Systems wird `init` aufgerufen, und die folgenden Schritte werden ausgeführt. Zuerst sucht `init` in `/etc/inittab` nach dem Eintrag `initdefault` (siehe `inittab(4)`). Ist ein solcher Eintrag vorhanden, benutzt `init` die darin angegebene Betriebsstufe als Startbetriebsstufe. Enthält `/etc/inittab` keinen Eintrag `initdefault`, fordert `init` den Benutzer auf, an der virtuellen Systemkonsole eine Betriebsstufe einzugeben. Wird `S` oder `s` eingegeben, geht `init` in den Einbenutzerbetrieb. In dieser Betriebsart wird die virtuelle Konsole dem Benutzerterminal zugewiesen und zum Lesen und Schreiben geöffnet; damit wird das Benutzerterminal zur Systemkonsole. Das Kommando `/sbin/su` wird aufgerufen, und auf der physischen Konsole (der ursprünglichen Konsole) erscheint eine Meldung darüber, wohin die virtuelle Konsole zugewiesen wurde. Mit `init` oder `telinit` kann `init` angewiesen werden, die Betriebsstufe des Systems zu ändern. Wird die Shell beendet (z.B. über ein Dateende), wird `init` nur dann neu im Einbenutzer-Modus initialisiert, wenn die Datei `/etc/inittab` nicht existiert.

Wird eine Zahl zwischen 0 und 6 eingegeben, aktiviert `init` die entsprechende Betriebsstufe. Die Betriebsstufen 0, 5 und 6 sind reserviert zum Herunterfahren des Systems. Die Betriebsstufen 2, 3 und 4 stehen als Mehrbenutzer-Betriebsarten zur Verfügung.

Ist `init` zum erstenmal seit dem Einschalten in eine andere Betriebsstufe als den Einbenutzerbetrieb gegangen, sucht das Programm zuerst in der Datei `/etc/inittab` nach den Einträgen `boot` und `bootwait` (siehe `inittab(4)`). Diese Einträge werden vor jeder anderen Verarbeitung von `/etc/inittab` ausgeführt,

vorausgesetzt, die eingegebene Betriebsstufe entspricht der in dem Eintrag. Auf diese Weise können spezielle Initialisierungsvorgänge des Betriebssystems, z.B. das Einhängen von Dateisystemen, ausgeführt werden, bevor die Benutzer auf das System zugreifen dürfen. Anschließend geht `init` die Datei `/etc/inittab` weiter durch und führt alle anderen Einträge aus, die für diese Betriebsstufe vorhanden sind.

Um alle Prozesse in `/etc/inittab` zu erzeugen, liest `init` alle Einträge und legt für jeden Eintrag, der neu erzeugt werden muß, einen Sohnprozeß an. Wenn alle in `/etc/inittab` angegebenen Prozesse erzeugt sind, wartet `init` darauf, daß einer seiner Sohnprozesse beendet wird, oder daß ein Stromausfallsignal oder ein Signal von einem anderen `init`- oder `telinit`-Prozeß die Betriebsstufe des Systems ändert. Wenn eine dieser Bedingungen auftritt, überprüft `init` die Datei `/etc/inittab` nochmals. Neue Einträge können jederzeit in `/etc/inittab` eingefügt werden; `init` wartet jedoch immer auf eine der drei oben genannten Bedingungen, bis es die Datei `/etc/inittab` erneut überprüft. Um dies zu umgehen, aktiviert das Kommando `init Q` oder `init q` das Programm `init`, damit es die Datei `/etc/inittab` sofort überprüft.

Wenn `init` beim Systemstart aktiviert wird oder wenn das System vom Einbenutzerbetrieb in eine andere Betriebsart wechselt, setzt `init` die `ioctl(2)`-Zustände der virtuellen Konsole auf die Modi, die in der Datei `/etc/ioctl.syscon` gespeichert sind. In diese Datei schreibt `init` immer dann, wenn der Einbenutzerbetrieb aktiviert wird.

Wird eine Aufforderung zum Wechseln der Betriebsstufe eingegeben, sendet `init` das Warnsignal `SIGTERM` an alle Prozesse, die in der angestrebten Betriebsstufe nicht definiert sind. `init` wartet fünf Sekunden, bevor es diese Prozesse mit dem Abbruchsignal `SIGKILL` zwangsweise beendet.

Wird `init` über ein Signal gemeldet, daß ein von ihm erzeugter Prozeß beendet ist, schreibt `init` dies und den Grund für die Beendigung in die Datei(en) `/var/adm/utmp` und `/var/adm/wtmp`, sofern vorhanden (siehe `who(1)`). Ein Protokoll der erzeugten Prozesse wird in `/var/adm/wtmp` geführt.

Empfängt `init` ein `powerfail`-Signal (`SIGPWR`), durchsucht es die Datei `/etc/inittab` auf besondere Einträge des Typs `powerfail` und `powerwait`. Diese Einträge werden aufgerufen (sofern die Betriebsstufen dies erlauben), bevor eine weitere Verarbeitung stattfindet. So kann `init` verschiedene Bereinigungs- und Aufzeichnungsfunktionen ausführen, kurz bevor das Betriebssystem ausgeschaltet wird. Auf der MX500 wird `powerfail` nicht über `init`, sondern über das Kommando `bp(1M)` abgehandelt.

telinit

/sbin/telinit verhält sich genauso wie /sbin/init.

DATEIEN

/etc/inittab
/var/adm/utmp
/var/adm/wtmp
/etc/ioctl.syscon
/dev/console

SIEHE AUCH

ttymon(1M), shutdown(1M), inittab(4), utmp(4), utmpx(4), termio(7).
login(1), sh(1), stty(1), who(1) in den *Kommandos*.
kill(2) im *Referenzhandbuch für Programmierer*.

FEHLERMELDUNGEN

Stellt `init` fest, daß ein Eintrag aus `/etc/inittab` innerhalb von zwei Minuten mehr als zehnmal erzeugt wird, nimmt es an, daß die Kommandozeichenkette in dem Eintrag einen Fehler enthält, und gibt eine Fehlermeldung auf der Systemkonsole aus. Zum erneuten Erzeugen dieses Prozesses muß eine Wartezeit von fünf Minuten verstreichen, oder es muß ein Signal von einem vom Benutzer erzeugten `init`- oder `telinit`-Prozeß eingehen. Dadurch verbraucht `init` keine Systemressourcen, wenn die Datei `inittab` einen Schreibfehler enthält oder ein Programm, auf das in `/etc/inittab` verwiesen wird, gelöscht wird.

Beim Starten des Systems kann es vorkommen, daß `init` nicht zur Eingabe einer neuen Betriebsstufe auffordern kann, weil die virtuelle Konsole mit einem anderen Gerät als der physischen Systemkonsole verbunden ist.

HINWEIS

`init` und `telinit` dürfen nur von einem privilegierten Benutzer verwendet werden.

Der Status `S` oder `s` sollte in der Datei `/etc/inittab` nicht leichtfertig verwendet werden. Es empfiehlt sich, ihn nur in der Zeile `initdefault` einzutragen.

Ist bei `initdefault` in der Datei `/etc/inittab` kein Standardstatus angegeben, wird in Status `6` übergegangen. Als Folge davon gelangt das System in eine Endlos-Schleife und versucht fortlaufend zu Booten.

Läßt sich die Datei `utmp` beim Laden des Systems nicht erstellen, geht das System beim Booten in die Betriebsstufe "`s`", unabhängig davon, was als `initdefault` in `/etc/inittab` angegeben ist. Dies kann vorkommen, wenn das Dateisystem `/var` nicht verfügbar ist.

Prozesse, die z.B. mit `at` gestartet wurden, werden beim Übergang vom Normalbenutzermodus in den Einbenutzermodus nicht abgebrochen. Um sicher zu sein, daß im Einbenutzermodus solche Prozesse nicht mehr laufen, sollte man so vorgehen: Zuerst `init 0` aufrufen, danach das System neu hochfahren und mit `init S` in den Einbenutzermodus wechseln.

BEZEICHNUNG

`install` – installiert Kommandos

ÜBERSICHT

```
/usr/sbin/install [-c vera] [-f verb] [-i] [-n verc] [-m zugr] [-u benutzer]
                  [-g gruppe] [-o] [-s] datei [verx ...]
```

BESCHREIBUNG

Das Kommando `install` wird am häufigsten beim Anlegen von Dateien ("makefiles") verwendet (siehe `make(1)`), um eine *datei* (aktualisierte Zieldatei) an einer bestimmten Position innerhalb eines Dateisystems zu installieren. Die *datei* wird dazu in das entsprechende Verzeichnis kopiert. Zugriffsrecht und Eigentümer des ursprünglichen Kommandos bleiben dadurch erhalten. Das Programm gibt Meldungen an den Benutzer aus, die genau auflisten, welche Dateien ersetzt oder angelegt werden und wohin sie geschrieben werden.

Sind keine Optionen oder Verzeichnisse (*verx* ...) angegeben, durchsucht `install` eine Reihe von Standardverzeichnissen (`/bin`, `/usr/bin`, `/etc`, `/lib` und `/usr/lib`, in dieser Reihenfolge) nach einer Datei mit demselben Namen wie *datei*. Wird die erste Übereinstimmung gefunden, teilt `install` in einer Meldung mit, daß diese Datei durch *datei* überschrieben wird, und führt diesen Schritt aus. Wird die Datei nicht gefunden, meldet das Programm dies und bricht ohne weitere Maßnahmen ab.

Sind eines oder mehrere Verzeichnisse (*verx* ...) nach *datei* angegeben, wird dieses/werden diese vor den in der Standardliste angegebenen Verzeichnissen durchsucht.

Die Optionen haben folgende Bedeutung:

- `-c vera` Installiert ein neues Kommando (*datei*) in dem mit *vera* bezeichneten Verzeichnis; dies geschieht allerdings nur dann, wenn es darin nicht gefunden wird. Wird diese Datei gefunden, gibt `install` die Meldung aus, daß die Datei bereits vorhanden ist, und bricht ab, ohne sie zu überschreiben. Diese Option kann allein oder zusammen mit der Option `-s` verwendet werden.
- `-f verb` Erzwingt die Installation der *datei* in dem angegebenen Verzeichnis, unabhängig davon, ob sie bereits existiert. Ist die zu installierende Datei noch nicht vorhanden, werden die Werte für Zugriffsrecht und Eigentümer für die neue Datei auf 755 bzw. `bin` gesetzt. Besteht die Datei bereits, gelten deren Werte für Zugriffsrecht und Eigentümer weiterhin. Kann allein oder zusammen mit der Option `-o` oder `-s` verwendet werden.
- `-i` Ignoriert die Standardverzeichnisliste und durchsucht nur die angegebenen Verzeichnisse (*verx* ...). Kann allein oder zusammen mit allen anderen Optionen außer `-c` und `-f` verwendet werden.
- `-n verc` Wird *datei* in keinem der durchsuchten Verzeichnisse gefunden, wird sie in das in *verc* angegebene Verzeichnis geschrieben. Die Werte für Zugriffsrecht und Eigentümer der neuen Datei werden auf 755 bzw. `bin` gesetzt. Kann allein oder zusammen mit allen anderen Optionen außer `-c` und `-f` verwendet werden.

install(1M)

install(1M)

- m *zugr* Der Wert für das Zugriffsrecht der neuen Datei wird auf *zugr* gesetzt.
- u *benutzer* Der Wert für den Eigentümer der neuen Datei wird auf *benutzer* gesetzt.
- g *gruppe* Der Wert für die Gruppennummer der neuen Datei wird auf *gruppe* gesetzt. Darf nur vom Superuser benutzt werden.
- o Wird *datei* gefunden, sichert diese Option die "gefundene" Datei, indem sie sie nach *OLDdatei* in dem Verzeichnis kopiert, in dem sie gefunden wurde. Diese Option ist beim Installieren häufig benutzter Dateien wie */bin/sh* oder */lib/saf/ttymon* von Nutzen, wenn die vorhandene Datei nicht entfernt werden kann. Kann allein oder zusammen mit allen anderen Optionen außer *-c* verwendet werden.
- s Unterdrückt die Ausgabe aller Meldungen außer den Fehlermeldungen. Kann allein oder zusammen mit anderen Optionen verwendet werden.

SIEHE AUCH

make(1).

killall (1M)

killall (1M)

BEZEICHNUNG

killall – bricht alle aktiven Prozesse ab

ÜBERSICHT

/usr/sbin/killall [*signal*]

BESCHREIBUNG

killall wird von /usr/sbin/shutdown dazu verwendet, alle aktiven Prozesse abzubrechen, die nicht direkt mit dem Systemabschluß zusammenhängen.

killall beendet alle Prozesse, die mit offenen Dateien arbeiten, so daß die eingehängten Dateisysteme nicht mehr benötigt werden und ausgehängt werden können.

killall sendet das *signal* (siehe kill(1)) an alle Prozesse, die nicht zu den oben genannten Ausnahmen gehören. Ist kein *signal* angegeben, dann wird der Standardwert SIGTERM (15) verwendet.

DATEIEN

/usr/sbin/shutdown

SIEHE AUCH

fuser(1M), shutdown(1M).

kill(1), ps(1) in den *Kommandos*.

signal(2) im *Referenzhandbuch für Programmierer*.

WARNUNG

Das Kommando killall kann nur von einem privilegierten Benutzer ausgeführt werden.

BEZEICHNUNG

labelit (generisch) – stellt Kennsätze für Dateisysteme zur Verfügung

ÜBERSICHT

labelit [-F *dateisystem-typ*] [-V] [*aktuelle_optionen*] [-o *spezifische_optionen*]
gerät [*operanden*]

BESCHREIBUNG

Mit labelit können Kennsätze für ausgehängte Dateisysteme auf Festplatten bzw. für Dateisysteme definiert werden, die auf Magnetband kopiert werden sollen.

Als Name für *gerät* sollte die Festplattenpartition (z.B. /dev/rdisk/0s3) oder das Magnetband (z.B. /dev/rmt/c0s0) angegeben werden. Das Gerät darf sich nicht auf einem fernen System befinden. *operanden* sind *dateisystem-typ*-spezifisch. Eine detaillierte Beschreibung entnehmen Sie dem Handbucheintrag zu dem *dateisystem-typ*-spezifischen labelit-Kommando.

aktuelle_optionen sind Optionen, die von dem s5-spezifischen Modul von labelit unterstützt werden. Andere *dateisystem-typen* unterstützen diese Optionen nicht unbedingt. Für *spezifische_optionen* wird eine Liste von durch Komma getrennten Unteroptionen und/oder Paaren von Schlüsselwort/Attribut angegeben, die von dem *dateisystemtyp*-spezifischen Modul des Kommandos interpretiert werden.

Die folgenden Optionen stehen zur Verfügung:

- F Gibt den *dateisystem-typ* an, auf den sich das Kommando beziehen soll. Der *dateisystem-typ* sollte hier angegeben werden. Ansonsten muß er aus /etc/vfstab hervorgehen. Hierzu muß *gerät* mit einem Eintrag in der Tabelle übereinstimmen.
- V Bewirkt, daß die ganze Kommandozeile auf dem Bildschirm ausgegeben wird. Diese Option wird verwendet, um die Kommandozeile auszuwerten und zu überprüfen. Zusätzliche Informationen, die eine Suche in /etc/vfstab ergibt, werden ebenfalls ausgegeben. Das Kommando wird nicht ausgeführt.
- o Gibt *dateisystem-typ*-spezifische Optionen an.

HINWEIS

Dieses Kommando wird möglicherweise nicht für alle Dateisystem-Typen unterstützt.

DATEIEN

/etc/vfstab Liste der Standardparameter für jedes Dateisystem

SIEHE AUCH

makefsys(1M), vfstab(4).

Handbucheinträge für die Dateisystemtyp-spezifischen Module von labelit.

BEZEICHNUNG

labelit (s5) – stellt Kennsätze für s5-Dateisysteme zur Verfügung

ÜBERSICHT

labelit [-F s5] [*generische_optionen*] [-n] *gerät* [*dsname datenträger*]

BESCHREIBUNG

generische_optionen sind Optionen, die von dem generischen Kommando labelit unterstützt werden.

Mit labelit können Kennsätze für ausgehängte s5-Dateisysteme auf Festplatten oder für s5-Dateisysteme definiert werden, die auf Magnetband kopiert werden sollen.

Werden die optionalen Argumente ausgelassen, gibt labelit die aktuellen Kennsatzwerte aus.

Als Name für *gerät* sollte die Festplattenpartition (z.B. /dev/rdisk/0s3) oder das Magnetband (z.B. /dev/rmt/c0s0) angegeben werden. Das Gerät darf sich nicht auf einem fernen System befinden.

Das Argument *dsname* steht für den Namen eines eingehängten Dateisystems (z.B. root, u1 etc.).

Mit dem Argument *datenträger* kann ein interner Name mit einem Datenträgernamen gleichgesetzt werden, der extern für die Festplatte, die Diskette oder das Magnetband gilt.

Bei Dateisystemen auf Festplatte werden *dsname* und *datenträger* im Superblock gespeichert.

Die folgenden Optionen stehen zur Verfügung:

- F s5 Gibt s5 als Dateisystem-Typ an. Hiermit wird sichergestellt, daß ein Kennsatz für ein s5-Dateisystem definiert wird.
- n Bewirkt, daß ein Kennsatz für ein Magnetband erstellt wird, wodurch der ursprüngliche Inhalt des Bands gelöscht wird.

SIEHE AUCH

generisches Kommando labelit(1M), makefsys(1M), s5spezifisches mount(1M). fs(4) im Referenzhandbuch für Programmierer.

labelit (1M)

(UFS)

labelit (1M)

BEZEICHNUNG

`labelit (ufs)` – stellt Etiketten für ufs-Dateisysteme zur Verfügung

ÜBERSICHT

`labelit [ -F ufs ] [generische_optionen] gerät [ dname datenträger ]`

BESCHREIBUNG

generische_optionen sind Optionen, die von dem generischen Kommando `labelit` unterstützt werden.

Mit `labelit` können Etiketten bzw. Kennsätze für ausgehängte Dateisysteme auf Festplatten oder für Dateisysteme definiert werden, die auf Magnetband kopiert werden sollen.

Wird weder *dname* noch *datenträger* angegeben, gibt `labelit` die aktuellen Werte aus.

Als Name für *gerät* sollte die Festplattenpartition (z.B. `/dev/dsk/0s3`) oder das Magnetband (z.B. `/dev/rmt/c0s0`) angegeben werden. Das Gerät darf sich nicht auf einem fernen System befinden.

Das Argument *dname* steht für den Namen eines eingehängten Dateisystems (z.B. `root`, `u1` etc.).

Mit *datenträger* kann ein interner Name mit einem Datenträgernamen gleichgesetzt werden, der extern für die Festplatte, die Diskette oder das Magnetband gilt.

Die Option lautet:

`-F ufs` Gibt ufs als Dateisystem-Typ an.

SIEHE AUCH

generisches Kommando `labelit(1M)`, `makefsys(1M)`, `ufs(4)`.

ldsysdump (1M)

ldsysdump (1M)

BEZEICHNUNG

ldsysdump – lädt Systemabzug von Disketten und Bändern

ÜBERSICHT

/usr/sbin/ldsysdump *zieldatei*

BESCHREIBUNG

ldsysdump lädt Dateien, die ein Speicherabbild enthalten, von einem Diskettensatz oder von einem Band, auf denen ein Systemabzug vorhanden ist, und schreibt sie in eine einzige Datei auf die Festplatte. Diese Datei kann dann mit dem Kommando crash geprüft werden. *zieldatei* ist der Name der Festplattendatei, in die ldsysdump das Speicherabbild schreiben soll.

ldsysdump startet einen Dialog, bei dem der Benutzer dazu aufgefordert wird, die Disketten bzw. das Band einzulegen. Der Benutzer kann die Sitzung jederzeit verlassen.

SIEHE AUCH

crash(1M), sysdump(8).
ulimit(2) im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

link, unlink – richtet Verweise (links) auf Dateien und Verzeichnisse ein und entfernt sie

ÜBERSICHT

```
/usr/sbin/link datei1 datei2  
/usr/sbin/unlink datei
```

BESCHREIBUNG

Mit dem Kommando `link` wird ein Dateiname eingerichtet, der auf eine andere Datei verweist. Verweise auf Dateien und Verzeichnisse können mit dem Kommando `unlink` entfernt werden. Es wird jedoch empfohlen, die Kommandos `rm` und `rmdir` anstelle des Kommandos `unlink` zu verwenden.

Der einzige Unterschied zwischen `ln`, `link` und `unlink` besteht darin, daß letztere genau das bewirken, was angegeben wird, wobei keine Fehlerprüfung vorgenommen wird. Dies ist darauf zurückzuführen, daß diese Kommandos die Systemaufrufe `link` und `unlink` direkt aufrufen.

SIEHE AUCH

`rm(1)` in den *Kommandos*.
`link(2)`, `unlink(2)` im *Referenzhandbuch für Programmierer*.

HINWEIS

Diese Kommandos können nur von einem privilegierten Benutzer ausgeführt werden.

listdgrp (1M)

listdgrp (1M)

BEZEICHNUNG

`listdgrp` – listet Mitglieder einer Gerätegruppe auf

ÜBERSICHT

`listdgrp ggruppe ...`

BESCHREIBUNG

Mit `listdgrp` werden die Mitglieder der in der Liste *ggruppe* aufgeführten Gerätegruppen aufgelistet.

EXIT-CODES

Die Exit-Codes von `listdgrp` haben folgende Bedeutung:

- 0 = erfolgreiche Ausführung der Aufgabe
- 1 = Kommandosyntax nicht korrekt, ungültige Option oder interner Fehler
- 2 = Gerätegruppentabelle konnte nicht zum Lesen geöffnet werden
- 3 = Gerätegruppe *ggruppe* konnte in der Gerätegruppentabelle nicht gefunden werden

BEISPIEL

Um Geräte aufzulisten, die zu der Gruppe `partitions` gehören, ist folgendes einzugeben:

```
$ listdgrp partitions
root
swap
usr
```

DATEIEN

`/etc/dgroup.tab`

SIEHE AUCH

`putdgrp(1)`.

BEZEICHNUNG

listen – Horchdämon des Netzwerks

ÜBERSICHT

/usr/lib/saf/listen [-m *gerstamm*] *netzspez*

BESCHREIBUNG

Der Prozeß *listen* "horcht", ob im Netzwerk eine Service-Anforderung vorliegt. Er nimmt eintreffende Anforderungen an und ruft danach entsprechende Server auf. Der Horchprozeß kann in jedem verbindungsorientierten Netzwerk, genauer gesagt, in jeder verbindungsorientierten Transportvorrichtung, verwendet werden, die der Transport Interface (TLI)-Spezifikation entspricht.

Für jede Verbindung generiert der Horchprozeß intern einen Pfadnamen für das Gerät. Dieser Pfadname wird in dem *utmp*-Eintrag für einen Service verwendet, falls einer angelegt wird. Standardmäßig wird dieser Pfadname aus der Verkettung des Präfixes */dev/netzspez* mit der Dezimaldarstellung der Gerätenummer gebildet. Wenn die Option *-m gerstamm* angegeben wird, verwendet der Horchprozeß *gerstamm* als Präfix für den Pfadnamen. In jedem Fall wird die Gerätenummer in der Form von mindestens zwei Ziffern (z.B. 05 oder 27) dargestellt. Sie kann jedoch auch länger sein, wenn dies erforderlich ist, um Gerätenummern über 99 darzustellen.

AUFRUF DES SERVERS

Wenn eine Verbindungsanforderung eingeht, legt der Horchprozeß einen neuen Transportendpunkt an und stellt die Verbindung an diesem Endpunkt her. Bevor der Dateideskriptor für diese neue Verbindung an den Server übergeben wird, werden alle angegebenen STREAMS-Module gepusht, und das Konfigurations-Skript wird, wenn vorhanden, ausgeführt. Dieser Dateideskriptor kann mit TLI (siehe insbesondere *t_sync(3N)*) oder der Socket-Schnittstellenbibliothek verwendet werden.

Standardmäßig wird der Server für jede Verbindung neu aufgerufen. Wenn der Server aufgerufen wird, bezeichnet der Dateideskriptor 0 den Transportendpunkt und ist für den Lese- und Schreibzugriff geöffnet. Die Dateideskriptoren 1 und 2 sind Kopien des Dateideskriptors 0. Es sind keine weiteren Dateideskriptoren geöffnet. Der Dienst wird mit der Benutzer- und Gruppennummer des Benutzernamens aufgerufen, unter der der Service beim Horchprozeß registriert wurde. Das aktuelle Verzeichnis ist hierbei das HOME-Verzeichnis dieses Benutzers.

Alternativ dazu kann ein Service auch so registriert werden, daß der Horchprozeß, über FIFO oder einen benannten STREAM, Verbindungen an einen permanenten Serverprozeß übergibt, anstatt den Server für jede Verbindung erneut aufzurufen. In diesem Fall wird die Verbindung in Form eines Dateideskriptors übergeben, der den neuen Transportendpunkt bezeichnet. Bevor der Dateideskriptor an den Server übergeben wird, interpretiert der Horchprozeß jedes Konfigurations-Skript, das für diesen Dienst registriert ist, mit Hilfe von *doconfig(3N)*, obwohl *doconfig* mit den Marken NORUN und NOASSIGN aufgerufen wird. Der Server empfängt den Dateideskriptor für die Verbindung in einer *strrecvfd*-Struktur über ein *l_RECVFD* *ioctl(2)*.

listen (1M)

listen (1M)

Weitere Informationen über die Verwendung und Verwaltung des Horchdienstes finden Sie unter `nlsadmin(1M)`.

DATEIEN

`/etc/saf/pmtag/*`

SIEHE AUCH

`nlsadmin(1M)`, `pmadm(1M)`, `sac(1M)`, `sacadm(1M)`,
`doconfig(3N)`, `nlsgetcall`, `nlsprovider(3N)`,
`streamio(7)`.
Network Programmer's Guide

HINWEIS

Wenn eine Verbindung an einen permanenten Server übergeben wird, handelt es sich bei der Benutzer- und Gruppennummer in der `strrecvfd`-Struktur um die für den Horchprozeß (d.h. beide sind 0). Der Benutzername, unter dem der Service beim Horchprozeß registriert wurde, geht aus diesen Nummern nicht hervor.

Wenn mehrere Horchprozesse auf einer Transportvorrichtung aktiv sind, kann beim Binden der Adressen während der Initialisierung der Horchprozesse ein Wettlaufzustand eintreten, wenn die Services dynamisch zugewiesene Adressen haben. Dieser Zustand äußert sich darin, daß der Horchprozeß einen Service mit statischer Adresse nicht seiner ansonsten gültigen Adresse zuweisen kann. Dies ist darauf zurückzuführen, daß ein Service mit dynamischer Adresse von einem anderen Horchprozeß dieser Adresse zugewiesen wurde.

BEZEICHNUNG

logins – zeigt Informationen über Benutzer- und Systemanmeldungen an

ÜBERSICHT

logins [-dmopstuxa] [-g gruppen] [-l *namen*]

BESCHREIBUNG

Dieses Kommando zeigt Informationen über Benutzer- und Systemanmeldungen an. Die Ausgabe wird mit den Kommandooptionen gesteuert und kann folgendes beinhalten: Benutzer- oder Systemnamen, Benutzernummer, den Wert des `/etc/passwd`-Abrechnungseintrags (Benutzername oder andere Informationen), den Namen der Primärgruppe, die Nummer der Primärgruppe, die Namen mehrerer Gruppen, die Nummern mehrerer Gruppen, das Home-Verzeichnis, die Anmelde-Shell und vier Parameter zur zeitlichen Begrenzung der Gültigkeit von Paßwörtern. Standardmäßig werden die folgenden Informationen ausgegeben: die Anmeldenummer, die Benutzernummer, der Name der Primärgruppe, die Nummer der Primärgruppe und der Wert des Abrechnungseintrags aus `/etc/passwd`. Die Ausgabe wird nach Benutzernummern sortiert, wobei zunächst die Systemanmeldungen und danach die Benutzeranmeldungen angezeigt werden.

- d Wählt Anmeldungen mit doppelten Benutzernummern aus.
- m Zeigt Informationen über die Mitgliedschaft in mehreren Gruppen an.
- o Die Ausgabe wird in Form einer Zeile mit durch Komma getrennten Feldern angezeigt.
- p Wählt Anmeldungen ohne Paßwörter aus.
- s Wählt alle Systemanmeldungen aus.
- t Sortiert die Ausgabe nach Benutzernamen anstelle der Benutzernummern.
- u Wählt alle Benutzeranmeldungen aus.
- x Gibt weiterführende Informationen über jeden ausgewählten Benutzer aus. Hierzu gehören das Home-Verzeichnis, die Anmelde-Shell und Informationen über die zeitliche Gültigkeitsbeschränkung des Paßworts. Diese Informationen werden auf jeweils einer Zeile angezeigt. Die Informationen über die Paßwörter beinhalten den Paßwortstatus (PS für paßwortgeschützt, NP für nicht paßwortgeschützt und LK für gesperrt). Wenn die Anmeldung paßwortgeschützt ist, wird neben dem Status angegeben, wann das Paßwort zuletzt geändert wurde, wieviele Tage zwischen zwei Paßwortänderungen liegen müssen und wieviele Tage zwischen zwei Änderungen liegen dürfen. Die Informationen über die zeitliche Gültigkeitsbeschränkung des Paßworts geben an, wann der Benutzer beim Anmelden die Meldung erhält, daß das Paßwort seine Gültigkeit verliert, bevor das Paßwort ungültig wird.
- a Fügt zwei Felder mit Informationen über die zeitliche Gültigkeitsbeschränkung des Paßworts zur Ausgabe hinzu. In den Feldern steht, wieviele Tage ein Paßwort unbenutzt bleiben kann, bis es automatisch ungültig wird, und wann das Paßwort seine Gültigkeit verliert.

logins (1M)

logins (1M)

- g Wählt nach Benutzernamen sortiert alle Benutzer aus, die zu der *gruppe* gehören. Mehrere Gruppen können, durch Komma getrennt, in Form einer Liste angegeben werden.
- l Wählt die angeforderte Anmeldung aus. Mehrere Anmeldungen können, durch Komma getrennt, in Form einer Liste angegeben werden.

HINWEIS

Die Optionen können zusammen verwendet werden. Wenn dies der Fall ist, werden alle Anmeldungen angezeigt, die einem beliebigen Kriterium entsprechen. Wenn die Optionen -l und -g kombiniert werden, wird ein Benutzer nur einmal aufgeführt, selbst dann, wenn er zu mehr als einer der ausgewählten Gruppen gehört.

BEZEICHNUNG

lpadmin – konfiguriert den LP-Druckdienst

ÜBERSICHT

lpadmin -p *drucker optionen*

lpadmin -x *ziel*

lpadmin -d [*ziel*]

lpadmin -S *typenrad* -A *alarmsignaltyp* [-W *minuten*] [-Q *anforderungen*]

BESCHREIBUNG

Das Dienstprogramm `lpadmin` definiert Drucker und Geräte, und konfiguriert so den LP-Druckdienst. Es dient zum Hinzufügen und Ändern von Druckern, zum Entfernen von Druckern aus dem Dienst, zum Definieren und Ändern des Standardziels für das System, zum Definieren von Alarmsignalen für Druckerfehler und zum Einlegen von Typenrädern.

Hinzufügen oder Ändern eines Druckers

Mit der ersten Form des Kommandos `lpadmin` (`lpadmin -p drucker optionen`) wird ein neuer Drucker konfiguriert oder die Konfiguration eines vorhandenen Druckers geändert. Die folgenden *optionen* dürfen in beliebiger Reihenfolge angegeben werden.

-A *alarmsignaltyp* [-W *minuten*]

Mit der Option -A wird ein Alarmsignal definiert, das den Verwalter informiert, wenn am Drucker ein Fehler festgestellt wird, und danach in regelmäßigen Abständen wiederholt wird, bis der Verwalter den Druckerfehler behebt. Folgende *alarmsignaltypen* stehen zur Verfügung:

mail Sendet die Alarmmeldung über elektronische Post (siehe `mail(1)`) an den Verwalter.

write Gibt die Meldung auf dem Terminal aus, an dem der Verwalter angemeldet ist. Ist der Verwalter an mehreren Terminals angemeldet, wird eines davon willkürlich ausgewählt.

quiet Gibt keine Meldungen über den aktuellen Zustand aus. Der Verwalter kann mit dieser Option die Ausgabe weiterer Meldungen zu einem bereits bekannten Zustand zeitweilig unterbrechen. Sobald der Fehler behoben ist und die Druckoperation weiterläuft, werden wieder Meldungen ausgegeben, wenn im Drucker ein anderer Fehler auftritt.

none Keine Meldungen senden; alle vorhandenen Definitionen für Alarmsignale werden gelöscht. Bei einem Fehler am Drucker werden erst dann wieder Alarmsignale ausgegeben, wenn ein anderer Alarmsignaltyp (außer `quiet`) angegeben wird.

shell-kommando

Das *shell-kommando* wird jedesmal ausgeführt, wenn ein Alarmsignal gesendet werden muß. Das Shell-Kommando erwartet die Meldung in der Standardeingabe. Enthält das Kommando Leerzeichen, muß es in Anführungszeichen gesetzt werden. Die *mail*- und *write*-Werte für diese Option entsprechen den Werten `mail benutzername` bzw. `write benutzername`. *benutzername* ist der aktuelle Name für den Verwalter, d.h. der Benutzername desjenigen, der dieses Kommando

eingibt. Dies gilt allerdings nicht, wenn diese Person mit dem Kommando `su` zu einer anderen Benutzernummer gewechselt hat. In diesem Fall wird der *benutzername* für die neue Benutzernummer verwendet.

`list` Zeigt den Typ des Alarmsignals an, das bei einem Druckerfehler ausgegeben wird. Das Signal selbst wird nicht verändert.

Folgende Meldung erscheint:

```
The printer drucker has stopped printing for the reason given below.
Fix the problem and bring the printer back on line. Printing has
stopped, but will be restarted in a few minutes; issue an enable
command if you want to restart sooner. Unless someone issues a
change request
```

```
lp -i anforderungsnummer -P ...
```

```
to change the page list to print, the current request will be
reprinted from the beginning.
```

```
The reason(s) it stopped (multiple reasons indicate reprinted
attempts):
```

grund

(Der Drucker *drucker* kann aus dem unten angegebenen Grund nicht weiterarbeiten. Beheben Sie den Fehler und aktivieren Sie den Drucker neu. Der Druckvorgang ist unterbrochen, wird in einigen Minuten jedoch wiederaufgenommen. Wenn der Druckvorgang früher beginnen soll, geben Sie ein Aktivierungskommando ein. Sofern keine Änderungsanforderung

```
lp -i anforderungsnummer -P ...
```

```
zum Ändern der Liste der zu druckenden Seiten eingegeben wurde, wird der
aktuelle Druckauftrag nochmals von Anfang an gedruckt.
```

```
Grund/Gründe für den Abbruch (mehrere Angaben bei Wiederholungen):
```

grund)

Der Druckdienst LP kann Druckerfehler nur finden, wenn ein geeignetes, schnelles Filterprogramm und ein Standard-Schnittstellenprogramm oder ein geeignetes angepaßtes Schnittstellenprogramm verwendet wird. Auch hängt das Wiederherstellungsvermögen nach einem Fehler von der Leistung des Filterprogramms ab.

Ist für den *drucker* der Wert `all` definiert, gilt die mit diesem Kommando definierte Alarmfunktion für alle vorhandenen Drucker.

Wird die Ausgabe von Alarmsignalen für *drucker* nicht mit der Option `-W` geregelt, wird standardmäßig eine Meldung pro Fehler an den Verwalter des *druckers* ausgegeben. Dies entspricht der Angabe `-W once` bzw. `-W 0`. Ist der Wert für *minuten* größer als null, wird das Alarmsignal in Abständen von *minuten* ausgegeben.

-c *klasse*

Fügt einen *drucker* in die angegebene *klasse* ein. Ist die *klasse* noch nicht vorhanden, wird sie angelegt.

-D *kommentar*

Sichert diesen *kommentar* und zeigt ihn an, wenn der Benutzer eine vollständige Beschreibung des *druckers* anfordert (siehe `lpstat(1)`). Der LP-Druckdienst interpretiert diesen Kommentar nicht.

-e *drucker₁*

Kopiert das Schnittstellenprogramm eines vorhandenen *druckers₁* als Schnittstellenprogramm für *drucker*. (Die Optionen `-i` und `-m` dürfen mit dieser Option nicht verwendet werden.)

-F *fehlerbehebung*

Mit dieser Option wird angegeben, welche Methode zur Fehlerbehebung für alle Druckanforderungen verwendet wird, die aufgrund eines Druckerfehlers abgebrochen wurden. Folgende Werte stehen für *fehlerbehebung* zur Verfügung.

continue

Oben auf der Seite, auf der der Druckvorgang unterbrochen wurde, wird weitergedruckt. Hierzu muß ein Filter definiert werden, damit das System erst dann automatisch weiterarbeitet, wenn der Fehler behoben ist.

beginning

Die Anforderung wird von Anfang an neu gedruckt.

wait Der *drucker* wird deaktiviert, bis der Verwalter oder ein Benutzer ihn wieder aktiviert.

Während der Wartezeit kann der Verwalter bzw. der Benutzer, der die Druckanforderung abgesetzt hat, in einer Änderungsanforderung angeben, an welcher Stelle weitergedruckt werden soll. (Siehe die Option `-i` zum Kommando `lp`.) Wird vor dem Reaktivieren des Drucks keine solche Änderungsanforderung ausgegeben, wird oben auf der Seite weitergedruckt, auf der der Druckvorgang unterbrochen wurde, sofern der Filter dies zuläßt; ansonsten wird die Druckanforderung vom Anfang an ausgeführt.

-f *allow:formularliste***-f *deny:formularliste***

Erlaubt oder verbietet das Drucken der Formulare in der *formularliste* auf dem *drucker*. Standardmäßig dürfen auf einem neuen Drucker keine Formulare gedruckt werden.

Für jeden Drucker unterhält der LP-Druckdienst zwei Listen: eine "Liste der gültigen Formulare" (Zulassungsliste) für Formulare, die auf dem Drucker ausgegeben werden dürfen, und eine "Liste der ungültigen Formulare" (Sperrliste) für Formulare, die auf dem Drucker nicht verwendet werden dürfen. Mit der Option `-f allow` werden die angegebenen Formulare in die Zulassungsliste aufgenommen und aus der Sperrliste gelöscht. Mit der Option `-f deny` werden die angegebenen Formulare in die Sperrliste aufgenommen und aus der Zulassungsliste gelöscht.

Ist die Zulassungsliste nicht leer, dürfen nur die darin verzeichneten Formulare auf dem Drucker verwendet werden, unabhängig vom Inhalt der Sperrliste. Ist die Zulassungsliste leer, die Sperrliste dagegen nicht, dürfen die in der Sperrliste angegebenen Formulare auf dem Drucker nicht verwendet werden. Mit der Option `-f deny:all` können alle Formulare für einen Drucker gesperrt werden. Mit der Option `-f allow:all` können alle Formulare auf einem Drucker ausgegeben werden, sofern der Drucker über alle Leistungsmerkmale zum Drucken der verschiedenen Formulare verfügt.

Diese Daten dienen dem LP-Druckdienst als Richtlinien, die angeben, auf welchem Drucker welche Formulare definiert werden können. Es steht den Verwaltern jedoch frei, Formulare auf jedem beliebigen Drucker zu verwenden. Wird ein Formular für einen Drucker definiert, auf dem es laut Zulassungs- oder Sperrliste nicht verwendet werden darf, erhält der Verwalter zwar eine Warnmeldung, das Formular wird aber dennoch akzeptiert. Wenn ein Benutzer versucht, eine Druck- oder Änderungsanforderung für eine ungültige Formular/Drucker-Kombination abzusetzen, wird die Anforderung jedoch nur dann akzeptiert, wenn das gewünschte Formular gerade für den Drucker definiert ist. Wird das Formular für den Drucker deaktiviert, bevor der Druckauftrag ausgeführt werden kann, wird die Anforderung storniert und der Benutzer benachrichtigt.

Wenn der Verwalter versucht, ein Formular für einen Drucker zu definieren, der nicht über die entsprechenden Voraussetzungen verfügt, wird das Kommando abgewiesen.

Bitte beachten Sie, daß die Option `-f` auch zusammen mit der Option `-M` verwendet werden kann und dort beschrieben wird.

- `-h` Gibt an, daß das mit dem Drucker verbundene Gerät fest verdrahtet ist. Die Optionen `-h` und `-l` schließen sich gegenseitig aus. Ist keine von beiden angegeben, wird die Option `-h` angenommen.

`-I inhaltstypliste`

Läßt Druckanforderungen auf *drucker* zu, deren Inhaltstyp in einer *inhaltstypliste* angegeben ist. Sind in der Liste mehrere Typen angegeben, müssen die Namen durch Kommas oder Leerzeichen getrennt sein. Bei Leerzeichen muß die gesamte Liste in doppelte Anführungszeichen eingeschlossen werden.

Der Typ `simple` (einfach) wird als Standard-Inhaltstyp für Dateien im SINIX-System erkannt. Bei dem Dateityp `simple` handelt es sich um einen Datenstrom, der nur druckbare ASCII-Zeichen und die folgenden Steuerzeichen enthält.

Steuerzeichen	Oktalwert	Bedeutung
Rückschritt	10 ₈	ein Zeichen zurück, jedoch nicht am Zeilenanfang
Tabulator	11 ₈	zum nächsten Tabulatorstop
Zeilenvorschub (Zeilenendezeichen)	12 ₈	zum Anfang der nächsten Zeile
Formularvorschub	14 ₈	zum Anfang der nächsten Seite
Zeilenschaltung	15 ₈	zum Anfang der aktuellen Zeile

Damit der Druckdienst `simple` nicht als gültigen Dateityp für den Drucker erkennt, muß entweder ein bestimmter Wert, z.B. der Druckertyp, in der *inhaltstyp*liste oder eine leere Liste angegeben werden. Soll `simple` neben anderen Typen gelten, muß dieser Wert in der *inhaltstyp*liste aufgeführt werden.

Abgesehen von `simple` kann jeder *inhaltstyp* vom Verwalter frei definiert werden. Wird der Druckertyp mit der Option `-T` definiert, gilt er implizit ebenfalls als gültiger Inhaltstyp.

`-i schnittstelle`

Definiert ein neues Schnittstellenprogramm für den *drucker*. *schnittstelle* ist der Pfadname des neuen Programms. Die Optionen `-e` und `-m` dürfen zu dieser Option nicht angegeben werden.

- `-l` Gibt an, daß es sich bei dem an den *drucker* angeschlossenen Gerät um ein Anmelde-Terminal handelt. Bei jedem Starten des LP-Schedulers (`lpsched`) werden die Anmelde-Terminals automatisch deaktiviert. Die Option `-h` darf nicht zusammen mit dieser Option angegeben werden.

`-M -f formularname [-a [-o filebreak]]`

Definiert das Formular *formularname* auf *drucker*. Druckenforderungen, die den Vordruck *formularname* benötigen, werden auf dem *drucker* ausgegeben. Ist das Formular für mehrere Drucker definiert, und hat der Benutzer zu der Option `-d` des Kommandos `lp any` (beliebig) als Druckerziel angegeben, wird der Druckauftrag auf dem Drucker ausgeführt, der auch den übrigen Anforderungen des Auftrags entspricht.

Die für das Formular benötigten Werte für Seitenlänge und -breite, Zeichendichte und Zeilenabstand werden mit den zulässigen Werten für den Drucker verglichen. Hierzu werden die Leistungsmerkmale des Druckertyps in der Datenbank `terminfo` überprüft. Bietet der Drucker nicht die für das Formular benötigten Merkmale, wird der Verwalter benachrichtigt, die Definition aber dennoch akzeptiert. Entspricht das Typenrad des Druckers nicht dem für das Formular vorgeschriebenen, wird ebenfalls eine Warnung an den Verwalter ausgegeben, die Definition aber trotzdem akzeptiert.

Wird die Option `-a` angegeben, wird ein Justierungsmuster ausgegeben, dem die gleiche Initialisierung des physischen Druckers vorausgeht wie einer normalen Druckenforderung, mit einer Ausnahme: Es wird keine Vorsatzseite gedruckt. Standardmäßig beginnt der Druck oben auf der ersten Seite des Formulars. Wenn das Muster ausgedruckt ist, kann der Verwalter das eingelegte Formular im Drucker ausrichten und mit der Eingabetaste ein weiteres Justierungsmuster drucken lassen. Für die Wiederholung ist keine

Initialisierung notwendig. Es können beliebig viele Justierungsmuster ausgegeben werden. Mit der Eingabe `q` kann der Verwalter den Druck von Justierungsmustern beenden.

Wird die Option `-o filebreak` angegeben, erfolgt zwischen den einzelnen Ausgaben des Justierungsmusters ein Formularvorschub. Normalerweise wird angenommen, daß ein Justierungsmuster ein Formular korrekt ausfüllt, so daß kein Formularvorschub erforderlich ist.

Um die Definition für ein Formular "auszuhängen", wird entweder eine neue Definition oder die Option `-f none` angegeben. Standardmäßig ist auf einem neuen Drucker kein Formular definiert.

Bitte beachten Sie, daß für die Option `-f` ohne die Option `-M` eine weitere, oben bereits beschriebene Verwendungsmöglichkeit besteht.

`-M -S typenrad`

Definiert das *typenrad* für *drucker*. Druckanforderungen, die dieses *typenrad* erfordern, werden auf diesem *drucker* verarbeitet. Ist das *typenrad* auf mehreren Druckern definiert und hat der Benutzer zu der Option `-d` des Kommandos `lp` das Argument *any* als Druckerziel angegeben, wird der Druckauftrag auf dem Drucker ausgeführt, der auch die übrigen Anforderungen des Auftrags erfüllt.

Ist das *typenrad* nicht als für den Drucker akzeptabel angegeben, wird eine Warnung an den Verwalter ausgegeben, die Definition aber trotzdem akzeptiert. Arbeitet der Drucker nicht mit Typenrädern, wird das Kommando abgewiesen.

Um die Definition für ein Typenrad "auszuhängen", wird sie entweder durch eine andere ersetzt, oder es wird die Option `-S none` angegeben. Standardmäßig ist für einen neuen Drucker kein Typenrad definiert.

Bitte beachten Sie die Verwendungsmöglichkeit der Option `-S` ohne die Option `-M`, die unten beschrieben wird.

`-m modell`

Wählt für den Drucker das zum LP-Druckdienst gehörende Schnittstellenprogramm *model* aus. Die Optionen `-e` und `-i` dürfen nicht zusammen mit dieser Option angegeben werden.

`-o druckoption`

Jede `-o`-Option in der nachstehenden Liste enthält den Standardwert, der für ein Schnittstellenprogramm gilt, sofern die Option nicht einer Definition für einen Vordruck entnommen oder von dem Benutzer angegeben wird, der die Anforderung absetzt (siehe `lp(1)`). Im folgenden werden die `-o`-Optionen angegeben, für die Standardwerte definiert werden können.

```
length=skalierte-dezimalzahl
width=skalierte-dezimalzahl
cpi=skalierte-dezimalzahl
lpi=skalierte-dezimalzahl
stty='stty-optionsliste'
```

Der Begriff "skalierte-dezimalzahl" bezieht sich auf eine nicht negative Zahl, die als Größenangabe verwendet wird. Den Typ der Größeneinheit zeigt ein

Buchstaben an, der an diese Zahl "angehängt" wird. Drei Typen von skalierten Dezimalzahlen können im LP-Druckdienst verwendet werden:

- Zahlen für Größenangaben in Zentimetern (gekennzeichnet durch ein angehängtes c);
- Zahlen für Größenangaben in Zoll (inches) (gekennzeichnet durch ein angehängtes i);
- Zahlen zur Angabe von anderen Einheiten (ohne angehängten Buchstaben), d.h. Zeilen, Zeichen, Zeilen pro Zoll oder Zeichen pro Zoll.

Die ersten vier Standard-Optionswerte müssen mit den Leistungsmerkmalen übereinstimmen, die in der Datenbank `terminfo` für den Typ des physischen Druckers angegeben sind. Anderenfalls wird das Kommando abgewiesen.

Die `stty-optionsliste` wird nicht auf zulässige Werte überprüft, sondern vom Standard-Schnittstellenprogramm direkt an das `stty`-Programm übergeben. Alle Fehlermeldungen, die `stty` ausgibt, während das Standard-Schnittstellenprogramm eine Anforderung verarbeitet, werden über elektronische Post an den Absender der Anforderung gesendet.

Für alle nicht eigens angegebenen Druckoptionen sind für die folgenden Attribute in der Datenbank `terminfo` Standardwerte für den angegebenen Druckertyp definiert.

```
length
width
cpi
lpi
```

Der Standardwert für `stty` ist

```
stty='9600 cs8 -cstopb -parenb ixon
-ixany opost -olcuc onlcr -ocrl -onocr
-onlret -ofill nl0 cr0 tab0 bs0 vt0 ff0'
```

Jede `-o`-Option kann mit dem für verschiedene Druckertypen unterschiedlichen Standardwert definiert werden, indem sie ohne Zuweisung eines Werts angegeben wird:

```
length=
width=
cpi=
lpi=
stty=
```

`-o nobanner`

Mit dieser Option kann ein Benutzer in einer Druckanforderung angeben, daß keine Vorsatzseite gedruckt wird.

`-o banner`

Erzwingt das Drucken einer Vorsatzseite bei jeder Druckanforderung, selbst wenn der Benutzer eine Ausgabe ohne diese Seite wünscht. Dies ist der Standardwert; wenn Benutzer die Möglichkeit haben sollen, zum Kommando `lp` die Option `-o nobanner` anzugeben, muß hier `-o nobanner` angegeben werden.

-r *klasse*

Löscht den *drucker* aus der angegebenen *klasse*. Ist *drucker* das letzte Element der *klasse*, wird die *klasse* ebenfalls gelöscht.

-S *liste*

Ermöglicht, daß die in der *liste* definierten Typenräder oder Aliasnamen für Zeichensätze auf dem Drucker verwendet werden.

Arbeitet der Drucker mit Typenrädern, sind die einzelnen Typenradnamen in der *liste* durch Kommas oder Leerzeichen getrennt. Werden Leerstellen als Trennzeichen verwendet, muß die Liste in Anführungszeichen eingeschlossen werden. Nur die in der Liste angegebenen Typenräder gelten als passend für den Drucker. Die Verwendung eines anderen Typenrads kann jedoch jederzeit erzwungen werden. Bevor mit dieser Option eine Liste der zulässigen Typenräder angelegt wird, gelten alle Typenräder als nicht verwendbar für den Drucker, und Druckanforderungen, die ein bestimmtes Typenrad für diesen Drucker angeben, werden zurückgewiesen.

Arbeitet der Drucker mit auswählbaren Zeichensätzen, enthält die *liste* durch Kommas oder Leerzeichen getrennte "Abbildeinträge" oder Aliasnamen für die gültigen Zeichensätze. Enthält die Liste Leerzeichen, muß sie in Anführungszeichen gesetzt werden. Alle "Abbildeinträge" haben die Form

bekannter_name=aliasname

Der *bekannte_name* entspricht der Nummer des Zeichensatzes, der die Buchstaben *cs* vorausgehen (z.B. *cs3* für Zeichensatz drei), oder dem Namen eines Zeichensatzes aus dem Terminfo-Datenbankeintrag *csnm*. (Siehe *terminfo(4)* im Referenzhandbuch für Programmierer.) Wird mit dieser Option keine Liste angegeben, gelten nur die aus der Terminfo-Datenbank bereits bekannten Namen bzw. die Nummern mit dem Präfix *cs* als akzeptabel für den Drucker.

Ist als *liste* das Wort *none* angegeben, werden alle vorhandenen Typenradlisten und Aliasnamen für Zeichensätze gelöscht.

Bitte beachten Sie die oben beschriebenen Verwendungsmöglichkeiten der Option **-S** mit der Option **-M**.

-s *systemname* [*!drucker-name*]

Stellt den Benutzern des lokalen Systems einen fernen Drucker, d.h. einen Drucker, auf den über ein anderes System zugegriffen wird, zur Verfügung. *systemname* ist der Name des fernen Systems, zu dem der ferne Drucker gehört; er muß in der Systemtabelle enthalten sein (*/etc/lp/Systems*). *drucker-name* ist der Name für den Drucker auf dem fernen System. Soll beispielsweise auf den *drucker₁* in *system₁* zugegriffen werden, und im lokalen System soll er *drucker₂* heißen, muß die Eingabe folgendermaßen lauten: **-p *drucker₂* -s *system₁FP!drucker₁***.

-T *druckertypenliste*

Kennzeichnet den Drucker als zu einem oder mehreren *druckertyp(en)* gehörig. Mit jedem *druckertyp* werden Daten aus der Datenbank *terminfo* abgerufen, mit denen der Drucker initialisiert wird, bevor die Anforderungen der einzelnen Benutzer abgearbeitet werden. Auch einige Filterprogramme können den *druckertyp* verwenden, um Dateien für den Drucker zu konvertieren. Wird diese

Option nicht benutzt, ist der standardmäßige *druckertyp* *unknown* (unbekannt); aus der Datenbank *terminfo* werden keine Daten abgerufen, so daß jede Druckanforderung ohne Initialisierung des Druckers ausgeführt wird. Diese Option muß auch angegeben werden, wenn die folgenden Optionen funktionieren sollen: *-o cpi*, *-o lpi*, *-o width* und *-o length* zu den Kommandos *lpadmin* und *lp* sowie *-S* und *-f* zum Kommando *lpadmin*.

Enthält die *druckertypenliste* mehrere Typen, muß die *inhaltstypenliste* der Option *-I* entweder als *simple*, leer (*-I ""*) oder gar nicht angegeben werden.

-u allow:anmeldenummernliste

-u deny:anmeldenummernliste

Gewährt oder verweigert den in der *anmeldenummernliste* genannten Benutzern den Zugriff auf den Drucker. Standardmäßig dürfen alle Benutzer einen neuen Drucker benutzen. Das Argument *anmeldenummernliste* kann eine, mehrere oder alle der folgenden Kombinationen enthalten:

anmeldenummer ein Benutzer in einem beliebigen System

systemname!anmeldenummer
ein Benutzer im System *systemname*

systemname!all alle Benutzer im System *systemname*

all!anmeldenummer ein Benutzer in allen Systemen

all alle Benutzer in allen Systemen

Für jeden Drucker führt der LP-Druckdienst zwei Benutzerlisten: eine "Zulassungsliste" mit den Benutzern, die den Drucker benutzen dürfen, und eine "Verweigerungsliste" mit den Benutzern, denen der Zugriff auf den Drucker verweigert wird. Mit der Option *-u allow* werden die angegebenen Benutzer in die Zulassungsliste aufgenommen und aus der Verweigerungsliste entfernt. Mit der Option *-u deny* werden die angegebenen Benutzer in die Verweigerungsliste aufgenommen und aus der Zulassungsliste gelöscht.

Ist die Zulassungsliste nicht leer, dürfen nur die dort genannten Benutzer auf den Drucker zugreifen, unabhängig vom Inhalt der Verweigerungsliste. Ist die Zulassungsliste leer, die Verweigerungsliste dagegen nicht, dürfen die in der Verweigerungsliste angegebenen Benutzer den Drucker nicht verwenden. Allen Benutzern kann der Zugriff auf den Drucker mit der Angabe *-u deny:all* verweigert werden. Alle Benutzer dürfen den Drucker verwenden, wenn *-u allow:all* angegeben wird.

-U wähl-info

Mit der Option *-U* kann der Druckdienst auf einen fernen Drucker zugreifen. Der lokale Druckdienst kann damit aber keinen fernen Druckdienst benutzen. Insbesondere werden dem Drucker mit *-U* die "Wähl"informationen *wähl-info* zugewiesen. *wähl-info* wird zusammen mit der Routine *dial* zum Aufrufen des Druckers verwendet. Gültig sind alle von Basic Networking Utilities unterstützten Netzwerkverbindungen. *wähl-info* kann eine Telefonnummer für eine Modemverbindung sein oder ein Systemname für andere Verbindungsarten. Wird *-U direct* angegeben, wird nicht gewählt, weil der Name *direct* für einen direkt angeschlossenen Drucker reserviert ist. Wird ein Systemname angegeben, wird damit in der Datei */etc/uucp/Systems* oder verwandten

Dateien nach Verbindungsdaten gesucht. Für diese Option müssen die Basic Networking Utilities installiert sein. Standardmäßig wird -U *direct* angenommen.

-v *gerät*

Verbindet ein *gerät* mit dem *drucker*. *gerät* ist der Pfadname einer Datei, in die lp schreiben kann. Bitte beachten Sie, daß ein *gerät* mehreren Druckern zugeordnet sein kann.

Einschränkungen

Beim Anlegen eines neuen Druckers muß eine von drei Optionen angegeben werden (-v, -U oder -s). Ferner darf nur eine der folgenden Optionen angegeben werden: -e, -i oder -m; ist keine dieser drei Optionen angegeben, wird der Standardwert für das Modell verwendet. Die Optionen -h und -l schließen sich gegenseitig aus. Drucker- und Klassennamen dürfen nicht länger als 14 Zeichen sein und nur aus den Zeichen A-Z, a-z, 0-9 und _ (Unterstrich) bestehen. Ist -s angegeben, sind die folgenden Optionen ungültig: -A, -e, -F, -h, -i, -l, -M, -m, -o, -U, -v und -W.

Löschen eines Druckerziels

Mit der Option -x *ziel* wird das Ziel *ziel* (ein Drucker oder eine Klasse) aus dem LP-Druckdienst entfernt. Handelt es sich bei *ziel* um einen Drucker und das einzige Element einer Klasse, wird die Klasse gleichfalls gelöscht. Ist der Wert für *ziel* *all*, werden alle Drucker und Klassen gelöscht. Andere *optionen* zu -x sind nicht zulässig.

Definieren/Ändern des Standardziels für das System

Mit der Option -d [*ziel*] wird *ziel*, ein vorhandener Drucker bzw. eine vorhandene Klasse, zum neuen Standardziel für das System. Wird *ziel* nicht angegeben, gibt es kein Standardziel für das System. Zu -d dürfen keine weiteren *optionen* angegeben werden.

Definieren eines Alarmsignals für das Typenrad

-S *typenrad* -A *alarmsignaltyp* [-W *minuten*] [-Q *anforderungen*]

Die Optionen -S *typenrad* und -A *alarmsignaltyp* dienen zum Definieren eines Alarmsignals, mit dem das Typenrad angefordert wird, wenn Aufträge dafür in der Warteschlange stehen. Wird dieses Kommando nicht verwendet, um ein Alarmsignal zum Wechseln des Typenrads zu definieren, wird kein Signal ausgegeben. Bitte beachten Sie die oben beschriebene andere Verwendungsmöglichkeit der Option -A mit der Option -p.

Folgende *alarmsignaltypen* stehen zur Verfügung:

- | | |
|-------|--|
| mail | Sendet die Alarmmeldung mit dem Kommando <i>mail</i> an den Verwalter. |
| write | Schreibt die Meldung mit dem Kommando <i>write</i> in das Terminal, an dem der Verwalter angemeldet ist. Ist der Verwalter an mehreren Terminals angemeldet, wird eines davon willkürlich ausgewählt. |
| quiet | Gibt keine Meldungen über den aktuellen Zustand aus. Der Verwalter kann mit dieser Option die Ausgabe weiterer Meldungen zu einem bereits bekannten Zustand zeitweilig unterbrechen. Nachdem das <i>typenrad</i> eingelegt und anschließend wieder entfernt wurde, werden erneut Meldungen gesendet, wenn die Anzahl der |

Druckanforderungen den mit der Option `-Q` definierten Schwellenwert erreicht.

`none` Erst dann Meldungen senden, wenn die Option `-A` mit einem anderen *alarmsignaltyp* (nicht `quiet`) erneut eingegeben wird.

shell-kommando

Das *shell-kommando* wird immer dann ausgeführt, wenn ein Alarmsignal gesendet werden muß. Das Shell-Kommando erwartet die Meldung in der Standardeingabe. Enthält das Kommando Leerzeichen, muß es in Anführungszeichen gesetzt werden. Die `mail-` und `write-`Werte für diese Option entsprechen den Werten `mail benutzername` und `write benutzername`, wobei *benutzername* der aktuelle Name für den Verwalter ist. Dies ist der Benutzername desjenigen, der dieses Kommando ausgibt, sofern nicht mit dem Kommando `su` zu einer anderen Benutzernummer gewechselt wurde. Wurde die Benutzernummer mit `su` geändert, wird der *benutzername* für die neue Nummer verwendet.

`list` Zeigt den Typ des Alarmsignals zum Wechseln des Typenrads auf der Standardausgabe an. Das Alarmsignal wird dadurch nicht geändert.

Folgende Meldung erscheint:

```
The print wheel typenrad needs to be mounted
on the printer(s):
drucker (ganzzahl1 requests)
ganzzahl2 print requests await this print wheel.
(Das Typenrad typenrad muß auf folgendem/n Drucker(n)
installiert werden:
drucker (ganzzahl1 Anforderungen)
ganzzahl2 Druckanforderungen warten auf dieses Typenrad.)
```

Die angegebenen Drucker entsprechen denen, die der Verwalter zuvor für die Verwendung mit diesem Typenrad definiert hat. Die neben jedem Drucker angegebene Zahl *ganzzahl*₁ gibt die Anzahl der für den jeweiligen Drucker anstehenden Anforderungen an. Die nach der Druckerliste angegebene Zahl *ganzzahl*₂ gibt die Gesamtzahl der Anforderungen an, die auf das Typenrad warten. Sie ist eventuell kleiner als die Summe der anderen Zahlen, wenn einige Anforderungen von mehreren Druckern ausgeführt werden können.

Hat *typenrad* den Wert `all`, wird das mit diesem Kommando definierte Alarmsignal für alle Typenräder ausgegeben, für die bereits ein Alarmsignal definiert wurde.

Wird die Option `-W` nicht angegeben, wird standardmäßig nur eine Meldung pro benötigtem Typenradwechsel ausgegeben. Wird die Option `-W` nicht angegeben, entspricht dies der Angabe `-W once` bzw. `-W 0`. Ist der Wert von *minuten* größer als null, wird das Alarmsignal alle *minuten* Minuten ausgegeben.

lpadmin (1M)

lpadmin (1M)

Wird auch die Option `-Q` angegeben, wird das Alarmsignal dann gesendet, wenn eine bestimmte, mit dem Argument *anforderungen* definierte, Anzahl von Druckanforderungen für das Typenrad in der Warteschlange steht. Wird die Option `-Q` nicht angegeben oder enthält *anforderungen* den Wert 1 oder das Wort *any* (beides Standardwerte), wird eine Meldung gesendet, sobald eine Anforderung für das nicht installierte Typenrad eingeht.

DATEIEN

`/var/spool/lp/*`
`/etc/lp`

SIEHE AUCH

`accept(1M)`, `lpsched(1M)` und `lpsystem(1M)`.
`enable(1)`, `lp(1)`, `lpstat(1)` und `stty(1)` in den *Kommandos*.
`dial(3C)`, `terminfo(4)` im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

`lpfilter` – verwaltet Filter, die zusammen mit dem LP-Druckdienst verwendet werden

ÜBERSICHT

```
lpfilter -f filtername -F pfadname
lpfilter -f filtername -
lpfilter -f filtername -i
lpfilter -f filtername -x
lpfilter -f filtername -l
```

BESCHREIBUNG

Mit dem Kommando `lpfilter` werden Filter hinzugefügt, geändert, gelöscht und aufgelistet, die zusammen mit dem LP-Druckdienst verwendet werden. Diese Filter dienen dazu, den Inhaltstyp einer Datei in einen anderen Inhaltstyp zu konvertieren, den der Drucker interpretieren kann. Eine der folgenden Optionen muß mit dem Kommando `lpfilter` verwendet werden: `-F pfadname` (oder `-` für Standardeingabe), wenn ein Filter hinzugefügt oder geändert werden soll, `-i`, wenn ein Filter auf seine ursprüngliche Einstellung zurückgesetzt werden soll, `-x`, wenn ein Filter gelöscht werden soll, oder `-l`, wenn eine Filterbeschreibung ausgegeben werden soll.

Das Argument `all` (alle) kann anstelle eines *filternamens* mit diesen Optionen verwendet werden. Wird `all` zusammen mit der Option `-F` oder `-` verwendet, wird die gewünschte Änderung an allen Filtern vorgenommen. Die Verwendung von `all` zusammen mit der Option `-i` bewirkt, daß die ursprünglichen Einstellungen aller Filter wiederhergestellt werden, für die ursprünglich vordefinierte Einstellungen galten. Zusammen mit der Option `-x` bewirkt das Argument `all`, daß alle Filter gelöscht werden, zusammen mit der Option `-l`, daß eine Liste aller Filter ausgegeben wird.

Filter hinzufügen oder ändern

Der in der Option `-f` angegebene Filter wird der Filtertabelle hinzugefügt. Ist der Filter bereits vorhanden, wird die Beschreibung mit den neu eingegebenen Informationen aktualisiert.

Die Filterbeschreibung wird aus dem *pfadnamen* entnommen, wenn die Option `-F` angegeben wird, oder aus der Standardeingabe, wenn die Option `-` verwendet wird. Eine der beiden Optionen muß angegeben werden, wenn ein Filter definiert oder geändert werden soll. Wenn es sich bei dem angegebenen Filter um einen Filter handelt, der ursprünglich zusammen mit dem LP-Druckdienst geliefert wurde, stellt die Option `-i` die ursprüngliche Filterbeschreibung wieder her.

Wenn ein bereits vorhandener Filter mit der Option `-F` oder `-` geändert wird, bleiben Angaben, die nicht in den neuen Informationen enthalten sind, unverändert. Wenn ein neuer Filter mit diesem Kommando hinzugefügt wird, werden nicht spezifizierten Angaben Standardwerte zugewiesen (siehe unten).

Filter dienen dazu, den Inhalt einer Anforderung in einen Datenstrom zu konvertieren, den der Drucker interpretieren kann. Für eine Druckanforderung müssen dem LP-Druckdienst die folgenden Informationen zur Verfügung stehen: der Inhaltstyp der Anforderung, der Name des Druckers, der Druckertyp, die Inhaltstypen, die vom Drucker unterstützt werden, und die Druckmodi, die von

demjenigen angefordert werden, der die Anforderung abgesetzt hat. Der Druckdienst verwendet diese Informationen dazu, einen Filter oder eine Filter-Pipe zu finden, die den Inhalt in einen vom Drucker unterstützten Typ konvertiert.

Im folgenden ist eine Liste von Angaben und deren Beschreibung abgebildet, die die Eingabe für dieses Kommando bilden. Handelt es sich bei der Eingabe um eine Liste, werden die einzelnen Elemente durch Komma oder Leerzeichen voneinander getrennt.

Input types: *inhaltstypenliste*
 Output types: *inhaltstypenliste*
 Printer types: *druckertypenliste*
 Printers: *druckerliste*
 Filter type: *filtertyp*
 Command: *shell-kommando*
 Options: *schablonenliste*

Input types Hiermit werden die Inhaltstypen angegeben, die von dem Filter interpretiert werden können. Der Standardwert ist *any* (beliebig).

Output types Hiermit werden die Inhaltstypen angegeben, in die der Filter die eingegebenen Inhaltstypen konvertieren kann. Der Standardwert ist *any*.

Printer types Hiermit werden die Druckertypen angegeben, für die der Filter verwendet werden kann. Der LP-Druckdienst schränkt den Gebrauch des Filters auf diese Druckertypen ein. Der Standardwert ist *any*.

Printers Hiermit werden die Namen der Drucker angegeben, für die der Filter verwendet werden kann. Der LP-Druckdienst schränkt den Gebrauch des Filters auf die angegebenen Drucker ein. Der Standardwert ist *any*.

Filter type Hiermit wird angegeben, ob der Filter *slow* (langsam) oder *fast* (schnell) ist. Bei langsamen Filtern dauert es normalerweise lange, bis die Eingabe konvertiert ist. Sie werden nicht mit dem Drucker verbunden, damit dieser nicht blockiert wird, während der Filter arbeitet. Wenn sich ein angegebener Drucker auf einem fernen System befindet, muß als Filtertyp *slow* angegeben werden. Bei schnellen Filtern handelt es sich normalerweise um Filter, bei denen die Eingabe schnell konvertiert wird oder die während der Ausführung mit dem Drucker verbunden werden müssen. Sie werden an das Schnittstellenprogramm übergeben, damit sie mit dem physischen Drucker verbunden ausgeführt werden.

Command Hiermit wird das Programm angegeben, das ausgeführt wird, um den Filter aufzurufen. Im *shell-kommando* müssen der vollständige Pfadname des Programms und die obligatorischen Optionen angegeben werden. Zusätzliche Optionen werden je nach den Eigenschaften der jeweiligen Druckanforderung und dem Feld *Options* angeführt. Für jeden Filter muß ein Kommando eingegeben werden.

Das Kommando muß einen Datenstrom als Standardeingabe akzeptieren und den konvertierten Datenstrom auf der Standardausgabe ausgeben. Dies ermöglicht es, Filter-Pipes aufzubauen, die Daten konvertieren, die nicht mit einem einzigen Filter bearbeitet werden können.

Options Hierbei handelt es sich um eine Liste von durch Komma getrennten Schablonen, die vom LP-Druckdienst dazu benutzt werden, anhand der Eigenschaften der jeweiligen Druckanforderung (siehe die Tabelle weiter unten) Optionen für den Filter zusammenzustellen.

Im allgemeinen weist jede Schablone die folgende Form auf:

schlüsselwort muster = ersatz

Das *schlüsselwort* gibt die Eigenschaften an, die die Schablone auf eine filterspezifische Option abzubilden versucht. Alle gültigen *schlüsselwörter* sind in der folgenden Tabelle aufgeführt. Bei einem *muster* kann es sich um folgendes handeln: ein genaues Muster einer der in der Tabelle aufgeführten Masken, ein einziger Asterisk (*) oder ein regulärer Ausdruck. Wenn das *muster* mit dem Wert der Eigenschaft übereinstimmt, handelt es sich um die passende Schablone, die dann dazu verwendet wird, eine filterspezifische Option zu generieren. Der *ersatz* gibt an, was als Option verwendet wird.

Reguläre Ausdrücke sind mit denen identisch, die in dem Kommando *ed(1)* oder *vi(1)* auftreten. Dazu gehören Konstruktionen der Form *\(...\)* und *\n*, mit denen Teile des *musters* extrahiert werden können, um sie in den *ersatz* zu kopieren. Das *&* kann dazu benutzt werden, das ganze *muster* in den *ersatz* zu kopieren.

Der *ersatz* kann auch das Zeichen * enthalten. Es wird ebenfalls durch das ganze *muster* ersetzt, wie das Zeichen & des Kommandos *ed(1)*.

lp Option	Eigenschaft	schlüsselwort	Mögliche muster
-T	Inhaltstyp (Eingabe)	INPUT	<i>inhaltstyp</i>
N/A	Inhaltstyp (Ausgabe)	OUTPUT	<i>inhaltstyp</i>
N/A	Druckertyp	TERM	<i>druckertyp</i>
-d	Druckername	PRINTER	<i>druckername</i>
-f, -o cpi=	Zeichendichte	CPI	<i>ganzzahl</i>
-f, -o lpi=	Zeilendichte	LPI	<i>ganzzahl</i>
-f, -o length=	Seitenlänge	LENGTH	<i>ganzzahl</i>
-f, -o width=	Seitenbreite	WIDTH	<i>ganzzahl</i>
-P	Zu druckende Seiten	PAGES	<i>seitenliste</i>
-S	Zeichensatz	CHARSET	<i>zeichensatzname</i>
	Typenrad	CHARSET	<i>typenradname</i>
-f	Maskenname	FORM	<i>maskenname</i>
-y	Modi	MODES	<i>modi</i>
-n	Anzahl der Kopien	COPIES	<i>ganzzahl</i>

Aus der Schablone

```
MODES landscape = -l
```

geht z.B. hervor, daß für den Filter die Option `-l` eingesetzt wird, wenn eine Druckaufforderung mit der Option `-y landscape` eingeht. Die Schablone

```
TERM * = -T *
```

gibt an, daß für jeden *druckertyp* in einer Druckaufforderung, bei der ein Filter verwendet wird, für diesen Filter die Option `-T druckertyp` eingesetzt wird.

Als letztes Beispiel soll die folgende Schablone erläutert werden:

```
MODES prwidth=\(.*\) = -w\1
```

Angenommen, ein Benutzer gibt das Kommando

```
lp -y prwidth=10
```

ein. Anhand der oben aufgeführten Tabelle stellt der LP-Druckdienst fest, daß die Option `-y` von einer `MODES`-Schablone bearbeitet wird. Die `MODES`-Schablone kann hier eingesetzt werden, das das *muster* `prwidth=\(.*\)` mit der Angabe `prwidth=10` übereinstimmt, die der Benutzer eingegeben hat. Der ersatz `-w\1` bewirkt, daß der LP-Druckdienst die Filteroption `-w10` generiert.

Falls nötig, baut der LP-Druckdienst eine Filter-Pipe auf. Dabei werden mehrere Filter verkettet, die dann die Datei des Benutzers und alle Druckoptionen bearbeiten. Die Beschreibung einer Pipe finden Sie unter `sh(1)`. Wenn der Druckdienst eine Filter-Pipe aufbaut, geben die `INPUT`- und `OUTPUT`-Werte, die für jeden Filter in der Pipe verwendet werden, die Ein- und Ausgabetypen für den jeweiligen Filter an, nicht für die ganze Pipe.

Einen Filter löschen

Die Option `-x` wird dazu verwendet, den mit *filtername* angegebenen Filter aus der LP-Filertabelle zu löschen.

Eine Filterbeschreibung löschen

Die Option `-l` wird dazu verwendet, die Beschreibung des mit *filtername* angegebenen Filters aufzulisten. Wird das Kommando erfolgreich ausgeführt, wird die folgende Meldung auf der Standardausgabe ausgegeben:

```
Input types: inhaltstypliste
Output types: inhaltstypliste
Printer types: druckertypenliste
Printers: druckerliste
Filter type: filtertyp
Command: shell-kommando
Options: schablonenliste
```

Schlägt die Ausführung des Kommandos fehl, wird eine Fehlermeldung an die Standardfehlerausgabe gesendet.

SIEHE AUCH

`lpadmin(1M)`.
`lp(1)` in den *Kommandos*.

BEZEICHNUNG

lpforms – verwaltet mit dem LP-Druckdienst verwendete Formulare

ÜBERSICHT

lpforms -f *formulardname* -F *pfadname* | - | -x | -l

lpforms -f *formulardname* -A *alarmsignaltyp* [-Q *minuten*] [-W *anforderungen*]

BESCHREIBUNG

Das Kommando lpforms dient zum Verwalten der verwendeten Vordrucke, z.B. Briefpapier mit Briefkopf, für den LP-Druckdienst. Ein Formular wird mit seinem *formulardnamen* angegeben. Benutzer können mit ihrer Druckanforderung ein Formular angeben (siehe lp(1)). Das Argument all kann anstelle von *formulardname* zu jeder der obengenannten Kommandozeilen angegeben werden. Mit der ersten Kommandozeile kann der Verwalter Formulardefinitionen hinzufügen, ändern und löschen, die Attribute eines vorhandenen Formulars auflisten und Benutzern den Zugriff auf bestimmte Formulare gewähren oder verweigern. Mit der zweiten Kommandozeile wird die Methode definiert, mit der der Verwalter benachrichtigt wird, wenn das Formular *formulardname* auf einem Drucker benötigt wird.

In der ersten Kommandozeile von lpforms muß eine der folgenden Optionen angegeben werden:

- F *pfadname* Zum Hinzufügen oder Ändern der Formulardefinition *formulardname* entsprechend den Angaben in *pfadname*.
- Zum Hinzufügen oder Ändern der Formulardefinition *formulardname* entsprechend den Angaben der Standardeingabe.
- x Zum Löschen der Formulardefinition *formulardname* (diese Option darf nur getrennt von allen anderen Optionen verwendet werden).
- l Zum Auflisten der Attribute des Formulars *formulardname*.

Hinzufügen oder Ändern einer Formulardefinition

Die Option -F *pfadname* dient zum Hinzufügen einer neuen Formulardefinition (im weiteren auch Formular genannt) *formulardname*, in den LP-Druckdienst oder zum Ändern der Attribute eines bestehenden Formulars. Die Beschreibung des Formulars wird dem *pfadnamen* entnommen, wenn die Option -F angegeben wurde, und der Standardeingabe, wenn die Option - verwendet wird. Eine dieser beiden Optionen muß zum Definieren oder Ändern eines Formulars immer angegeben werden. *pfadname* ist der Pfadname einer Datei, die alle oder einen Teil der folgenden Angaben zu dem Formular enthält.

Page length: *skalierte – dezimalzahl*₁
 Page width: *skalierte – dezimalzahl*₂
 Number of pages: *ganzzahl*
 Line pitch: *skalierte – dezimalzahl*₃
 Character pitch: *skalierte – dezimalzahl*₄
 Character set choice: *zeichensatz/typenrad* [mandatory]
 Ribbon color: *farbbandfarbe*
 Comment: *kommentar*
 Alignment pattern: [*inhaltstyp*] *inhalt*

Der Begriff "skalierte-dezimalzahl" bezieht sich auf eine nicht negative Zahl, mit der eine Größeneinheit angegeben wird. Um welche Größeneinheit es sich handelt, geht aus einem Buchstaben hervor, der an die Zahl "angehängt" ist. Drei Typen von skalierten Dezimalzahlen können im LP-Druckdienst verwendet werden:

- Zahlen für Größenangaben in Zentimetern (gekennzeichnet durch ein angehängtes c);
- Zahlen für Größenangaben in Zoll (inches) (gekennzeichnet durch ein angehängtes i);
- Zahlen zur Angabe von anderen Einheiten (ohne angehängten Buchstaben), d.h. Zeilen, Zeichen, Zeilen pro Zoll oder Zeichen pro Zoll.

Mit Ausnahme der beiden letzten Zeilen können die oben gezeigten Elemente in beliebiger Reihenfolge angegeben sein. Die Zeilen `comment:` und `kommentar` müssen nacheinander erscheinen, können aber vor den übrigen Punkten angegeben werden, und die Elemente `alignment pattern:` und `inhalt` müssen nacheinander am Ende der Datei erscheinen. Auch darf der Punkt `kommentar` nur dann Zeilen enthalten, die mit einem der anderen Schlüsselausdrücke beginnen, wenn dem Schlüsselausdruck eine spitze Klammer (:) vorausgeht. Beim Anzeigen des `kommentars` werden vorangestellte Spitzklammern gelöscht. Die Groß- und Kleinschreibung bei Schlüsselausdrücken wird ignoriert.

Mit diesem Kommando wird die mit *formularname* bezeichnete Formulardefinition in die Liste der Formulare aufgenommen. Ist die Definition bereits vorhanden, wird sie mit den neuen Werten aktualisiert. Sobald das Formular aufgenommen ist, kann es in einer Druckanforderung verwendet werden, mit Ausnahme der Fälle, in denen der Zugriff beschränkt wird. Dies wird im Zusammenhang mit der Option `-u` näher beschrieben. Die Verwendung eines Formulars kann auch auf bestimmte Drucker beschränkt werden.

Im folgenden werden die einzelnen Formularattribute beschrieben:

`Page length and Page Width` (Seitenlänge und -breite)

Bevor eine Druckanforderung für dieses Formular ausgeführt wird, initialisiert das generische Schnittstellenprogramm, das mit dem LP-Druckdienst geliefert wird, den physischen Drucker für das Drucken von Seiten mit der Länge *skalierte-dezimalzahl*₁ und der Breite *skalierte-dezimalzahl*₂. Als Zugang zur Datenbank `terminfo` dient der Druckertyp.

Seitenlänge und -breite werden nach Möglichkeit ebenfalls an alle Filter übergeben, die in einer Anforderung für dieses Formular verwendet werden.

`Number of pages` (Seitenzahl)

Bei jeder Ausgabe des Justierungsmusters versucht der LP-Druckdienst, den *inhalt* auf ein Blatt zu beschränken, indem es, wenn möglich, an jeden Filter als Seitenbereich *1-ganzzahl* angibt.

`Line pitch und Character pitch` (Zeilen- und Zeichendichte)

Bevor eine Druckanforderung für dieses Formular ausgeführt wird, initialisieren die Schnittstellenprogramme, die mit dem LP-Druckdienst geliefert werden, den physischen Drucker für das Drucken mit diesen Werten. Als Zugang zur Datenbank `terminfo` dient der Druckertyp. Auch werden diese Werte, wenn möglich, an alle Filter übergeben, die für das Drucken

dieses Formulars verwendet werden. Mit *skalierte-dezimalzahl*₃ werden die Zeilen pro Zentimeter angegeben, wenn der Zahl der Buchstabe *c* folgt, anderenfalls die Zeilen pro Zoll. Entsprechend wird mit *skalierte-dezimalzahl*₄ die Anzahl der Zeichen pro Zentimeter angegeben, wenn der Zahl ein *c* folgt, andernfalls bezieht sich der Wert auf Zeichen pro Zoll. Die Zeichendichte kann auch mit *elite* (12 Zeichen pro Zoll), *pica* (10 Zeichen pro Zoll) oder *compressed* (so viele Zeichen pro Zoll wie möglich) angegeben werden.

Character set choice (Auswahl des Zeichensatzes)

Wenn der LP-Druckdienst einen Verwalter anweist, dieses Formular zu aktivieren, meldet er auch, daß das Typenrad *typenrad* auf den Druckern, die dafür ausgelegt sind, verwendet werden soll. Soll das Formular auf einem Drucker ausgegeben werden, der statt über Typenräder über wähl- oder ladbare Zeichensätze verfügt, wählen bzw. laden die mit dem LP-Druckdienst gelieferten Schnittstellenprogramme den richtigen Zeichensatz automatisch. Ist *mandatory* (verbindlich) angegeben, darf der Benutzer für das Formular keinen anderen Zeichensatz auswählen; ansonsten handelt es sich bei dem angegebenen Zeichensatz bzw. dem genannten Typenrad um einen Vorschlag und einen Standardwert.

Ribbon color (Farbbandfarbe)

Wenn der LP-Druckdienst den Verwalter benachrichtigt, weil dieses Formular benutzt werden soll, meldet er auch, daß ein Farbband mit der Farbe *farbbandfarbe* benutzt werden soll.

Comment (Kommentar)

Der LP-Druckdienst zeigt den *kommentar* unverändert an, wenn ein Benutzer Fragen zu diesem Formular stellt (siehe *lpstat(1)*).

Alignment pattern (Justierungsmuster)

Beim Einlegen des Formulars kann der Verwalter den *inhalt* des Druckauftrags mehrmals ausgeben lassen. Dies hilft ihm, den Vordruck richtig auszurichten. Mit der wahlfreien Angabe *inhaltstyp* wird der Druckertyp definiert, für den der *inhalt* generiert wurde. Wird kein *inhaltstyp* angegeben, wird standardmäßig *simple* verwendet. Es ist zu beachten, daß der *inhalt* wie eingegeben gespeichert wird und nur vom Benutzer *lp* gelesen werden kann.

Wird mit diesem Kommando eine bestehende Formulardefinition geändert, werden Elemente, die in der neuen Version fehlen, unverändert beibehalten. Wird mit diesem Kommando eine Formulardefinition hinzugefügt, gelten für fehlende Elemente die folgenden Standardwerte:

Seitenlänge: 66
 Seitenbreite: 80
 Seitenanzahl: 1
 Zeilendichte: 6
 Zeichendichte: 10
 Zeichensätze: any
 Farbbandfarbe: any

Löschen einer Formulardefinition

Mit der Option `-x` wird das Formular *formulardname* aus dem LP-Druckdienst gelöscht.

Liste der Formularattribute

Mit der Option `-l` wird eine Attributliste für das bestehende Formular *formulardname* erstellt. Die möglichen Attribute wurden bereits im Abschnitt "Hinzufügen" oder "Ändern einer Formulardefinition" beschrieben. Da das Justierungsmuster u.U. vertraulich ist, darf nur der Verwalter das Formular mit diesem Kommando überprüfen. Andere Benutzer können mit dem Kommando `lpstat` den nicht vertraulichen Teil der Formularbeschreibung prüfen.

Zugriff auf ein Formular gewähren oder verweigern

Mit der Option `-u`, gefolgt von dem Argument `allow:anmeldenummernliste` oder `-u deny:anmeldenummernliste`, kann definiert werden, welche Benutzer in einer Druckanforderung ein bestimmtes Formular angeben dürfen. Diese Option kann mit der Option `-F` oder `-` verwendet werden, die bereits unter "Hinzufügen" oder "Ändern einer Formulardefinition" beschrieben wurden.

Das Argument *anmeldenummernliste* darf eine, mehrere oder alle der folgenden Kombinationen enthalten:

<i>anmeldenummer</i>	Ein Benutzer auf einem beliebigen System
<i>systemname!anmeldenummer</i>	Ein Benutzer auf System <i>systemname</i>
<i>systemname!all</i>	Alle Benutzer auf System <i>systemname</i>
<i>all!anmeldenummer</i>	Ein Benutzer auf allen Systemen
<i>all</i>	Alle Benutzer auf allen Systemen

Der LP-Druckdienst führt für jedes Formular zwei Benutzerlisten: eine "Zulassungsliste" mit denjenigen, die das Formular benutzen dürfen, und eine "Verweigerungsliste" mit den Benutzern, die das Formular nicht benutzen dürfen. Mit der Option `-u allow` werden die angegebenen Benutzer in die Zulassungsliste aufgenommen und aus der Verweigerungsliste gelöscht. Mit der Option `-u deny` werden die angegebenen Benutzer in die Verweigerungsliste aufgenommen und aus der Zulassungsliste gelöscht. (Beide Formen der Option `-u` können zusammen mit der Option `-F` oder `-` verwendet werden.)

Ist die Zulassungsliste nicht leer, dürfen nur die dort genannten Benutzer auf das Formular zugreifen, unabhängig vom Inhalt der Verweigerungsliste. Ist die Zulassungsliste leer, die Verweigerungsliste dagegen nicht, dürfen die in der Verweigerungsliste angegebenen Benutzer das Formular nicht verwenden, alle anderen Benutzer dürfen jedoch darauf zugreifen. Allen Benutzern kann mit der Angabe `-f deny:all` der Zugriff auf das Formular verweigert werden. Alle Benutzer dürfen den Drucker verwenden, wenn `-f allow:all` angegeben wird. Dies ist der Standardwert.

Definieren eines Alarmsignals für das Formular

Mit den Optionen `-f formularname` und `-A alarmsignaltyp` wird ein Alarmsignal definiert, mit dem das Formular angefordert wird, wenn Aufträge dafür in der Warteschlange stehen. Wird mit dieser Option kein Alarmsignal für dieses Formular definiert, wird kein Signal ausgegeben.

Nach welcher Methode das Alarmsignal gesendet wird, hängt vom Wert des Arguments *alarmsignaltyp* zu der Option `-A` ab. Folgende *alarmsignaltypen* stehen zur Verfügung:

- `mail` Sendet die Alarmmeldung mit dem Kommando `mail` an den Verwalter.
- `write` Schreibt die Meldung mit dem Kommando `write` an das Terminal, an dem der Verwalter angemeldet ist. Ist der Verwalter an mehreren Terminals angemeldet, wird eines davon willkürlich ausgewählt.
- `quiet` Gibt keine Meldungen über den aktuellen Zustand aus. Der Verwalter kann mit dieser Option die Ausgabe weiterer Meldungen zu einem bereits bekannten Zustand zeitweilig unterbrechen. Nachdem das Formular *formularname* eingelegt und anschließend wieder entfernt wurde, werden erneut Meldungen gesendet, wenn die Anzahl der Druckanforderungen den mit der Option `-Q` definierten Schwellenwert erreicht.
- `none` Erst dann Meldungen senden, wenn die Option `-A` mit einem anderen *alarmsignaltyp* (nicht `quiet`) erneut eingegeben wird.

shell-kommando

Das *shell-kommando* wird immer dann ausgeführt, wenn ein Alarmsignal gesendet werden muß. Das Shell-Kommando erwartet die Meldung in der Standardeingabe. Enthält das Kommando Leerzeichen, muß es in Anführungszeichen gesetzt werden. Die `mail`- und `write`-Werte für diese Option entsprechen den Werten `mail benutzername` und `write benutzername`, wobei *benutzername* der aktuelle Name für den Verwalter ist. Dies ist der Benutzername desjenigen, der dieses Kommando ausgibt, sofern nicht mit dem Kommando `su` zu einer anderen Benutzernummer gewechselt wurde. Wurde die Benutzernummer mit `su` geändert, wird der *benutzername* für die neue Nummer verwendet.

- `list` Zeigt den Typ des Alarmsignals zum Wechseln des Typenrads auf der Standardausgabe an. Das Alarmsignal wird dadurch nicht geändert.

Folgende Meldung erscheint:

```
The form formularname needs to be mounted
on the printer(s):
drucker (ganzzahl1 anforderungen).
ganzzahl2 print requests await this form.
Use the farbbandfarbe ribbon.
Use the typenrad print wheel, if appropriate.
(Das Formular formularname muß in den/die
Drucker eingelegt werden:
```

drucker (*ganzzahl*₁ Anforderungen).
*ganzzahl*₂ warten auf dieses Formular.
 Farbband mit der Farbe *farbbandfarbe* verwenden.
 Ggf. Typenrad *typenrad* verwenden.

Die angegebenen Drucker entsprechen denen, die der Verwalter zuvor für die Verwendung mit diesem Formular definiert hat. Die neben jedem Drucker angegebene Zahl *ganzzahl*₁ gibt die Anzahl der für den jeweiligen Drucker anstehenden Anforderungen an. Die nach der Druckerliste angegebene Zahl *ganzzahl*₂ gibt die Gesamtzahl der Anforderungen an, die auf das Formular warten. Sie ist eventuell kleiner als die Summe der anderen Zahlen, wenn einige Anforderungen von mehreren Druckern ausgeführt werden können. Für *farbbandfarbe* und *typenrad* gelten die in der Formulardefinition angegebenen Werte. Die letzte Zeile der Meldung wird immer gesendet, auch wenn keiner der Drucker mit Typenrädern arbeitet, da der Verwalter das Formular eventuell auf einem Drucker verwendet, der Typenräder benutzt.

Wenn beliebige Farbbänder und Typenräder zulässig sind, lauten die Anweisungen:

Use any ribbon.
 Use any print-wheel.
 (Farbband: beliebig.
 Typenrad: beliebig.)

Gilt für *formularnamen* der Wert *any*, gilt das mit diesem Kommando definierte Alarmsignal für alle Formulare, für die bisher kein solches Signal definiert war. Hat *formular* den Wert *all*, wird das mit diesem Kommando definierte Alarmsignal für alle Formulare ausgegeben.

Wird die Option *-w* nicht angegeben, wird standardmäßig nur einmal eine Meldung ausgegeben, wenn ein Formular eingelegt werden muß. Wird die Option *-w* nicht angegeben, entspricht dies der Angabe *-w once* bzw. *-w 0*. Ist der Wert von *minuten* größer als null, wird das Alarmsignal in Abständen von *minuten* ausgegeben.

Wird auch die Option *-Q* angegeben, wird das Alarmsignal dann gesendet, wenn eine bestimmte, mit dem Argument *anforderungen* definierte, Anzahl von Druckanforderungen für das Formular in der Warteschlange steht. Wird die Option *-Q* nicht angegeben oder enthält *anforderungen* den Wert 1 oder das Wort *any* (beides Standardwerte), wird eine Meldung gesendet, sobald eine Anforderung für das nicht eingelegte Formular eingeht.

Anzeigen des aktuellen Alarmsignaltyps

Mit den Optionen `-f` und `-A` und dem Argument `list` wird der Typ des Alarmsignals ausgegeben, der für das angegebene Formular *formularname* definiert wurde. Das Signal selbst wird nicht verändert. Wird *formularname* vom LP-Druckdienst erkannt, wird, je nach dem definierten Alarmsystemtyp, eine der folgenden Meldungen in die Standardausgabe geschrieben.

- When *anforderungen* requests are queued:
alert with *shell-kommando* every *minuten* minutes
(Wenn *anforderungen* Anforderungen in der Warteschlange stehen:
Alarmsignal mit *shell-kommando* alle *minuten* Minuten ausgeben)
- When *anforderungen* requests are queued:
write to *benutzername* every *minuten* minutes
(Wenn *anforderungen* Anforderungen in der Warteschlange stehen:
Meldung an *benutzername* alle *minuten* Minuten)
- When *anforderungen* requests are queued:
mail to *benutzername* every *minuten* minutes
(Wenn *anforderungen* Anforderungen in der Warteschlange stehen:
Mitteilung an *benutzername* alle *minuten* Minuten)
- No alert (Kein Alarmsignal)

Der Ausdruck `every minutes minutes` wird durch `once` (einmal) ersetzt, wenn der Wert *minuten* (`-W minuten`) gleich null ist.

Ausschalten eines aktiven Alarmsignals

Mit der Option `-A quiet` werden die Meldungen über den aktuellen Zustand ausgeschaltet. Damit kann der Verwalter die Ausgabe weiterer Meldungen zu einem bereits bekannten Zustand zeitweilig unterbrechen. Nachdem das Formular eingelegt und anschließend wieder entfernt wurde, werden erneut Meldungen gesendet, wenn die Anzahl der Druckanforderungen den definierten Schwellenwert *anforderungen* erreicht.

Löschen einer Definition für ein Alarmsignal

Wurde die Option `-A none` angegeben, werden keine Alarmmeldungen gesendet, bis die Option `-A` mit einem anderen *alarmsystemtyp* erneut eingegeben wird. Damit kann die Ausgabe weiterer Meldungen dauerhaft unterbunden werden, wenn alle vorhandenen Definitionen für Alarmsignale für das Formular gelöscht werden.

SIEHE AUCH

`lpadmin(1M)`, `terminfo(4)`.
`lp(1)` in den *Kommandos*.

BEZEICHNUNG

lpsched, lpshut, lpmove – startet/stoppt den LP-Druckdienst und leitet Anforderungen um

ÜBERSICHT

```
/usr/lib/lp/lpsched
lpshut
lpmove anforderungen ziel
lpmove ziel1 ziel2
```

BESCHREIBUNG

lpsched startet den LP-Druckdienst; dies ist nur mit root oder lp möglich.

lpshut beendet den Druckdienst. Alle Drucker, die beim Aufruf des Kommandos lpshut laufen, werden deaktiviert. Beim erneuten Starten von lpsched werden Anforderungen, die zum Zeitpunkt des Abbruchs bearbeitet wurden, vom Anfang an neu gedruckt.

Mit lpmove werden Anforderungen, die lp in die Warteschlange eingereicht hat, zwischen den LP-Zielen umgeleitet. Die erste Form des Kommandos lpmove (unter **ÜBERSICHT**) sendet die angegebenen *anforderungen* zum LP-Ziel *ziel*. Bei *anforderungen* handelt es sich um Anforderungsnummern, die von lp zurückgegeben werden. Mit der zweiten Form des Kommandos lpmove wird versucht, alle Anforderungen für Ziel *ziel*₁ zu Ziel *ziel*₂ umzuleiten. Danach weist lp alle neuen Anforderungen für *ziel*₁ zurück.

Es ist zu beachten, daß lpmove beim Umleiten von Anforderungen nie den Annahme-Status (siehe accept(1M)) des neuen Ziels überprüft. Auch werden die Anforderungsnummern der umgeleiteten Anforderung nicht geändert, so daß die Benutzer ihre Anforderungen wiederfinden können. Das Kommando lpmove leitet keine Anforderungen um, deren Optionen (Inhaltstyp, benötigtes Formular usw.) auf dem neuen Ziel nicht verarbeitet werden können.

Wurde eine Anforderung ursprünglich in die Warteschlange für eine Klasse oder das Sonderziel any eingereicht und die erste Form des Kommandos lpmove wurde verwendet, wird das Ziel der Anforderung in *neues-ziel* geändert. Eine solche Anforderung kann nur auf dem *neuen-ziel* und nicht auf anderen Mitgliedern der Klasse oder anderen akzeptablen Druckern ausgegeben werden, wenn das ursprüngliche Ziel any war.

DATEIEN

```
/var/spool/lp/*
```

SIEHE AUCH

accept(1M), lpadmin(1M).
enable(1), lp(1), lpstat(1) in den *Kommandos*.

BEZEICHNUNG

`lpsystem` – Eintragen ferner Systeme beim Druckdienst

ÜBERSICHT

```
lpsystem [-t typ] [-T wartezeit] [-R pause] [-y "kommentar"]
          systemname [systemname ...]

lpsystem -l [systemname ...]

lpsystem -r systemname [systemname ...]

lpsystem -A
```

BESCHREIBUNG

Das Kommando `lpsystem` dient zum Definieren von Parametern für den Druckdienst LP, und zwar für die Kommunikation mit fernen Systemen (über Hochgeschwindigkeits-Netzwerke wie STARLAN oder TCP/IP). Nur ein privilegierter Benutzer (d.h. der Eigentümer des Logins `root`) darf das Kommando `lpsystem` benutzen.

Das Kommando `lpsystem` dient vor allem zur Definition von fernen Systemen, mit denen der lokale Druckdienst LP Druckanforderungen austauschen kann. Diese fernen Systeme werden gegenüber dem lokalen Druckdienst LP in Form mehrerer Parameter zur Kommunikationssteuerung definiert: Typ, Pause und Wartezeit. Diese Parameter sind in der Datei `/etc/lp/Systems` definiert. Sie kann mit einem Text-Editor (z.B. `vi`) editiert werden; dies ist jedoch nicht empfehlenswert.

-t *typ* Der Parameter *typ* definiert den Typ des fernen Systems. Zwei Typenangaben sind möglich: `s5` (System V Release 4) oder `bsd` (SunOS). Der Standardtyp ist `s5`.

-T *wartezeit*

Der Parameter *wartezeit* gibt an, wie lange (in Minuten) eine Netzwerkverbindung sich im Leerlauf befinden darf. Gültige Werte sind `n`, `0` und `N`; dabei ist `N` eine ganze Zahl größer `0`. Befindet sich die Netzwerkverbindung zu dem fernen System `N` Minuten lang im Leerlauf, d.h. findet während dieser Zeit keine Datenübertragung statt, wird sie abgebrochen. Sie wird wiederhergestellt, wenn erneut eine Datenübertragung ansteht. Der Wert `n` steht für "unbegrenzte Wartezeit"; `0` bedeutet "die Verbindung abbrechen, sobald sie keine Daten überträgt". Der Standardwert ist `n`.

-R *pause*

Der Parameter *pause* gibt an, wie lange nach einem außerplanmäßigen Abbruch (d.h. einem Netzwerkfehler) gewartet werden muß, bis die Verbindung zu dem fernen System wiederhergestellt werden kann. Gültige Werte sind `n`, `0` und `N`; dabei ist `N` eine ganze Zahl größer `0`, die folgende Bedeutung hat: "Verbindung nach `N` Minuten wiederherstellen". (Der Standardwert ist 10 Minuten.) Der Wert `n` bedeutet "Abgebrochene Verbindung erst dann wiederherstellen, wenn weitere Arbeit ansteht"; `0` bedeutet "Verbindung sofort wieder aufbauen".

-y "*kommentar*"

Durch das Argument *kommentar* kann ein frei gestalteter Kommentar mit dem Systemeintrag verbunden werden. Dieser wird angezeigt, wenn `lpsystem -l` verwendet wird.

- l Ist zusätzlich ein Systemname *systemname* angegeben, dann gibt *lpsystem* eine Beschreibung der zu diesem System gehörenden Parameter aus. Fehlt die Angabe eines Systemnamens, dann gibt *lpsystem* die Parameter aller Systeme in der Datenbank aus.
- r *lpsystem* entfernt den zu *systemname* gehörenden Eintrag. Der Druckdienst nimmt von diesem System keine Aufträge mehr an und sendet keine Aufträge mehr dorthin, auch wenn der ferne Drucker auf dem lokalen System noch definiert ist.
- A *lpsystem* gibt die TCP/IP - Adresse des lokalen Rechners aus. Die Ausgabe erfolgt in einem Format, das dazu geeignet ist, den lokalen Port Monitor so zu konfigurieren, daß er Anforderungen von einem SunOS-System akzeptiert.

systemname

systemname ist der Name des fernen Systems, von dem der lokale Druckdienst Aufträge empfangen und an das er Aufträge senden soll.

HINWEIS

Diese Informationen sind im Hinblick auf */etc/lp/Systems* und die Steuerung der Kommunikation im Netzwerk nicht sonderlich aussagekräftig. Netzwerkadressen und -dienste werden von den Einrichtungen *Netconfig* und *Netdir* verwaltet (Angaben zu Netzwerkadressen und -diensten finden Sie im Kapitel "Einstellen der Netzkonfiguration" im *Leitfaden für Systemverwalter*). Port Monitore sind für das Horchen nach Anforderungen ferner Dienstanforderungen und das Routing der Verbindung zum Druckdienst zuständig (in dem Kapitel "Anschluß-Verwaltung" im *Leitfaden für Systemverwalter* finden Sie nähere Angaben zu Port Monitoren).

Sind die Einrichtungen *Netconfig* und *Netdir* nicht ordnungsgemäß konfiguriert, kann der abgehende ferne Druckdienst wahrscheinlich nicht arbeiten. Sind die lokalen Port Monitore nicht dafür konfiguriert, ferne Druckanforderungen zum Druckdienst zu leiten, steht der Dienst für ferne Systeme nicht zur Verfügung. (Nähere Angaben zum ordnungsgemäßen Vorgehen finden Sie im Kapitel "Druckerverwaltung" des *Leitfadens für Systemverwalter*.)

Für die Semantik der Werte *wartezeit* und *pause* gilt, daß der Druckdienst je einen Prozeß für jedes ferne System verwendet, mit dem er kommuniziert, und daß er mit einem fernen System nur dann kommuniziert, wenn ein Auftrag auf diesem System ausgeführt oder von ihm gesendet werden soll.

Das System, das die Verbindung aufbaut, wird als "Master-", das System, das die Verbindung akzeptiert, wird als "Slave"-Prozeß bezeichnet. Diese Bezeichnung soll lediglich angeben, welcher Prozeß beendet wird (der Slave), wenn eine Verbindung abgebrochen wird. So wird verhindert, daß mehrere Prozesse mit einem fernen System kommunizieren. Davon abgesehen sind alle Verbindungen, unabhängig von der Bezeichnung als Master oder Slave, Zweiwege-Verbindungen. Die Kennzeichnung Master/Slave läßt sich nicht steuern. Daraus ergibt sich, daß bei einer Wartezeitüberschreitung bei einem Master sowohl Master- als auch Slave-Prozeß beendet werden. Wird die Wartezeit für einen Slave-Prozeß überschritten, kann der Master dennoch aktiv bleiben und nach der Wiederholungspause die Verbindung erneut herstellen. Daher kann die Strategie der Ressourcenverwaltungs eines Systems die Strategie eines anderen Systems beeinflussen.

Zu `lpsystem -A`: Ein SunOS-System (angegeben als `-t bsd`) darf an das System des Benutzers nur über TCP/IP angeschlossen werden, Druckanforderungen eines SunOS-Systems an das lokale System können nur über einen Sonderanschluß (515) ankommen. Die von `lpsystem` zugewiesene Adresse entspricht der Adresse des Systems und des Anschlusses 515. Diese Adresse verwendet das TCP/IP-Anschlußüberwachungsprogramm (siehe `sacadm(1M)` und `nlsadmin(1M)`), um diese Adresse und den Anschluß "abzuhören" und Verbindungen zum Druckdienst zu leiten. (Diese Prozedur wird im Kapitel "Anschluß-Verwaltung" des *Leitfadens für Systemverwalter* erläutert.) Wichtig ist dabei, daß an dieser Stelle in dieser Prozedur auf die Adresse verwiesen wird.

Das Kommando `lpsystem -A` kann nicht ausgeführt werden, wenn der Systemname und die IP-Adresse nicht in der Datei `/etc/inet/hosts` und der Druckerdienst nicht in der Datei `/etc/inet/services` enthalten sind.

DATEIEN

`/var/spool/lp/* /etc/lp/*`

SIEHE AUCH

`netconfig(4)`

Leitfaden für Programmierer, Netzwerk-Schnittstellen

Leitfaden für Systemverwalter

BEZEICHNUNG

lpusers – legt die Prioritäten in der Druckwarteschlange fest

ÜBERSICHT

```
lpusers -d prioritätsstufe
lpusers -q prioritätsstufe -u benutzernummernliste
lpusers -u benutzernummernliste
lpusers -q prioritätsstufe
lpusers -l
```

BESCHREIBUNG

Mit dem Kommando `lpusers` wird die Prioritätsstufe in der Warteschlange eingeschränkt, die die Benutzer des LP-Druckdienstes einem Auftrag zuweisen können.

Mit der ersten Form des Kommandos (mit `-d`) wird der im System gültige Standardwert für die Priorität auf *prioritätsstufe* eingestellt, wobei für *prioritätsstufe* ein Wert zwischen 0 und 39 angegeben werden kann. 0 ist die höchste Prioritätsstufe. Wenn ein Benutzer für eine Druckanforderung keine Prioritätsstufe angibt (siehe `lp(1)`), gilt der Standardwert für die Prioritätsstufe, der anfangs bei 20 liegt.

Mit der zweiten Form des Kommandos (mit `-q` und `-u`) wird der höchste Standardwert für die *prioritätsstufe* (0-39) eingestellt, den die Benutzer in *benutzernummernliste* für eine Druckanforderung eingeben können. Das Argument *benutzernummernliste* kann eine beliebige oder alle der folgenden Angaben enthalten:

<i>benutzernummer</i>	Ein Benutzer auf einem beliebigen System
<i>systemname</i> ! <i>benutzernummer</i>	Ein Benutzer auf dem System <i>systemname</i>
<i>systemname</i> !all	Alle Benutzer auf dem System <i>systemname</i>
all! <i>benutzernummer</i>	Ein Benutzer auf allen Systemen
all	Alle Benutzer auf allen Systemen

Benutzer, für die ein Grenzwert festgelegt wurde, können keine Druckanforderung mit einer höheren Prioritätsstufe absetzen, als für sie zulässig ist. Sie können auch für eine bereits abgesetzte Anforderung keine höhere Priorität festlegen. Allen Druckanforderungen, die mit einer höheren Prioritätsstufe als zulässig abgesetzt werden, wird die höchste zulässige Priorität zugewiesen.

Mit der dritten Form des Kommandos (mit `-u`) werden alle explizit für die angegebenen Benutzer festgelegten Prioritätsstufen gelöscht.

Mit der vierten Form des Kommandos (mit `-q`) wird der höchste Standardwert für die Prioritätsstufe allen Benutzern zugewiesen, die nicht explizit mit der zweiten Form des Kommandos erfaßt wurden.

Mit der letzten Form des Kommandos (mit `-l`) werden die standardmäßige Prioritätsstufe und die den Benutzern zugewiesenen Prioritätsgrenzen aufgelistet.

SIEHE AUCH

`lp(1)` in den *Kommandos*.

BEZEICHNUNG

`mail_pipe` – ruft das Kommando auf, das ankommende Post empfängt

ÜBERSICHT

`mail_pipe [-x fehlersuchstufe] -r empfänger -R pfad_zum_sender -c inhaltstyp
-S subjekt`

BESCHREIBUNG

Wenn eine neue Postmeldung ankommt, überprüft das Kommando `mail` zunächst, ob der elektronische Briefkasten des Empfängers angibt, daß die Meldung weitergeleitet werden soll (an einen anderen Empfänger oder als Eingabe für ein Kommando). Soll die Meldung über eine Pipe an ein vom Empfänger angegebenes Kommando weitergeleitet werden, ruft `mail` zunächst `mail_pipe` auf. Es werden einige Prüfungen vorgenommen und das Kommando wird im Kontext des Empfängers ausgeführt.

In der Kommandozeile können die folgenden Argumente eingesetzt werden:

- x *fehlersuchstufe* Die Fehlersuche für diesen Aufruf wird aktiviert. Detaillierte Informationen finden Sie unter der Beschreibung der Option -x für das Kommando `mail`.
- r *empfänger* Die Benutzernummer des Empfängers.
- R *pfad_zum_sender* Die Rückkehradresse zum Sender der Meldung.
- c *inhaltstyp* Der Wert des Content-Type (Inhaltstyps): Kopfzeile in der Meldung.
- S *subjekt* Der Wert des Subject (Subjekts): Kopfzeile in der Meldung, falls vorhanden.

`mail_pipe` ist als ein Prozeß installiert, bei dem das s-Bit für Benutzer auf root gesetzt ist. Er kann daher, wenn nötig, seine Benutzer- und Gruppennummer in die des Empfängers ändern.

Wenn `mail_pipe` aufgerufen wird, führt der Prozeß die folgenden Schritte durch (schlägt ein Schritt fehl, wird der Exit-Code [N] festgehalten):

- Die beim Aufruf angegebenen Argumente werden geprüft [1].
- Es wird überprüft, ob der Name des Empfängers ≤ 14 Zeichen lang ist [2].
- Es wird überprüft, ob die Marke zum Setzen des s-Bits für Gruppe für den empfangenden Briefkasten gesetzt ist [3].
- Die Datei `/var/mail/empfänger` [4] wird geöffnet.
- Es wird überprüft, ob der Briefkasten des Empfängers mit der Zeichenkette `Forward to` [5] (Weiterleiten an) anfängt.
- Das Pipe-Symbol, das den Anfang der Kommandozeichenkette im empfangenden Briefkasten angibt, wird gesucht [6].
- In der Datei `/etc/passwd` [7] wird ein Eintrag für den Empfänger gesucht.
- Das s-Bit für Gruppe wird auf die Gruppennummer des Empfängers gesetzt [8].
- Das s-Bit für Benutzer wird auf die Benutzernummer des Empfängers gesetzt [9].
- Aus dem aktuellen Verzeichnis wird in das Anmeldeverzeichnis des Empfängers gewechselt [10].
- Es wird Speicherplatz für die mit dem Kommando `exec` neu eingerichtete Umgebung für das empfangende Kommando zugewiesen [11].

mail_pipe(1M)

mail_pipe(1M)

- Das empfangende Kommando wird analysiert, wobei die erforderliche `%schlüsselwort`-Expansion vorgenommen wird. Weitere Informationen über die `%schlüsselwort`-Expansion [12] finden Sie in dem Abschnitt 'Forwarding mail' unter `mail(1)`.
- Das empfangende Kommando wird ausgeführt [13, wenn `exec` fehlschlägt, andernfalls der Exit-Code des empfangenden Kommandos selbst].

DATEIEN

<code>/etc/passwd</code>	um den Sender zu identifizieren und den Empfänger zu finden
<code>/var/mail/<i>empfänger</i></code>	ankommende Post für den <i>empfänger</i> , d.h. die Post-datei
<code>/tmp/MLDBG*</code>	Verfolgungsdatei für Fehlersuche
<code>/usr/lib/mail/mail_pipe</code>	mail_pipe-Programm

SIEHE AUCH

`mail(1)`, `notify(1)`, `vacation(1)`

BEZEICHNUNG

`makefsys` – erzeugt ein Dateisystem

ÜBERSICHT

`makefsys`

BESCHREIBUNG

Mit dem Kommando `makefsys` kann ein Dateisystem angelegt werden.

Das Kommando ruft eine Menü-Schnittstelle auf (das `make`-Kommando, das über das Kommando `sysadm` zur Verfügung steht).

Zunächst kann der Benutzer auswählen, auf welchem Gerät das Dateisystem erstellt werden soll. Danach wird zur Eingabe einiger weiterer Informationen aufgefordert, bevor das Dateisystem angelegt wird.

Dieselbe Funktion ist auch unter dem `sysadm`-Menü zu finden:

```
sysadm make
```

EXIT-CODES

Die Exit-Codes von `makefsys` haben folgende Bedeutung:

- 0 = normales Beenden des Kommandos
- 2 = ungültige Kommandosyntax; eine Meldung mit der richtigen Syntax wird angezeigt
- 7 = die Menü-Schnittstelle für dieses Kommando steht nicht zur Verfügung, da `fmli` nicht aufgerufen werden kann; das FMLI-Paket ist nicht installiert oder beschädigt

SIEHE AUCH

`checkfsys(1M)`, `labelit(1M)`, `mkfs(1M)`, `mountfsys(1M)`, `sysadm(1M)`.

BEZEICHNUNG

migration – verschieben eines Archivs von einem auf einen anderen Datenträger

ÜBERSICHT

migration -B [-dlmotuvAENS] siaufnr udname udsgcr udskenr beschr

BESCHREIBUNG

migration wird als Sohnprozeß von *bkdaemon(1M)* aufgerufen, um ein vorhandenes, mit einer beliebigen Methode erstelltes Archiv, auf einen anderen Datenträger zu verlagern. Der vorhandene Sicherungsprotokolleintrag für das Archiv wird mit den Angaben zu den neuen Datenträgern sowie Angaben zum Ziel des Archivs aktualisiert.

siaufnr ist die von *backup(1M)* zugewiesene Auftragsnummer. *udsgcr* ist der Name des Geräts im Raw-Modus, auf dem sich das Archiv befindet. *udskenr* ist das Datenträgerkennzeichen des Archivs (siehe *labelit(1M)*). *beschr* ist eine Beschreibung für ein Zielgerät in der Form:

ggruppe:gname:gchar:gkenn

ggruppe gibt eine Gerätegruppe an. *gname* gibt einen Gerätenamen an. *gchar* gibt Merkmale für das genannte Gerät und die genannte Gruppe an (eine ausführliche Beschreibung der Gerätemerkmale finden Sie unter *device.tab(4)*). *gkenn* gibt die Namen für die Datenträger an, die für das Archiv benutzt werden sollen.

Optionen

- d** Eintrag im Sicherungsprotokoll für das Archiv nicht aktualisieren.
- l** Erstellen einer ausführlichen Form des Sicherungsprotokolls, das ein Inhaltsverzeichnis des Archivs enthält. Hierzu gehören die Daten, die zum Erstellen einer *ls -l*-artigen Liste aller Dateien im Archiv benötigt werden.
- m** Einhängen des ursprünglichen Dateisystems mit Leseberechtigung vor dem Sichern und erneutes Einhängen mit den ursprünglichen Zugriffsberechtigungen nach Abschluß der Sicherungsoperation.
- o* Ermöglicht es dem Benutzer, Aufforderungen zum Einlegen von Datenträgern außer Kraft zu setzen (siehe *getvol(1M) -o*).
- t** Erstellen eines Inhaltsverzeichnisses für die Sicherung auf zusätzlichen Datenträgern anstatt im Sicherungsprotokoll.
- u** Aushängen des ursprünglichen Dateisystems vor dem Sichern und Wiedereinhängen nach Abschluß der Sicherungsoperation mit den ursprünglichen Zugriffsberechtigungen.
- v** Prüfen des Archivs beim Schreiben. Während das Archiv geschrieben wird, wird eine Prüfsumme berechnet; sobald die einzelnen Datenträger abgeschlossen sind, wird diese Prüfsumme erneut gelesen und neu berechnet, um sicherzugehen, daß alle Blöcke lesbar und korrekt sind.

Schlägt eine dieser Prüfungen fehl, gilt der Datenträger als nicht lesbar. Wurde die Option `-A` angegeben, schlägt die Archivierungsoperation fehl; anderenfalls wird der Bediener aufgefordert, den fehlerhaften Datenträger zu ersetzen.

- A Keine Eingabeaufforderung an den Benutzer bei Operationen mit auswechselbaren Datenträgern (automatischer Betrieb).
- E* Gibt eine Schätzung des Datenträgerbedarfs für das Archiv aus; anschließend wird die Sicherungsoperation ausgeführt.
- N* Gibt eine Schätzung des Datenträgerbedarfs für das Archiv aus; die Sicherung wird nicht durchgeführt.
- S* Schreibt auf dem Zielgerät einen Punkt (.) nach jeweils 100 Blöcken (zu 512 Byte), die aus dem Archiv gelesen oder darin geschrieben wurden.

Dialog mit dem Benutzer

Die Komplexität der Verbindung zwischen einer Archivierungsmethode und `backup(1M)` geht über eine einfache `fork/exec`- oder `Pipe`-Beziehung hinaus. `backup(1M)` ist für den gesamten Dialog mit dem Benutzer zuständig, sei er direkt oder über `bkoper(1M)`. Daher liest `migration` weder aus der Standardeingabe, noch schreibt es in die Standardausgabe oder die Standardfehlerausgabe. Zum Übertragen von Berichten (Schätzungen, Punkte, Status usw.) an `backup(1M)` muß eine Verfabrensbibliothek benutzt werden (siehe `libbrmeth(3)`).

EXIT-CODES

Schließt `migration` seine Aufgabe erfolgreich ab, wird es mit dem Exit-Code 0 beendet. Ist einer der Parameter zu `migration` ungültig, lautet der Exit-Code 1. Tritt ein Fehler auf, durch den `migration` nicht *alle* Teile seiner Aufgabe erfüllen kann, lautet der Exit-Code 2.

Fehler werden gemeldet, wenn eine der folgenden Bedingungen gegeben ist:

1. `-t` wurde zusammen mit `-A` angegeben.
2. `-A` wurde zusammen mit `-o` angegeben.
3. `-t` wurde angegeben, und das Zielgerät unterstützt auswechselbare Datenträger nicht.
4. `-A` wurde angegeben, und es werden mehrere auswechselbare Datenträger benötigt.
5. Nicht behebbare Fehler traten bei dem Versuch auf, auf dem Zielgerät zu lesen oder zu schreiben.
6. `-m` wurde angegeben, und das ursprüngliche Dateisystem konnte nicht mit Leseberechtigung eingehängt werden.
7. `-m` wurde eingehängt, und das ursprüngliche Dateisystem konnte nicht ausgehängt werden.
8. `-o` wurde nicht angegeben, und in *beschr* wurden unzureichende Datenträgernamen angegeben.

migration(1M)

migration(1M)

9. -u wurde angegeben, und das Dateisystem konnte nicht ausgehängt werden.
10. -u wurde angegeben, und das Dateisystem konnte nicht wieder eingehängt werden.

DATEIEN

/usr/oam/bkrs/tables/bkhist.tab
\$TMP/filelist\$\$

SIEHE AUCH

awk(1), backup(1M), device.tab(4), getvol(1M), grep(1), labelit(1M), lib-
brmeth(3), ls(1), prtvtoc(1M), restore(1M), rsoper(1M), sed(1), time(2),
urestore(1M).

BEZEICHNUNG

`mkfifo` – erstellt FIFO-Gerätedateien

ÜBERSICHT

`mkfifo pfad ...`

BESCHREIBUNG

`mkfifo` erstellt die FIFO-Gerätedateien, die in der Argumentliste angegeben werden. Die Argumente werden sequentiell in der angegebenen Reihenfolge ausgewertet. Jede FIFO-Gerätedatei wird entweder vollständig oder, im Falle eines Fehlers oder Signals, überhaupt nicht erstellt.

Für jedes *pfad*-Argument verhält sich das Kommando `mkfifo` so, als ob die Funktion `mkfifo` (siehe `mkfifo(3C)`) mit dem auf *pfad* eingestellten Argument *pfad* und dem auf das bitweise inklusive ODER von `S_IRUSR`, `S_IWUSR`, `S_IRGRP`, `S_IWGRP`, `S_IROTH` und `S_IWOTH` eingestellten *modus* aufgerufen wird.

Treten beim Erstellen einer der Gerätedateien Fehler auf, schreibt `mkfifo` eine Fehlermeldung in die Standardfehlerausgabe und fährt mit den übrigen Argumenten, falls vorhanden, fort.

SIEHE AUCH

`mkfifo(3C)` im *Referenzhandbuch für Programmierer*.

EXIT-CODES

`mkfifo` gibt den Exit-Code 0 zurück, wenn alle FIFO-Gerätedateien normal erstellt wurden. Andernfalls wird eine Fehlermeldung ausgegeben und ein Wert größer 0 zurückgegeben.

BEZEICHNUNG

mkfs (generisch) – erstellt ein Dateisystem

ÜBERSICHT

mkfs [-F *dstyp*] [-V] [-m] [*aktuelle_optionen*] [-o *spezifische_optionen*] *gerät* [*operanden*]

BESCHREIBUNG

Mit *mkfs* wird für die Gerätedatei *gerät* ein Dateisystem eines bestimmten Typs angelegt; *gerät* muß nach den Optionen als erstes Argument angegeben werden. Das Dateisystem wird angelegt entsprechend den Angaben *dstyp*, *spezifische_optionen* und *operanden*, die in der Kommandozeile stehen. *mkfs* wartet 10 Sekunden, bevor das Dateisystem erstellt wird. Während dieses Intervalls kann das Kommando durch Drücken der DEL-Taste abgebrochen werden.

operanden sind *dstyp*-spezifisch. Eine detaillierte Beschreibung ist dem *dstyp*-spezifischen Handbucheintrag zu *mkfs* zu entnehmen. Abhängig vom Typ des Dateisystems sind bestimmte Operanden zwingend anzugeben.

aktuelle_optionen sind die Optionen, die von dem *s5*-spezifischen Modul von *mkfs* unterstützt werden. Andere Dateisystemtypen unterstützen diese Optionen nicht unbedingt.

Für *spezifische_optionen* ist eine Liste von durch Komma getrennten Unteroptionen und/oder Paaren von Schlüsselwortattributen anzugeben, die von dem *dstyp*-spezifischen Modul des Kommandos interpretiert werden.

Die folgenden Optionen stehen zur Verfügung:

- F Gibt den Typ *dstyp* des Dateisystems an, das angelegt werden soll. Wird der Dateisystemtyp hier nicht angegeben, dann muß er aus der Tabelle */etc/vfstab* bestimmbar sein; dazu muß *gerät* zu einem Eintrag in dieser Tabelle passen.
- V Bewirkt, daß die ganze Kommandozeile am Bildschirm ausgegeben wird. Das Kommando wird dabei nicht ausgeführt. Die Kommandozeile wird generiert anhand der angegebenen Optionen und Argumente sowie anhand von Informationen aus der Datei */etc/vfstab*. Diese Option wird dazu verwendet, die Kommandozeile auszuwerten und zu überprüfen.
- m Gibt die Kommandozeile aus, mit der das Dateisystem erstellt wurde. Das Dateisystem muß bereits vorhanden sein. Diese Option ermöglicht es festzustellen, mit welchem Kommando das Dateisystem erstellt wurde. Sie kann nicht zusammen mit den Optionen *aktuelle_optionen*, *spezifische_optionen* oder *operanden* verwendet werden.
- o Gibt *dstyp*-spezifische Optionen an.

HINWEIS

Dieses Kommando wird möglicherweise nicht von allen Dateisystemtypen unterstützt.

mkfs(1M)

mkfs(1M)

DATEIEN

/etc/vfstab

Liste der Standardparameter für jedes Dateisystem, d.h. Beschreibung der einzelnen Dateisysteme

SIEHE AUCH

makefsys(1M), vfstab(4).

Handbucheinträge für die *dstyp*-spezifischen Module von `mkfs`.

mkfs (1M)

(bfs)

mkfs (1M)

BEZEICHNUNG

mkfs (bfs) – erstellt ein Boot-Dateisystem

ÜBERSICHT

mkfs [-F bfs] *gerät* *blöcke* [*i-nodes*]

BESCHREIBUNG

Mit mkfs wird ein Boot-Dateisystem erstellt. Hierbei handelt es sich um ein zusammenhängendes lineares (nicht hierarchisches) Dateisystem, in dem die bootbaren Programme und die für das Booten notwendigen Datendateien gespeichert werden.

Das Argument *gerät* gibt die Gerätedatei an, also die Slice, für die das Dateisystem erstellt werden soll. Mit dem Argument *blöcke* wird die Größe des Dateisystems angegeben. Als Blockgröße gilt automatisch 512 Byte.

Mit dem Argument *i-nodes* wird die Anzahl der Dateien angegeben, die in dem Dateisystem angelegt werden können.

HINWEIS

Es empfiehlt sich nicht, das erstellte Boot-Dateisystem für allgemeine Zwecke einzusetzen.

SIEHE AUCH

Weitere Informationen über das Boot-Dateisystem finden Sie im *Leitfaden für Systemverwalter*.

BEZEICHNUNG

mkfs (s5) – erstellt ein s5-Dateisystem

ÜBERSICHT

```
mkfs [-F s5] [gen_opt] gerät
mkfs [-F s5] [gen_opt] [-b block_größe] gerät blöcke[:i-nodes] [zw_raum blöcke/zykl]
mkfs [-F s5] [gen_opt] [-b block_größe] gerät proto [zw_raum blöcke/zykl]
```

BESCHREIBUNG

gen_opt sind Optionen, die von dem generischen mkfs-Kommando unterstützt werden.

mkfs legt für die Gerätedatei *gerät* anhand der übrigen Argumente in der Kommandozeile ein s5-Dateisystem an, das über ein root- und ein lost+found-Verzeichnis verfügt.

Die folgenden Optionen stehen zur Verfügung:

-F s5 Gibt s5 als Dateisystemtyp an.

-b *blockgröße*

Gibt die logische Blockgröße für das Dateisystem an. Die logische Blockgröße ist die Anzahl der Byte, die von dem Betriebssystem in einer einzigen E/A-Operation gelesen oder geschrieben werden. Gültige Werte für *blockgröße* sind 512, 1024 und 2048. Der Standardwert ist 1024.

Wenn das zweite Argument für mkfs eine Ziffernfolge ist, gilt als Größe des Dateisystems der als Dezimalzahl interpretierte Wert des Arguments *blöcke*. Dies ist die Anzahl der *physischen* (512 Byte) Blöcke, die das Dateisystem umfassen soll. Wird die Anzahl der I-Nodes nicht angegeben, gilt als Standardwert ungefähr die Anzahl der *logischen* Blöcke geteilt durch 4. mkfs erstellt ein Dateisystem mit einem einzigen leeren Verzeichnis, lost+found. Der Block für das Boot-Programm (Block Null) wird nicht initialisiert.

Wenn als zweites Argument der Name einer Datei angegeben wird, die geöffnet werden kann, geht mkfs davon aus, daß es sich hierbei um eine Prototypdatei, *proto*, handelt und entnimmt die Anweisungen aus dieser Datei. Die Prototypdatei enthält durch Leer- oder Zeilenendezeichen getrennte Token. Im folgenden ist ein Beispiel für eine Prototypspezifikation abgebildet (die Zeilennummern wurden hinzugefügt, um die Erläuterung übersichtlicher zu machen):

```
1.      4872 110
2.      d--777 3 1
3.      usr  d--777 3 1
4.          sh   ---755 3 1 /sbin/sh
5.          ken  d--755 6 1
6.          $
7.          b0   b--644 3 1 0 0
8.          c0   c--644 3 1 0 0
9.          slnk 1--777 2 2 /var/tmp
10.         $
11.      $
```

Zeile 1 gibt die Anzahl der *physischen* (512 Byte) Blöcke an, die das Dateisystem belegen soll, und die Anzahl der I-Nodes im Dateisystem.

Die Zeilen 2-9 geben *mkfs* an, welche Dateien und Verzeichnisse in dieses Dateisystem aufgenommen werden sollen.

Zeile 2 gibt das Root-Verzeichnis an.

Die Zeilen 3-5 und 7-9 geben andere Verzeichnisse und Dateien an.

Zeile 9 gibt den symbolischen Verweis (link) *slnk* an, der in */usr* steht und */var/tmp* enthält.

Das Zeichen \$ in Zeile 6 weist *mkfs* an, den aktuellen Zweig des Dateisystems zu verlassen und vom nächsthöheren Verzeichnis aus fortzufahren. Das \$ in den Zeilen 10 und 11 beendet den Prozeß, da keine weiteren Spezifikationen folgen.

Die Dateispezifikationen geben den Modus, die Benutzernummer, die Gruppennummer und den anfänglichen Inhalt der Datei an. Die für das Inhaltsfeld gültige Syntax hängt vom ersten Zeichen des Modus ab.

Der Modus für eine Datei wird mit einer aus 6 Zeichen bestehenden Zeichenkette angegeben. Das erste Zeichen gibt den Dateityp an. Mögliche Zeichen sind -, b, c, d bzw. l, um reguläre Dateien, blockorientierte Gerätedateien, zeichenorientierte Gerätedateien, Verzeichnisse bzw. Dateien mit symbolischen Verweisen zu spezifizieren. Das zweite Zeichen der Moduszeichenkette ist entweder u oder -. Hiermit wird angegeben, ob das s-Bit für Benutzer gesetzt wird. Mit dem dritten Zeichen, g oder -, wird angegeben, ob das s-Bit für Gruppe gesetzt wird. Der restliche Modus besteht aus einer dreistelligen Oktalzahl, die die Lese-, Schreib- und Ausführungsrechte für Eigentümer, Gruppe und Andere angeben (siehe *chmod(1)*).

Zwei Token in der Form von Dezimalzahlen folgen auf den Modus. Sie geben die Benutzer- und Gruppennummer des Eigentümers der Datei an.

Wenn es sich bei der Datei um eine reguläre Datei handelt, kann das nächste Token der Spezifikation ein Pfadname sein, der angibt, woher der Inhalt und die Größe kopiert werden. Ist die Datei eine blockorientierte oder zeichenorientierte Gerätedatei, folgen zwei Dezimalzahlen, die die Geräte- und die Geräteklassennummer angeben. Handelt es sich bei der Datei um ein Verzeichnis, legt *mkfs* die Einträge . und .. an und liest dann eine Liste von Namen und (rekursiv) Dateispezifikationen für die Einträge in dem Verzeichnis. Wie oben bereits erwähnt, wird die Suche mit dem Token \$ beendet.

Das Argument *zw_raum blöcke/zyl* gibt in beiden Formen des Kommandos den Blockzwischenraum und die Anzahl der Blöcke je Zylinder an. Werden der *zw_raum* und die *blöcke/zyl* nicht angegeben oder als ungültige Werte erkannt, gilt für die Größe des Zwischenraums der Standardwert 7 und für die Anzahl der Blöcke je Zylinder der Wert 400.

HINWEIS

Bei einer Prototypdatei ist es nicht möglich, "hard links" anzugeben.

Es können höchstens 65500 I-Nodes konfiguriert werden.

mkfs(1M)

(s5)

mkfs(1M)

DATEIEN

/etc/vfstab

SIEHE AUCH

generisches mkfs(1M), dir(4), fs(4).
chmod(1) in den *Kommandos*.

BEZEICHNUNG

mkfs (ufs) – erstellt ein ufs-Dateisystem

ÜBERSICHT

```
mkfs [ -F ufs ] [gen_opt] gerät
mkfs [ -F ufs ] [gen_opt] [-o spezifische_optionen] gerät größe
mkfs [ -F ufs ] [gen_opt] gerät größe nsect ntrack bsize
      fragsize cgsiz e free rps nbpi opt apc gap
```

BESCHREIBUNG

gen_opt sind Optionen, die von dem generischen `mkfs`-Kommando unterstützt werden.

Die spezifischen Optionen, außer `N`, können Sie auf zwei verschiedene Arten angeben. Entweder müssen die Werte aller Optionen in der richtigen Reihenfolge aufgeführt werden. Die Werte sind durch Leerzeichen voneinander zu trennen. Ein möglicher Aufruf ist z.B.

```
mkfs /dev/xxx 3000 12 15 4096 1024 16 ... 4
```

Oder Sie müssen, um nur bestimmte Optionen mit Werten zu belegen, diese nach `-o` als Zeichenkette spezifizieren. Die Zeichenkette muß von Anführungszeichen " eingeschlossen sein. Vor jedem Wert muß der Name der Option, durch ein Gleichheitszeichen getrennt, stehen. Ein möglicher Aufruf ist dann z.B.

```
mkfs -o 'ntrack=15,nsect=12,...' /dev/xxx 3000
```

`mkfs` legt ein ufs-Dateisystem in der Gerätedatei *gerät* an, es sei denn, die Option `'-o N'` wurde angegeben. Die numerische *größe* gibt die Anzahl der Sektoren im Dateisystem an. `mkfs` erstellt ein Dateisystem mit einem Root- und einem `lost+found`-Verzeichnis (siehe `fsck(1M)`). Die Anzahl der I-Nodes wird als Funktion der Dateisystemgröße errechnet.

Die folgenden Optionen stehen zur Verfügung:

`-F ufs`

Gibt `ufs` als Dateisystemtyp an.

`-o` Gibt Optionen an, die für das ufs-Dateisystem spezifisch sind. Die folgenden Optionen stehen zur Verfügung:

`N` Das Dateisystem wird nicht in der Gerätedatei *gerät* angelegt. Diese Option liefert alle Informationen, die zum Erstellen eines Dateisystems erforderlich sind, legt es aber nicht an.

`nsect` Die Anzahl der Sektoren je Spur auf der Festplatte. Der Standardwert ist 18.

`ntrack` Die Anzahl der Spuren je Zylinder auf der Festplatte. Der Standardwert ist 9.

`bsize` Die Größe des Primärblocks für Dateien im Dateisystem. Es muß eine Zweierpotenz sein, und zwar entweder 4096 oder 8192 (der Standardwert).

fragsize

Die Fragmentgröße für Dateien im Dateisystem. Die *fragsize* gibt den kleinsten Festplattenspeicherplatz an, der einer Datei zugewiesen wird. Es muß eine Zweierpotenz im Bereich zwischen 512 und 8192 sein. Der Standardwert ist 1024.

cgsiz

Die Anzahl der Plattenzylinder je Zylindergruppe. Diese Zahl muß zwischen 1 und 32 liegen. Der Standardwert ist 16.

free

Der kleinste zulässige Prozentsatz an freiem Speicherplatz auf der Festplatte. Sobald das Dateisystem diese Grenze erreicht, darf nur noch ein privilegierter Benutzer Plattenblöcke zuweisen. Der Standardwert ist 10%.

rps

Die Rotationsgeschwindigkeit der Festplatte in Umdrehungen pro Sekunden. Der Standardwert ist 60.

nbpi

Die Anzahl der Byte, für die ein I-Node-Block zugewiesen ist. Dieser Parameter ist zur Zeit auf einen I-Node-Block für jeweils 2048 Byte gesetzt.

opt

Ermöglicht es dem Benutzer auszuwählen, ob die Zugriffszeit oder der Speicherplatz optimiert werden soll. Mit *s* wird die Optimierung des Speicherplatzes ausgewählt, mit *t* die Optimierung der Zugriffszeit. Der Standardwert ist *t*.

apc

Die Anzahl der Ersatzsektoren pro Zylinder (nur SCSI-Geräte). Der Standardwert ist 0.

gap

Die Zeit (in Millisekunden), die erwartungsgemäß für die Abarbeitung eines Übertragungsabschluß-Interrupts und der Initialisierung einer neuen Übertragung auf derselben Festplatte benötigt wird. Hiermit wird bestimmt, wie groß die Rotationszwischenräume zwischen den Blöcken einer Datei sein müssen. Der Standardwert ist 4 bei der MX500 und der WX200, 1 bei der MX300.

HINWEIS

Der Wert des Operanden *nbpi* in der Ausgabe von *mkfs -m* ist immer 2048, selbst wenn das Dateisystem mit einem anderen Wert erstellt wurde.

SIEHE AUCH

fsck(1M), generisches *mkfs(1M)*, *dir(4)*, *ufs(4)*.

BEZEICHNUNG

mknod – legt eine Gerätedatei an

ÜBERSICHT

```
/sbin/mknod name [b | c ] geräteklasse gerätenummer  
/sbin/mknod name p  
/sbin/mknod name m  
/sbin/mknod name s
```

BESCHREIBUNG

mknod erstellt einen Verzeichniseintrag für eine Gerätedatei.

In der ersten Form des Kommandos gibt *name* die zu erstellende Gerätedatei an. Das zweite Argument ist b, um eine blockorientierte Gerätedatei, oder c, um eine zeichenorientierte Gerätedatei anzugeben. Die letzten beiden Argumente sind Zahlen, mit denen die *geräteklasse* und die *gerätenummer* angegeben werden. Diese können in Dezimal- oder Oktalschreibweise eingegeben werden. Bei Oktalschreibweise muß eine führende Null angegeben werden. Die Zuweisung der Geräteklassennummer ist systemspezifisch. Diese Form des Kommandos kann nur ein privilegierter Benutzer ausführen.

Mit der zweiten Form des Kommandos wird eine FIFO-Datei (benannte Pipe) erstellt.

Mit der dritten Form werden gemeinsam benutzte XENIX-Speicher-Handles erstellt.

Mit der vierten Form werden XENIX-Semaphor-Handles erstellt.

SIEHE AUCH

mknod(2) im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

mkpart – Dienstprogramm zur Festplattenverwaltung

ÜBERSICHT

```
/etc/mkpart [ -f dateiname ] [ -p slice ] ... [ -P slice ] ... [ -b ]
[ -B dateiname ] [ -A sektor ] ... [ -i ] [ -x datei ]
[ -t [ vpa ] ] gerät
```

BESCHREIBUNG

Dieses Programm ermöglicht es dem Systemverwalter, die Datenstrukturen anzuzeigen und zu ändern, die die Aufteilung der Platte in Slices und Partitionen beschreiben. Die Strukturen informieren über Anzahl, Größe und Typ der Partitionen, sowie die physikalischen Eigenschaften des Plattenlaufwerks selbst.

Die Informationen sind in einer Datei abgelegt, die der Benutzer ändern kann. Diese Datei enthält zwei Arten von Einträgen, die jeweils aus mehreren Zeilen bestehen. Die eine Zeilengruppe beschreibt die Plattenpartition, die andere die Plattengeräte. Zeilen aus einer Zeilengruppe können sich auf Zeilen einer anderen Gruppe desselben Typs beziehen, so daß allgemeine Geräte- oder Partitionstypen maßgeschneidert werden können. Standardmäßig stehen die Informationen in der Datei `/etc/partitions`.

Der erforderliche Parameter *gerät* gibt an, welcher Eintrag (d.h. welche Zeilengruppe in der Informationsdatei) und damit welches Gerät bearbeitet werden soll. *gerät* ist ein symbolischer Name, wie z.B. `disk01`, der in der Informationsdatei verwendet wird, um ein Gerät oder eine Partition zu bezeichnen.

Die folgenden Optionen stehen für `mkpart` zur Verfügung:

- `-f dateiname` Bezeichnet die Datei, in der die Gerätespezifikationen und Informationen stehen. Wird die Option nicht angegeben, gilt standardmäßig `/etc/partitions`.
- `-p slice` Löscht eine Slice aus dem VTOC des angegebenen Geräts. *slice* ist die Nummer der Slice. Diese Nummer steht in der Informationsdatei in der Zeilengruppe, die zu der Slice gehört, die gelöscht werden soll (s.u.). Es werden keine Attributvergleiche vorgenommen.
Hinweis: Ersatzslices können nicht gelöscht werden.
- `-P slice` Fügt eine Slice zum VTOC des angegebenen Geräts hinzu. *slice* bezeichnet eine Zeilengruppe, die andere Zeilengruppen enthält und/oder sich auf solche bezieht, in denen alle notwendigen Parameter für eine VTOC-Slice stehen.
- `-b` Bewirkt, daß nur das Ladeprogramm aktualisiert wird, es sei denn, weitere Optionen werden angegeben.
- `-B dateiname` Gibt ein anderes Ladeprogramm an (vgl. Parameter `boot` der Gerätspezifikation).
- `-A sektor` Markiert den angegebenen Sektor als fehlerhaft und weist, falls möglich, einen Ersatzsektor zu. *sektor* ist die, von Null (dem Anfang der Festplatte) ausgehend gezählte, absolute Sektorennummer. Wenn der Zylinder, der Kopf und der (von Null ausgehend gezählte) Sektor in der Spur bekannt sind, kann eine bestimmte Sektorennummer mit der Formel

Zylinder * (Sektoren-pro-Spur * Köpfe-pro-Zylinder) + Kopf * (Sektoren-pro-Spur) + Sektor
berechnet werden. Für SCSI-Platten sind die Anzahl Köpfe, Zylinder usw. nicht von Bedeutung. Beim Aufruf von mkpart wird die Blocknummer angegeben, die der Treiber bei einem Zugriff als fehlerhaft erkannt und gemeldet hat.

-i Initialisiert das VTOC auf dem Laufwerk mit Standardwerten, wobei alle eventuell vorhandenen Informationen über Slices und fehlerhafte Sektoren gelöscht werden. Dies stellt die einzige Möglichkeit dar, eine Ersatzslice zu löschen. Mit dieser Option kann ein Laufwerk neu initialisiert werden, auf dem sich veraltete oder falsche VTOC-Daten befinden.

-x *datei* Mit dieser Option werden alle Zeilengruppen, die das angegebene Gerät betreffen, in die Datei *datei* geschrieben.
Hinweis: Die Marken in der Datei sind Pseudo-Namen, mit denen die Slice identifiziert wird.

-t [*vpa*] Listet das aktuelle VTOC auf. Die Unterparameter benennen die auszugebenden Informationen: a - Ersatzsektoren, p - Partitionen und v - VTOC, sowie damit verbundene Strukturen. -ta listet die ausgelagerten Blöcke in folgendem Format auf:

bn = 4264, cn = —, tn = —, sn = — M

.

.

bn = 10135, cn = —, tn = —, sn = — U

bn = 10137, cn = —, tn = —, sn = — U

bn = 11194, cn = —, tn = —, sn = — U

Ein M am Zeilenende bedeutet "manufactured", ein U "user". Damit wird unterschieden zwischen Blöcken, die der Hersteller bereits ausgelagert hat und solchen, die der Benutzer ausgelagert hat.

Die Datei `/etc/partitions` besteht aus Zeilengruppen, die durch Leerzeilen getrennt werden. Als Kommentar gilt ein Text, der zwischen dem Zeichen '#' und einem Neuezeile-Zeichen steht. Zeilengruppen fangen mit einem Namen gefolgt von Doppelpunkt ':' an. Danach folgt eine Liste von durch Komma getrennten Parametern. Jeder Parameter besteht aus einem Schlüsselwort gefolgt von einem '=' und einem Wert. Der Wert kann, je nach Schlüsselwort, eine Zahl, der Name einer anderen Zeilengruppe, ein in doppelte Anführungszeichen eingeschlossene Zeichenkette oder eine in Klammern eingeschlossene Liste von durch Komma getrennten Zahlen oder Zahlenbereichen sein. Zahlen können, wie in C, in der Form von dezimalen, oktalen oder hexadezimalen Konstanten angegeben werden.

Zeilengruppen mit Gerätespezifikationen können die folgenden Parameter enthalten:

usedevice = *name* Bewirkt, daß die Parameter der angegebenen Zeilengruppe in die Gerätedefinition aufgenommen werden.

<code>boot = <i>zeichenkette</i></code>	Gibt an, daß die Zeichenkette der Dateiname eines Urladeprogramms ist, das auf der Festplatte installiert werden soll (gilt nicht für SCSI-Platten am ADP32 Controller).
<code>device = <i>zeichenkette</i></code>	Gibt den Dateinamen des zeichenorientierten Special Device für die Festplatte an.
<code>heads = <i>anzahl</i></code>	Gibt die Anzahl der Spuren je Zylinder auf der Festplatte an.
<code>cyls = <i>anzahl</i></code>	Gibt die Anzahl der Zylinder auf der Festplatte an.
<code>sectors = <i>anzahl</i></code>	Gibt die Anzahl der Sektoren je Spur an.
<code>bpsec = <i>anzahl</i></code>	Gibt die Anzahl der Byte je Sektor an.
<code>badsec = <i>nummernliste</i></code>	Listet die bekannten fehlerhaften Sektoren auf. Die Liste wird ergänzt, wenn solche Sektoren in der Kommandozeile angegeben oder während einer Oberflächenprüfung gefunden werden.

Zeilengruppen mit Informationen über Partitionen können die folgenden Parameter haben:

<code>usepart = <i>name</i></code>	Bezieht sich auf eine andere Partitionszeilengruppe.
<code>partition = <i>nummer</i></code>	Gibt die Nummer des Eintrags dieser Partition im VTOC an.
<code>tag = <i>tagname</i></code>	Ein Partitionstag gibt den Zweck der Partition an. Die <i>tagnamen</i> sind reservierte Wörter, die zur Identifizierung verwendet werden: BACKUP bezeichnet die ganze Festplatte. ROOT ist eine Partition für ein Root-Dateisystem. BOOT ist eine Partition für den Urlader. SWAP ist eine Partition, die kein Dateisystem enthält. USR ist eine Partition, die ein Dateisystem enthält. OTHER ist eine Partition, die das UNIX-System nicht verwalten kann, wie z.B. den Speicherbereich für MS-DOS.
<code>perm = <i>berechtigung</i></code>	Spezifiziert einen Berechtigungstyp für die Partition. Zugriffsberechtigungen schließen sich nicht gegenseitig aus. RO gibt an, daß die Partition schreibgeschützt ist. Normalerweise wird trotz der standardmäßigen Dateizugriffsberechtigungen des UNIX-Systems der Schreibzugriff gewährt. NOMOUNT bewirkt, daß der Treiber das Dateisystem, das in der Partition steht, nicht einhängen kann. VALID zeigt an, daß die Partition gültige Daten enthält. Jede Partition, die mit der Option -A hinzugefügt wird, wird als VALID markiert.

`start = nummer`

ist die Nummer des Anfangssektors der Partition.
Die erste Slice sollte innerhalb einer Partition frühestens auf der zweiten Spur dieser Partition beginnen. Ausgenommen ist dabei die Slice 0, die die gesamte Partition umfaßt. In der ersten Spur einer Partition werden die `pinfo`- und die `vtoc`-Struktur für diese Partition untergebracht und diese sollten nicht versehentlich durch Zugriffe auf andere Slices der Partition überschrieben und damit vernichtet werden.

`size = nummer`

Gibt die Größe der Partition in Sektoren an.

Wenn `mkpart` ausgeführt wird, versucht es zunächst, die `fdisk`-Tabelle, das VTOC und die Tabelle mit den alternativen Sektoren zu lesen. Wenn eine dieser Strukturen ungültig ist oder nicht gelesen werden kann oder wenn die Option `-i` angegeben wird, werden die internen Tabellen mit Standardwerten für das spezifizierte Gerät initialisiert. Diese Werte werden der Gerätezeilengruppe in der Partitionsdatei entnommen. Danach werden, falls erforderlich, Partitionen gelöscht oder hinzugefügt. Im nächsten Schritt werden alle fehlerhaften Sektoren, die in der Partitionsdatei aufgeführt sind oder in der Kommandozeile mit der Option `-A` angegeben werden, in die Tabelle mit den alternativen Sektoren aufgenommen. Zuletzt wird das Ladeprogramm in Zylinder 0 geschrieben. Wenn nicht nur `-b` als Parameter angegeben wurde, werden das aktualisierte VTOC und alternative Tabellen geschrieben, und der Plattentreiber wird angewiesen, die Tabellen erneut zu lesen, wenn das Laufwerk das nächste Mal aktiviert wird. Wurde nur `-t` angegeben, wird nur eine Auflistung erstellt, aber keine Aktualisierung vorgenommen.

DATEIEN

```
/etc/partitions
/etc/boot
/dev/rdisk/shd?t?s0
```

HINWEIS

Zur Zeit wird kaum eine Konsistenzprüfung vorgenommen. Es wird z.B. nicht überprüft, ob die `fdisk`-Partitionstabelle mit den Partitionen des UNIX-Systems im VTOC konsistent ist.

BEZEICHNUNG

mkpart – Dienstprogramm zur Festplattenverwaltung

ÜBERSICHT

```
/etc/mkpart [ -f dateiname ] [ -p slice ] ... [ -P slice ] ... [ -b ]
[ -B dateiname ] [ -A sektor ] ... [ -V ] [ -v ] [ -i ] [ -x datei ]
[ -t [ vpa ] ] gerät
```

```
/etc/mkpart -F verzahnung zeichorient gerät
```

BESCHREIBUNG

Dieses Programm ermöglicht es dem Systemverwalter, die Datenstrukturen anzuzeigen und zu ändern, die die Aufteilung der Platte in Slices und Partitionen beschreiben. Die Strukturen informieren über Anzahl, Größe und Typ der Partitionen, sowie die physikalischen Eigenschaften des Plattenlaufwerks selbst.

Die Informationen sind in einer Datei abgelegt, die der Benutzer ändern kann. Diese Datei enthält zwei Arten von Einträgen, die jeweils aus mehreren Zeilen bestehen. Die eine Zeilengruppe beschreibt die Plattenpartition, die andere die Plattengeräte. Zeilen aus einer Zeilengruppe können sich auf Zeilen einer anderen Gruppe desselben Typs beziehen, so daß allgemeine Geräte- oder Partitionstypen maßgeschneidert werden können. Standardmäßig stehen die Informationen in der Datei `/etc/partitions`.

Der erforderliche Parameter *gerät* gibt an, welcher Eintrag (d.h. welche Zeilengruppe in der Informationsdatei) und damit welches Gerät bearbeitet werden soll. *gerät* ist ein symbolischer Name, wie z.B. `disk01`, der in der Informationsdatei verwendet wird, um ein Gerät oder eine Partition zu bezeichnen.

Die folgenden Optionen stehen für `mkpart` zur Verfügung:

- `-f dateiname` Bezeichnet die Datei, in der die Gerätespezifikationen und Informationen stehen. Wird die Option nicht angegeben, gilt standardmäßig `/etc/partitions`.
- `-p slice` Löscht eine Slice aus dem VTOC des angegebenen Geräts. *slice* ist die Nummer der Slice. Diese Nummer steht in der Informationsdatei in der Zeilengruppe, die zu der Slice gehört, die gelöscht werden soll (s.u.). Es werden keine Attributvergleiche vorgenommen.
Hinweis: Ersatzslices können nicht gelöscht werden.
- `-P slice` Fügt eine Slice zum VTOC des angegebenen Geräts hinzu. *slice* bezeichnet eine Zeilengruppe, die andere Zeilengruppen enthält und/oder sich auf solche bezieht, in denen alle notwendigen Parameter für eine VTOC-Slice stehen.
- `-b` Bewirkt, daß nur das Ladeprogramm aktualisiert wird, es sei denn, weitere Optionen werden angegeben.
- `-B dateiname` Gibt ein anderes Ladeprogramm an (vgl. Parameter `boot` der Gerätespezifikation).
- `-A sektor` Markiert den angegebenen Sektor als fehlerhaft und weist, falls möglich, einen Ersatzsektor zu. *sektor* ist die, von Null (dem Anfang der Festplatte) ausgehend gezählte, absolute Sektorennummer. Wenn der Zylinder, der Kopf und der (von Null ausgehend gezählte) Sektor

in der Spur bekannt sind, kann eine bestimmte Sektorennummer mit der Formel

$\text{Zylinder} * (\text{Sektoren-pro-Spur} * \text{Köpfe-pro-Zylinder}) + \text{Kopf} * (\text{Sektoren-pro-Spur}) + \text{Sektor}$
berechnet werden. Für SCSI-Platten sind die Anzahl Köpfe, Zylinder usw. nicht von Bedeutung. Beim Aufruf von `mkpart` wird die Blocknummer angegeben, die der Treiber bei einem Zugriff als fehlerhaft erkannt und gemeldet hat.

- V Bewirkt, daß eine vollständige Oberflächenprüfung durchgeführt wird. Hierbei wird ein Datenmuster (zur Zeit 0xe5 in jedem Byte) in jeden Sektor auf der Platte geschrieben, und danach wird der Sektor gelesen. Alle Fehler werden festgehalten, und die fehlerhaften Sektoren werden, falls möglich, zu der Tabelle mit den Ersatzsektoren hinzugefügt.
- v Bewirkt eine nicht destruktive Oberflächenprüfung. Hierbei werden die Sektoren auf der Festplatte lediglich gelesen. Fehlerhafte Sektoren werden wie bei dem oben genannten Kommando festgehalten.
- i Initialisiert das Datenträgerinhaltsverzeichnis (VTOC) auf dem Laufwerk mit Standardwerten, wobei alle eventuell vorhandenen Informationen über Slices und fehlerhafte Sektoren gelöscht werden. Dies stellt die einzige Möglichkeit dar, eine Ersatzslice zu löschen. Mit dieser Option kann ein Laufwerk neu initialisiert werden, auf dem sich veraltete oder falsche VTOC-Daten befinden.
- x *datei* Mit dieser Option werden alle Zeilengruppen, die das angegebene Gerät betreffen, in die Datei *datei* geschrieben.
Hinweis: Die Marken in der Datei sind Pseudo-Namen, mit denen die Slice identifiziert wird.
- t [*vpa*] Listet das aktuelle VTOC auf. Die Unterparameter benennen die auszugebenden Informationen: a - Ersatzsektoren, p - Partitionen und v - VTOC, sowie damit verbundene Strukturen.

Die Datei `/etc/partitions` besteht aus Zeilengruppen, die durch Leerzeilen getrennt werden. Als Kommentar gilt ein Text, der zwischen dem Zeichen '#' und einem Neuezeile-Zeichen steht. Zeilengruppen fangen mit einem Namen gefolgt von Doppelpunkt ':' an. Danach folgt eine Liste von durch Komma getrennten Parametern. Jeder Parameter besteht aus einem Schlüsselwort gefolgt von einem '=' und einem Wert. Der Wert kann, je nach Schlüsselwort, eine Zahl, der Name einer anderen Zeilengruppe, ein in doppelte Anführungszeichen eingeschlossene Zeichenkette oder eine in Klammern eingeschlossene Liste von durch Komma getrennten Zahlen oder Zahlenbereichen sein. Zahlen können, wie in C, in der Form von dezimalen, oktalen oder hexadezimalen Konstanten angegeben werden.

Zeilengruppen mit Gerätespezifikationen können die folgenden Parameter enthalten:

- `usedevice = name` Bewirkt, daß die Parameter der angegebenen Zeilengruppe in die Gerätedefinition aufgenommen werden.

<code>boot = zeichenkette</code>	Gibt an, daß die Zeichenkette der Dateiname eines Urladeprogramms ist, das auf der Festplatte installiert werden soll (gilt nicht für SCSI-Platten am ADP32 Controller).
<code>device = zeichenkette</code>	Gibt den Dateinamen des zeichenorientierten Special Device für die Festplatte an.
<code>heads = anzahl</code>	Gibt die Anzahl der Spuren je Zylinder auf der Festplatte an.
<code>cyls = anzahl</code>	Gibt die Anzahl der Zylinder auf der Festplatte an.
<code>sectors = anzahl</code>	Gibt die Anzahl der Sektoren je Spur an.
<code>bpssec = anzahl</code>	Gibt die Anzahl der Byte je Sektor an.
<code>vtocsec = nummer</code>	Gibt die Sektorennummer an, die für das Datenträgerinhaltsverzeichnis verwendet werden soll.
<code>altsec = nummer</code>	Gibt den Sektor an, der für die Tabelle mit den Ersatzblöcken zu verwenden ist.
<code>badsec = nummernliste</code>	Listet die bekannten fehlerhaften Sektoren auf. Die Liste wird ergänzt, wenn solche Sektoren in der Kommandozeile angegeben oder während einer Oberflächenprüfung gefunden werden.

Zeilengruppen mit Informationen über Partitionen können die folgenden Parameter haben:

<code>usepart = name</code>	Bezieht sich auf eine andere Partitionszeilengruppe.
<code>partition = nummer</code>	Gibt die Nummer des Eintrags dieser Partition im VTOC an.
<code>tag = tagname</code>	Ein Partitionstag gibt den Zweck der Partition an. Die <i>tagnamen</i> sind reservierte Wörter, die zur Identifizierung verwendet werden: BACKUP bezeichnet die ganze Festplatte. ROOT ist eine Partition für ein Root-Dateisystem. BOOT ist eine Partition für den Urlader. SWAP ist eine Partition, die kein Dateisystem enthält. USR ist eine Partition, die ein Dateisystem enthält. ALTS enthält alternative Sektoren, auf die der Treiber fehlerhafte Sektoren neu abbildet. Zur Zeit werden höchstens 62 alternative Sektoren unterstützt. OTHER ist eine Partition, die das UNIX-System nicht verwalten kann, wie z.B. den Speicherbereich für MS-DOS.
<code>perm = berechtigung</code>	Spezifiziert einen Berechtigungstyp für die Partition. Zugriffsberechtigungen schließen sich nicht gegenseitig aus. RO gibt an, daß die Partition schreibgeschützt ist. Normalerweise wird trotz der standardmäßigen Dateizugriffsberechtigungen des UNIX-Systems der Schreibzugriff

gewährt.

NOMOUNT bewirkt, daß der Treiber das Dateisystem, das in der Partition steht, nicht einhängen kann.

VALID zeigt an, daß die Partition gültige Daten enthält. Jede Partition, die mit der Option -A hinzugefügt wird, wird als VALID markiert.

`start = nummer`

ist die Nummer des Anfangssektors der Partition.

Die erste Slice sollte innerhalb einer Partition frühestens auf der zweiten Spur dieser Partition beginnen. Ausgenommen ist dabei die Slice 0, die die gesamte Partition umfaßt. In der ersten Spur einer Partition werden die `pdinfo`- und die `vtoc`-Struktur für diese Partition untergebracht und diese sollten nicht versehentlich durch Zugriffe auf andere Slices der Partition überschrieben und damit vernichtet werden.

`size = nummer`

Gibt die Größe der Partition in Sektoren an.

Wenn `mkpart` ausgeführt wird, versucht es zunächst, die `fdisk`-Tabelle, das VTOC und die Tabelle mit den alternativen Sektoren zu lesen. Wenn eine dieser Strukturen ungültig ist oder nicht gelesen werden kann oder wenn die Option `-i` angegeben wird, werden die internen Tabellen mit Standardwerten für das spezifizierte Gerät initialisiert. Diese Werte werden der Gerätezeilengruppe in der Partitionsdatei entnommen. Wird die Option `-F` angegeben, wird das Gerät formatiert. Wird die Option `-v` oder `-V` angegeben, wird die entsprechende Oberflächenprüfung durchgeführt. Danach werden, falls erforderlich, Partitionen gelöscht oder hinzugefügt. Im nächsten Schritt werden alle fehlerhaften Sektoren, die in der Partitionsdatei aufgeführt sind, bei der Oberflächenprüfung gefunden wurden oder in der Kommandozeile mit der Option `-A` angegeben werden, in die Tabelle mit den alternativen Sektoren aufgenommen. Es ist zu beachten, daß alternative Partitionen vorhanden sein müssen, damit eine Markierung der fehlerhaften Sektoren erfolgen kann, da den fehlerhaften Sektoren an dieser Stelle alternative Sektoren zugewiesen werden. Zuletzt wird das Ladeprogramm in Zylinder 0 geschrieben, in dem die aktive `fdisk`-Partition des UNIX-Systems anfängt. Wenn nicht nur `-b` als Parameter angegeben wurde, werden das aktualisierte VTOC und alternative Tabellen geschrieben, und der Plattentreiber wird angewiesen, die Tabellen erneut zu lesen, wenn das Laufwerk das nächste Mal aktiviert wird. Wurde nur `-t` angegeben, wird nur eine Auflistung erstellt, aber keine Aktualisierung vorgenommen.

`-F` *verzahnung*

Bewirkt, daß das ganze Gerät hardwaremäßig formatiert wird. Hierbei werden alle Sektorenköpfe in allen Spuren auf der Platte neu geschrieben. Dies ermöglicht später normale Lese- und Schreibzugriffe. *verzahnung* gibt, in physischen Sektoren, den Abstand zwischen den aufeinanderfolgenden logischen Sektoren an. Normalerweise gilt der Wert 1 für Spur-Cache-Controller und ein Wert zwischen 3 und 4 für Standard-Controller. Bei dieser Option muß als Gerät ein zeichenorientiertes UNIX-Systemgerät angegeben werden. Die Option `-F` schließt alle anderen Optionen aus und sollte daher allein verwendet werden.

DATEIEN

```
/etc/partitions  
/etc/boot  
/dev/rdisk/shd?t?s0
```

HINWEIS

Zur Zeit wird kaum eine Konsistenzprüfung vorgenommen. Es wird z.B. nicht überprüft, ob die `fdisk`-Partitionstabelle mit den Partitionen des UNIX-Systems im VTOC konsistent ist. Wenn eine `fdisk`-Partition des Betriebssystems DOS vom Zylinder 0 gestartet wird, überschreibt DOS das VTOC des UNIX-Systems.

BEZEICHNUNG

monitor – Aktivität des Systems darstellen

ÜBERSICHT

```
/sbin/monitor[ -c n ][ -i n ][ -p n ][ -t ][ -f ]
```

BESCHREIBUNG

monitor gibt verschiedene Meßwerte über die Aktivität des Systems aus. Der erste Teil der Ausgabe besteht aus mehreren Balkengraphiken, die die Aktivität der einzelnen Prozessoren darstellen. Für jeden Prozessor wird ein Balken angezeigt. Zusätzlich gibt es noch einen Balken mit der Bezeichnung "Total" für das ganze System. Jeder Balken gibt den Prozentsatz der Zeit an, die der jeweilige Prozessor aktiv war. Die Balken bestehen aus zwei Symbolen, nämlich "=" für die Rechenzeit im Systemmodus und "-" für die Rechenzeit im Benutzermodus.

Die weitere Ausgabe von monitor besteht aus verschiedenen Informationen zur Systemaktivität (sog. Aktivitätsmonitoren), die Sie einzeln auswählen können. Sie verlassen monitor, indem Sie entweder Wagenrücklauf, Zeilenvorschub oder den Buchstaben q drücken. Mit ^L (Ctrl-L) bauen Sie den Bildschirm neu auf.

monitor kennt folgende Optionen:

- c *n* (columns) gibt die Anzahl der Spalten an, die eine Balkengraphik belegen soll. Zulässige Werte für *n* sind 1, 2 und 3.
- i *n* gibt die Länge des Zeitintervalls an, das zwischen zwei Aktualisierungen der Ausgabe liegen soll. Alle *n* Sekunden wird der Bildschirm neu aufgebaut und die Graphiken und Informationen werden aktualisiert. Standardwert ist 1.
- p *n* Wenn der monitor-Prozeß mit genügend hohen Privilegien läuft, wird er dem Prozessor mit der Nummer *n* zugeordnet. Standardwert ist Prozessor 0 (siehe tmp_affinity(2)).
- t (terse) monitor gibt nur knappe Beschreibungen der Aktivitäten aus. Dadurch haben mehr Informationen auf dem Bildschirm Platz.
- f (flip) Es werden zwei verschiedene Bildschirmdarstellungen verwendet, eine Darstellungsart für die Prozessorgraphen und eine für die Aktivitätsmonitore. Durch Eingabe von f können Sie zwischen den beiden Darstellungsarten wechseln.

Es gibt folgende Aktivitätsmonitore (die Angaben in Klammern gelten für die Option -t):

Total User Time (usr %)
Zeit, die das System im Benutzermodus verbrachte, ausgedrückt als Prozentsatz eines Prozessors.

Total System Time (sys %)
Zeit, die das System im Systemmodus verbrachte, ausgedrückt als Prozentsatz eines Prozessors.

Total Time (tot %)
Summe der Benutzer- und Systemzeit, ausgedrückt als Prozentsatz eines Prozessors.

Number of Procs (procs)
Anzahl der Prozesse im System (belegte Einträge in der Prozeßtafel).

Running Procs (on p)
Anzahl der Prozesse, die gerade auf einem Prozessor laufen.

Runnable Procs (runq)
Anzahl der Prozesse, die ausführbar sind und in der Warteschlange für ablauffähige Prozesse darauf warten, einem Prozessor zugeordnet zu werden.

Sleeping Procs (sleep)
Anzahl der schlafenden Prozesse.

Swapped Procs (swapped)
Anzahl der Prozesse, die auf Festplatte ausgelagert sind.

Swapped Runnable (swprun)
Anzahl der lauffähigen Prozesse, die auf Festplatte ausgelagert sind.

System Calls (scall)
Anzahl der Systemaufrufe pro Sekunde.

Context Switches (csw)
Anzahl der Kontextwechsel pro Sekunde.

Run Queue (runq)
Anzahl der Prozesse in der Ablauf-Warteschlange.

Swap Queue (swpq)
Anzahl der Prozesse in der Swap-Warteschlange.

Forks (fork)
Anzahl der Verzweigungen (forks) pro Sekunde.

Execs (exec)
Anzahl der exec-Systemaufrufe pro Sekunde.

TTY Chars In (ttyin)
Anzahl der Zeichen pro Sekunde, die auf Leitungen von Datensichtstationen empfangen wurden.

TTY Chars Out (ttyout)
Anzahl der Zeichen pro Sekunde, die auf Leitungen zu Datensichtstationen übertragen wurden.

Packets In (pkt in)
Anzahl der pro Sekunde empfangenen Pakete.

Packets Out (pkt out)
Anzahl der pro Sekunde übertragenen Pakete.

Packets Collision (pkt col)
Anzahl der Kollisionen pro Sekunde.

Packets In Error (PktIerr)
Anzahl der bei empfangenen Paketen aufgetretenen Fehler pro Sekunde.

Packets Out Error (PktOerr)
Anzahl der bei übertragenen Paketen aufgetretenen Fehler pro Sekunde.

Read Calls (Reads)
Anzahl der read-Systemaufrufe pro Sekunde.

Write Calls (Writes)
Anzahl der write-Systemaufrufe pro Sekunde.

Iget Calls (Iget)
Anzahl der Aufrufe der Routine iget pro Sekunde.

Namei Calls (Namei)
Anzahl der Aufrufe der Routine namei pro Sekunde.

Direct Blks (Dirblks)
Anzahl der Zugriffe auf direkte Blöcke pro Sekunde.

V Faults (vf)
Anzahl der Seitenfehler pro Sekunde (valid faults).

P Faults (pf)
Anzahl der Zugriffsfehler (protection faults) pro Sekunde.

Pages In (pgin)
Anzahl der Anforderungen für einzulagernde Seiten (major faults) pro Sekunde.

Pages Out (pgout)
Anzahl der Anforderungen für auszulagernde Seiten pro Sekunde.

Pages Swap In (pswpin)
Anzahl der eingelagerten Seiten pro Sekunde.

Pages Swap Out (pswpout)
Anzahl der ausgelagerten Seiten pro Sekunde.

Pages Reclaimed (pgrclm)
Anzahl der aus einer Liste freier Seiten wiederverwendeten Seite (minor faults) pro Sekunde.

Proc Swap In (swpin)
Anzahl der pro Sekunde eingelagerten Prozesse.

Proc Swap Out (swpout)
Anzahl der pro Sekunde ausgelagerten Prozesse.

Free Memory (freem)
Freier Speicherplatz in Kilobytes.

Free Buffers (buffs)
Anzahl der Headers für freie Puffer.

Free Swap (frees)

Freier Auslagerungsplatz in Kilobytes.

FS Log Reads (fs lrd)

Anzahl der logischen Lesevorgänge, die durch das Dateisystem initiiert wurden.

FS Log Writes (fs lwt)

Anzahl der logischen Schreibvorgänge, die durch das Dateisystem initiiert wurden.

FS Blk Reads (fs brd)

Anzahl der tatsächlichen Lesevorgänge von der Festplatte, die durch das Dateisystem initiiert wurden.

FS Blk Writes (fs bwt)

Anzahl der tatsächlichen Schreibvorgänge von der Festplatte, die durch das Dateisystem initiiert wurden.

FS Read Hit (fs rdh)

Prozentsatz der Leseanforderungen an das Dateisystem, die aus den Cache-Puffern erledigt werden konnten.

FS Write Hit (fs wth)

Prozentsatz der Schreibenanforderungen auf das Dateisystem, die keinen tatsächlichen Schreibvorgang zur Folge hatten.

Raw Reads (raw rd)

Anzahl der Lesevorgänge im Raw-Modus für alle Geräte.

Raw Writes (raw wt)

Anzahl der Schreibvorgänge im Raw-Modus für alle Geräte.

SIEHE AUCH

sar(1M)

FEHLER

monitor bereitet Daten auf, die sich ständig ändern. Dabei können ab und zu ungewöhnliche Effekte auftreten. Am auffälligsten ist, daß die Anzahl der Prozesse auf einem Prozessor manchmal größer ist als die Anzahl der Prozessoren im System.

BEZEICHNUNG

montbl – Erstellen einer Währungsdatenbank

ÜBERSICHT

montbl [-o *ausgdatei*] *eingdatei*

BESCHREIBUNG

Die Eingabe für das Kommando `montbl` ist eine Spezifikationsdatei, *eingdatei*, die die Konventionen für Währungsangaben in einem bestimmten Land enthält.

-o *ausgdatei* Schreibt die Ausgabe in *ausgdatei*; anderenfalls wird die Ausgabe in eine Datei namens `LC_MONETARY` geschrieben.

Die Ausgabe von `montbl` eignet sich zur Verwendung durch die Funktion `localeconv()` (siehe `localeconv(3C)`). Bevor `localeconv()` die *ausgdatei* benutzen kann, muß diese Datei von einem privilegierten Benutzer oder einem Mitglied der Gruppe `bin` unter dem Namen `LC_MONETARY` im Verzeichnis `/usr/lib/locale/land` installiert werden. *land* ist das Land, dessen Konventionen für Währungsangaben in der Datei *eingdatei* beschrieben wird. Für diese Datei müssen Benutzer, Gruppe und andere Lesezugriff haben; andere Zugriffsrechte dürfen nicht vergeben werden. Um die in dieser Datei beschriebenen Konventionen für Währungsangaben verwenden zu können, wird mit `setlocale(3C)` das Land für die Kategorie `LC_MONETARY` in *land* geändert (siehe `setlocale(3C)`).

Sobald diese Datei installiert ist, wird sie von der Funktion `localeconv(3C)` zum Initialisieren der währungsspezifischen Felder einer Struktur des Typs `struct lconv` verwendet. Eine Beschreibung aller Felder in dieser Struktur finden Sie unter `localeconv(3C)`.

```
struct    lconv    {
    char *decimal_point;        /* "." */
    char *thousands_sep;       /* "" (Zeichenkette mit Länge Null) */
    char *grouping;             /* "" */
    char *int_curr_symbol;      /* "" */
    char *currency_symbol;      /* "" */
    char *mon_decimal_point;    /* "" */
    char *mon_thousands_sep;    /* "" */
    char *mon_grouping;         /* "" */
    char *positive_sign;        /* "" */
    char *negative_sign;        /* "" */
    char int_frac_digits;       /* CHAR_MAX */
    char frac_digits;           /* CHAR_MAX */
    char p_cs_precedes;         /* CHAR_MAX */
    char p_sep_by_space;        /* CHAR_MAX */
    char n_cs_precedes;         /* CHAR_MAX */
    char n_sep_by_space;        /* CHAR_MAX */
    char p_sign_posn;           /* CHAR_MAX */
    char n_sign_posn;           /* CHAR_MAX */
};
```


Die Spezifikationsdatei gibt den Wert für jedes Element von `struct lconv` außer den ersten drei, *decimal_point* (Dezimaltrennzeichen), *thousands_sep* (Tausendertrennzeichen) und *grouping* (Gruppierung), an. Diese drei Elemente werden von der Kategorie `LC_NUMERIC` von `setlocale(3C)` definiert. Der Wert der einzelnen Elemente wird in einer Zeile im folgenden Format angegeben:

schlüsselwort <leer-/tabulatorzeichen> *wert*

Dabei ist *schlüsselwort* identisch mit dem Feldnamen von `struct lconv`. *wert* ist bei Feldern, die mit `char *` gekennzeichnet sind, eine in Anführungszeichen stehende Zeichenkette und bei Feldern, die mit `int` markiert sind, eine Ganzzahl. Mit

```
int_curr_symbol      "ITL."
int_frac_digits      0
```

z.B. werden das internationale Währungssymbol und die Anzahl der Stellen nach dem Komma in einer international formatierten Währungsangabe als `ITL.` bzw. `0` formatiert.

Leerzeilen und Zeilen, die mit dem Zeichen `#` beginnen, gelten als Kommentar und werden ignoriert. Ein Zeichen in einer Zeichenkette kann oktal oder hexadezimal dargestellt werden. Z.B. könnte `\141` oder `\x61` zur Darstellung des Buchstabens 'a' verwendet werden. Ist für ein bestimmtes Element der Struktur keine Spezifikationszeile vorhanden, wird der Standardwert von 'C' für dieses Element verwendet (siehe die als Kommentar angegebenen Werte in der obigen `struct lconv`-Definition).

Es folgt ein Beispiel einer Spezifikationsdatei für Italien:

```
# Italy

int_curr_symbol      "ITL."
currency_symbol      "L."
mon_decimal_point    "."
mon_thousands_sep    " "
mon_grouping         "\3"
positive_sign        ""
negative_sign        "-"
int_frac_digits      0
frac_digits          0
p_cs_precedes        1
p_sep_by_space       0
n_cs_precedes        1
n_sep_by_space       0
p_sign_posn          1
n_sign_posn          1
```

montbl(1M)

montbl(1M)

DATEIEN

/usr/lib/locale/*land*/LC_MONETARY

LC_MONETARY Datenbank für *land*

/usr/lib/locale/C/montbl_C

Eingabedatei zum Erstellen von LC_MONETARY für das Standard-
land

SIEHE AUCH

localeconv(3C), setlocale(3C) im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

mount, umount – Einhängen oder Aushängen von Dateisystemen und fernen Ressourcen

ÜBERSICHT

mount [-v | -p]

mount [-F *dateisystem-typ*] [-V] [*akt_optionen*] [-o *spezifische_optionen*]
 {*sondergerät* | *einhängepunkt*}

mount [-F *dateisystem-typ*] [-V] [*akt_optionen*] [-o *spezifische_optionen*]
 sondergerät *einhängepunkt*

umount [-V] [-o *spezifische_optionen*] {*sondergerät* | *einhängepunkt*}

BESCHREIBUNG

Alle Dateisysteme außer `root ( / )` gelten als temporär in dem Sinne, daß sie für Benutzer verfügbar oder nicht verfügbar sein können. `mount` benachrichtigt das System darüber, daß *sondergerät*, ein blockorientiertes Gerät oder eine ferne Ressource, über den *einhängepunkt*, der bereits vorhanden sein muß, Benutzern zur Verfügung steht. Er wird zum Namen der Root des neu eingehängten *sondergeräts* bzw. der Ressource.

Wird `mount` mit Argumenten eingegeben, überprüft es alle Argumente außer dem Gerätenamen und ruft ein *dateisystem-typ*-spezifisches `mount`-Modul auf. Ohne Argumente listet `mount` alle eingehängten Dateisysteme aus der Einhängetabelle auf. Wird `mount` mit einem Argument der folgenden Argumentlisten aufgerufen, z.B. *sondergerät* oder *einhängepunkt*, oder wenn beide Argumente, aber kein *dateisystem-typ* angegeben ist, durchsucht `mount` `/etc/vfstab` nach den fehlenden Argumenten: *dateisystem-typ*, *sondergerät*, *einhängepunkt* und *spezifische_optionen*. Anschließend ruft es das *dateisystem-typ*-spezifische `mount`-Modul auf.

Für die meisten *dateisystem-typen* gibt es kein `umount`-spezifisches Modul. Das vorhandene wird ausgeführt; ansonsten wird das Dateisystem mit dem generischen Modul ausgehängt. Ist die Option `-o` angegeben, wird das `umount`-spezifische Modul immer ausgeführt.

akt_optionen sind die Optionen, die von dem `s5`-spezifischen Modul von `mount` und `umount` unterstützt werden. Andere *dateisystem-typen* unterstützen diese Optionen nicht immer. *spezifische_optionen* sind Unteroptionen, die in einer Liste von durch Komma getrennten Unteroptionen und/oder Paaren von Schlüsselwort/Attribut zur Interpretation durch das *dateisystem-typ*-spezifische Modul des Kommandos zusammengefaßt sind.

Es handelt sich um folgende Optionen:

- v Gibt die Ausgaben in einer neuen Darstellung aus. Bei der neuen Ausgabe werden *dateisystem-typ* und Optionen zusätzlich zu den Angaben der alten Ausgabe angezeigt. Die Felder *einhängepunkt* und *sondergerät* sind vertauscht.
- p Gibt die Liste der eingehängten Dateisysteme im Format von `/etc/vfstab` aus.

mount(1M)

mount(1M)

- F Dient zum Angeben des zu bearbeitenden *dateisystem-typs*. Der *dateisystem-typ* muß angegeben sein, oder er muß sich aus */etc/vfstab* beim Einhängen eines Dateisystems bestimmen lassen. Ist *dateisystem-typ* nicht angegeben oder existiert er in */etc/vfstab* nicht, wird standardmäßig *s5* als *dateisystem-typ* angenommen.
- V Gibt die gesamte Kommandozeile auf dem Bildschirm aus, führt das Kommando jedoch nicht aus. Die Kommandozeile wird mit den vom Benutzer eingegebenen Optionen und Argumenten sowie aus */etc/vfstab* abgeleiteten Werten erstellt. Diese Option sollte verwendet werden, um die Kommandozeile einer allgemeinen Prüfung und einer Gültigkeitsprüfung zu unterziehen.
- o dient zum Angeben *dateisystem-typ*-spezifischer Optionen.

`mount` kann von allen Benutzern angewandt werden, um eine Liste der eingehängten Dateisysteme und Ressourcen zu erstellen. Doch nur ein privilegierter Benutzer darf Dateisysteme ein- und aushängen.

HINWEIS

Das bisherige Format der Ausgabe wird in künftigen Versionen nicht mehr unterstützt. Die gesamte Ausgabe hat dann das neue *-v*-Format. Die wichtigsten Änderungen betreffen zwei neue Felder, in denen *dateisystem-typ* und Optionen angegeben werden, sowie die umgekehrte Anordnung der Angaben *einhängpunkt* und *sondergerät*.

Mit `mount` wird ein Eintrag in der Einhängetabelle */etc/mnttab* hinzugefügt; mit `umount` wird ein Eintrag aus der Tabelle gelöscht.

DATEIEN

<i>/etc/mnttab</i>	Einhängetabelle
<i>/etc/vfstab</i>	Liste der Standardparameter für die einzelnen Dateisysteme.

SIEHE AUCH

`setmnt(1M)`, `mountfsys(1M)`, `umountfsys(1M)`, `mnttab(4)`, `vfstab(4)`.

Die Handbucheinträge zu den *dateisystem-typ*-spezifischen Modulen von `mount`.

mount(1M)

(bfs)

mount(1M)

BEZEICHNUNG

`mount` (bfs) - Einhängen von bfs-Dateisystemen

ÜBERSICHT

```
mount [-F bfs] [generische_optionen] [-r] [-o spezif_optionen]
      {sondergerät | einhängepunkt}
```

```
mount [-F bfs] [generische_optionen] [-r] [-o spezif_optionen]
      sondergerät | einhängepunkt
```

BESCHREIBUNG

generische_optionen sind Optionen, die vom generischen Kommando `mount` unterstützt werden.

Mit `mount` wird ein bfs-Dateisystem in die Dateisystemhierarchie an der Position des Pfadnamens *einhängepunkt* eingehängt. Dieser Punkt muß bereits vorhanden sein. Hat der *einhängepunkt* bereits vor der `mount`-Operation einen Inhalt, wird er verdeckt, bis das Dateisystem ausgehängt wird.

Folgende Optionen sind verfügbar:

- F bfs Gibt bfs als Dateisystem-Typ an
- r Einhängen des Dateisystems mit Leseberechtigung
- o Gibt bfs-spezifischen Unteroptionen an. Es handelt sich um folgende:
 rw | ro - Lesen/Schreiben oder Lesen
 Standardwert ist Lesen/Schreiben.

Nur ein privilegierter Benutzer darf Dateisysteme einhängen.

DATEIEN

/etc/mnttab

SIEHE AUCH

generisches `mount(1M)`, `mountfsys(1M)`, `umountfsys(1M)`, `mount(2)`, `mnttab(4)`

BEZEICHNUNG

mount – Einhängen ferner NFS-Ressourcen

ÜBERSICHT

mount [-F nfs] [-r] [-o *spezif_optionen*] [*ressource einhängepunkt*]

BESCHREIBUNG

Mit dem Kommando `mount` wird eine angegebene *ressource* in die Dateisystemhierarchie an der Position mit dem Pfadnamen *einhängepunkt*, die bereits vorhanden sein muß, eingehängt. Hat *einhängepunkt* vor der `mount`-Operation bereits einen Inhalt, wird dieser verdeckt, bis die *ressource* wieder ausgehängt wird.

Ist die Ressource in der Datei `vfstab` verzeichnet, kann in der Kommandozeile entweder *ressource* oder *einhängepunkt* angegeben werden, und `mount` sucht in `vfstab` nach weiteren Angaben. Ist die Option `-F` nicht vorhanden, verwendet `mount` den in `vfstab` angegebenen Dateisystem-Typ.

`mount` trägt hinzugefügte Dateisysteme in die Tabelle der eingehängten Dateisysteme `/etc/mnttab` ein, wie unter `mnttab(4)` beschrieben.

Folgende Optionen stehen für das Kommando `mount` zur Verfügung:

`-r` Einhängen des angegebenen Dateisystems mit Leseberechtigung.

`-o spezif_optionen`

Angaben der dateisystemspezifischen Optionen in einer Liste durch Komma getrennter Wörter aus der nachfolgenden Liste.

`rw|ro` *ressource* wird mit Schreib- oder Leseberechtigung eingehängt. Der Standardwert ist `rw`.

`suid|nosuid` Ausführung von "s-Bit für Benutzer setzen" aktiviert oder deaktiviert. Der Standardwert ist `suid`.

`remount` Ist ein Dateisystem nur mit Leseberechtigung eingehängt, wird es mit Lese-/Schreibberechtigung neu eingehängt.

`bg|fg` Wenn der erste Versuch fehlschlägt, neuer Versuch im Hintergrund oder im Vordergrund. Der Standardwert ist `fg`.

`retry=n` Anzahl der Wiederholungen für die Einhängoperation. Der Standardwert ist 10000.

`port=n` Die Server-IP-Anschlußnummer. Der Standardwert ist `NFS_PORT`.

`grpuid` Anlegen einer Datei, deren Gruppennummer (GID) der effektiven GID des aufrufenden Prozesses entspricht. Diese Einstellung kann pro Verzeichnis dadurch außer Kraft gesetzt werden, daß das s-Bit für die Gruppe des Vatterverzeichnisses gesetzt wird; in diesem Fall entspricht die Gruppennummer der des Vatterverzeichnisses (siehe `open(2)` und `mkdir(2)`). Dateien, die in Dateisystemen angelegt werden, die *nicht* mit der Option `grpuid` eingehängt werden, unterliegen der BSD-Semantik, d.h. die GID wird unbedingt von der des Vatterverzeichnisses übernommen.

`rsize=n` Definiert die Größe des Lesepuffers mit *n* Byte.

mount (1M)

(NFS)

mount (1M)

<code>wsiz=<i>n</i></code>	Definiert die Größe des Schreibpuffers mit <i>n</i> Byte.
<code>timeo=<i>n</i></code>	Setzt den Wert für die Wartezeit von NFS auf <i>n</i> Zehntelsekunden fest.
<code>retrans=<i>n</i></code>	Setzt die Anzahl der nochmaligen Übertragungen von NFS auf <i>n</i> fest.
<code>soft hard</code>	Gibt einen Fehler zurück, wenn der Server nicht antwortet, oder sendet die Wiederholungsanforderung so lange, bis der Server antwortet.
<code>intr</code>	Ermöglicht, daß ein Prozeß, der hängt, während er auf eine Antwort auf einem fest eingehängten Dateisystem wartet, von Tastatur-Interrupts abgebrochen werden kann.
<code>secure</code>	Für NFS-Transaktionen wird ein Protokoll mit höherer Sicherheit verwendet.
<code>noac</code>	Unterdrückt die Cache-Pufferung von Attributen.
<code>acregmin=<i>n</i></code>	Hält Attribute nach dem Ändern der Datei mindestens <i>n</i> Sekunden lang im Cache-Puffer.
<code>acregmax=<i>n</i></code>	Hält Attribute nach dem Ändern der Datei höchstens <i>n</i> Sekunden lang im Cache-Puffer.
<code>acdirmin=<i>n</i></code>	Hält Attribute nach der Aktualisierung eines Verzeichnisses mindestens <i>n</i> Sekunden lang im Cache-Puffer.
<code>acdirmax=<i>n</i></code>	Hält Attribute nach der Aktualisierung eines Verzeichnisses höchstens <i>n</i> Sekunden lang im Cache-Puffer.
<code>actimeo=<i>n</i></code>	Definiert <i>mindest-</i> und <i>höchst-</i> Zeiten für normale Dateien und Verzeichnisse mit <i>n</i> Sekunden.

NFS-DATEISYSTEME

Einhängen im Hintergrund oder Vordergrund

Werden Dateisysteme mit der Option `bg` eingehängt, bedeutet dies, daß `mount` die Einhängeoperation im Hintergrund wiederholen soll, wenn der Einhängedämon des Servers (`mountd(1M)`) nicht antwortet. `mount` wiederholt die Anforderung so oft, wie in der Option `retry=n` angegeben. Sobald das Dateisystem eingehängt ist, warten alle NFS-Anforderungen an den Kernel `timeo=n` Zehntelsekunden auf Antwort. Trifft keine Antwort ein, wird die Wartezeit mit 2 multipliziert, und die Anforderung wird erneut übertragen. Hat die Anzahl der Wiederholungen die in der Option `retrans=n` angegebene Zahl erreicht, gibt ein mit der Option `soft` eingehängtes Dateisystem einen Fehler für die Anforderung zurück; wurde das Dateisystem mit der Option `hard` eingehängt, gibt es eine Warnung aus und wiederholt weiterhin die Anforderung.

Lese-/Schreibzugriff oder Lesezugriff

Mit `rw` (Lese-/Schreibzugriff) eingehängte Dateisysteme müssen die Option `hard` verwenden.

Sichere Dateisysteme

Die Option `secure` muß angegeben werden, wenn der Server sicheres Einhängen für das Dateisystem fordert.

mount(1M)

(NFS)

mount(1M)

Dateiattribute

Im Cache-Puffer werden Dateiattribute für den Client zwischengespeichert. Attribute zu einer Datei werden nach einer bestimmten Zeit gelöscht. Wird eine Datei geändert, bevor der Attribut-Puffer geleert wird, wird das Leerungsintervall um die Zeitspanne seit der letzten Änderung verlängert; dabei wird vorausgesetzt, daß vor kurzem geänderte Dateien bald wieder geändert werden. Für normale Dateien und für Verzeichnisse bestehen Mindest- und Höchstwerte für die Verlängerung der Leerungsintervalle. Mit `actimeo=n` wird das Leerungsintervall für reguläre Dateien und Verzeichnisse um *n* Sekunden verlängert.

BEISPIELE

Zum Einhängen eines fernen Dateisystems:

```
mount -F nfs serv:/usr/src /usr/src
```

Zum festen Einhängen eines fernen Dateisystems:

```
mount -o hard serv:/usr/src /usr/src
```

DATEIEN

<code>/etc/mnttab</code>	Tabelle der eingehängten Dateisysteme
<code>/etc/dfs/fstypes</code>	standardmäßiger verteilter Dateisystem-Typ
<code>/etc/vfstab</code>	Tabelle der automatisch eingehängten Ressourcen

SIEHE AUCH

`mountall(1M)`, `mount(2)`, `umount(2)`, `mnttab(4)`.

HINWEIS

Handelt es sich bei dem Verzeichnis, in das ein Dateisystem eingehängt werden soll, um einen symbolischen Verweis, wird das Dateisystem in *das Verzeichnis, auf das sich der symbolische Verweis bezieht*, eingehängt und nicht zusätzlich zu dem symbolischen Verweis.

BEZEICHNUNG

mount (s5) – Einhängen eines s5-Dateisystems

ÜBERSICHT

```
mount [-F s5] [generische_optionen] [-r] [-o spezif_optionen]
      {sondergerät | einhängepunkt}
```

```
mount [-F s5] [generische_optionen] [-r] [-o spezif_optionen]
      sondergerät einhängepunkt
```

BESCHREIBUNG

generische_optionen sind Optionen, die das generische Kommando `mount` unterstützt.

`mount` benachrichtigt das System davon, daß *sondergerät*, ein s5-blockorientiertes Gerät, den Benutzern am *einhängepunkt* zur Verfügung steht. Der Einhängepunkt muß vorhanden sein, ehe `mount` aufgerufen wird; er erhält den Namen der Root des neu eingehängten *sondergeräts*.

Folgende Optionen stehen zur Verfügung:

- F s5 Gibt s5 als Dateisystem-Typ an.
- r Einhängen des Dateisystems mit Lesezugriff.
- o Angeben von s5-spezifischen Unteroptionen. Folgende Unteroptionen stehen zur Verfügung:
 - rw | ro Schreib- oder Lesezugriff. Standardwert ist *rw*.
 - suid | nosuid s-Bit für Benutzer wird bei der Ausführung gesetzt bzw. nicht gesetzt. Standardwert ist *suid*.
 - remount Wird zusammen mit *rw* verwendet. Ein mit Lesezugriff eingehängtes Dateisystem kann mit Lese-/Schreibzugriff *neu eingehängt* werden. Schlägt fehl, wenn das Dateisystem aktuell nicht oder mit *rw* eingehängt ist. Die Option ist nur in Kraft, wenn sie angegeben wurde.

Nur ein privilegierter Benutzer darf Dateisysteme einhängen.

DATEIEN

/etc/mnttab Einhängetabelle

SIEHE AUCH

Generisches Kommando `mount(1M)`, `mountfsys(1M)`, `setmnt(1M)`.
`mount(2)`, `setuid(2)`, `mnttab(4)` im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

mount (ufs) – Einhängen von ufs-Dateisystemen

ÜBERSICHT

```
mount [-F ufs] [generische_optionen] [-r] [-o spezif_optionen]
      {sondergerät | einhängepunkt}
```

```
mount [-F ufs] [generische_optionen] [-r] [-o spezif_optionen]
      sondergerät einhängepunkt
```

BESCHREIBUNG

mount gliedert ein ufs-Dateisystem in die Dateisystemhierarchie an der Pfadnamenposition *einhängepunkt* ein, die bereits vorhanden sein muß. Hat *einhängepunkt* vor der mount-Operation bereits einen Inhalt, bleibt dieser verdeckt, bis das Dateisystem wieder ausgehängt wird.

Folgende Optionen stehen zur Verfügung:

- F ufs Gibt ufs als Dateisystem-Typ an.
- r Einhängen des Dateisystems mit Lesezugriff.
- o Gibt ufs-dateisystemspezifische Optionen an. Werden ungültige Optionen angegeben, wird eine Warnung ausgegeben, und die ungültigen Optionen werden ignoriert. Folgende Optionen sind verfügbar:
 - f Täuscht einen /etc/mnttab-Eintrag vor, hängt aber keine Dateisysteme ein. Parameter werden nicht überprüft.
 - n Einhängen des Dateisystems, ohne einen Eintrag in /etc/mnttab zu stellen.
 - rw | ro Lese-/Schreib- oder Lesezugriff. Standardwert ist rw.
 - nosuid Standardmäßig wird das Dateisystem so eingehängt, daß das s-Bit für Benutzer gesetzt wird. Durch Angeben von nosuid wird der Standardwert außer Kraft gesetzt, und das Dateisystem wird ohne Setzen des s-Bits für Benutzer eingehängt.
 - remount Wird zusammen mit rw verwendet. Ein mit Lesezugriff eingehängtes Dateisystem kann mit Lese-/Schreibzugriff *neu eingehängt* werden. Diese Option schlägt fehl, wenn das Dateisystem aktuell nicht oder mit rw eingehängt ist.

HINWEIS

Handelt es sich bei dem Verzeichnis, in das ein Dateisystem eingehängt werden soll, um einen symbolischen Verweis, wird das Dateisystem in das Verzeichnis, auf das sich der symbolische Verweis bezieht, eingehängt und nicht zusätzlich zu dem symbolischen Verweis.

mount(1M)

(UFS)

mount(1M)

DATEIEN

/etc/mnttab

Tabelle der eingehängten Dateisysteme

SIEHE AUCH

generisches Kommando `mount(1M)`, `mountfsys(1M)`, `umountfsys(1M)`, `mkdir(2)`,
`mount(2)`, `unmount(2)`, `open(2)`, `mnttab(4)`.

BEZEICHNUNG

mountall, mountallp, umountall – Einhängen, Aushängen mehrerer Dateisysteme

ÜBERSICHT

```
mountall [-F dateisystem-typ] [-l | -r] [ dateisystemtabelle ]
mountallp [-F dateisystem-typ] [-l | -r] [ dateisystemtabelle ]
umountall [-F dateisystem-typ] [-k] [-l | -r]
```

BESCHREIBUNG

Diese Kommandos dürfen nur von einem privilegierten Benutzer ausgeführt werden.

mountall dient zum Einhängen von Dateisystemen entsprechend einer *dateisystemtabelle* (/etc/vfstab ist die Standarddateisystemtabelle). Der Gerätedateiname "-" liest aus der Standardeingabe. Ist der Bindestrich angegeben, muß die Standardeingabe dasselbe Format haben wie /etc/vfstab. Ohne Argumente beschränkt mountall das Einhängen auf alle Systeme, bei denen das Feld automnt in der *dateisystemtabelle* auf yes gesetzt ist.

Bevor die einzelnen Dateisysteme eingehängt werden, wird mit fsck (siehe fsck(1M)) eine Plausibilitätsprüfung durchgeführt, um festzustellen, ob das System einhängbar erscheint. Ist das Dateisystem nicht einhängbar, wird es mit fsck korrigiert, bevor versucht wird, es einzuhängen.

mountallp verhält sich bis auf die Plausibilitätsprüfung der Dateisysteme wie mountall. Die Plausibilitätsprüfung wird jedoch parallel für alle Dateisysteme ausgeführt, die die gleiche Vorzugsnummer haben. Die Vorzugsnummer steht in der Datei /etc/vfstab. Soll beim Hochfahren des Systems mountallp nicht verwendet werden, dann muß in der Datei /etc/default/mountall die Zeile

```
MOUNTALL=NOTPARALLEL
```

eingetragen sein. Standardmäßig steht dort

```
MOUNTALL=PARALLEL
```

umountall bewirkt das Aushängen aller eingehängten Dateisysteme außer root, /proc, /var und /usr. Ist der *dateisystem-Typ* angegeben, beschränkt sich die Wirkung von mountall und umountall auf Dateisysteme des angegebenen Typs.

Folgende Optionen stehen zur Verfügung:

- F Angabe des Dateisystem-Typs, der eingehängt oder ausgehängt werden soll. Ist *dateisystem-typ* angegeben, beschränkt sich der Vorgang auf Dateisysteme des angegebenen Typs.
- l Begrenzt den Vorgang auf lokale Dateisysteme.
- r Begrenzt den Vorgang auf ferne Dateisystem-Typen.
- k Sendet das Signal SIGKILL an Prozesse, die Dateien im Dateisystem geöffnet haben.

mountall(1M)

mountall(1M)

FEHLERMELDUNGEN

Wenn die Dateisysteme einhängbar und fehlerfrei sind, wird keine Meldung ausgegeben.

Fehler- und Warnmeldungen stammen von `fsck(1M)` und `mount(1M)`.

SIEHE AUCH

`fsck(1M)`, `fuser(1M)`, `mount(1M)`, `vfstab(4)`, `mnttab(4)`,
`signal(2)` im *Referenzhandbuch für Programmierer*.

mountd (1M)

(NFS)

mountd (1M)

BEZEICHNUNG

`mountd` – Server für Anforderungen zum Einhängen eines NFS-Dateisystems

ÜBERSICHT

`mountd [ -n ]`

BESCHREIBUNG

`mountd` ist ein RPC-Server, der auf Anforderungen zum Einhängen von Dateisystemen reagiert. Dazu liest er die Datei `/etc/dfs/sharetab` (die unter `sharetab(4)` beschrieben ist). So bestimmt dieser Server, welche Dateisysteme auf welchen Rechnern zum Einhängen zur Verfügung stehen. Er macht auch Angaben darüber, welche Dateisysteme von welchen Clients eingehängt wurden. Diese Daten können mit dem Kommando `dfmounts(1M)` ausgegeben werden.

Der `mountd`-Dämon wird in Betriebsstufe 3 automatisch aufgerufen.

Mit der Option `-n` prüft `mountd` nicht, ob die Clients unter der Benutzerkennung `root` arbeiten. Zwar wird dadurch die Sicherheit beeinträchtigt, aber es können so auch ältere Versionen (vor 3.0) von Clients-NFS einbezogen werden.

DATEIEN

`/etc/dfs/sharetab`

SIEHE AUCH

`dfmounts(1M)`, `sharetab(4)`.

BEZEICHNUNG

`mountfsys`, `umountfsys` – Einhängen, Aushängen eines Dateisystems

ÜBERSICHT

`mountfsys`
`umountfsys`

BESCHREIBUNG

Mit dem Kommando `mountfsys` wird ein Dateisystem geladen, so daß Benutzer darin Lese- und Schreiboperationen ausführen können. Mit dem Kommando `umountfsys` wird das Dateisystem ausgehängt.

Das Kommando ruft eine Menü-Schnittstelle auf (die über das Kommando `sysadm` verfügbaren Aufgaben `mount-` oder `unmount`).

Bei der ersten Eingabeaufforderung der beiden Kommandos kann das Gerät ausgewählt werden, auf dem bzw. von dem das Dateisystem ein- bzw. ausgehängt werden soll.

Bei dem Kommando `mountfsys` muß angegeben werden, wie das Dateisystem eingehängt werden soll, z.B. mit Lesezugriff oder mit Lese-/Schreibzugriff.

Die gleichen Funktionen stehen auch im `sysadm`-Menü zur Verfügung:

`sysadm mount`
`sysadm unmount`

EXIT-CODES

Die Exit-Codes von `mountfsys` und `umountfsys` haben folgende Bedeutung:

- 0 = normaler Abschluß
- 2 = ungültige Kommandosyntax; es wird eine Auslastungsmeldung ausgegeben
- 7 = die Menü-Schnittstelle für dieses Kommando ist nicht verfügbar, weil `fmli` sich nicht aufrufen läßt (das `fmli`-Paket ist nicht installiert oder defekt)

HINWEIS

Werden Dateisysteme auf herausnehmbaren Datenträgern verwendet, darf eine Diskette mit einem einmal eingehängten Dateisystem erst nach dem Aushängen des Dateisystems aus dem Laufwerk genommen werden. Wird eine Diskette aus dem Laufwerk genommen, während das Dateisystem noch eingehängt ist, kann dies schwere Beschädigungen der Daten auf der Diskette zur Folge haben.

SIEHE AUCH

`checkfsys(1M)`, `labelit(1M)`, `makefsys(1M)`, `mkfs(1M)`, `mount(1M)`, `sysadm(1M)`.

BEZEICHNUNG

`mvmir` – Verschieben eines Verzeichnisses

ÜBERSICHT

`/usr/sbin/mvmir vername name`

BESCHREIBUNG

`mvmir` verschiebt Verzeichnisse innerhalb eines Dateisystems. *vername* muß ein Verzeichnis sein. Existiert *name* noch nicht, wird es als Verzeichnis angelegt. Existiert *name*, und ist es ein Verzeichnis, wird *vername* als *name/vername* angelegt. *vername* und *name* dürfen sich nicht auf demselben Pfad befinden; d.h. das eine Verzeichnis darf dem anderen nicht untergeordnet sein. Zum Beispiel:

```
mvmir x/y x/z
```

ist zulässig,

```
mvmir x/y x/y/z
```

dagegen nicht.

SIEHE AUCH

`mkmir(1)`, `mv(1)` in den *Kommandos*.

WARNUNG

Nur ein privilegierter Benutzer darf das Kommando `mvmir` benutzen.

BEZEICHNUNG

named, in.named – Internet Name Server

ÜBERSICHT

in.named [-d *umfang*] [-p *anschluß*] [[-b] *ladedatei*]

BESCHREIBUNG

in.named ist der Name-Server des Internet-Bereichs (domain). Das Kommando wird von Hosts im Internet benutzt, um den Zugriff auf die verteilte Name-Datenbank des Internet zu ermöglichen. Weitere Angaben finden Sie unter RFC 1034 und RFC 1035. Werden keine Argumente angegeben, durchsucht in.named die Datei /etc/named.boot nach Eingabedaten. Der Name-Server wartet dann auf Anfragen an einem privilegierten Anschluß.

Folgende Optionen stehen zur Verfügung:

- d *umfang* Gibt Informationen zur Fehlersuche aus. *umfang* ist eine Zahl, die den Umfang der auszugebenden Meldungen angibt.
- p *anschluß* Benutzt einen anderen *anschluß* (andere Nummer).
- b *ladedatei* Benutzt die *ladedatei* anstelle von /etc/named.boot.

BEISPIEL

```

;
;   boot file for name server
;
; type           domain           source file or host
;
domain          berkeley.edu
primary         berkeley.edu      named.db
secondary       cc.berkeley.edu   10.2.0.78 128.32.0.10
cache           named.ca
```

Die Zeile domain (Bereich) gibt an, daß der Bereich (domain) des betreffenden Servers berkeley.edu ist.

Die Zeile primary (primär) gibt an, daß die Datei named.db autoritative Daten für berkeley.edu enthält. Die Datei named.db enthält Daten im Master-Dateiformat, das in RFC 1035 beschrieben wird. Alle Bereichsnamen werden jedoch relativ zum Ursprung angegeben, in diesem Fall berkeley.edu (eine genauere Beschreibung folgt).

Die Zeile secondary (sekundär) gibt an, daß alle autoritativen Daten in cc.berkeley.edu vom Name-Server an der Adresse 10.2.0.78 übertragen werden. Schlägt die Übertragung fehl, wird die Adresse 128.32.0.10 benutzt, und zwar in bis zu zehn Versuchen. Auch diese Daten sind autoritative Daten für den Bereich.

Die Zeile cache gibt an, daß die Daten in named.ca in den Cache geschrieben werden sollen (in der Regel Daten wie der Standort von Root-Bereichs-Servern). Die Datei named.ca hat das gleiche Format wie named.db.

Die Master-Datei besteht aus Einträgen in der Form:

```
$INCLUDE < dateiname >
$ORIGIN < bereich >
< bereich > < opt_z > < opt_klasse > < typ > < ressourcendaten >
```

Dabei steht *bereich* für den Bereich (domain), und zwar . für die Root, @ für den aktuellen Ursprung, oder es handelt sich um einen Standardbereichsnamen. Ist *bereich* ein Standardbereichsname, der nicht mit . endet, wird der aktuelle Ursprung an den Bereich angehängt. Bereichsnamen, die mit . enden, werden nicht geändert.

Das Feld *opt_z* repräsentiert eine optionale ganze Zahl für das Zeitfeld. Der Standardwert lautet Null.

Das Feld *opt_klasse* ist im Augenblick ein Token, und zwar IN für Internet.

Das Feld *typ* ist eines der folgenden Token; die Daten im Feld *ressourcendaten* stehen in Klammern.

A	Hostadresse (durch "." getrenntes 4-Tupel)
NS	Autoritativer Name-Server (Bereich)
MX	Postaustauschprogramm (Bereich)
CNAME	Standardname für einen Aliasnamen (Bereich)
SOA	Markiert den Beginn einer Berechtigungszone (5 Zahlen). Siehe RFC 1035.
MB	Bereichsname eines elektronischen Briefkastens (Bereich)
MG	Mitglied eines Verteilers (Bereich)
MR	Bereichsname eines elektronischen Briefkastens (umbenannt) (Bereich)
NULL	Nullsatz für die Ressourcen (kein Format, keine Daten)
WKS	Bekannte Service-Beschreibung (noch nicht implementiert)
PTR	Bereichsnamen-Zeiger (Bereich)
HINFO	Host-Informationen (cpu_typ BS_typ).
MINFO	Informationen zum elektronischen Briefkasten oder zur Verteilerliste (anforderung_bereich fehler_bereich).

DATEIEN

/etc/named.boot	Ladedatei für die Konfiguration des Name-Servers
/etc/named.pid	Prozeßnummer
/var/tmp/named.run	Ausgabe der Fehlersuche
/var/tmp/named_dump.db	Dump der Datenbank des Namen-Servers

named (1M)

named (1M)

SIEHE AUCH

`kill(1)`, `signal(3)`, `resolver(3N)`, `resolve.conf(4)`.

Mockapetris, Paul, *Domain Names - Concepts and Facilities*, RFC 1034, Network Information Center, SRI International, Menlo Park, Kalif., November 1987.

Mockapetris, Paul, *Domain Names - Implementation and Specification*, RFC 1035, Network Information Center, SRI International, Menlo Park, Kalif., November 1987.

Mockapetris, Paul, *Domain System Changes and Observations*, RFC 973, Network Information Center, SRI International, Menlo Park, Kalif., Januar 1986.

Partridge, Craig, *Mail Routing and the Domain System*, RFC 974, Network Information Center, SRI International, Menlo Park, Kalif., Januar 1986.

HINWEIS

Die folgenden Signale haben die angegebene Wirkung, wenn sie mit dem Kommando `kill(1)` an den Server-Prozeß gesendet werden.

SIGHUP	Liest <code>/etc/named.boot</code> und lädt die Datenbank neu.
SIGINT	Schreibt einen Dump der aktuellen Datenbank und des Caches in <code>/var/tmp/named_dump.db</code> .
SIGUSR1	Aktiviert die Fehlersuche; jedes weitere Signal SIGUSR1 erhöht die Ebene der Fehlersuche.
SIGUSR2	Deaktiviert die Fehlersuche.

BEZEICHNUNG

`ncheck` (generisch) – listet Pfadnamen und zugeordnete I-Nummern auf

ÜBERSICHT

`ncheck [-F dateisystem-typ] [-V] [aktuelle_optionen] [-o spezifische_optionen] [gerät...]`

BESCHREIBUNG

`ncheck` ohne Optionen generiert eine Liste von Pfadnamen und zugeordneten I-Nummern aller Dateien auf *gerät*. Wird *gerät* nicht in der Kommandozeile angegeben, wird die Liste für all die *geräte* in `/etc/vfstab` generiert, deren Einträge einen numerischen Wert für *fsckpass* aufweisen. *gerät* ist ein blockorientiertes Gerät, auf dem sich das Dateisystem befindet.

aktuelle_optionen sind Optionen, die vom `s5`-spezifischen Modul von `ncheck` unterstützt werden. Andere *dateisystem-typen* unterstützen diese Optionen nicht in jedem Fall. *spezifische_optionen* sind Unteroptionen, die durch Komma getrennt aufgelistet werden, und zwar in Form von Unteroptionen und/oder Paaren von Schlüsselwort/Attribut, die vom *dateisystem-typ*-spezifischen Modul des Kommandos interpretiert werden.

Die Optionen lauten:

- F Gibt den *dateisystem-typ* an, auf den sich das Kommando beziehen soll. Der *dateisystem-typ* sollte entweder hier angegeben werden oder sich aus `/etc/vfstab` ergeben, indem ein Tabelleneintrag gefunden wird, der ein numerisches *fsckpass*-Feld und - wenn angegeben - ein entsprechendes *gerät* enthält.
- V Gibt die gesamte Kommandozeile auf dem Bildschirm aus, führt das Kommando aber nicht aus. Die Kommandozeile wird aus den Optionen und Argumenten, die der Benutzer angibt, ergänzt durch zusätzliche Informationen, die sich aus `/etc/vfstab` ergeben, generiert. Diese Option dient dazu, die Kommandozeile auszuwerten und ggf. auf ihre Gültigkeit zu überprüfen.
- o dient dazu, ggf. *dateisystem-typ*-spezifische Optionen anzugeben.

HINWEIS

Dieses Kommando wird nicht für alle *dateisystem-typen* unterstützt.

DATEIEN

`/etc/vfstab` Liste der Standardparameter für die einzelnen Dateisysteme

SIEHE AUCH

`vfstab(4)`

Handbucheinträge für die *dateisystem-typ*-spezifischen Module von `ncheck`.

BEZEICHNUNG

ncheck (s5) – generiert Pfadnamen und zugeordnete I-Nummern für s5-Dateisysteme

ÜBERSICHT

ncheck [-F s5] [*generische_optionen*] [-i *i-nummer...*] [-a] [-s] [*gerät...*]

BESCHREIBUNG

generische_optionen sind Optionen, die von dem generischen Kommando ncheck unterstützt werden.

ncheck generiert eine Liste von Pfadnamen und zugeordneten I-Nummern aller Dateien auf dem/den zugehörigen *gerät(en)*. Auf Namen von Verzeichnisdateien folgt ein `"/. ".`

Die Optionen lauten:

-F s5 Gibt s5 als Dateisystem-typ an.

-i *i-nummer*

Beschränkt den Bericht auf die Dateien mit den angegebenen *i-nummern*. Die *i-nummern* werden durch Komma voneinander getrennt. Leerzeichen sind nicht erlaubt.

-a Ermöglicht die Ausgabe der Namen `"/.` und `"/..",` die normalerweise nicht ausgegeben werden.

-s Beschränkt den Bericht auf Gerätedateien und Dateien mit dem Modus set-user-ID (s-Bit für Benutzer setzen). Diese Option kann dazu benutzt werden, Verstöße gegen die Sicherheitsregeln festzustellen.

FEHLERMELDUNGEN

Bei nicht konsistenten Dateisystemstrukturen steht `??` für den Vater einer vaterlosen Datei und ein Pfadname, der mit `...` beginnt, für eine Schleife.

SIEHE AUCH

generisches Kommando ncheck(1M).

BEZEICHNUNG

ncheck (ufs) – generiert Pfadnamen mit zugeordneten I-Nummern für ufs-Dateisysteme

ÜBERSICHT

ncheck [-F ufs] [*generische_Optionen*] [-i *i-liste*] [-a] [-s] [-o m] [*gerät ...*]

BESCHREIBUNG

generische_optionen sind Optionen, die vom generischen Kommando ncheck unterstützt werden.

ncheck generiert eine Liste von Pfadnamen mit zugeordneter I-Nummer für das ufs-Dateisystem. Namen von Verzeichnisdateien werden gefolgt von "/. ".

Die Optionen lauten:

-F ufs

Gibt ufs als Dateisystem-Typ an.

-i *i-list*

Beschränkt den Bericht auf die Dateien in der folgenden *i-liste*. Die Einträge der *i-liste* müssen durch Komma, jedoch ohne Leerzeichen, getrennt werden.

-a

Ermöglicht die Ausgabe der Namen "." und "..", die sonst unterdrückt werden.

-s

Beschränkt den Bericht auf Gerätedateien und Dateien im Modus set-user-ID (s-Bit für Benutzer setzen). Diese Option dient dazu, ggf. Verstöße gegen die Sicherheitsregeln festzustellen.

-o

Gibt für ufs spezifische Optionen aus. Die entsprechende Option lautet:

m

Gibt Modusinformationen aus.

FEHLERMELDUNGEN

Im Falle einer fehlerhaften Dateisystemstruktur bezeichnet ?? den Vater einer vaterlosen Datei und ein Pfadname, der mit . . . beginnt, eine Schleife.

SIEHE AUCH

generisches Kommando ncheck(1M)

BEZEICHNUNG

netstat – zeigt den Netzwerkstatus an

ÜBERSICHT

```
netstat [ -aAn ] [ -f adr_familie ] [ system ] [ core ]
```

```
netstat [ -n ] [ -s ] [ -i | -r ] [ -f adr_familie ] [ system ] [ core ]
```

```
netstat [ -n ] [ -I schnittstelle ] intervall [ system ] [ core ]
```

BESCHREIBUNG

netstat zeigt den Inhalt von verschiedenen Datenstrukturen zum Netzwerk in verschiedenen Formaten an, je nach den ausgewählten Optionen.

Die erste Form des Kommandos erzeugt eine Liste der aktiven Socket-Schnittstellen für jedes Protokoll. Bei der zweiten Form des Kommandos wird eine von mehreren Netzwerkdatenstrukturen ausgewählt. Bei der dritten Form des Kommandos werden laufende Statistiken zum Paketaustausch auf den konfigurierten Netzwerkschnittstellen angezeigt; das Argument *intervall* gibt an, wieviele Sekunden lang zwischen den einzelnen Anzeigen statistische Daten gesammelt werden sollen.

Der Standardwert für das Argument *system* lautet `/stand/unix`; für *core* ist der Standard `/dev/kmem`.

Folgende Optionen stehen zur Verfügung:

- a Zeigt den Status aller Socket-Schnittstellen an; in der Regel werden Sockets, die von Server-Prozessen benutzt werden, nicht angezeigt.
- A Zeigt die Adresse aller Protokollsteuerblöcke an, die den Socket-Schnittstellen zugeordnet sind; diese Option wird zur Fehlersuche benutzt.
- i Zeigt den Status von selbstkonfigurierten Schnittstellen an. Schnittstellen, die statisch in einem System konfiguriert sind, aber zur Zeit des Ladens nicht lokalisiert werden können, werden nicht angezeigt.
- n Zeigt die Netzwerkadressen in Form von Zahlen an. In der Regel zeigt netstat Adressen als Symbole an. Diese Option kann mit jedem Anzeigeformat verwendet werden.
- r Zeigt die Routing-Tabelle an. Wenn diese Option mit der Option -s verwendet wird, werden statt dessen Routing-Statistiken angezeigt.
- s Zeigt Statistiken getrennt nach Protokollen an. Wenn diese Option mit der Option -r verwendet wird, werden Routing-Statistiken angezeigt.
- f *adr_familie*
Beschränkt Statistiken oder Adreßsteuerblockberichte auf die der angegebenen *adr_familie*. Dabei gibt es zwei Möglichkeiten:

inet	Für die AF_INET Adreßfamilie oder
unix	Für die AF_UNIX Familie.

-I *schnittstelle*

Hebt Informationen zur angegebenen *schnittstelle* in einer eigenen Spalte hervor; der Standardwert bei der dritten Form des Kommandos ist die Schnittstelle, die seit dem letzten Laden des Systems am intensivsten genutzt wurde. *schnittstelle* kann jede gültige Schnittstelle sein, die in der Systemkonfigurationsdatei aufgelistet ist, zum Beispiel *em0* oder *lo0*.

ANZEIGE

Aktive Sockets (Erste Form)

Die Anzeige für jede aktive Socket-Schnittstelle enthält die lokale und die ferne Adresse, die Größe der Sende- und der Empfangswarteschlange in Byte, das Protokoll und den internen Status des Protokolls.

Das symbolische Format, das normalerweise zur Anzeige von Socket-Adressen benutzt wird, lautet entweder

hostname.anschluß

wenn der Name des Host angegeben wird, oder

netzwerk.anschluß

wenn eine Socket-Adresse ein Netzwerk, aber keinen bestimmten Host angibt. Jeder *hostname* bzw. jedes *netzwerk* wird je nach seinem Eintrag in der Datei */etc/hosts* bzw. */etc/networks* angezeigt.

Sind das Netzwerk oder der Hostname für eine Adresse nicht bekannt oder wurde die Option *-n* angegeben, wird die numerische Netzwerkadresse angezeigt. Nicht spezifizierte Adressen und Anschlüsse, d.h. solche mit Platzhalterzeichen, erscheinen als ***. Weitere Angaben zu den Namenskonventionen von Internet finden Sie unter *inet(7)*.

TCP-Sockets

Die möglichen Statuswerte für TCP-Sockets lauten:

CLOSED	Geschlossen. Die Socket-Schnittstelle wird nicht benutzt.
LISTEN	Horcht auf ankommende Verbindungen.
SYN_SENT	Versucht aktiv, eine Verbindung herzustellen.
SYN_RECEIVED	Anfangssynchronisierung der Verbindung läuft.
ESTABLISHED	Verbindung ist hergestellt.
CLOSE_WAIT	Fernes Abschalten; wartet, daß die Socket-Schnittstelle geschlossen wird.
FIN_WAIT_1	Socket geschlossen; Verbindung wird gerade beendet.
CLOSING	Geschlossen, dann fernes Abschalten; wartet auf Bestätigung.
LAST_ACK	Fernes Abschalten, dann geschlossen; wartet auf Bestätigung.
FIN_WAIT_2	Socket-Schnittstelle geschlossen; wartet auf Abschalten der fernen Station.
TIME_WAIT	Wartet nach Schließen auf Signal, daß fernes Abschalten erfolgte.

Datenstrukturen im Netzwerk (Zweite Form)

Die Form der Anzeige hängt davon ab, welche der Optionen `-i` oder `-r` gewählt wurde. Werden beide Optionen angegeben, wählt `netstat` eine in der hier angegebenen Reihenfolge aus.

Anzeige der Routingtabelle

Beim Anzeigen der Routing-Tabelle werden die verfügbaren Routen und deren Status aufgelistet. Jede Route besteht aus einem Ziel-Host bzw. -Netzwerk und einem Gateway, über den Pakete versendet werden sollen. Die Spalte mit den Flags zeigt den Status der Route (U für aktiv); zeigt, ob die Route über einen Gateway führt (G) und ob die Route dynamisch über eine Umleitung generiert wurde (D).

Direkte Routen werden für jede Schnittstelle generiert, die an den lokalen Host angeschlossen ist; das Feld Gateway für diese Einträge zeigt die Adresse der Sendeschnittstelle.

Die Spalte `refcnt` gibt die aktuelle, aktive Belegung der Route an (in Zahlen). Verbindungsorientierte Protokolle benutzen in der Regel für die Dauer einer Verbindung nur eine einzige Route, während nicht-verbindungsorientierte Protokolle für die Übertragung an ein bestimmtes Ziel jeweils eine Route aufbauen.

Die Spalte `use` zeigt an, wieviele Pakete pro Route verschickt wurden.

Der Eintrag *schnittstelle* zeigt die Netzwerkschnittstelle an, die für die Route benutzt wird.

Kumulierte Auslastungsstatistiken (Dritte Form)

Wenn das Argument *intervall* angegeben wird, zeigt `netstat` eine Tabelle kumulierter Statistiken an. Diese enthalten Angaben zu den versendeten Paketen, zu Fehlern und Kollisionen, zu Netzwerkadressen für die Schnittstelle und zu der größten Übertragungseinheit (mtu). Die erste angezeigte Datenzeile und danach jeweils die 24. Zeile enthält kumulierte Statistiken zu den Aktivitäten des Systems seit dem letzten Laden. Jede folgende Zeile zeigt statistische Angaben zu dem *intervall* (angegeben in der Kommandozeile) seit der letzten Anzeige.

SIEHE AUCH

`iostat(1M)`, `trpt(1M)`, `vmstat(1M)`, `hosts(4)`, `networks(4)`, `protocols(4)`, `services(4)`.

HINWEIS

Bei der Fehlerbehandlung gibt es Probleme.

Die Tabellen des Kernels können sich ändern, während `netstat` sie untersucht, so daß sich falsche oder unvollständige Anzeigen ergeben.

BEZEICHNUNG

`newgrp` – Anmelden in einer neuen Gruppe

ÜBERSICHT

`newgrp [-] [group]`

BESCHREIBUNG

`newgrp` ändert die reale und die effektive Gruppennummer des Benutzers. Dieser bleibt angemeldet, und das aktuelle Verzeichnis wird nicht geändert. Der Benutzer erhält immer eine neue Shell. `newgrp` ersetzt die aktuelle Shell, unabhängig davon, ob sie ordnungsgemäß oder aufgrund einer Fehlerbedingung (d.h. unbekannte Gruppe) beendet wurde.

Exportierte Variablen behalten nach dem Aufrufen von `newgrp` ihre Werte; dagegen werden alle nicht exportierten Variablen entweder auf ihren Standardwert oder auf Null zurückgesetzt. Systemvariablen (z.B. `PS1`, `PS2`, `PATH`, `MAIL` und `HOME`) werden auf die Standardwerte zurückgesetzt, sofern sie nicht vom System oder ausdrücklich vom Benutzer exportiert wurden. Als Beispiel wird angenommen, ein Benutzer habe als primäre Eingabeaufforderung (`PS1`) eine andere Zeichenkette als `$` (Standard) und habe `PS1` nicht exportiert. Wenn `newgrp` aufgerufen wird, ob erfolgreich oder erfolglos, wird die `PS1` des Benutzers dadurch auf die standardmäßige Eingabeaufforderung `$` zurückgesetzt. Bitte beachten Sie, daß Variablen mit dem Shell-Kommando `export` (siehe den Handbucheintrag `sh(1)`) exportiert werden müssen, damit sie beim Aufrufen neuer Shells den zugewiesenen Wert behalten.

Ohne Argumente ändert `newgrp` die (reale und effektive) Gruppennummer des Benutzers in den Wert zurück, der in dem entsprechenden Eintrag in der Paßwortdatei für den Benutzer angegeben ist. So kann die Wirkung eines zuvor eingegebenen Kommandos `newgrp` aufgehoben werden.

Ist das erste Argument zu `newgrp` ein Minuszeichen `-`, wird die Umgebung so geändert, als hätte sich der Benutzer als Mitglied der neuen Gruppe neu angemeldet.

Gilt für die Gruppe ein Paßwort und ist der Benutzer in der Datei `/etc/group` nicht als Mitglied dieser Gruppe angegeben, wird er aufgefordert, dieses Paßwort einzugeben.

DATEIEN

`/etc/group` Gruppendatei des Systems

`/etc/passwd` Paßwortdatei des Systems

SIEHE AUCH

`login(1)`, `sh(1)` in den *Kommandos*.

`group(4)`, `passwd(4)`, `environ(5)` im *Referenzhandbuch für Systemverwalter*.

sowie den Abschnitt `intro(2)` "Effective User ID and Effective Group ID" im *Referenzhandbuch für Programmierer*

BEZEICHNUNG

nfsd – NFS-Dämon

ÜBERSICHT

nfsd [-a] [-p *protokoll*] [-t *gerät*] [*anzserver*]

BESCHREIBUNG

nfsd aktiviert die Dämonen, die Dateisystemanforderungen von Clients ausführen.

Folgende Optionen werden erkannt:

- a Aktiviert nfsds für alle verfügbaren verbindungslosen Transportprotokolle.
- p *protokoll* Aktiviert nfsds für das angegebene Protokoll.
- t *gerät* Aktiviert nfsds für die von dem angegebenen Gerät spezifizierten Transportprotokolle.
- anzserver* Die Anzahl der zu aktivierenden Dämonen. *anzserver* ist entsprechend der Belastung anzugeben, die für diesen Server erwartet wird. Der übliche Wert für *anzserver* ist vier.

In Betriebsstufe 3 werden die nfsd-Dämonen automatisch aktiviert.

DATEIEN

.nfsXXX Zeiger des Clientsystems auf eine offene Datei ohne Verweis

SIEHE AUCH

biod(1M), mountd(1M), sharetab(4).

BEZEICHNUNG

nlsadmin – Netzwerk-Service-Verwaltung

ÜBERSICHT

```
/usr/sbin/nlsadmin -x
/usr/sbin/nlsadmin [ optionen ] netz_bez
/usr/sbin/nlsadmin [ optionen ] -N anschl_monitor_etikett
/usr/sbin/nlsadmin -V
/usr/sbin/nlsadmin -c kom | -o streamname [ -p module ] \
    [ -A adresse | -D ] [ -R prognr:versnr ]
```

BESCHREIBUNG

nlsadmin ist das Verwalterkommando für einen oder mehrere Horchprozesse für Netzwerke auf einem Rechner. Jedem Netzwerk wird der Netzwerkhorchprozeß mindestens einmal zugewiesen; jede Zuweisung, und damit jedes Netzwerk, wird einzeln konfiguriert. Der Horchprozeß "horcht" im Netzwerk nach Service-Anforderungen, nimmt sie an und ruft die entsprechenden Server auf. Der Netzwerkhorchprozeß kann in jedem Netz verwendet werden (mit jedem verbindungsorientierten Transport-Provider), das der Spezifikation für Transport-Provider entspricht.

nlsadmin kann einen Horchprozeß für ein bestimmtes Netzwerk einrichten, die spezifischen Attribute dieses Prozesses definieren und den Horchprozeß für dieses Netzwerk aktivieren und deaktivieren. nlsadmin kann auch Einzel- oder Sammelberichte über die Horchprozesse auf einem Rechner erstellen.

Die nachstehende Liste gibt die Einsatzmöglichkeiten für nlsadmin an. In dieser Liste steht *netz_bez* für einen bestimmten Horchprozeß. *netz_bez* ist der relative Pfadname des Eintrags in */dev* für ein bestimmtes Netzwerk (d.h. eines Transport-Providers). *adresse* ist die Adresse eines Transport-Providers, an der gehorcht wird. Sie wird anhand einer Syntax interpretiert, die zahlreiche Adreßformate zuläßt. Standardmäßig wird *adresse* als symbolische ASCII-Darstellung der Adresse eines Transport-Providers interpretiert. Ist der *adresse* ein *\x* vorangestellt, kann eine Adresse im Hexadezimalformat eingegeben werden. *adresse* darf für die Shell nur als Einzelwort zu erkennen sein und muß in Anführungszeichen stehen, wenn sie Leerzeichen enthält.

Änderungen in der Liste der vom Horchdienst erbrachten Dienste oder deren Adressen werden sofort wirksam.

nlsadmin kann mit den folgenden Kombinationen von Optionen und Argumenten verwendet werden:

```
nlsadmin      Gibt eine kurze Auslastungsmeldung aus.
nlsadmin -x   Erstellt einen Statusbericht für alle auf diesem Rechner installierten
               Horchprozesse.
nlsadmin netz_bez
               Gibt den Status des Horchprozesses für netz_bez aus.
```

- `nlsadmin -q netz_bez`
 Fragt den Status des Horchprozesses für das angegebene Netzwerk ab und erfaßt das Ergebnis im Exit-Code. Ist ein Horchprozeß aktiv, endet `nlsadmin` mit dem Exit-Code 0; ist kein Prozeß aktiv, lautet der Exit-Code 1; bei einem Fehler ist der Exit-Code größer als 1.
- `nlsadmin -v netz_bez`
 Gibt einen ausführlichen Bericht über die mit `netz_bez` verbundenen Server aus. Dieser enthält den Dienstcode, Status, das Kommando und den Kommentar für jeden Server. Ebenso enthält er die `uid`, unter der der Server arbeitet, und eventuell die Liste der Module, die gepusht werden müssen, bevor der Server aktiviert wird.
- `nlsadmin -z dienstcode netz_bez`
 Gibt einen Bericht über den mit `netz_bez` verbundenen Server aus, der den Dienstcode `dienstcode` hat. Dieser Bericht enthält dieselben Angaben wie der mit der Option `-v` erstellte.
- `nlsadmin -q -z dienstcode netz_bez`
 Fragt den Status des Dienstes mit dem Dienstcode `dienstcode` auf dem Netzwerk `netz_bez` ab und endet mit Exit-Code 0, wenn dieser Dienst aktiv ist, mit 1, wenn der Dienst deaktiviert ist, und größer als 1 bei einem Fehler.
- `nlsadmin -l adresse netz_bez`
 Ändert oder definiert die Transport-Adresse, an der der Horchprozeß horcht (der allgemeine Horchdienst). Diese Adresse kann von fernen Prozessen verwendet werden, um über diesen Horchprozeß auf die verfügbaren Server zuzugreifen (siehe Option `-a`).
 Ist `adresse` lediglich ein Bindestrich ("-"), ändert `nlsadmin` die aktuell konfigurierte Adresse nicht, sondern meldet sie.
 Eine Adreßänderung wird sofort wirksam.
- `nlsadmin -t adresse netz_bez`
 Ändert oder definiert die Adresse, an der der Horchprozeß nach Anforderungen für Terminaldienste lauscht und ähnelt ansonsten der bereits beschriebenen Option `-l`. Eine Terminaldienstadresse darf nur definiert werden, wenn die geeignete Software für Fern-Login vorhanden ist; in diesem Falle muß die Software mit Dienstcode 1 definiert sein (siehe die Option `-a`).
- `nlsadmin -i netz_bez`
 Initialisiert einen weiteren Horchprozeß für das mit `netz_bez` angegebene Netzwerk; d.h. die vom Horchprozeß benötigten Dateien werden angelegt und initialisiert; der zusätzliche Horchprozeß wird aktiviert. Ein bestimmter weiterer Horchprozeß darf nur einmal initialisiert werden. Der Horchprozeß muß initialisiert werden, ehe Adressen oder Dienste zugewiesen werden.
- `nlsadmin -a dienstcode [-p module] [-w name] -c kom -y kommentar netz_bez`
 Fügt einen neuen Dienst in die Liste der über den angegebenen Horchprozeß verfügbaren Dienste ein. `dienstcode` ist der Code für den Dienst, `kom` ist das Kommando, das auf diesen Dienstcode hin

aufgerufen werden soll. Es muß aus dem vollständigen Pfadnamen des Servers und seinen Argumenten bestehen. *kommentar* ist eine kurze Beschreibung des Dienstes, der in den verschiedenen Berichten verwendet werden soll (in beliebigem Format). Bitte beachten Sie, daß *kom* gegenüber der Shell als Einzelwort erscheinen muß; werden Argumente benötigt, müssen *kom* und die Argumente in Anführungszeichen eingeschlossen werden. Der *kommentar* muß gegenüber der Shell ebenfalls als Einzelwort erscheinen. Wird ein Dienst hinzugefügt, wird er sofort aktiviert (siehe die Optionen *-e* und *-d*).

Dienstcodes sind alphanumerische Zeichenketten, die von AT&T verwaltet werden. Die numerischen Dienstcodes von 0 bis 100 sind für den internen Gebrauch im Horchprozeß reserviert. Dienstcode 0 ist dem nlps-Server zugewiesen, dem Dienst, der für die allgemeine Horchadresse aufgerufen wird. Code 1 ist dem Fernlogin-Service zugewiesen, der bei Verbindungen zur Terminal-Anmeldeadresse automatisch aufgerufen wird.

Ist die Option *-p* angegeben, werden *module* als eine Liste von STREAMS-Modulen interpretiert, die der Horchprozeß pushen muß, bevor der hinzugefügte Dienst aktiviert wird. Die Module werden in der angegebenen Reihenfolge gepusht. *module* muß eine Liste von durch Komma getrennten Modulen sein, die keine Leerzeichen oder Tabulatoren enthalten darf.

Ist die Option *-w* angegeben, wird *name* als der Benutzername aus der Datei */etc/passwd* interpretiert, den der Horchprozeß suchen soll. Über den Benutzernamen erhält der Horchprozeß die Benutzernummer, die Gruppennummer(n) und das Home-Verzeichnis zur Verwendung durch den Server. Ist *-w* nicht angegeben, wird standardmäßig der Benutzername *listen* verwendet.

Ein Dienst muß dem Horchprozeß für jedes Netzwerk, auf dem er verfügbar sein soll, ausdrücklich hinzugefügt werden. Diese Operation wird normalerweise nur dann ausgeführt, wenn der Dienst auf einem Rechner installiert ist, oder wenn die Liste der Dienste für ein neues Netzwerk zusammengestellt wird.

```
nlsadmin -r dienstcode netz_bez
```

Entfernt den Eintrag für den *dienstcode* aus der Dienstliste dieses Horchprozesses. Dies geschieht normalerweise nur im Zusammenhang mit der Deinstallation eines Dienstes aus einem Rechner.

```
nlsadmin -e dienstcode netz_bez
```

```
nlsadmin -d dienstcode netz_bez
```

Aktiviert bzw. deaktiviert den mit *dienstcode* bezeichneten Dienst für das genannte Netzwerk. Der Dienst muß zuvor in den Horchprozeß für dieses Netzwerk eingefügt worden sein (siehe die bereits beschriebene Option *-a*). Durch das Deaktivieren des Dienstes werden spätere Anforderungen für diesen Dienst abgelehnt, aber noch laufende Prozesse aus früheren Dienstanforderungen arbeiten unbeeinträchtigt weiter.

```
nlsadmin -s netz_bez
nlsadmin -k netz_bez
```

Startet bzw. beendet den Horchprozeß für das angegebene Netzwerk. Diese Operationen werden normalerweise als Teil der Systemstart- bzw. -abschlußprozeduren ausgeführt. Bevor ein Horchprozeß für ein bestimmtes Netzwerk aktiviert werden kann, muß er initialisiert werden (siehe die Option *-i*). Wird ein Horchprozeß beendet, arbeiten durch frühere Dienstanforderungen aktivierte, noch laufende Prozesse unbeeinträchtigt weiter.

Mit der Service Access Facility (Administrations-Datenbank für den TCP/IP Netzwerk-Service-Prozeß) können mehrere Exemplare des Horchprozesses auf einer *netz_bez* aktiv sein. Bei allen oben beschriebenen Kommandos kann die Option *-N anschl_monitor_etikett* anstelle des Arguments *netz_bez* benutzt werden. Dieses Argument gibt das Etikett an, an dem ein Exemplar des Horchprozesses von der Service Access Facility zu erkennen ist. Wird die Option *-N* nicht angegeben (d.h. *netz_bez* wird im Aufruf angegeben), wird angenommen, daß das letzte Element von *netz_bez* das Etikett des Horchprozesses darstellt, für den die Operation durchgeführt wird. Mit anderen Worten, es wird angenommen, daß in einer angegebenen *netz_bez* mindestens ein Horchprozeß existiert und daß dessen Etikett dem letzten Element von *netz_bez* entspricht. Dieser Horchprozeß kann als Primär- oder Standardhorchprozeß für eine bestimmte *netz_bez* angesehen werden.

nlsadmin wird auch zusammen mit den Kommandos der Service Access Facility verwendet. Dabei können die folgenden Kombinationen von Optionen verwendet werden:

```
nlsadmin -V
```

Schreibt die aktuelle Versionsnummer der Verwaltungsdatei des Horchprozesses in die Standardausgabe. Diese Angabe bildet einen Teil der *sacadm*-Kommandozeile, wenn *sacadm* einen Anschlußmonitor in das System einfügt.

```
nlsadmin -c kom | [-o streamname [-p module] [-A adresse | -D] [-R prognr:versnr]
```

Formatiert anschußmonitor-spezifische Daten, so daß sie als Argument zu *pmadm*(1M) verwendet werden können.

Option *-c* gibt den vollständigen Pfadnamen des Servers und seine Argumente an. *kom* muß gegenüber der Shell als Einzelwort erscheinen; die Argumente müssen daher in Anführungszeichen eingeschlossen sein.

Die Option *-o* gibt den vollständigen Pfadnamen einer FIFO-Datei oder eines benannten STREAM an, durch die die Verbindung zu einem permanenten Server tatsächlich hergestellt wird.

Ist die Option *-p* angegeben, werden die *module* als eine Liste von STREAMS-Modulen interpretiert, die der Horchprozeß pushen muß, bevor der hinzugefügte Dienst gestartet werden kann. Die Module werden in der Reihenfolge gepusht, in der sie eingegeben wurden. *module* müssen in der Liste durch Komma getrennt sein, Leerzeichen sind nicht zulässig.

Ist die Option `-A` angegeben, wird *adresse* als private Adresse des Servers interpretiert. Der Horchprozeß überwacht diese Adresse für den Dienst und leitet alle Aufrufe, die dort ankommen, direkt an den angegebenen Dienst weiter. Diese Option darf nicht zusammen mit der Option `-D` verwendet werden.

Wird die Option `-D` angegeben, wird dem Dienst dynamisch eine private Adresse zugewiesen, d.h. der Horchprozeß läßt den Transport-Provider die Adresse auswählen, wenn er für diesen Dienst zu horchen beginnt. Bei RPC-Diensten wird diese Option oft zusammen mit der Option `-R` verwendet, um die dynamisch zugewiesene Adresse im `rpcbinder` einzutragen. Diese Option darf nicht zusammen mit der Option `-A` verwendet werden.

Wird die Option `-R` angegeben, handelt es sich um einen RPC-Dienst, dessen Adresse, Programmnummer und Versionsnummer für diesen Transport-Provider im `rpcbinder` eingetragen werden müssen. Dies geschieht immer dann, wenn der Horchprozeß für den Dienst seine Arbeit aufnimmt. *prognum* und *versnum* sind Programm- bzw. Versionsnummer des RPC-Dienstes.

`nlsadmin` kann von allen Benutzern zum Erstellen von Berichten verwendet werden. Operationen, die den Status oder die Konfiguration eines Horchprozesses verändern, dürfen jedoch nur von privilegierten Benutzern ausgeführt werden.

Die spezifischen Optionen der Service Access Facility dürfen nicht zusammen mit anderen Optionen angegeben werden.

SIEHE AUCH

`listen(1M)`, `pmadm(1M)`, `rpcbind(1M)`, `sacadm(1M)`
Network Programmer's Guide

HINWEIS

Dynamisch zugewiesene Adressen werden nicht wie statisch zugewiesene in Berichten angezeigt.

BEZEICHNUNG

nslookup – Dialogabfrage von Name-Servern

ÜBERSICHT

nslookup [-l] [*adresse*]

BESCHREIBUNG

nslookup ist ein Dialogprogramm zum Abfragen von Name-Servern von ARPA Internet-Domänen. Der Benutzer kann sich mit den Servern in Verbindung setzen, um Daten zu einem bestimmten Host anzufordern oder eine Liste der Hosts der Domäne auszugeben.

OPTIONEN

- l Verwendet den Server des lokalen Hosts anstelle der Server in /etc/resolve.conf (existiert die Datei /etc/resolve.conf nicht oder enthält sie keine Server-Daten, bleibt die Option -l ohne Wirkung).
- adresse* Verwendet den Name-Server auf dem Host-System mit der angegebenen Internet-Adresse.

VERWENDUNG**Übersicht**

Der Adreßbereich der Internet-Domäne hat eine Baumstruktur mit derzeit vier Domänen auf oberster Ebene:

- COM Wirtschaftsunternehmen
- EDU Bildungseinrichtungen
- GOV Behörden
- MIL MILNET-Hosts

Wird ein bestimmter Host gesucht, müssen Informationen über dessen Zugehörigkeit zu Organisationen bekannt sein, um die oberste Domäne zu bestimmen. Wird beispielsweise die Internet-Adresse eines Systems der UCLA (Universität von Kalifornien in Los Angeles) gesucht, ist folgendermaßen vorzugehen:

- Über das Kommando `root` die Verbindung zum Root-Server herstellen. Der Root-Server des Adreßbereichs enthält Angaben zu den obersten Domänen.
- Da es sich bei UCLA um eine Universität handelt, lautet der Domänenname `ucla.edu`. Die Verbindung zu einem Server für die Domäne `ucla.edu` wird mit dem Kommando `server ucla.edu` hergestellt. Daraufhin werden die Namen von Hosts ausgegeben, die als Server für diese Domäne dienen. Bitte beachten Sie: Der Root-Server enthält keine Informationen über `ucla.edu`, sondern die Namen und Adressen der entsprechenden Hosts. Sobald der Root-Server den Name-Server der UCLA gefunden hat, werden alle weiteren Abfragen dorthin geleitet.

- Zum Anfordern von Daten zu einem bestimmten Host in der Domäne (z.B. lok) wird lediglich der Name des Hosts eingegeben. Zum Anfordern einer Liste der Hosts in der Domäne UCLA wird das Kommando `ls` verwendet. Zu diesem Kommando muß als Argument ein Domänenname angegeben werden (in diesem Fall `ucla.edu`).

Besteht eine Verbindung mit einem Name-Server, der für mehrere Domänen arbeitet, muß bei jeder Suche nach Host-Namen die entsprechende Domäne angegeben werden. Die Domäne `harvard.edu` beispielsweise wird von dem Server `seismo.css.gov` bedient, der auch die Domänen `css.gov` und `cornell.edu` bearbeitet. Eine Suchanforderung für den Host `aiken` in der Domäne `harvard.edu` muß als `aiken.harvard.edu` angegeben werden. Mit den Kommandos

```
set domain=name
```

und

```
set defname
```

kann an jede Anforderung automatisch ein Domänenname angehängt werden.

Wurde ein Host gefunden, kann mit dem Kommando `finger` angezeigt werden, wer in dem System arbeitet, oder es können Angaben zu einer bestimmten Person abgerufen werden. Um weitere Informationen über den Host abzufragen, wird das Kommando

```
set querytype=wert
```

eingegeben, mit dem der Typ der gewünschten Daten angegeben und ein weiterer Suchdurchgang angefordert wird. Bei dem Kommando `finger` muß der Datentyp `A` sein.

Kommandos

Zum Beenden wird Ctrl-D (EOF - Ende der Datei) eingegeben. Die Kommandozeile muß weniger als 80 Zeichen lang sein. Ein nicht erkanntes Kommando wird als Host-Name interpretiert.

host [*server*]

Sucht Informationen zum *host* über den aktuellen Standard-Server oder über *server*, sofern hier ein Wert angegeben ist.

server *domäne*

lserver *domäne*

Ändert den Standard-Server in *domäne*. *lserver* benutzt den ursprünglichen Server, um Informationen zu *domäne* zu suchen, während *server* den aktuellen Standard-Server benutzt. Ist eine autoritative Antwort nicht zu finden, werden die Namen von Servern ausgegeben, in denen die Antwort eventuell enthalten ist.

root Ändert den Standard-Server in den Server für die Root des Adreßbereichs der Domäne. Aktuell wird der Host `sri-nic.arpa` benutzt; dieses Kommando ist ein Synonym für `lserver sri-nic.arpa`. Der Name des Root-Server läßt sich mit dem Kommando `set root` ändern.

`finger [ name ]`

Stellt eine Verbindung zum finger-Server auf dem aktuellen Host her, der bei einer früheren erfolgreichen Suche nach Adreßdaten für einen Host definiert wurde (siehe das Kommando `set querytype = A`). Wie bei der Shell kann die Ausgabe mit `>` und `>>` in eine angegebene Datei umgeleitet werden.

`ls [-ah]`

Listet die für *domäne* vorhandenen Informationen auf. Die Standardausgabe umfaßt die Host-Namen und deren Internet-Adressen. Die Option `-a` listet die Aliasnamen der Hosts in der Domäne auf. Die Option `-h` listet die Informationen zur CPU und zum Betriebssystem für die Domäne auf. Wie bei der Shell kann die Ausgabe mit `>` und `>>` in eine angegebene Datei umgeleitet werden. Wird die Ausgabe in eine Datei geleitet, werden nach jeweils 50 vom Server empfangenen Datensätzen Hash-Markierungen ausgegeben.

`view dateiname`

Sortiert die Ergebnisse des Kommandos `ls` mit `more(1)` und listet sie auf.

`help`

`?`

Gibt eine kurze Übersicht über die Kommandos aus.

`set schlüsselwort [ = wert ]`

Dieses Kommando dient zum Ändern von Statusinformationen, die die Suche beeinflussen. Die gültigen Schlüsselwörter lauten:

`all` Gibt alle aktuellen Werte der verschiedenen Optionen zu `set` aus. Angaben zum aktuellen Standardserver und -Host werden ebenfalls ausgegeben.

`[ no ] deb [ug]`

Aktivieren des Fehlersuchmodus. Hier werden wesentlich ausführlichere Informationen über das an den Server gesendete Paket und die Antwort darauf ausgegeben. Der Standardwert ist `nodebug`.

`[ no ] def [name]`

Anhängen des Standard-Domänennamens an jede Suchanforderung. Der Standardwert ist `nodefname`.

`do [main ]= dateiname`

Ändern des Standard-Domänennamens in *dateiname*. Der Standarddomänenname wird an alle Suchanforderungen angehängt, wenn die Option `defname` angegeben ist. Der Standardwert ist der Wert in `/etc/resolve.conf`.

`q [ querytype ] = wert`

Ändern des bei einer Abfrage zurückzugebenden Informationstyps:

A Internet-Adresse des Hosts (Standardwert).
 CNAME Standardname für einen Aliasnamen.
 HINFO CPU- und Betriebssystemtyp des Hosts.
 MD Ziel der Post.
 MX Nachrichtenaustauscher.
 MB Domänenname des elektronischen Briefkastens.
 MG Mitglied des Postverteilers.
 MINFO Informationen zum elektronischen Briefkasten bzw. zum Verteiler.

(Andere in dem RFC883-Dokument angegebene Typen sind gültig, aber nicht sonderlich nützlich.)

`[ no ] recurse`

Weist den Name-Server an, andere Server abzufragen, wenn er die Informationen nicht enthält. Der Standardwert ist `recurse`.

`ret [ry] = zähler`

Legt in `zähler` fest, wie oft eine Anforderung wiederholt werden soll. Geht innerhalb einer bestimmten Zeitspanne (die mit `set timeout` definiert wird) keine Antwort auf eine Anforderung ein, wird die Anforderung erneut gesendet. Der Standardwert für `zähler` ist 2.

`ro [ot] = host`

Ändern des Namens des Root-Servers in `host`. Dies wirkt sich auf das Kommando `root` aus. Der standardmäßige Root-Server ist `scri-nic.arpa`.

`t [timeout] = wartezeit`

Ändern der Wartezeit für eine Antwort auf `wartezeit` Sekunden. Die standardmäßige `wartezeit` ist 10 Sekunden.

`[ no ] v [c]`

Beim Senden von Anforderungen an den Server wird immer eine virtuelle Verbindung verwendet. Der Standardwert ist `novc`.

DATEIEN

`/etc/resolve.conf` ursprünglicher Domänenname und Adressen von Name-Servern.

SIEHE AUCH

`named(1M)`, `resolver(3N)`, `resolve.conf(4)`, RFC 882, RFC 883.

FEHLERMELDUNGEN

Ist die Suchanforderung fehlgeschlagen, wird eine Fehlermeldung ausgegeben. Folgende Fehler sind möglich:

Time-out

Der Server reagierte auf eine Anforderung nicht innerhalb einer bestimmten Zeitspanne (mit `set timeout = wert` zu ändern) und nach einer bestimmten Anzahl von Wiederholungen (mit `set retry = wert` zu ändern).

No information

Je nach dem mit dem Kommando `set querytype` eingestellten Abfragetyp waren keine Angaben zum Host zu finden, obwohl der Host-Name gültig ist.

Non-existent domain

Der Host- oder Domänenname existiert nicht.

Connection refused**Network is unreachable**

Die Verbindung zum Name- oder finger-Server konnte zum aktuellen Zeitpunkt nicht hergestellt werden. Dieser Fehler kommt oft bei finger-Anforderungen vor.

Server failure

Der Name-Server stellte eine interne Diskrepanz in seiner Datenbank fest und konnte keine gültige Antwort zurückgeben.

Refused

Der Name-Server hat die Anforderung abgelehnt.

Der folgende Fehler dürfte nicht vorkommen, denn er deutet auf einen Programmfehler hin.

Format error

Der Name-Server stellte fest, daß das Anforderungspaket nicht das richtige Format hatte.

BEZEICHNUNG

`nvram` – Anzeigen und Ändern von Daten im NVRAM

ÜBERSICHT

```
nvram [-fsamq] [-d "[ dumpdev (minor# [, offset] ) ] dumpdev_unix
      [ [ -D specdumpdev (minor#) ] -U specdumpdev_unix [ -a [ -o ] ] " ]
      [-u default_loc ] [-b boot_loc ] [-t args ] [ device ]
```

BESCHREIBUNG

Auf der CPU-Baugruppe existiert ein nichtflüchtiger Speicherbereich in Form eines 32 KB großen NVRAM (non-volatile RAM). Das NVRAM ist batteriegepuffert, so daß sein Inhalt auch ohne Netzspannung erhalten bleibt. Mit dem Kommando `nvram` können Informationen, die im NVRAM abgelegt sind, angezeigt oder auch geändert werden.

Diese Informationen umfassen:

- die Konfigurationsparameter der Firmware bezüglich Konsol- und Teleservice-Schnittstelle. Sie können vom Kommando `nvram` gelesen werden. Die Einstellungen der Teleservice-Schnittstelle können auch verändert werden.
- die "Boot Load Location". Das sind Name und Adresse des Geräts, von dem die Firmware den Boot laden soll. Die Boot Load Location kann vom Kommando `nvram` gelesen und geschrieben werden. Die Location wird als Geräteadresse im Firmware-Format angegeben (s.u.).
- die "Boot Active Location". Das sind Name und Adresse des Geräts, von dem die Firmware den Boot tatsächlich geladen hat. Die Boot Active Location kann vom Kommando `nvram` nur gelesen werden. Sie wird als Geräteadresse im Firmware-Format angegeben (s.u.).
- die "Boot Default File Location". Das sind Name und Geräteadresse der Datei, die die Default-Parametereinstellungen für den Boot enthält. Die Boot Default File Location kann vom Kommando `nvram` gelesen und geschrieben werden.
- die "Dump Location". Das sind Name und Adresse des Geräts, auf das ggf. ein Speicherabzug zu schreiben ist. Die Dump Location kann vom Kommando `nvram` gelesen und geschrieben werden.
- den Nachrichtenpuffer. Dieser Bereich kann vom Kommando `nvram` nur gelesen, nicht aber beschrieben werden. In ihm werden Meldungen der Firmware, des Boots und des SINIX-Kerns abgespeichert.

Die Adressen von "Boot Load Location" und "Boot Active Location" sind im Firmware-Format angegeben. Dessen Syntax ist

```
is (geraet dist)
```

`is` bezeichnet den Multicontroller Interphase Storer III. Nur dieses Boot-Gerät wird z.Zt. von der Firmware unterstützt. `geraet` gibt die Gerätenummer an.

Die Gerätenummer besteht aus bis zu 3 Ziffern.

```

123
|||
||+- Scheiben-Nummer
|+- Laufwerksnummer
+- Controller-Nummer

```

Die Laufwerksnummer ist

```

0 für die erste Festplatte
1 für die zweite Festplatte
2 für die erste Diskette
3 für die zweite Diskette

```

dist gibt eine Distanzadresse auf dem Gerät in KBytes an.

Die Numerierung beginnt jeweils mit 0.

Beispiel

is(120,0) bezeichnet die Adresse 0 auf der ersten Diskette am zweiten Storage Controller.

Achtung

Dieses Format wird nur von der Firmware verwendet. Im Standalone-Format sehen die Adressen anders aus! (vgl. *boot*(1M)).

nvr_{am} kennt folgende Optionen:

- f gibt die Einstellungen der Firmware für die Konsol- und die Teleservice-Schnittstelle aus.
 - s gibt die Boot Load Location, die Boot Active Location, die Boot Default File Location und die Dump Location aus.
 - a wirkt wie -f und -s zusammen.
 - m gibt den Inhalt des Nachrichtenpuffers aus. Vor dem eigentlichen Inhalt werden die Größe und Schreibposition des Nachrichtenpuffers angezeigt.
 - q Fragt die "Boot Active Location" ab und gibt im Firmware-Format aus, von wo gebootet wurde.
 - d "[[*dumpdev* (*minor#* [, *offset*])] *dumpdev_unix*]
[[-D *specdumpdev* (*minor#*)] -U *specdumpdev_unix* [-a] [-o]]"
- stellt die Dump Location neu ein. Die einzelnen Parameter zu -d können in beliebiger Reihenfolge angegeben werden. Ist *dumpdev* nicht angegeben, so wird standardmäßig zuerst nach /dev/rdump, dann nach /dev/rswap gesucht.

dumpdev (*minor#*)

Gerät, auf das der Speicherabzug geschrieben werden soll, z.B. *hd*(2). Dieses Gerät ist die Standardpartition, die auch verwendet wird, wenn beim Dumpen auf das mit den Schaltern -D und -U bezeichnete Gerät ein Fehler auftritt.

- offset* Offset. Bei `/dev/rdump` wird immer der Offset 0 genommen. Bei `/dev/rswap` und bei `specdumpdev` wird ein angegebener Offset ignoriert. Bei allen anderen Geräten wird eine Warnung ausgegeben. Wird ein anderer Name für `/dev/rswap` oder `/dev/rdump` verwendet, so wird der Offset nicht entsprechend korrigiert, sondern die übliche Warnung ausgegeben.
- dumpdev _unix* Name der Gerätedatei für ein logisches Laufwerk, z.B. `/dev/rswap`. In diesem Laufwerk findet `savecore` den Speicherabzug, wenn nicht die Schalter `-D` und `-U` gesetzt sind oder ein Fehler bei dem Gerät auftritt, das diese Schalter bezeichnen. Es handelt sich um dasselbe Gerät wie `dumpdev(minor#)`, nur diesmal in SINIX-Darstellung. Das Gerät muß zeichenorientiert sein (character device). Andernfalls wird mit einer Fehlermeldung abgebrochen. Der Name der Gerätedatei muß mit `/dev` beginnen. Bezeichnen `dumpdev_unix` und `dumpdev(minor#)` nicht dasselbe Gerät, so wird ebenfalls abgebrochen.
- `-D specdumpdev(minor#)` Gibt ein spezielles Dumpgerät an, auf das der Speicherabzug geschrieben werden soll, z.B. `ts(0)`, `ex(0)` oder `hd(1036)`. Bei Streamer- oder Exabyte-Geräten muß ein Rewind-Gerätename angegeben werden. Sonst wird eine Warnung ausgegeben, eine automatische Korrektur erfolgt nicht.
- `-U specdumpdev _unix` Name der Gerätedatei für ein logisches Laufwerk, z.B. `/dev/tapen`, `/dev/exa8` oder `/dev/rdisk/c0d1s1L`. In diesem Laufwerk findet `savecore` den Speicherabzug, wenn dieser auf ein spezielles Gerät geschrieben wurde. Es handelt sich um dasselbe Gerät wie `specdumpdev(minor#)`, nur diesmal in SINIX-Darstellung. Das Gerät muß mit `savecore` initialisiert werden. Das Gerät muß zeichenorientiert sein (character device). Andernfalls, wird mit einer Fehlermeldung abgebrochen. Der Name der Gerätedatei muß mit `/dev` beginnen. Bezeichnen `dumpdev_unix` und `dumpdev(minor#)` nicht dasselbe Gerät, so wird ebenfalls abgebrochen. Bei Streamer- oder Exabyte-Geräten muß ein No-Rewind-Gerätename angegeben werden.
- `-a` Ein Dialog mit dem Benutzer findet statt, wenn auf Streamer oder Exabyte gedumpt werden soll und sich keine Kassette im Laufwerk befindet.
- `-o` Ein bereits gesicherter Hauptspeicherabzug darf überschrieben werden.
- `-u default_loc` stellt die Boot Default File Location neu ein. `default_loc` wird im Standalone-Format angegeben.
- `-b boot_loc` stellt die Boot Load Location neu ein. `boot_loc` wird im Firmware-Format angegeben.
- `-t args` ändert die Einstellungen der Teleservice-Schnittstelle. `args` ist eine Zeichenkette der Form `arg0:arg1:::argn`

nvrnm (1M)**(MX300)****nvrnm (1M)**

argi kann folgende Werte annehmen (jeweils maximal einen aus einer Zeile):

300,600,1200,2400,4800,9600,19200,38400	(Baudrate)
cs7,cs8	(7/8 Bit)
stop	(2 Stoppbits)
parodd,parevn,parno	(Parität)
tse	(Teleservice aktiv)
s1	(Enable Teleservice)
s2	(Bewertung des V24-Signals S1)
s4	(Bewertung des V24-Signals S2)
m1	(Bewertung des V24-Signals S4)
m2	(Bewertung des V24-Signals M1)
m3	(Bewertung des V24-Signals M2)
m4	(Bewertung des V24-Signals M3)
m5	(Bewertung des V24-Signals M4)
	(Bewertung des V24-Signals M5)

Der Parameter *device* bezeichnet die Gerätedatei des NVRAMs und braucht im Normalfall nicht angegeben zu werden. Standardwert ist `/dev/nvrnm`.

BEISPIELE

```
nvrnm -d "/dev/rdump -o"
nvrnm -d "hd(2) /dev/rswap -o"
nvrnm -d "hd(2) /dev/rswap -D ex(0) -U /dev/exa8 -a"
nvrnm -d "-U /dev/rdisk/c0d1s12"
nvrnm -d "hd(2) /dev/rswap -D hd(1036) -U /dev/rdisk/c0d1s12"
nvrnm -d "-U /dev/tapen"
nvrnm -d "hd(2) "
```

Aus Kompatibilitätsgründen ist die Angabe `nvrnm -d "hd(2) "` erlaubt.

SIEHE AUCH

`boot(1M)` `savecore(1M)`

BEZEICHNUNG

`offline` – Zustand der Prozessoren anzeigen oder ändern

ÜBERSICHT

`/etc/offline [-a] [-v] [prozessornummer ...]`

BESCHREIBUNG

`offline` zeigt den Zustand eines oder mehrerer Prozessoren auf dem Multiprozessor-Board an oder ändert deren Zustand. Gültige Prozessornummern sind 0, 1, .. bis zur Anzahl der aktiven Prozessoren.

HINWEIS: Mit dem Kommando `wpc N=N` des Einschaltmonitors (power on monitor) können Prozessoren vor dem Laden des Betriebssystems dekonfiguriert werden.

Bei `offline` können Sie folgende Optionen angeben:

- `-a` Alle Prozessoren bis auf Prozessor 0 deaktivieren.
- `-v` Report über den augenblicklichen Zustand und Ablauf ausgeben.

SIEHE AUCH

`online(1M)`

BEZEICHNUNG

online – Zustand der Prozessoren anzeigen oder ändern

ÜBERSICHT

/etc/online [-a] [-v] [prozessornummer ...]

/etc/online -N

/etc/online -S

BESCHREIBUNG

online zeigt den Zustand eines oder mehrerer Prozessoren auf dem Multiprozessor-Board an oder ändert deren Zustand. Gültige Prozessornummern sind 0, 1, .. bis zur Anzahl der aktiven Prozessoren.

HINWEIS: Mit dem Kommando `wpc N=N` des Einschaltmonitors (power on monitor) können Prozessoren vor dem Laden des Betriebssystems dekonfiguriert werden.

Bei online können Sie folgende Optionen angeben:

- a Alle Prozessoren aktivieren.
- v Report über den augenblicklichen Zustand und Ablauf ausgeben.
- S Liste der aktiven Prozessoren ausgeben.
- N Anzahl der aktiven Prozessoren ausgeben.

SIEHE AUCH

offline(1M)

BEZEICHNUNG

passmgmt – Verwaltung der Paßwortdateien

ÜBERSICHT

passmgmt -a *optionen name*

passmgmt -m *optionen name*

passmgmt -d *name*

BESCHREIBUNG

Das Kommando `passmgmt` aktualisiert die Daten in den Paßwortdateien. Es arbeitet in den Dateien `/etc/passwd` und `/etc/shadow`.

Mit `passmgmt -a` wird ein Eintrag für den Benutzer *name* in die Paßwortdateien eingefügt. Dieses Kommando legt kein Verzeichnis für den neuen Benutzer an, und die neue Anmeldung bleibt gesperrt (im Paßwortfeld steht die Zeichenkette `*LK*`) bis mit dem Kommando `passwd(1)` das Paßwort definiert wurde.

Mit `passmgmt -m` wird der Eintrag für den Benutzer *name* in den Paßwortdateien geändert. Das Namensfeld eines Eintrags in der Datei `/etc/shadow` und alle Felder, außer dem Paßwortfeld, eines Eintrags in der Datei `/etc/passwd` können mit diesem Kommando geändert werden. Es werden nur die Felder geändert, die in der Kommandozeile angegeben werden.

Mit `passmgmt -d` wird der Eintrag für den Benutzer *name* aus den Paßwortdateien gelöscht. Dateien, die dem Benutzer gehören, werden dadurch jedoch nicht entfernt; sie müssen manuell gelöscht werden.

Folgende Optionen stehen zur Verfügung:

- c *kommentar* Eine Kurzbeschreibung der Anmeldung. Sie darf 128 Zeichen lang sein und ist standardmäßig ein leeres Feld.
- h *home-ver* Home-Verzeichnis von *name*. Ist auf eine Länge von maximal 256 Zeichen begrenzt und hat den Standardwert `/home/name`.
- u *bnr* Benutzernummer von *name*. Diese Nummer muß im Bereich zwischen 0 und dem größten nicht negativen Wert für das System liegen. Der Standardwert ist die erste verfügbare Benutzernummer über 99. Ohne die Option -o wird die Eindeutigkeit der Benutzernummer durchgesetzt.
- o Durch diese Option entfällt der Zwang, daß eine Benutzernummer eindeutig sein muß. Sie wird nur zusammen mit der Option -u verwendet.
- g *gnr* Gruppennummer von *name*. Diese Nummer muß im Bereich zwischen 0 und dem größten nicht negativen Wert für das System liegen. Der Standardwert ist 1.
- s *shell* Login-Shell für *name*. Hier sollte der vollständige Pfadname des Programms angegeben werden, das ausgeführt wird, wenn sich der Benutzer anmeldet. *shell* darf maximal 256 Zeichen groß sein. Der Standardwert ist ein leeres Feld, das als `/usr/bin/sh` interpretiert wird.

-l logname

Diese Option ändert den *namen* in logname. Sie wird nur zusammen mit der Option -m verwendet.

Die Gesamtgröße der einzelnen Login-Einträge ist auf 511 Byte in jeder der beiden Paßwortdateien begrenzt.

DATEIEN

/etc/passwd
/etc/shadow
/etc/opasswd
/etc/oshadow

SIEHE AUCH

useradd(1M), userdel(1M), usermod(1M), passwd(4), und shadow(4) im *Referenzhandbuch für Systemverwalter*.
passwd(1) in den *Kommandos*.

EXIT-CODES

Die Exit-Codes von passmgmt haben folgende Bedeutung:

- 0 = erfolgreiche Ausführung
- 1 = Berechtigung verweigert
- 2 = ungültige Kommandosyntax; es wird die Belegungsmeldung des Kommandos passmgmt angezeigt
- 3 = ungültiges Argument zur Option angegeben
- 4 = Benutzernummer wird benutzt
- 5 = Diskrepanzen in den Paßwortdateien (z.B. *name* ist in der Datei /etc/passwd und nicht in der Datei /etc/shadow vorhanden oder umgekehrt)
- 6 = unerwarteter Fehler; Paßwortdateien unverändert
- 7 = unerwarteter Fehler; keine Paßwortdatei(en) vorhanden
- 8 = Paßwortdatei(en) belegt; später erneut ansprechen
- 9 = *name* existiert nicht (bei Option -m oder -d), besteht bereits (bei Option -a) oder logname besteht bereits (bei Option -m -l)

HINWEIS

Doppelpunkt oder Wagenrücklaufzeichen dürfen nicht als Teil eines Arguments verwendet werden, weil diese Zeichen in der Paßwortdatei als Feldtrennzeichen interpretiert werden.

Dieses Kommando ist in künftigen Versionen nicht mehr vorhanden. Seine Funktionen wurden durch die Kommandos useradd, userdel und usermod ersetzt und erweitert. Diese Kommandos sind bereits verfügbar.

BEZEICHNUNG

ping – sendet ICMP ECHO_REQUEST-Pakete an Netzwerk-Hosts

ÜBERSICHT

ping *host* [*wartezeit*]

/usr/sbin/ping [-s] [-lrRv] .I *host* [.I *paketgröße*] [.I *zähler*]

BESCHREIBUNG

ping verwendet das Datagramm ECHO_REQUEST des ICMP-Protokolls, um von dem angegebenen *host* oder Netzwerk-Gateway eine ICMP ECHO_RESPONSE zu erhalten. Antwortet *host*, gibt ping auf der Standardausgabe die folgende Meldung aus und endet dann: *host is alive* (*host* ist aktiv). Anderenfalls wartet das Kommando *wartezeit* Sekunden und gibt dann folgende Meldung aus: *no answer from host* (*host* antwortet nicht). Der Standardwert für *wartezeit* beträgt 20 Sekunden.

Wird die Option *-s* angegeben, sendet ping ein Datagramm pro Sekunde und gibt eine Ausgabezeile pro empfangener ECHO_RESPONSE-Antwort aus. Geht keine Antwort ein, wird keine Ausgabe erzeugt. In dieser zweiten Form berechnet ping die Umlaufzeiten und Paketverluststatistiken; am Ende oder bei einer Überschreitung der Wartezeit wird eine Zusammenfassung dieser Daten ausgegeben. Die Standardgröße des Datagrammpakets beträgt 64 Byte; mit dem Argument *paketgröße* kann jedoch auch ein anderer Wert definiert werden. Wird ein optionaler *zähler* angegeben, sendet ping nur die entsprechende Anzahl von Anfragen.

Wird ping zur Fehlereingrenzung verwendet, muß zuerst der lokale Host mit diesem Kommando überprüft werden, um sicherzugehen, daß die lokale Netzwerkschnittstelle aktiv ist.

OPTIONEN

- l Beliebiger Quellenweg. Mit dieser Option im IP-Kopf wird das Paket zu dem angegebenen Host und zurück gesendet. Wird normalerweise zusammen mit der Option *-R* verwendet.
- r Umgeht die normalen Routingtabellen und sendet direkt an einen Host in einem angeschlossenen Netzwerk. Wenn der Host sich nicht in einem direkt angeschlossenen Netz befindet, wird ein Fehler zurückgegeben. Mit dieser Option kann ein lokaler Host über eine Schnittstelle, die vom Routingdämon aufgegeben wurde, mit dem Kommando ping überprüft werden (siehe *routed(1M)*).
- R Aufzeichnen der Route. Setzt die IP-Option zum Aufzeichnen der Route, wodurch die Route des Pakets im IP-Kopf gespeichert wird. Der Inhalt der aufgezeichneten Route wird nur ausgegeben, wenn die Option *-v* angegeben wird, und nur dann für zurückgesendete Pakete angegeben, wenn der Ziel-Host die Option zum Aufzeichnen der Route beibehält, oder wenn die Option *-l* angegeben wird.
- v Ausführliche Ausgabe. Listet alle empfangenen ICMP-Pakete auf, die nicht ECHO_RESPONSE sind.

SIEHE AUCH

ifconfig(1M), netstat(1M), rpcinfo(1M), icmp(7).

BEZEICHNUNG

pkgadd – überträgt Software-Pakete auf das System

ÜBERSICHT

```
/usr/sbin/pkgadd [-d gerät] [-r antwort] [-n] [-a verw] [pktabl ...]
```

```
/usr/sbin/pkgadd -s spool [-d gerät] [pktabl ...]
```

BESCHREIBUNG

Mit pkgadd installieren Sie ein Software-Paket auf dem System. Dazu überträgt pkgadd den Inhalt eines Software-Pakets vom Datenträger oder -verzeichnis, über den/das die Software verteilt wird, auf das System. Ohne die Option *-d* sucht pkgadd das Paket im Standard-Spool-Verzeichnis */var/spool/pkg*. Wird die Option *-s* angegeben, liest pkgadd das Paket in ein Spool-Verzeichnis *spool* ein, anstatt das Paket zu installieren.

- d *gerät** Installiert oder kopiert ein Paket von *gerät*. *gerät* kann ein vollständiger Pfadname zu einem Verzeichnis oder ein Bezeichner für Magnetband, Diskette oder auswechselbare Platte (z.B. */var/tmp*, */dev/dsk/f0* oder */dev/dsk/f1*) sein. Auch die Angabe des Aliasnamens für das Gerät ist zulässig, wie z.B. *diskette1* oder *ctapel*.
- r *antwort** Bezeichnet eine Datei oder ein Verzeichnis, *antwort*, die die Ausgabe aus einer früheren pkgask-Sitzung enthält. Diese Datei enthält die Dialogantworten, die das Paket im Dialogbetrieb benötigt. *antwort* muß ein vollständiger Pfadname sein.
- n* Die Installation wird nicht interaktiv durchgeführt. Der Standardwert ist jedoch Dialogbetrieb.
- a *verw** Definiert eine Installationsverwaltungsdatei, *verw*, die anstelle der Standardverwaltungsdatei verwendet werden soll. Mit dem Terminalsymbol *none* wird die Verwendung einer Installationsverwaltungsdatei unterbunden und so der Dialog mit dem Benutzer erzwungen. Wird kein vollständiger Pfadname angegeben, sucht pkgadd die Datei im Verzeichnis */var/sadm/install/admin*.
- pktabl* Gibt das Exemplar des Pakets oder die Liste der Exemplare an, das/die installiert werden soll(en). Mit dem Terminalsymbol *all* können alle auf dem Quelldatenträger vorhandenen Pakete angegeben werden. Mit der Angabe *pktabl.** können alle Exemplare eines Pakets bezeichnet werden.
- s* Liest das Paket in das Verzeichnis *spool* ein, anstatt es zu installieren.

Wird das Kommando pkgadd ohne Optionen ausgeführt, verwendet es */var/spool/pkg*, das Standard-Spool-Verzeichnis.

HINWEIS

Bei der Übertragung eines Pakets auf ein Spool-Verzeichnis dürfen die Optionen *-r*, *-n* und *-a* nicht verwendet werden.

Die Option *-r* kann zum Angeben eines Verzeichnis- oder eines Dateinamens benutzt werden. Das Verzeichnis kann mehrere *antwort*-Dateien enthalten, die jeweils den Namen des Pakets tragen, dem sie zugeordnet sind. Diese Möglichkeit ließe sich beispielsweise nutzen, wenn mehrere Pakete im Dialogbetrieb mit einem

Aufruf von `pkgadd` installiert werden sollen. Jedes Paket kann eine *antwort*-Datei benötigen. Sollen die erstellten Antwortdateien denselben Namen haben wie das jeweilige Paket, (d.h. *paket1* und *paket2*), dann ist das Verzeichnis, in dem diese Dateien stehen, mit der Option `-r` anzugeben.

Die Option `-n` bewirkt, daß die Installation angehalten wird, wenn ergänzende Benutzereingaben erforderlich sind.

Ist es nicht gelungen, ein Paket zu installieren (Meldung: `package installation failed`), dann sind u.U. noch Teile des Pakets im System vorhanden, die Sie mit `pkgrm` löschen können. `pkginfo` bietet die Möglichkeit, nicht erfolgreich installierte Pakete zu erkennen.

BEZEICHNUNG

pkgask – speichert Antworten auf ein Anforderungs-Skript

ÜBERSICHT

/usr/sbin/pkgask [-d *gerät*] -r *antwort* [*pktabl* ...]

BESCHREIBUNG

Mit pkgask kann der Verwalter Antworten auf ein Paket, das im Dialog mit dem Benutzer installiert wird (ein Paket mit einem Anforderungs-Skript), speichern. Beim Aufruf dieses Kommandos wird eine Datei *antwort* angelegt, die bei der Installation als Eingabe verwendet wird. Dadurch werden bei der Installation keine Benutzereingaben erforderlich, denn die Datei *antwort* enthält bereits alle Informationen, die das Paket braucht.

- d *gerät* Aktiviert das Anforderungs-Skript für ein Paket auf *gerät*. *gerät* kann der Pfadname eines Verzeichnisses sein oder auch Bezeichner für Magnetband, Diskette oder auswechselbare Platte (z.B. /var/tmp, /dev/dsk/0s2 und /dev/dsk/f0t). Das Standardgerät ist das Installations-Spool-Verzeichnis.
- r *antwort* Bezeichnet eine Datei oder ein Verzeichnis, das angelegt werden muß, um die Antworten für den Dialog mit dem Paket aufzunehmen. Als Name muß ein vollständiger Pfadname angegeben werden. Die Datei bzw. das Verzeichnis kann später als Eingabe für das Kommando pkgadd verwendet werden.
- pktabl* Gibt die Paket-Variante bzw. die Liste der Varianten an, für die Anforderungs-Skripts erstellt werden. Mit dem Terminalsymbol `all` können alle auf dem Quelldatenträger verfügbaren Pakete erfaßt werden.

HINWEIS

Die Option -r kann zum Angeben eines Verzeichnis- bzw. Dateinamens verwendet werden. Der Verzeichnisname dient zum Anlegen mehrerer *antwort*-Dateien, die denselben Namen haben wie die Pakete, zu denen sie gehören. Diese werden beispielsweise verwendet, um mehrere Pakete, die einen Dialog mit dem Benutzer führen, mit einem Aufruf von pkgadd hinzuzufügen. Jedes Paket benötigt eine *antwort*-Datei. Zum Anlegen mehrerer Antwortdateien, die denselben Namen haben wie die zugehörige Paket-Variante, werden das Verzeichnis, in dem die Dateien angelegt werden sollen, und mehrere Variantennamen mit dem Kommando pkgask angegeben. Bei der Installation der Pakete kann dieses Verzeichnis als Argument zu dem Kommando pkgadd angegeben werden.

SIEHE AUCH

installf(1M), pkgadd(1M), pkgchk(1), pkgmk(1), pkginfo(1), pkgparam(1), pkgproto(1), pkgtrans(1), pkgrm(1M), removef(1M).

BEZEICHNUNG

pkgchk – prüft den Zustand der Installation

ÜBERSICHT

```
/usr/sbin/pkgchk [-l | -acfqv] [-nx] [-p pfad1[, pfad2 ...]] [-i datei] [pktvar...]
```

```
/usr/sbin/pkgchk -d gerät [-l | v] [-p pfad1[, pfad2 ...]] [-i datei] [pktvar...]
```

```
/usr/sbin/pkgchk -m pktabb [-e umgdat] [-l | -acfqv] [-nx] [-i datei]  
[-p pfad1[, pfad2 ...]]
```

BESCHREIBUNG

pkgchk prüft den Zustand installierter Dateien oder zeigt, wenn die Option `-l` verwendet wird, Informationen zu Paketdateien an. Das Kommando prüft die Integrität der Verzeichnisstrukturen und der Dateien. Diskrepanzen und eine ausführliche Erklärung des Problems werden in die Datei `stderr` geschrieben.

Mit der ersten Syntaxübersicht wird der Inhalt und/oder Attribute von aktuell auf dem System installierten Objekten aufgelistet oder überprüft. In der Kommandozeile können Paketnamen angegeben werden, da standardmäßig der gesamte Inhalt aller Speichergeräte des Rechners überprüft wird.

Mit der zweiten Syntax wird der Inhalt eines Pakets aufgelistet oder geprüft, das sich auf dem angegebenen Gerät befindet, aber nicht installiert wurde. Bitte beachten Sie, daß bei gespoolten Paketen Argumente nicht überprüft werden können.

Mit der dritten Syntax werden Inhalt und/oder Attribute von Objekten aufgelistet oder geprüft, die in der angegebenen *pktabb* beschrieben werden.

Die Optionen werden folgendermaßen definiert:

- l Erstellt eine Liste mit Daten über die ausgewählten Dateien, die ein Paket bilden. Ist nicht kompatibel mit den Optionen `a`, `c`, `f`, `g` und `v`.
- a Prüft nur die Dateiattribute, nicht den Inhalt der Datei. Standardmäßig wird beides überprüft.
- c Prüft nur den Dateiinhalt, nicht die Dateiattribute. Standardmäßig wird beides überprüft.
- f Korrigiert die Dateiattribute nach Möglichkeit. Wird zusätzlich die Option `-x` angegeben, werden verdeckte Dateien gelöscht. Wenn pkgchk mit dieser Option aufgerufen wird, legt das Kommando Verzeichnisse, benannte Pipes, Verweise und Geräte an, sofern sie noch nicht vorhanden sind.
- q Gibt keine Meldungen zu fehlenden Dateien aus.
- v Protokoll-Modus. Die Dateien werden bei der Überprüfung angezeigt.
- n Prüft temporäre oder editierbare Dateien nicht. Diese Option ist bei den meisten Prüfungen nach der Installation zu verwenden.
- x Durchsucht exklusive Verzeichnisse nach vorhandenen Dateien, die nicht in der Datenbank für Installations-Software oder der angegebenen *pktabb*-Datei enthalten sind.

pkgchk (1M)

pkgchk (1M)

- p Prüft nur die Genauigkeit des oder der aufgeführten Pfadnamen(s). *pfad* kann ein Name oder mehrere, durch Komma (oder durch Leerzeichen, wenn die Liste in Anführungszeichen steht) getrennte Pfadnamen angeben.
- i Liest eine Liste von Pfadnamen aus *datei* und vergleicht sie mit der Datenbank für Installations-Software oder der angegebenen Datei *pktabb*. Pfadnamen, die nicht in *eingabedatei* enthalten sind, werden nicht geprüft.
- d Gibt das Gerät an, auf dem sich ein gespooltes Paket befindet. *gerät* kann den Pfadnamen eines Verzeichnisses oder die Bezeichner für Magnetband, Diskette oder auswechselbare Platte angeben (z.B. */var/tmp* oder */dev/diskette*).
- m Fordert die Prüfung des Pakets anhand der *pktabb*-Datei.
- e Fordert, daß die als *umgd* bezeichnete *pktabb*-Datei zum Auflösen von Parametern verwendet wird, die in der angegebenen *pktabb*-Datei verzeichnet sind.
- pktvar* Gibt die zu prüfende Paketvariante(n) an. Mit dem Format *pktvar.** können alle Varianten eines Pakets überprüft werden. Standardmäßig werden alle Daten zu allen installierten Paketen angezeigt.

SIEHE AUCH

pkgadd(1M), pkgask(1M), pkginfo(1), pkgm(1M), pkgtrans(1).

BEZEICHNUNG

pkginfo – zeigt Informationen über Software-Pakete an

ÜBERSICHT

```
pkginfo [-q|x|l] [-p|i] [-a arch] [-v version]
        [-c kategorie[,kategorie ...]] [pktvar[,pktvar ...]]

pkginfo [-d gerät [-q|x|l] [-a arch] [-v version]
        [-c kategorie[,kategorie ...]] [pktvar[,pktvar ...]]
```

BESCHREIBUNG

pkginfo zeigt Informationen über Software-Pakete an, die im System installiert sind (erste Form des Kommandos), oder über Pakete, die auf einem bestimmten Gerät oder in einem bestimmten Verzeichnis stehen (zweite Form des Kommandos). Es werden nur der Name des Pakets und die Abkürzung für Pakete vor SVR4 angezeigt.

Die folgenden Optionen stehen zur Verfügung:

- q Mit dieser Option werden keine Informationen angezeigt, sondern es wird innerhalb eines Programms überprüft, ob ein Paket installiert wurde.
- x Bewirkt, daß Teilinformationen über die Pakete aufgelistet werden. Zu diesen Informationen gehören die Abkürzung für das Paket, der Name des Pakets, die Paketarchitektur (falls verfügbar) und die Paketversion (falls verfügbar).
- l Bewirkt, daß alle verfügbaren Informationen über die angegebenen Pakete ausgegeben werden.
- p Bewirkt, daß nur Informationen über teilweise installierte Pakete ausgegeben werden.
- i Bewirkt, daß nur Informationen über vollständig installierte Pakete ausgegeben werden.
- a *arch* Gibt die Architektur *arch* an. Hiermit werden die anzuzeigenden Pakete anhand der Architektur *arch* des Pakets ausgewählt.
- v *version* Hiermit werden die anzuzeigenden Pakete anhand der Version *version* ausgewählt. "Alle kompatiblen Versionen" können aufgeführt werden, indem der Versionsbezeichnung eine Tilde (~) vorangestellt wird. Mehrere Leer-/Tabulatorzeichen werden durch ein einziges Leerzeichen ersetzt, während die Versionen verglichen werden.
- c *kategorie* Hiermit werden die anzuzeigenden Pakete anhand der Kategorie *kategorie* ausgewählt. Kategorien sind im Kategorienfeld der pkginfo-Datei definiert. Werden mehrere Kategorien angegeben, muß das Paket nur mit einer dieser Kategorien übereinstimmen. Bei diesem Vergleich wird nicht zwischen Groß- und Kleinschreibung unterschieden.

pkginfo(1)

pkginfo(1)

<i>pktvar</i>	Bezeichnet die Variante eines Pakets. Eine Variante kann die Abkürzung des Pakets oder eine bestimmte Variante sein (zum Beispiel <code>inst.1</code> oder <code>inst.beta</code>). Alle Varianten eines Pakets können mit <code>inst.*</code> aufgeführt werden.
<code>-d gerät</code>	Hiermit wird ein Gerät, <i>gerät</i> , definiert, auf dem die Software gespeichert ist. <i>gerät</i> kann der Pfadname eines Verzeichnisses oder der Bezeichner für ein Bandlaufwerk, ein Diskettenlaufwerk, eine auswechselbare Festplatte <i>etc.</i> sein. Mit dem speziellen Token "spool" kann das Spool-Verzeichnis angegeben werden, das standardmäßig für die Installation verwendet wird.

HINWEIS

Ohne Optionen listet `pkginfo` die Primärkategorie, die Paketvariante und den Namen aller vollständig und teilweise installierten Pakete auf. Für jedes ausgewählte Paket wird eine Zeile ausgegeben.

Die Optionen `-p` und `-i` sind bedeutungslos, wenn sie zusammen mit der Option `-d` verwendet werden.

Die Optionen `-q`, `-x` und `-l` schließen sich gegenseitig aus.

`pkginfo` kann nicht feststellen, ob ein Paket einer Vorgängerversion von SINIX V5.4? (SVR4) nur teilweise installiert ist. Es geht davon aus, daß solche Pakete vollständig installiert sind.

SIEHE AUCH

`pkgadd(1M)`, `pkgask(1M)`, `pkgchk(1M)`, `pkgm(1M)`, `pkgtrans(1)`.

BEZEICHNUNG

`pkgparam` – zeigt Werte für Paketparameter an

ÜBERSICHT

`pkgparam [-v] [-d gerät] pktvar [param ...]`

`pkgparam -f datei [-v] [param ...]`

BESCHREIBUNG

`pkgparam` zeigt den zu dem/den in der Kommandozeile angeforderten Parameter(n) gehörenden Wert an. Die Werte befinden sich bei *pktvar* in der Datei *pkginfo* oder in der mit der Option `-f` angegebenen Datei.

Pro Zeile wird ein Parameterwert angezeigt. Ohne die Option `-v` wird nur der Wert des Parameters angezeigt. Wird diese Option angegeben, hat die Ausgabe dieses Kommandos das folgende Format:

```
parameter1='wert1'
parameter2='wert2'
parameter3='wert3'
```

Werden in der Kommandozeile keine Parameter angegeben, werden die Werte für alle zu dem Paket gehörenden Parameter ausgegeben.

Zu diesem Kommando gibt es folgende Optionen und Argumente:

- `-v` Gibt den Protokollmodus an. Angezeigt werden Name und Wert des Parameters.
- `-d gerät` Gibt das *gerät* an, auf dem *pktvar* gespeichert ist. Dies kann ein Verzeichnisname sein oder ein Bezeichner für Magnetband, Diskette oder auswechselbare Platte (z.B. `/var/tmp`, `/dev/dsk/f0t` und `/dev/dsk/0s2`). Das Standardgerät ist das Installations-Spool-Verzeichnis. Wird keine Variante angegeben, werden Parameterdaten für alle Pakete auf dem *gerät* angezeigt.
- `-f datei` Fordert, daß das Kommando in *datei* nach Parameterwerten sucht.
- pktvar* Definiert eine bestimmte Paketvariante, für die Parameterwerte angezeigt werden sollen.
- param* Definiert einen bestimmten Parameter, dessen Wert angezeigt werden soll.

EXIT-CODE

Sind für das angegebene Paket keine Parameterdaten vorhanden, schließt das Kommando mit einem anderen Exit-Code als Null.

HINWEIS

Bei der Syntax mit `-f` kann die Datei angegeben werden, aus der die Parameterwerte entnommen werden sollen. Diese Datei muß dasselbe Format haben wie eine *pkginfo*-Datei. Eine solche Datei könnte z.B. während der Entwicklung des Pakets angelegt und beim Testen der Software verwendet werden.

SIEHE AUCH

`pkgmk(1)`, `pkgparam(3x)`, `pkgproto(1)`, `pkgtrans(1)`.

BEZEICHNUNG

`pkgrm` – entfernt ein Paket aus dem System

ÜBERSICHT

`pkgrm [-n] [-a verw] [pktabl ... ]`

`pkgrm -s spool [pktabl]`

BESCHREIBUNG

`pkgrm` entfernt ein zuvor installiertes oder teilweise installiertes Paket aus dem System. Es wird geprüft, ob andere Pakete von dem zu löschenden Paket abhängig sind. Die erforderlichen Maßnahmen bei einer Abhängigkeit sind in der Datei *verw* definiert.

Der standardmäßige Status für das Kommando ist der Dialogmodus, d.h. während der Verarbeitung werden Eingabeaufforderungen ausgegeben, so daß der Verwalter die durchzuführenden Maßnahmen bestätigen kann. Mit der Option `-n` kann der Nicht-Dialogmodus gewählt werden.

Mit der Option `-s` kann das Spool-Verzeichnis angegeben werden, aus dem gespoolte Pakete entfernt werden sollen.

Zu diesem Kommando stehen folgende Optionen und Argumente zur Verfügung:

- `-n` Nicht-Dialogmodus. Ist der Dialogbetrieb erforderlich, wird das Kommando beendet. Bei dieser Option muß beim Aufrufen des Kommandos mindestens ein Paketexemplar angegeben werden.
- `-a verw` Definiert eine Installationsverwaltungsdatei, *verw*, die anstelle der standardmäßigen *verw*-Datei verwendet werden soll.
- `-s spool` Entfernt das/die angegebene(n) Paket(e) aus dem Verzeichnis *spool*.
- pktabl* Gibt das zu entfernende Paket an. Mit dem Format *pktabl*. * können alle Exemplare eines Pakets gelöscht werden.

SIEHE AUCH

`installf(1M)`, `pkgadd(1M)`, `pkgask(1M)`, `pkgchk(1)`, `pkginfo(1)`, `pkgmk(1)`, `pkgparam(1)`, `pkgproto(1)`, `pkgtrans(1)`, `removef(1M)`.

HINWEIS

Ist es nicht gelungen, ein Paket zu installieren (Meldung: `package installation failed`), dann sind u.U. noch Teile des Pakets im System vorhanden, die Sie mit `pkgrm` löschen können. `pkginfo` bietet die Möglichkeit, nicht erfolgreich installierte Pakete zu erkennen.

BEZEICHNUNG

pkgtrans – Umwandeln von Paketformaten

ÜBERSICHT

pkgtrans [-ions] *gerät1* *gerät2* [*pktvar* ...]

BESCHREIBUNG

pkgtrans wandelt das Format eines installierbaren Pakets um, und zwar:

ein Dateisystemformat in einen Datenstrom

einen Datenstrom in ein Dateisystemformat

ein Dateisystemformat in ein anderes Dateisystemformat

Folgende Optionen und Argumente stehen für dieses Kommando zur Verfügung:

- i Kopiert nur die *pkginfo*- und *pkgmap*-Dateien.
- o Überschreibt eine identische Variante auf dem Ziel-Gerät; bereits vorhandene Paketvarianten werden ebenfalls überschrieben.
- n Erzeugt eine neue Variante, falls von diesem Paket bereits Varianten vorhanden sind.
- s Gibt an, daß das Paket als Datenstrom anstatt als Dateisystem auf *gerät2* geschrieben werden soll. Standardmäßig wird ein Dateisystemformat auf Geräte geschrieben, die beide Formate unterstützen.
- gerät1* Gibt das Ursprungsgerät an. Das Paket bzw. die Pakete auf diesem Gerät werden umgewandelt und auf *gerät2* geschrieben.
- gerät2* Gibt das Ziel-Gerät an. Umgewandelte Pakete werden auf dieses Gerät geschrieben.
- pktvar* Gibt an, welche Paketvariante(n) auf *gerät1* umgewandelt werden soll(en). Mit dem Terminalsymbol *all* können alle Pakete bezeichnet werden. Mit *pktvar*.*** können alle Varianten eines Pakets bezeichnet werden. Sind keine Pakete definiert, zeigt eine Eingabeaufforderung alle Pakete auf dem Gerät an und fragt, welche davon umgewandelt werden sollen.

HINWEIS

Geräte können entweder mit dem Geräte-Knotennamen (*/dev/diskette*) oder dem Geräte-Aliasnamen (*diskette1*) angegeben werden. Das Gerät *spool* gibt das Standard-Spool-Verzeichnis an. Ursprungs- und Ziel-Gerät können unterschiedlich sein.

Standardmäßig überträgt *pkgtrans* keine Variante eines Pakets, wenn eine Variante auf dem Ziel-Gerät bereits vorhanden ist. Wurde die Option *-n* angegeben, wird eine neue Variante angelegt, wenn von einem Paket bereits eine Variante vorhanden ist. Bei der Option *-o* wird eine solche vorhandene Variante überschrieben. Keine dieser Optionen ist sinnvoll, wenn es sich bei dem Ziel-Gerät um einen Datenstrom handelt.

BEISPIEL

Mit dem folgenden Beispiel werden alle Pakete auf dem Diskettenlaufwerk `/dev/diskette` umgewandelt und die Umwandlungen in `/tmp` abgelegt.

```
pkgtrans /dev/diskette /tmp all
```

Beim nächsten Beispiel werden die Pakete `pkg1` und `pkg2` auf `/tmp` umgewandelt und die Umwandlungen, d.h. ein Datenstrom, auf dem Ausgabegerät `9track1` gespeichert.

```
pkgtrans /tmp 9track1 pkg1 pkg2
```

Im folgenden Beispiel werden `pkg1` und `pkg2` auf `tmp` umgewandelt und im Datenstromformat auf eine Diskette ausgegeben.

```
pkgtrans -s /tmp /dev/diskette pkg1 pkg2
```

SIEHE AUCH

`installf(1M)`, `pkgadd(1M)`, `pkgask(1M)`, `pkginfo(1)`, `pkgmk(1)`, `pkgparam(1)`, `pkgproto(1)`, `pkgrm(1M)`, `removef(1M)`.

BEZEICHNUNG

pmadm – Verwaltung von Port Monitoren

ÜBERSICHT

```
pmadm -a [-p pmtag | -t typ] -s svctag -i id -m pmspezif
        -v ver [-f xu] [-y kommentar] [-z skript]
```

```
pmadm -r -p pmtag -s svctag
```

```
pmadm -e -p pmtag -s svctag
```

```
pmadm -d -p pmtag -s svctag
```

```
pmadm -l [-t typ | -p pmtag] [-s svctag]
```

```
pmadm -L [-t typ | -p pmtag] [-s svctag]
```

```
pmadm -g -p pmtag -s svctag [-z skript]
```

```
pmadm -g -s svctag -t typ -z skript
```

BESCHREIBUNG

pmadm ist das Systemverwalterkommando für die untere Ebene der Hierarchie der "Service Access Facility". Ein Port kann mit nur einem Service verbunden sein, der Service kann aber über mehrere Ports verfügbar sein. Um einen bestimmten Service eindeutig erkennen zu können, muß das Kommando pmadm sowohl den/die Port Monitor(e), über den/die der Service verfügbar ist (-p oder -t), als auch den Service (-s) identifizieren. Näheres ergibt sich aus den nachfolgenden Beschreibungen der Optionen.

pmadm erfüllt die folgenden Funktionen:

- Hinzufügen oder Entfernen eines Services
- Aktivieren oder Deaktivieren eines Services
- Installieren oder Ersetzen eines pro Service angelegten Konfigurations-Skripts
- Ausgeben von Informationen zu einem Service

Alle Benutzer des Systems können pmadm aufrufen, um sich den Service-Status (-l oder -L) anzeigen zu lassen oder um die einem Service zugeordneten Konfigurations-Skripts auszugeben (-g ohne die Option -z). Andere Optionen darf nur ein privilegierter Benutzer verwenden.

Die Optionen haben folgende Bedeutung:

- a Einen Service hinzufügen. pmadm fügt einen Eintrag für den neuen Service in die Verwaltungsdatei des Port Monitors ein. Aufgrund der Komplexität der Optionen und Argumente, die auf die Option -a folgen, ist es günstiger, zum Hinzufügen von Services ein Kommando-Skript oder das Menüsystem zu verwenden. Beim Menüsystem gibt man sysadm ports ein und wählt dann die Option port_services aus.
- d Einen Service deaktivieren. In der Verwaltungsdatei für den Port Monitor wird in dem Markenfeld des Eintrags für den Service svctag ein x hinzugefügt. Diesen Eintrag benutzt der Port Monitor pmtag. Eine Beschreibung der verfügbaren Marken finden Sie in der Erläuterung zu der Option -f.

- e Einen Service aktivieren. Das *x* wird aus dem Markenfeld des Eintrags für den Service *svctag* in der Verwaltungsdatei des Port Monitors gelöscht. Diesen Eintrag benutzt der Port Monitor *pmtag*. Eine Beschreibung der verfügbaren Marken finden Sie in der Erläuterung zu der Option -f.
- f xu Zusammen mit der Option -f werden eine oder beide der folgenden Marken angegeben, die daraufhin in das Markenfeld des Eintrags für den neuen Service aufgenommen werden. Dieser Eintrag steht in der Verwaltungsdatei für den Port Monitor. Fehlt die Option -f, werden keine Marken gesetzt, und die Standardbedingungen gelten weiterhin. Standardmäßig wird ein neuer Service aktiviert, und es wird kein *utmp*-Eintrag dafür angelegt. Die Option -f ohne eines der folgenden Argumente ist unzulässig.
 - x Den über den Port Monitor *pmtag* verfügbaren Service *svctag* nicht aktivieren.
 - u Einen *utmp*-Eintrag anlegen für den über den Port Monitor *pmtag* verfügbaren Service *svctag*.
- g Ein Konfigurations-Skript, das einem Service zugeordnet ist, ausgeben, installieren oder ersetzen. Wird die Option -g durch die Optionen -p und -s ergänzt, wird das Konfigurations-Skript ausgegeben, das dem über den Port Monitor *pmtag* verfügbaren Service *svctag* zugeordnet ist. Die Kombination der Option -g mit den Optionen -p, -s und -z bewirkt die Installation des Konfigurations-Skripts in der Datei *skript* als Konfigurations-Skript für den Service *svctag*, der über den Port Monitor *pmtag* verfügbar ist. Kombiniert man die Option -g mit den Optionen -s, -t und -z, wird die Datei *skript* als Konfigurations-Skript für den Service *svctag* installiert, der über einen beliebigen Port Monitor des Typs *typ* verfügbar ist. Andere Kombinationen von Optionen mit der Option -g sind unzulässig.
- i *id* *id* ist die Kennung, die dem Service *svctag* beim Start zugewiesen wird. *id* muß ein Eintrag in */etc/passwd* sein.
- l Mit der Option -l werden Service-Daten angefordert. Allein und mit den nachstehend beschriebenen Optionen stellt diese Option einen Filter zum Extrahieren unterschiedlicher Datenkombinationen dar.
 - l Allein erstellt die Option -l eine Liste aller Services auf dem System.
 - l -p *pmtag* Gibt alle Services an, die über den Port Monitor *pmtag* erreichbar sind.
 - l -s *svctag* Gibt alle Services mit der Marke *svctag* an.
 - l -p *pmtag* -s *svctag* Gibt den Service *svctag* an.
 - l -t *typ* Listet alle Services auf, die über Port Monitore des Typs *typ* verfügbar sind.

- `-l -t typ -s svctag`
 Gibt alle Services mit der Marke *svctag* aus, die über einen Port Monitor des Typs *typ* verfügbar sind.
 Andere Kombinationen von Optionen mit `-l` sind unzulässig.
- `-L` Die Option `-L` ist mit `-l` identisch, abgesehen davon, daß bei `-L` die Ausgabe komprimiert wird.
- `-m pmspezif`
pmspezif ist der port-monitor-spezifische Teil des Eintrags für den Service in der Verwaltungsdatei des Port Monitors.
- `-p pmtag`
 Gibt die Marke zu dem Port Monitor an, durch den ein Service (angegeben als `-s svctag`) verfügbar ist.
- `-r` Entfernen eines Service. Wenn `pmadm` einen Service entfernt, wird der zugehörige Eintrag aus der Verwaltungsdatei zum Port Monitor gelöscht.
- `-s svctag`
 Gibt die Service-Marke zu einem bestimmten Service an. Die Service-Marke wird vom Systemverwalter vergeben und ist Teil des Eintrags für den Service in der Verwaltungsdatei zum Port Monitor.
- `-t typ` Gibt den Typ des Port Monitors an.
- `-v ver` Gibt die Versionsnummer der Verwaltungsdatei zum Port Monitor an, und zwar als
`-v `pmspec -V``
 Dabei ist *pmspec* das spezielle Systemverwalterkommando für den Port Monitor *pmtag*. Dieses spezielle Kommando lautet `ttysadm` für `ttymon` und `nlsadmin` für `listen`. Das Kommando kennt den Versionsstempel des Port Monitors und gibt ihn zurück, wenn *pmspec* mit der Option `-V` aufgerufen wird.
- `-y kommentar`
 Verbindet *kommentar* mit dem Service-Eintrag in der Verwaltungsdatei zum Port Monitor.
- `-z skript`
 Wird zusammen mit der Option `-g` zum Angeben der Datei verwendet, die das dem Service zugeordnete Konfigurations-Skript enthält. Das Ändern eines Konfigurations-Skripts geschieht in drei Schritten. Zunächst wird eine Kopie des vorhandenen Skripts erstellt (`-g` allein). Anschließend wird die Kopie editiert. Schließlich wird das vorhandene Skript mit der editierten Version überschrieben (`-g` mit `-z`).

EXIT-STATUS

Bei erfolgreicher Ausführung wird `pmadm` mit einem Exit-Status 0 abgeschlossen. Schlägt die Ausführung fehl, hat der Exit-Status einen anderen Wert.

Optionen, die Daten anfordern, schreiben die Daten auf die Standardausgabe. Wurde die Datenanforderung mit der Option `-l` versehen, werden Spaltenüberschriften ausgegeben und die Daten unter den entsprechenden Überschriften aufgeführt. In diesem Format wird ein fehlender Wert durch einen Bindestrich angezeigt. Wurden Daten mit der Option `-L` in komprimiertem Format abgefragt, werden sie in durch Doppelpunkt getrennten Feldern ausgegeben. Fehlende Felder werden durch zwei Doppelpunkte hintereinander dargestellt. `#` ist das Kommentarzeichen.

BEISPIELE

Ein Service soll einem Port Monitor mit der Marke `pmtag` hinzugefügt werden. Der Service erhält die Marke `svctag`. Für den Port Monitor spezifische Angaben werden mit `pmspec` erstellt. Der als `svctag` definierte Service wird mit der Kennung `root` aufgerufen.

```
pmadm -a -p pmtag -s svctag -i root -m `pmspec -a arg1 -b arg2` \
-v `pmspec -V`
```

Ein Service mit Service-Marke `svctag`, Kennung `guest` und mit `pmspec` erstellte port-monitor-spezifische Daten werden allen Port Monitoren des Typs `type` hinzugefügt:

```
pmadm -a -s svctag -i guest -t type -m `pmspec -a arg1 -b arg2` \
-v `pmspec -V`
```

Der Service `svctag` wird aus dem Port Monitor `pmtag` entfernt:

```
pmadm -r -p pmtag -s svctag
```

Aktivieren des Service `svctag`, der über den Port Monitor `pmtag` verfügbar ist:

```
pmadm -e -p pmtag -s svctag
```

Deaktivieren des Service `svctag`, der über den Port Monitor `pmtag` verfügbar ist:

```
pmadm -d -p pmtag -s svctag
```

Erstellen einer Liste mit Statusangaben für alle Services:

```
pmadm -l
```

Erstellen einer Liste mit Statusangaben für alle Services, die über den Port Monitor mit Marke `ports` verfügbar sind:

```
pmadm -l -p ports
```

Erstellen derselben Liste in komprimierter Form:

```
pmadm -L -p ports
```

Erstellen einer Liste mit Statusangaben für alle Services, die über Port Monitore des Typs `listen` verfügbar sind:

```
pmadm -l -t listen
```

Ausgeben des Konfigurations-Skripts zum Service `svctag`, der über den Port Monitor `pmtag` verfügbar ist:

```
pmadm -g -p pmtag -s svctag
```

pmadm (1M)

pmadm (1M)

DATEIEN

/etc/saf/*pmtag*/_config
/etc/saf/*pmtag*/*svctag*
/var/saf/*pmtag*/*

SIEHE AUCH

doconfig(3n), sacadm(1M), sac(1M).

BEZEICHNUNG

profiler: prfld, prfstat, prfdc, prfsnap, prfpr – auswerten der System-Aktivitäten

ÜBERSICHT

```
/usr/sbin/prfld [ sysnamensliste ]
/usr/sbin/prfstat on
/usr/sbin/prfstat off
/usr/sbin/prfdc datei [ periode [ stoppzeit ] ]
/usr/sbin/prfsnap datei
/usr/sbin/prfpr file [ grenzwert [ sysnamensliste ] ]
```

BESCHREIBUNG

prfld, prfstat, prfdc, prfsnap und prfpr bilden ein Programmsystem, das die Aktivitätsanalyse des Betriebssystems SINIX erleichtert.

Mit prfld wird der Aufzeichnungsmechanismus des Systems initialisiert. Es generiert eine Tabelle mit den Anfangsadressen aller Unterrouinen des Systems, die aus *sysnamensliste* hervorgehen.

Mit prfstat wird die Aufzeichnung aktiviert bzw. deaktiviert. Der zusätzliche Steuerungsaufwand für die Profiling-Programme beträgt für 500 Textadressen weniger als 1%. prfstat gibt darüber hinaus Aufschluß über die Anzahl der erfaßten Textadressen.

prfdc und prfsnap erfüllen die Datenerfassungsfunktion bei der Analysenerstellung. Sie kopieren den aktuellen Wert aller Textadreßzähler in eine Datei, in der die Daten analysiert werden können. prfdc speichert die Zählerwerte im Abstand von *periode* Minuten in einer *datei* und wird zur *stoppzeit* ausgeschaltet. Gültige Werte für *stoppzeit* sind 0-24. prfsnap erfaßt die Daten nur beim Aufruf und hängt die Zählerwerte an die *datei* an.

prfpr formatiert die von prfdc oder prfsnap erfaßten Daten. Jede Textadresse wird in das nächstliegende Textsymbol konvertiert, das aus der *sysnamensliste* hervorgeht, und wird ausgegeben, wenn die prozentuale Aktivität für diesen Bereich größer als *grenzwert* ist.

DATEIEN

/dev/prf	Schnittstelle für Analysedaten und Textadressen
/stand/unix	Standardname für die Datei mit den Systemnamen

BEZEICHNUNG

prvtoc – gibt das VTOC einer Platte aus

ÜBERSICHT

/usr/sbin/prvtoc *gerät* [-a] [-p] [-e] [-f *datei*]

BESCHREIBUNG

Mit dem Kommando prvtoc wird der Inhalt des VTOC (VTOC: volume table of contents, Inhaltsverzeichnis des Datenträgers) ausgegeben. Dieses Kommando kann nur von einem privilegierten Benutzer ausgeführt werden.

Der Name des *geräts* muß der Dateiname einer Platte sein, auf die im Raw-Modus zugegriffen werden kann:

/dev/rdisk/cxdys0 für ESDI-Platten am Interphase Storager oder am ESDI-Controller, für SCSI-Platten am AHA1542B

/dev/rdisk/shdxtys0 für SCSI-Platten am NCR Host Adapter (ADP32 Controller)

Die Optionen haben folgende Bedeutung:

- a prvtoc gibt den Inhalt der "alternate table" aus. Diese Tabelle enthält die alternativen Spuren und Sektoren, die fehlerhafte ersetzen. Die Option liefert keine sinnvollen Angaben bei Platten, die ihre defekten Sektoren selbst verwalten.
- e prvtoc erstellt oder aktualisiert die Datei /etc/partitions, die Informationen über die aktuellen Partitionen und Slices enthält.
- p prvtoc schreibt die Plattengeometrie, die in der Datenstruktur pdinfo angegeben ist, auf die Standardausgabe.
- f *datei* prvtoc schreibt das aktuelle VTOC in die Datei *datei*.

BEISPIEL

Die unten abgebildeten Kommandozeilen und die Antwort des Systems gelten für eine Festplatte Micropolis 1568 mit 630 MB:

prvtoc(1M)

prvtoc(1M)

```
# prvtoc -f datei /dev/rdisk/c0d0s0
# cat datei

#SLICE  TAG    FLAGS  START  SIZE
  0     0x5    0x201    795 1292670
  1     0x2    0x200   44573   41340
  2     0x3    0x201   11183   33390
  3     0x4    0x200   85913  133560
  4     0xb    0x200  219473  611355
  5     0x0     0x0      0      0
  6     0x0     0x0      0      0
  7     0x1    0x201    795     34
  8     0x0     0x0      0      0
  9     0x0     0x0      0      0
 10     0x9    0x200    848   10335
 11     0xa    0x200  830828  246450
 12     0x4    0x200 1077278  216187
 13     0x0     0x0      0      0
 14     0x0     0x0      0      0
 15     0x0     0x0      0      0
```

Codes für TAG sind:

NAME	NUMMER
UNASSIGNED	0x0
BOOT	0x1
ROOT	0x2
SWAP	0x3
USR	0x4
DISK or BACKUP	0x5
STAND	0x9
VAR	0xa
HOME	0xb

FLAGS gibt an, wie die Partition eingehängt werden soll.

NAME	NUMMER
MOUNTABLE, READ AND WRITE	0x200
NOT MOUNTABLE	0x201
MOUNTABLE, READ ONLY	0x210

SIEHE AUCH

edvtoc(1M), disksetup(1M), mkpart(1M)

prvtoc(1M)

prvtoc(1M)

HINWEIS

Das Kommando `mount` prüft nicht das Bit, das angibt, daß die Partition nicht eingehängt werden kann.

BEZEICHNUNG

putdev – editiert die Gerätetabelle

ÜBERSICHT

putdev -a *aliasname* [*attribut=wert* ...]

putdev -m *gerät* *attribut=wert* [*attribut=wert* ...]

putdev -d *gerät* [*attribut* ...]

BESCHREIBUNG

Mit putdev kann ein neues Gerät zur Gerätetabelle hinzugefügt, eine vorhandene Gerätebeschreibung modifiziert oder ein Geräteeintrag aus der Tabelle gelöscht werden. Mit der ersten Form des Kommandos wird ein Gerät hinzugefügt, mit der zweiten werden vorhandene Einträge durch Hinzufügen oder Ändern von Attributen modifiziert. Ist ein angegebenes Attribut nicht definiert, wird dieses Attribut mit dieser Option zu der Gerätedefinition hinzugefügt. Ist das Attribut bereits definiert, wird mit dieser Option die Attributdefinition geändert. Mit der dritten Form des Kommandos kann ein ganzer Geräteeintrag gelöscht werden. Wird das Argument *attribut* verwendet, kann mit dieser Form ein dem Gerät zugewiesenes Attribut gelöscht werden.

Bei diesem Kommando stehen die folgenden Optionen und Argumente zur Verfügung:

- a Fügt einen Geräteeintrag zu der Gerätetabelle hinzu, wobei die angegebenen Attribute verwendet werden. Das Gerät muß mit dem *aliasnamen* angegeben werden.
- m Modifiziert einen Geräteeintrag in der Gerätetabelle. Ist der Eintrag bereits vorhanden, werden alle angegebenen Attribute, die noch nicht definiert sind, hinzugefügt. Wurden Attributen bereits Werte zugewiesen, können diese Werte durch die mit diesem Kommando angegebenen Werte ersetzt werden.
- d Mit dieser Option wird ein Geräteeintrag aus der Gerätetabelle gelöscht, wenn sie ohne das Argument *attribut* verwendet wird. Zusammen mit dem Argument *attribut* verwendet, löscht diese Option die angegebene Attributspezifikation für das *gerät* aus der Tabelle.
- aliasname* Gibt den Aliasnamen des hinzuzufügenden Geräts an.
- gerät* Gibt den Pfad- oder Aliasnamen des Geräts an, in dessen Eintrag ein Attribut hinzugefügt, modifiziert oder gelöscht werden soll.
- attribut* Gibt ein Geräteattribut an, das hinzugefügt oder modifiziert werden soll. Hierbei kann es sich um ein beliebiges der unter HINWEIS beschriebenen Attribute außer dem Aliasnamen handeln. Hierdurch wird verhindert, daß der Aliasname des Geräts versehentlich modifiziert oder aus der Tabelle gelöscht wird.
- wert* Gibt den Wert an, der dem Geräteattribut zugewiesen werden soll.

HINWEIS

Im folgenden sind alle Attribute aufgelistet, die für ein Gerät definiert werden können:

putdev(1)

putdev(1)

alias	Der eindeutige Name, der dem Gerät zugewiesen ist. In der Datenbank darf ein Aliasname nur jeweils einem Gerät zugewiesen sein. Der Name darf höchstens 14 Zeichen lang sein und sollte nur alphanumerische Zeichen enthalten. Die folgenden Sonderzeichen können ebenfalls verwendet werden, sofern sie mit einem Gegenschrägstrich versehen werden: Unterstreichungszeichen (<u> </u>), Dollarzeichen (\$), Bindestrich (-) und Punkt (.
bdevice	Der Pfadname zu dem Knoten des blockorientierten Sondergeräts, der dem Gerät zugewiesen ist, falls vorhanden. Die dem Gerät zugewiesene Geräteklasse- und Gerätenummer sollte in der Datenbank eindeutig sein und mit den Werten in dem Feld cdevice übereinstimmen, falls dieses vorhanden ist. Es ist die Aufgabe des Systemverwalters, sicherzustellen, daß die major/minor-Nummern in der Datenbank eindeutig sind.
capacity	Die Kapazität des Geräts oder des üblichen Datenträgers, falls dieser auswechselbar ist.
cdevice	Der Pfadname zum Knoten des zeichenorientierten Sondergeräts, der dem Gerät zugewiesen ist, falls vorhanden. Die dem Gerät zugewiesene Geräteklasse- und Gerätenummer sollte in der Datenbank eindeutig sein und mit den Werten in dem Feld bdevice übereinstimmen, falls dieses vorhanden ist. Es ist die Aufgabe des Systemverwalters, sicherzustellen, daß die major/minor-Nummern in der Datenbank eindeutig sind.
cyl	Wird von dem Kommando verwendet, das in dem Attribut mkfscmd angegeben wird.
desc	Die Beschreibung einer beliebigen Variante eines Datenträgers, die dem Gerät zugewiesen ist, wie z.B. eine Diskette.
dpartlist	Die Liste der Plattenpartitionen, die dem Gerät zugewiesen sind. Wird nur verwendet, wenn type=disk ist. Die Liste sollte Aliasnamen von Geräten enthalten. Für jedes Gerät muß type=dpart sein.
dparttype	Der Typ der Plattenpartition für dieses Gerät. Wird nur verwendet, wenn type=dpart ist. Als Typ sollte fs (für Dateisystem) oder dp (für Datenpartition) angegeben werden.
erasecmd	Die Kommandokette, mit der das Gerät aus der Gerätetabelle gelöscht werden kann.
fntcmd	Die Kommandokette, mit der das Gerät formatiert werden kann.
fsname	Der Name des Dateisystems, das auf dieser Partition verwaltet und an das Kommando /usr/sbin/labelit übergeben wird. Dieses Attribut wird nur angegeben, wenn type=dpart und dparttype=fs ist.
gap	Wird von dem Kommando verwendet, das in dem Attribut mkfscmd angegeben wird.

putdev(1)**putdev(1)**

mkfscmd	Die Kommandokette, mit der ein Dateisystem auf ein zuvor formatiertes Gerät geschrieben werden kann.
mountpt	Der Standardeinhängpunkt für das Gerät. Wird nur verwendet, wenn das Gerät eingehängt werden kann. Bei Plattenpartitionen, für die <code>type=dpart</code> und <code>dparttype=fs</code> ist, sollte dieses Attribut die Speicheradresse angeben, an der die Partition normalerweise eingehängt wird.
nblocks	Die Anzahl der Blöcke in dem Dateisystem, das auf dieser Partition verwaltet wird. Wird nur verwendet, wenn <code>type=dpart</code> und <code>dparttype=fs</code> ist.
ninodes	Die Anzahl der I-Nodes in dem Dateisystem, das auf dieser Partition verwaltet wird. Wird nur verwendet, wenn <code>type=dpart</code> und <code>dparttype=fs</code> ist.
norewind	Der Name des Knoten des zeichenorientierten Sondergeräts, der den Zugriff auf das serielle Gerät ermöglicht, ohne daß beim Schließen des Geräts zurückgespult wird.
pathname	Definiert den Pfadnamen zu einem I-Node, der das Gerät beschreibt. Wird für Pfadnamen nicht blockorientierter oder zeichenorientierter Geräte, wie z.B. Verzeichnisse, verwendet.
type	Ein Token, das für Eigenschaften des Geräts steht. Zu den Standardtypen gehören: <code>9-track</code> (9-spurig), <code>ctape</code> , <code>disk</code> (Festplatte), <code>directory</code> (Verzeichnis), <code>diskette</code> (Diskette), <code>dpart</code> und <code>qtape</code> .
volname	Der Datenträgername auf dem Dateisystem, das auf dieser Partition verwaltet und an das Kommando <code>/usr/sbin/labelit</code> übergeben wird. Wird nur verwendet, wenn <code>type=dpart</code> und <code>dparttype=fs</code> ist.
volume	Eine Textzeichenkette, mit der jede Variante eines Datenträgers beschrieben wird, die dem Gerät zugewiesen ist. Dieses Attribut sollte für Geräte, die nicht auswechselbar sind, nicht definiert werden.

EXIT-CODES

Die Exit-Codes von `putdev` haben folgende Bedeutung:

- 0 = Aufgabe erfolgreich ausgeführt
- 1 = Kommandosyntax ist falsch, es wurde eine ungültige Option verwendet oder es ist ein interner Fehler aufgetreten
- 2 = Gerätetabelle konnte nicht gelesen werden oder es konnte keine neue Gerätetabelle angelegt werden
- 3 = Wurde das Kommando zusammen mit der Option `-a` verwendet, gibt dieser Exit-Code an, daß in der Gerätetabelle bereits ein Eintrag mit dem *aliasnamen* vorhanden ist. Wurde das Kommando mit der Option `-m` oder `-d` ausgeführt, gibt dieser Wert an, daß für das *gerät* kein Eintrag vorhanden ist.
- 4 = Gibt an, daß `-d` angefordert wurde und mindestens eines der angegebenen Attribute für das Gerät nicht definiert wurde.

DATEIEN

`/etc/device.tab`

SIEHE AUCH

`devattr(1)`, `putdgrp(1M)`.

BEZEICHNUNG

putdgrp – Editieren der Gerätegruppentabelle

ÜBERSICHT

putdgrp [-d] *ggruppe* [*gerät* . . .]

BESCHREIBUNG

Mit *putdgrp* läßt sich die Gerätegruppentabelle auf zwei mögliche Arten ändern. Einerseits kann eine neue Gerätegruppe angelegt oder eine vorhandene gelöscht werden. Andererseits kann auch die Gruppendifinition durch Hinzufügen oder Löschen eines Geräts geändert werden.

Wird das Kommando nur mit der Angabe *ggruppe* aufgerufen, trägt es den angegebenen Gruppennamen in die Gerätegruppentabelle ein, sofern er noch nicht vorhanden ist. Wird zusätzlich die Option -d angegeben, löscht das Kommando die Gruppe aus der Tabelle.

Wird das Kommando mit den Angaben *ggruppe* und *gerät* aufgerufen, wird der angegebene Gerätenamen (bzw. werden die Namen) in die Gruppendifinition eingefügt. Wird das Kommando mit beiden Argumenten und der Option -d eingegeben, löscht es den/die Gerätenamen aus der Gruppendifinition.

Wird das Kommando mit den Angaben *ggruppe* und *gerät* aufgerufen und die Gerätegruppe existiert nicht, legt das Kommando die Gruppe an und fügt die angegebenen Geräte dort ein.

Folgende Optionen und Argumente stehen zu diesem Kommando zur Verfügung:

- d Löscht die Gruppe oder, wenn *gerät* angegeben wurde, das Gerät aus einer Gruppendifinition.
- ggruppe* Gibt einen Gerätegruppennamen an.
- gerät* Gibt den Pfadnamen oder Aliasnamen des Geräts an, das in die Gerätegruppe aufgenommen oder daraus gelöscht werden soll.

EXIT-CODES

Die Exit-Codes von *putdgrp* haben folgende Bedeutung:

- 0 = erfolgreiche Ausführung der Aufgabe
- 1 = falsche Kommandosyntax, ungültige Option oder interner Fehler
- 2 = Gerätegruppentabelle konnte nicht zum Lesen geöffnet werden, oder eine neue Gerätegruppentabelle konnte nicht angelegt werden
- 3 = Wurde das Kommando mit der Option -d ausgeführt, bedeutet dieser Wert, daß ein Eintrag in der Gerätegruppentabelle für die Gerätegruppe *ggruppe* nicht existiert und daher nicht gelöscht werden kann. Anderenfalls bedeutet der Wert, daß die Gerätegruppe *ggruppe* schon vorhanden ist und deshalb nicht hinzugefügt werden kann.
- 4 = Wurde die Option -d benutzt, bedeutet dieser Wert, daß eines oder mehrere der angegebenen Geräte keine Elemente der Gerätegruppe *ggruppe* sind. Anderenfalls bedeutet der Wert, daß eines oder mehrere der angegebenen Geräte bereits Elemente der Gerätegruppe *ggruppe* sind.

putdgrp(1)

putdgrp(1)

BEISPIEL

Hinzufügen einer neuen Gerätegruppe:

```
putdgrp floppies
```

Hinzufügen eines Geräts zu einer Gerätegruppe:

```
putdgrp floppies diskette2
```

Löschen einer Gerätegruppe:

```
putdgrp -d floppies
```

Löschen eines Geräts aus einer Gerätegruppe:

```
putdgrp -d floppies diskette2
```

DATEIEN

```
/etc/dgroup.tab
```

SIEHE AUCH

```
listdgrp(1), putdev(1M).
```


BEZEICHNUNG

pwck, grpck – Prüfprogramm für Paßwort-/Gruppendatei

ÜBERSICHT

/usr/sbin/pwck [datei]

/usr/sbin/grpck [datei]

BESCHREIBUNG

pwck durchsucht die Paßwortdatei und zeichnet Diskrepanzen auf. Geprüft wird die Richtigkeit der Felderzahl, des Benutzernamens, der Benutzernummer, der Gruppennummer ebenso wie die Frage, ob das Anmeldeverzeichnis und das als Shell zu verwendende Programm vorhanden sind. Die Standard-Paßwortdatei ist /etc/passwd.

grpck prüft alle Einträge in der Gruppendatei. Geprüft werden die Anzahl der Felder, der Gruppenname, die Gruppennummer. Geprüft wird auch, ob ein Benutzername zu mehr als NGROUPS_MAX Gruppen gehört und ob alle Benutzernamen in der Paßwortdatei erscheinen. Die Standard-Gruppendatei ist /etc/group.

DATEIEN

/etc/group

/etc/passwd

SIEHE AUCH

group(4), passwd(4).

FEHLERMELDUNGEN

Gruppeneinträge in /etc/group ohne Benutzernamen werden mit einer Marke gekennzeichnet.

BEZEICHNUNG

pwconv – Installation und Aktualisierung von /etc/shadow

ÜBERSICHT

pwconv

BESCHREIBUNG

Das Kommando pwconv erzeugt und aktualisiert /etc/shadow mit Daten aus /etc/passwd.

Ist die Datei /etc/shadow nicht vorhanden, wird sie von diesem Kommando mit Daten aus /etc/passwd erstellt. Das Kommando gibt in /etc/shadow den Benutzernamen, das Paßwort sowie Daten zur zeitlichen Begrenzung der Gültigkeit von Paßwörtern ein. Enthält /etc/passwd keine Daten zur zeitlichen Begrenzung der Gültigkeit von Paßwörtern für einen bestimmten Benutzer, werden keine solchen Daten zu /etc/shadow hinzugefügt. Daten zur letzten Änderung werden jedoch immer aktualisiert.

Ist die Datei /etc/shadow vorhanden, werden die folgenden Aufgaben ausgeführt:

Einträge, die in der Datei /etc/passwd, aber nicht in /etc/shadow stehen, werden in die Datei /etc/shadow aufgenommen.

Einträge, die in /etc/shadow, aber nicht in /etc/passwd stehen, werden aus der Datei /etc/shadow gelöscht.

Paßwort-Attribute, z.B. Paßwortdaten und Daten zur zeitlichen Begrenzung der Gültigkeit von Paßwörtern, die in einem Eintrag von /etc/passwd stehen, werden in den entsprechenden Eintrag in /etc/shadow übernommen.

Das Programm pwconv darf nur von privilegierten Benutzern ausgeführt werden.

DATEIEN

/etc/passwd
/etc/shadow
/etc/passwd
/etc/shadow

SIEHE AUCH

passwd(1), passmgmt(1M)

EXIT-CODES

Die Exit-Codes von pwconv haben folgende Bedeutung:

- 0 = erfolgreiche Ausführung
- 1 = Zugriffsberechtigung verweigert
- 2 = ungültige Kommandosyntax
- 3 = unerwarteter Fehler; Übertragung nicht ausgeführt
- 4 = unerwarteter Fehler; Paßwortdatei(en) fehlt/fehlen
- 5 = Paßwortdatei(en) belegt; Kommando später wiederholen

quot (1M)

(UFS)

quot (1M)

BEZEICHNUNG

quot – gibt eine Übersicht über den Eigentümerstatus für ein Dateisystem

ÜBERSICHT

quot [-acfhnv] [*dateisystem*]

BESCHREIBUNG

quot zeigt die Anzahl der jedem Benutzer momentan gehörenden Blöcke (1024 Byte) in dem angegebenen *dateisystem* an. Der Höchstwert liegt bei 2048 Blöcken. Dateien, die größer sind, werden als Dateien zu 2048 Blöcken gezählt; bei der Zählung der Blockzahl werden dagegen wieder die tatsächlichen Werte verwendet.

Folgende Optionen stehen zur Verfügung:

- a Erzeugt einen Bericht über alle eingehängten Dateisysteme.
- c Zeigt drei Spalten an, in denen die Dateigröße in Blöcken, die Anzahl der Dateien dieser Größe sowie die Summe der Blöcke, die Dateien dieser Größe oder kleinere Dateien enthalten, angegeben werden.
- f Zeigt die jedem Benutzer gehörende Anzahl von Dateien und den von ihm belegten Plattenplatz an. Diese Option darf nicht mit den Optionen -c und -v verwendet werden.
- h Schätzt die Anzahl der Blöcke in der Datei — Dateien mit Lücken werden hier nicht berücksichtigt.
- n Vergibt Namen an die Liste der von der Standardeingabe gelesenen Dateien. quot -n kann nicht allein verwendet werden, weil Daten aus der Standardeingabe erwartet werden. Die Pipe
ncheck filesystem | sort +0n | quot -n filesystem
beispielsweise erstellt eine Liste aller Dateien und ihrer Eigentümer. Diese Option darf mit keiner anderen Option verwendet werden.
- v Zeigt zusätzlich zu den Standardangaben drei Spalten mit der Anzahl der Blöcke an, die seit 30, 60 oder 90 Tagen nicht mehr benutzt wurden.

HINWEIS

Dieses Kommando darf nur von einem privilegierten Benutzer verwendet werden.

DATEIEN

/etc/mnttab

eingehängte Dateisysteme

/etc/passwd

zum Abrufen von Benutzernamen

SIEHE AUCH

du(1M)

BEZEICHNUNG

quota – zeigt Plattenquote und Plattenbelegung für einen Benutzer an

ÜBERSICHT

quota [-v] [*benutzername*]

BESCHREIBUNG

quota zeigt die Plattenbelegung durch einen Benutzer und die Höchstwerte des ihm zur Verfügung stehenden Plattenspeicherplatzes an. Nur ein privilegierter Benutzer darf mit dem optionalen Argument *benutzername* die Höchstwerte für die anderen Benutzer anzeigen lassen.

Wird quota ohne Optionen verwendet, werden lediglich Warnmeldungen zu eingehängten Dateisystemen angezeigt, deren Auslastung den Höchstwert überschreitet. Über Netz eingehängte Dateisysteme, für die keine Quoten bestehen, werden ignoriert.

benutzername kann numerisch sein, entsprechend der *uid* eines Benutzers.

Mit der Option -v werden die Benutzerquoten in allen eingehängten Dateisystemen angezeigt, für die Quoten gesetzt sind.

DATEIEN

/etc/mnttab

Liste der aktuell eingehängten Dateisysteme

SIEHE AUCH

edquota(1M), quotaon(1M), quotactl(2).

BEZEICHNUNG

quotacheck – Konsistenzprüfung für Dateisystemquoten

ÜBERSICHT

quotacheck [-v] [-p] *dateisystem...*

quotacheck [-apv]

BESCHREIBUNG

quotacheck prüft jedes Dateisystem, erstellt eine Tabelle der aktuellen Plattenbelegung und vergleicht sie mit der Tabelle, die in der Plattenquotendatei für das Dateisystem gespeichert ist. Werden Diskrepanzen festgestellt, werden sowohl die Quotendatei als auch die aktuelle Systemkopie der falschen Quoten aktualisiert (dieser letzte Schritt wird nur ausgeführt, wenn ein aktives Dateisystem geprüft wird).

quotacheck geht davon aus, daß für jedes zu prüfende Dateisystem eine Quotendatei namens *quotas* im Root-Verzeichnis vorhanden ist. Ist dies nicht der Fall, ignoriert quotacheck das Dateisystem.

quotacheck greift bei der Berechnung der tatsächlichen Plattenbelegung pro Benutzer auf das zeichenorientierte Gerät zu. Daher müssen sich die überprüften Dateisysteme im Ruhezustand befinden, während quotacheck läuft.

Folgende Optionen sind vorhanden:

- v Gibt die berechneten Plattenquoten für jeden Benutzer in einem bestimmten Dateisystem an. quotacheck meldet normalerweise nur die geänderten Quoten.
- a Prüft die in */etc/mnttab* mit Lese-/Schreib-Zugriff und Plattenquoten angegebenen Dateisysteme. Geprüft werden nur die Dateisysteme, bei denen das Feld *mntopts* der Datei */etc/vfstab* ein "rq" enthält.
- p Führt in den benötigten Dateisystemen parallele Prüfläufe durch.

DATEIEN

/etc/mnttab

eingehängte Dateisysteme

/etc/vfstab

Liste der Standardparameter für die einzelnen Dateisysteme

SIEHE AUCH

quotaon(1M), quotactl(2).

BEZEICHNUNG

quotaon, quotaoff – Aktivieren und Deaktivieren der Dateisystemquoten

ÜBERSICHT

quotaon [-v] *dateisystem...*

quotaon [-av]

quotaoff [-v] *dateisystem...*

quotaoff [-av]

BESCHREIBUNG

quotaon meldet dem System, daß in einem oder mehreren Dateisystemen Plattenquoten aktiviert werden müssen. Die angegebenen Dateisysteme müssen zu diesem Zeitpunkt eingehängt sein. Die Quotendateien für die Dateisysteme müssen im Root-Verzeichnis des angegebenen Dateisystems unter dem Namen *quotas* vorhanden sein.

quotaoff meldet dem System, daß die Plattenquoten für die angegebenen Dateisysteme deaktiviert werden müssen.

Folgende Optionen stehen für *quotaon* zur Verfügung:

- a Die Quoten aller in */etc/mnttab* mit Lese-/Schreib-Zugriff und Quoten geführten Dateisysteme werden aktiviert. Diese Option wird normalerweise beim Laden des Systems verwendet, um Quoten zu aktivieren. Sie gilt nur für Dateisysteme, bei denen in dem Feld *mntopts* der Datei */etc/vfstab* "rq" angegeben ist.
- v Gibt eine Meldung für alle Dateisysteme aus, für die Quoten aktiviert sind.

Folgende Optionen stehen für *quotaoff* zur Verfügung:

- a Erzwingt die Deaktivierung der Quoten für alle Dateisysteme in */etc/mnttab*. Diese Option gilt nur für Dateisysteme, bei denen in dem Feld *mntopts* der Datei */etc/vfstab* "rq" angegeben ist.
- v Zeigt für jedes betroffene Dateisystem eine Meldung an.

Mit diesen Kommandos wird das Statusfeld von Geräten in */etc/mnttab* aktualisiert. Dieses Feld zeigt an, ob für die einzelnen Dateisysteme Quoten aktiviert oder deaktiviert sind.

DATEIEN

/etc/mnttab

eingehängte Dateisysteme

/etc/vfstab

Liste der Standardparameter für die einzelnen Dateisysteme

SIEHE AUCH

mnttab(4), *vfstab*(4)

BEZEICHNUNG

rarpd – Server für das DARPA-Reverse Address Resolution Protocol

ÜBERSICHT

rarpd *schnittstelle* [*hostname*]

/usr/sbin/rarpd -a

BESCHREIBUNG

rarpd aktiviert einen Dämon, der auf RARP-Anforderungen antwortet. Der Dämon erzeugt eine Kopie von sich, die im Hintergrund läuft. Er muß unter der Benutzererkennung Root ablaufen.

RARP wird vom Rechner beim Laden verwendet, um seine Internet Protocol (IP)-Adresse festzustellen. Der ladende Rechner gibt in der RARP-Anforderungsmeldung seine Ethernet-Adresse an. Anhand der Datenbanken *ethers* und *hosts* ordnet rarpd diese Ethernet-Adresse der entsprechenden IP-Adresse zu und gibt diese in einer RARP-Antwortmeldung an den Rechner zurück. Der ladende Rechner muß in beiden Datenbanken vorhanden sein, damit rarpd seine IP-Adresse finden kann. Kann rarpd eine IP-Adresse nicht finden, wird keine Antwort ausgegeben.

In der ersten Kommandoübersicht bezeichnet *schnittstelle* die Netzwerk-Schnittstelle, die rarpd auf Anforderungen hin abhört. Der Parameter *schnittstelle* hat die Form "name einheit", die *ifconfig*(1M) verwendet. Das zweite Argument, *hostname*, dient zum Feststellen der IP-Adresse dieser Schnittstelle. Bei diesem Argument kann die IP-Adresse in "Dezimal-Punkt"-Schreibweise angegeben werden. Wird *hostname* nicht angegeben, wird die Adresse der Schnittstelle vom Systemkern bezogen. Wird die erste Form des Kommandos benutzt, muß rarpd für jede Schnittstelle, auf der der RARP-Service unterstützt werden soll, einzeln aktiviert werden. Ist der Rechner ein Router, kann er rarpd mehrmals aufrufen, z.B.:

```
/usr/sbin/rarpd emd1 host
/usr/sbin/rarpd emd2 host-backbone
```

Mit der zweiten Kommandosyntax sucht rarpd alle Netzwerk-Schnittstellen im System und aktiviert einen Dämonprozeß für jede, die RARP unterstützt.

DATEIEN

/etc/ethers
/etc/hosts

SIEHE AUCH

ifconfig(1M), *ethers*(4), *hosts*(4), *netconfig*(4), *boot*(8).

Finlayson, Ross, Timothy Mann, Jeffrey Mogul, and Marvin Theimer, *A Reverse Address Resolution Protocol*, RFC 903, Network Information Center, SRI International, Menlo Park, Kalif., Juni 1984.

BEZEICHNUNG

rc0 – aktiviert Kommandos zum Stoppen des Betriebssystems

ÜBERSICHT

/etc/rc0

BESCHREIBUNG

Diese Datei wird bei allen Änderungen des Systemzustands (Systemstatus) ausgeführt, bei denen das System inaktiv sein muß. Sie ist für die Maßnahmen zuständig, die das System in einen Ruhezustand versetzen und gewöhnlich als "shutdown" bezeichnet werden.

Drei Systemzustände erfordern ein solches Vorgehen, und zwar Status 0 (Systemunterbrechung), Status 5 (Firmware-Status) und Status 6 (Neustart-Status). Bei jedem Wechsel in einen dieser Zustände wird die Prozedur rc0 durchgeführt. Der Eintrag in /etc/inittab könnte folgendermaßen aussehen:

```
r0:056:wait:/etc/rc0 off 1>/dev/sysmrg 2>&1 </dev/console
```

Einige der von rc0 aktivierten Aktionen werden von Dateien im Verzeichnis /usr/etc/shutdown.d sowie mit K beginnenden Dateien in /etc/rc0.d ausgeführt. Diese Dateien werden in ASCII-Reihenfolge ausgeführt (Näheres hierzu finden Sie im Abschnitt DATEIEN) und beenden einige Systemdienste. Aus der Kombination von Kommandos in rc0 mit Dateien in /usr/etc/shutdown.d und /etc/rc0.d ergibt sich die Art des Systemabschlusses.

Folgende Reihenfolge empfiehlt sich für rc0:

Stoppen der Systemdienste und -dämonen.

Verschiedene Systemdienste (z.B. 3BNET LAN oder LP-Spooler) werden ordnungsgemäß beendet.

Wenn neue Dienste hinzugefügt werden, die beim Systemabschluß beendet werden müssen, werden die entsprechenden Dateien in /usr/etc/shutdown.d und /etc/rc0.d gespeichert.

Beenden von Prozessen

killall(1M) sendet SIGTERM-Signale an alle laufenden Prozesse. Daraufhin stoppen die Prozesse ordnungsgemäß.

Abbrechen von Prozessen

An alle verbleibenden Prozesse werden SIGKILL-Signale gesendet, denen alle Prozesse folgen müssen.

Zu diesem Zeitpunkt sind nur noch Prozesse im Zusammenhang mit rc0 sowie die Prozesse 0 und 1 aktiv, die zum Betriebssystem gehören.

Aushängen aller Dateisysteme

Nur das Dateisystem root (/) bleibt eingehängt.

Je nach dem Status, in dem sich die Systeme dann befinden (0, 5 oder 6), bestimmen die Einträge in /etc/inittab das weitere Vorgehen. Sind in /etc/inittab keine weiteren auszuführenden Schritte definiert, wie bei Systemstatus 0, ist das System inaktiv. Es muß unmöglich sein, es zu aktivieren.

rc0(1M)

rc0(1M)

Die einzige Möglichkeit wäre, den Rechner auszuschalten oder einen Firmware-Monitor zu aktivieren. Dieses Kommando darf nur von einem privilegierten Benutzer verwendet werden.

DATEIEN

Alle Dateien in `/usr/etc/shutdown.d` werden von `/usr/bin/sh` nach der ASCII-Sortierfolge ausgeführt. Näheres finden Sie unter `rc2(1M)`.

SIEHE AUCH

`killall(1M)`, `rc2(1M)`, `shutdown(1M)`.

BEZEICHNUNG

rc2 – aktiviert Kommandos für die Mehrbenutzerumgebung

ÜBERSICHT

/etc/rc2

BESCHREIBUNG

Diese Datei wird über einen Eintrag in /etc/inittab ausgeführt und ist dafür zuständig, die Initialisierungsvorgänge zu aktivieren, die das System in einen betriebsbereiten Zustand, normalerweise Status 2 oder Mehrbenutzer-Status, bringen.

Die von rc2 ausgeführten Aktionen sind in Dateien im Verzeichnis /etc/rc.d sowie mit S beginnenden Dateien in /etc/rc2.d enthalten. Sie werden von /usr/bin/sh in ASCII-Sortierfolge ausgeführt (Näheres hierzu finden Sie im Abschnitt DATEIEN). Wenn Funktionen hinzugefügt werden, die beim Aktivieren des Mehrbenutzer-Betriebs auf dem System initialisiert werden müssen, ist eine entsprechende Datei in /etc/rc2.d aufzunehmen.

Folgende Funktionen werden von rc2- und entsprechenden /etc/rc2.d-Dateien ausgeführt:

Definieren und Exportieren der Variablen TIMEZONE.

Installieren und Einhängen des Dateisystems (/usr).

Bereinigen (Neuerstellen) der Verzeichnisse /tmp und /var/tmp.

Laden der Netzwerkschnittstellen- und Anschlußkarten mit Programmdateien und Starten der zugehörigen Prozesse.

Starten des Dämons cron durch Ausführen von /usr/etc/cron.

Bereinigen (Löschen) des Sperrstatus für uuap und temporärer Dateien im Verzeichnis /var/spool/uuap.

Zur Unterstützung der Hardware- und Software-Einrichtungen können nach Bedarf weitere Funktionen hinzugefügt werden.

BEISPIELE

Die folgenden Dateien sind typisch für die Dateien in /etc/rc2.d. Der Dateiname hat als Präfix ein S und enthält eine Zahl, die angibt, in welcher Reihenfolge die Dateien ausgeführt werden.

```

MOUNTFILESYS
#   Installieren und Einhängen von Dateisystemen

cd /
/etc/mountall /etc/fstab

RMTMPFILES
#   clean up /tmp
rm -rf /tmp
mkdir /tmp
chmod 777 /tmp
chgrp sys /tmp
chown sys /tmp

uucp
#   Bereinigen von uucp-Sperren, Status und temporären
Dateien.

rm -rf /var/spool/locks/*

```

Die Datei `/etc/TIMEZONE` wird in `rc2` früh aufgenommen, so daß die Standardzeitzone für alle folgenden Kommandos gilt.

DATEIEN

Hier einige Hinweise zu Dateien in `/etc/rc.d`:

Es ist wichtig, in welcher Reihenfolge die Dateien ausgeführt werden. Da sie nach der ASCII-Sortierfolge ausgeführt werden, läßt sich die richtige Reihenfolge am besten dadurch einhalten, daß der erste Buchstabe des Dateinamens als Bezugszeichen verwendet wird. Dadurch ergäbe sich die folgende Reihenfolge für Dateien mit den Anfangszeichen

```

[0-9].  zuerst
[A-Z].  danach
[a-z].  zuletzt

```

Dateien in `/etc/rc.d`, deren Namen mit einem Punkt (.) beginnen, werden nicht ausgeführt. Mit dieser Funktion lassen sich Dateien verstecken, die zum gegebenen Zeitpunkt nicht ausgeführt, aber auch nicht gelöscht werden sollen. Dieses Kommando darf nur von einem privilegierten Benutzer verwendet werden.

Dateien in `/etc/rc2.d` müssen mit `S` oder `K` beginnen, gefolgt von einer Zahl und dem restlichen Dateinamen. Wird Betriebsstufe 2 aktiviert, werden mit `S` beginnende Dateien mit der Option `start` ausgeführt. Mit `K` beginnende Dateien werden mit der Option `stop` ausgeführt. Mit anderen Zeichen beginnende Dateien werden ignoriert.

SIEHE AUCH

`shutdown(1M)`.

BEZEICHNUNG

rc6 – aktiviert Kommandos zum Stoppen und Neuladen des Betriebssystems

ÜBERSICHT

/etc/rc6

BESCHREIBUNG

Das Shell-Skript rc6 wird ausgeführt, wenn ein Übergang zu Betriebsart 6 von `init 6` oder `shutdown -i6` angefordert wird.

Die einzelnen Schritte von rc6 werden in folgender Reihenfolge ausgeführt:

- prüfen, ob ein neues ladbares Betriebssystem (/stand/unix) erstellt werden muß; wenn ja, wird es mit dem Kommando `buildsys` aufgebaut
- Aushängen aller Dateisysteme

Nun führt `init` den Eintrag `initdefault` in der Datei `/etc/inittab` aus, um das System in den dort definierten Betriebszustand zu versetzen.

Tritt ein Fehler auf, während `buildsys` ein neues, ladbares Betriebssystem aufbaut, wird eine Shell erzeugt, die nur mit dem Firmware-Status beendet werden kann; (siehe `buildsys(1M)`).

SIEHE AUCH

`buildsys(1M)`, `cunix(1M)`, `init(1M)`, `rc0(1M)`, `rc2(1M)`, `shutdown(1M)`, `inittab(4)`.

Leitfaden für Systemverwalter

rdate (1M)

rdate (1M)

BEZEICHNUNG

rdate – Einstellen des Datums von einem fernen Host aus

ÜBERSICHT

rdate *hostname*

BESCHREIBUNG

rdate stellt Datum und Uhrzeit eines Hosts von dem als Argument angegebenen *hostnamen* aus ein. Die Berechtigung dazu hat nur ein privilegierter Benutzer für das lokale System. Normalerweise wird rdate als Teil des Start-Skripts verwendet.

BEZEICHNUNG

relogin – Umbenennen des Login-Eintrags zum Anzeigen des aktuellen Shell-Fensters

ÜBERSICHT

/usr/lib/layer^{sys}/relogin [-s] [*leitung*]

BESCHREIBUNG

Das Kommando `relogin` ändert das Feld für die Terminal-*leitung* im `utmp`-Eintrag für einen Benutzer in den Namen des Shell-Fensters, das mit der Standardeingabe verbunden ist. An den Benutzer gesendete `write`-Meldungen gehen an dieses Fenster. Darüber hinaus kann mit dem Kommando `who` der zu diesem Fenster gehörende Benutzer angezeigt werden. `relogin` darf nur unter `layers` aufgerufen werden.

`relogin` wird von `layers` automatisch aufgerufen, um im `utmp`-Eintrag beim Start die Terminal-Leitung des ersten angelegten Fensters zu definieren und am Ende den `utmp`-Eintrag auf die reale Leitung zurückzusetzen. Das Kommando kann von einem Benutzer aufgerufen werden, um die Schicht zum Empfang von `write`-Meldungen zu wechseln.

-s Unterdrücken von Fehlermeldungen.

leitung Angabe, welcher `utmp`-Eintrag geändert werden soll. Die Datei `utmp` wird nach einem Eintrag mit dem entsprechenden Feld *leitung* durchsucht. In dieses Feld wird die Leitung eingetragen, die mit der Standardeingabe verbunden ist. Um festzustellen, welche Leitungen zu einem bestimmten Benutzer gehören, z.B. `hmay`, ist `ps -f -u hmay` einzugeben und auf die Werte im Feld TTY zu achten (siehe `ps(1)`)).

DATEIEN

/var/adm/utmp Datenbank der Benutzer gegenüber Terminals

SIEHE AUCH

`layers(1)`, `mesg(1)`, `ps(1)`, `who(1)`, `write(1)` in den *Kommandos*.
`utmp(4)`.

EXIT-CODES

Bei erfolgreicher Ausführung ist der Exit-Code 0, ansonsten 1.

HINWEIS

`relogin` schlägt fehl, wenn *leitung* nicht dem Benutzer gehört, der das Kommando `relogin` absetzt, oder wenn die Standardeingabe nicht mit einem Terminal verbunden ist.

BEZEICHNUNG

rep – Konfigurationsreport ausgeben

ÜBERSICHT

rep

BESCHREIBUNG

rep informiert Sie über die aktuelle Konfiguration. Sie benötigen diese Information u.a., wenn Sie die aktuelle Platten- oder Systemkonfiguration ändern wollen. Bei der folgenden Beschreibung erhalten Sie deshalb auch einen Überblick über die Möglichkeiten, Ihr System neu zu konfigurieren oder die ursprüngliche Konfiguration wiederherzustellen.

Die Informationen, die rep ausgibt sind in 5 Reportabschnitte unterteilt:

- A General
- B Setup
- C Disks and Filesystems
- D Keyboard
- E Link Kit

Jeder Abschnitt besteht wiederum aus Unterabschnitten, z.B. A.1, A.2, C.01. In dieser Beschreibung sind Unterabschnitte mit * gekennzeichnet, die Angaben liefern, die Sie ändern oder neu konfigurieren können. Beispielsweise bedeutet A.4*, daß Sie das Datum ändern können, das rep im Abschnitt A.4 ausgibt.

A GENERAL

Der Reportabschnitt A enthält folgende Informationen:

- A.1 Copyright-Angabe
- A.2 Version des Betriebssystems
- A.3 Hostname
- A.4 Datum. Mit dem Kommando date(1) können Sie das Datum ändern.

B SETUP

Einige der Informationen, die dieser Abschnitt liefert, sind sehr vom verwendeten BIOS abhängig:

- B.0 BIOS-Version
- B.1 CMOS RAM Status
- B.2 Angaben zur CPU-Geschwindigkeit
- B.3 CPU-Typ
- B.4 Angaben zu Cache- und Shadow-Speicher. Beim Systemtest ist eine Änderung mit CTRL-ALT-INS (Setup-Menü) möglich.
- B.5 Angaben zum arithmetischen Koprozessor
- B.6 Speichergrößen. Die Summe der beiden Werte Base Memory Size und Extended Memory Size ergibt die Hauptspeichergröße.
- B.7 Angaben zu den Diskettenlaufwerken. Beim Systemtest ist eine Änderung mit CTRL-ALT-INS (Setup-Menü) möglich.

- B.8 Angaben zur Festplattenkonfiguration (ESDI, ST506, RLL und AT-BUS). Beim Systemtest ist eine Änderung mit CTRL-ALT-INS (Setup-Menü) möglich. rep liefert in diesem Abschnitt keine Informationen über SCSI-Platten; vielmehr erfolgt die Meldung `Not Installed`.
- B.9 Video Adapter Konfiguration. Beim Systemtest ist eine Änderung mit CTRL-ALT-INS (Setup-Menü) möglich.

C DISKS AND FILESYSTEMS

rep informiert in diesem Reportabschnitt über maximal 6 Festplatten. Zu jeder installierten Festplatte gibt es weitere Unterabschnitte mit Informationen zu den einzelnen Partitionen. Mit `fdisk(1M)` und `disksetup(1M)` können Sie die Aufteilung einer Platte in Partitionen und Slices ändern. Eine Beschreibung der Neuaufteilung einer Platte finden Sie in der *Installationsanleitung*.

- C.00 Angaben zur ersten ESDI-, ST506-, RLL- bzw. AT-BUS-Platte
- C.01 Angaben zur zweiten ESDI-, ST506-, RLL- bzw. AT-BUS-Platte
- C.80 Angaben zur ersten SCSI-Platte
- C.81 Angaben zur zweiten SCSI-Platte
- C.90 Angaben zur dritten SCSI-Platte
- C.91 Angaben zur vierten SCSI-Platte

D KEYBOARD

rep gibt im Reportabschnitt D nur dann korrekte Angaben aus, wenn Sie das Kommando von der Konsole aus aufrufen.

- D* Angaben zur Tastatur. Sie ändern den Tastaturmodus (keyboard mode) mit `kbmode`. Die Default-Mapkey-Datei ändern Sie mit `mapkey`, die Default-Mapchan-Datei mit `mapchan`.

E LINK KIT

- E.1 Interrupt- und DMA-Konfiguration
- E.2* Informationen zur SCSI System-Konfiguration.
 - E.2.1* SCSI Adapter
 - E.2.2* SCSI Platten
 - E.2.3* SCSI Magnetbandlaufwerke
 - E.2.4* SCSI Worm-Laufwerke
 - E.2.5* SCSI Plattenwechsler
 - E.2.6* SCSI CD-ROM-Laufwerke
- E.3* WD 8003 Controller Konfiguration.
- E.4* EXOS 105 Controller Konfiguration.

BEISPIEL

SYSYEM CONFIGURATION REPORT

A GENERAL

A.1 COPYRIGHT
Copyright (C) Siemens Nixdorf Informationssysteme AG 1991
All rights reserved

A.2 SYSTEM
SINIX-D Release 5.41 Version A0003

A.3 HOSTNAME
almeria

A.4 DATE
Mon Feb 10 14:21:14 MET 1992

B SETUP

B.0 BIOS
BIOS Version 3.10 Rev. 2.02
COPYRIGHT 1984,1989 Award Software Inc.ALL RIGHTS RESERVED

B.1 CMOS RAM STATUS
CMOS RAM Diagnostic Status OK.
ERROR HALT on ALL ERRORS.

B.2 CPU SPEED SELECTOR
CPU is running at High Speed Setting.

B.3 CPU TYPE
CPU: 80486 SLOT-CPU.

B.4 CACHE AND SHADOW MEMORY
CACHE MEMORY Enabled.

B.5 ARITHMETIC COPROCESSOR
Coprocessor installed; Typ: 80387

B.6 MEMORY SIZES
Base Memory Size: 640 KBytes.
Extended Memory Size: 31744 KBytes.

B.7 FLOPPY DISK CONFIGURATION
One Floppy Disk Drive installed.
Floppy Disk Drive 0: 3.5" HD (1.44 MByte).
Floppy Disk Drive 1: Not Installed.

B.8 HARD DISK CONFIGURATION (ESDI, ST506, RLL and AT-BUS)
SCSI-Disks please look below.

Hard Disk Drive 00: Not Installed.

Hard Disk Drive 01: Not Installed.

B.9 VIDEO ADAPTER CONFIGURATION

Primary CRT Controller: EGA/VGA

C DISKS AND FILESYSTEMS

C.00 Disk 00 : Not installed

C.01 Disk 01 : Not installed

C.80 Disk 80 :

Partition 1: "root" size: 61440
start: 145456 last: 206895
tag: ROOT permission: VALID

Partition 2: "swap" size: 131072
start: 14384 last: 145455
tag: SWAP permission: NOMOUNT

Partition 3: "usr" size: 112640
start: 206896 last: 319535
tag: USR permission: VALID

Partition 4: "home4" size: 372736
start: 319536 last: 692271
tag: HOME permission: VALID

Partition 7: "reserved" size: 34
start: 1 last: 34
tag: BOOT permission: NOMOUNT

Partition 10: "stand" size: 14336
start: 48 last: 14383
tag: STAND permission: VALID

Partition 11: "var" size: 274432
start: 692272 last: 966703
tag: VAR permission: VALID

Partition 12: "home12" size: 336848
start: 966704 last: 1303551
tag: USR permission: VALID

rep (1M)

(WX200)

rep (1M)

C.81 Disk 81 :
Partition 1: "home1" size: 2029520
start: 48 last: 2029567
tag: USR permission: VALID
Partition 7: "reserved" size: 34
start: 1 last: 34
tag: BOOT permission: NOMOUNT

C.90 Disk 90 : Not installed

C.91 Disk 91 : Not installed

D KEYBOARD

Program running from /dev/pts/3
Keyboard is in raw mode.
Default mapkey file is /usr/lib/keyboard/us.wx200.
Cannot determine default mapchan file

E LINK KIT

E.1 INTERRUPT AND DMA CONFIGURATION

Driver	Interrupt	DMA Channel
kd	1	-
fas	3	-
fas	4	-
wd	5	-
fd	6	2
lp	7	-
rtc	8	-
scsi	11	5
scsi	12	5
fp	13	-
hd	14	-

rep(1M)

(WX200)

rep(1M)

```
E.2  SCSI SUBSYSTEM CONFIGURATION
E.2.1 SCSI ADAPTERS
      Adapter: 0, SCSI Id: 7,      (typ Adaptec 1542)
      Inter. Vector: 11,      Base I/O Addr.: 330
E.2.2 SCSI DISKS
      2 SCSI disks are configured.
      Device: MICROP 1588-15MB1036511      Target id: 0
      Host adapter: 0      #LUS: 1
      Device: MICROP 1598-15MD1052405      Target id: 1
      Host adapter: 0      #LUS: 1
E.2.3 SCSI TAPE DRIVES
      1 SCSI tape drive is configured.
      Device: TANDBERG TDC 3600      Target id: 2
      Host adapter: 0      #LUS: 1
E.2.4 SCSI WORM DRIVES
      No SCSI WORM drive is configured.
E.2.5 SCSI MEDIA CHANGERS
      No SCSI media changer is configured.
E.2.6 SCSI CDROM DRIVES
      No SCSI CDROM drive is configured.

E.3  WD 8003 CONTROLLER CONFIGURATION
      1 WD 8003 board is configured
      Board 0:  int 5,  io 240-25F, dual port ram D0000-D1FFF

E.4  EXOS 105 CONTROLLER CONFIGURATION
      No EXOS 105 board is configurd.
```

DATEIEN

/etc/rep

repquota (1M)

(UFS)

repquota (1M)

BEZEICHNUNG

repquota – Überblick über die Speicherplatzbelegung und Plattenquoten von Benutzern für ein Dateisystem

ÜBERSICHT

repquota [-v] *dateisystem*...

repquota [-av]

BESCHREIBUNG

repquota gibt eine Übersicht über Plattenbelegung und Quoten für die angegebenen Dateisysteme aus. Für jeden Benutzer wird die aktuelle Anzahl seiner Dateien und die Summe des belegten Plattenplatzes (in KB) ausgegeben, dazu alle mit edquota zugewiesenen Quoten.

Folgende Optionen stehen zur Verfügung:

-a Bericht über alle Dateisysteme, bei denen "rq" im Feld `mntopts` der Datei `/etc/vfstab` steht.

-v Ausgabe aller Quoten, auch wenn kein Plattenplatz beansprucht wird.

Nur privilegierte Benutzer dürfen sich fremde Quoten anzeigen lassen.

SIEHE AUCH

edquota(1M), quota(1M), quotacheck(1M), quotaon(1M)

restore (1M)

restore (1M)

BEZEICHNUNG

`restore` – startet die Wiederherstellung von Dateisystemen, Datenpartitionen oder Platten.

ÜBERSICHT

`restore [-o ziel] [-d datum] [-mn] [-s | v] -P partger`

`restore [-o ziel] [-d datum] [-mn] [-s | v] -S agerät`

`restore [-o ziel] [-d datum] [-mn] [-s | v] -A partger`

BESCHREIBUNG

`restore` setzt Anforderungen zum Wiederherstellen einer Datenpartition, einer Dateisystempartition oder einer Platte aus vom System unterhaltenen Archiven ab. Handelt es sich bei dem Archiv, das die benötigte Partition enthält, um ein Online-Archiv, wird die Partition sofort wiederhergestellt. Anderenfalls wird eine Anforderung zum Wiederherstellen des betreffenden Archivs für die Partition an eine Tabelle für Wiederherstellungs-Status abgesetzt. Diese Tabelle trägt den Namen `/etc/bkup/rsstatus.tab`. Die Anforderung zum Wiederherstellen erhält eine *restore-auftragsnummer*, mit der der Fortschritt der Wiederherstellung überwacht oder diese annulliert werden kann. Eine abgesetzte Anforderung zum Wiederherstellen muß später von einem Bediener aufgelöst werden (siehe `rsoper(1M)`).

`restore` darf nur von einem privilegierten Benutzer benutzt werden.

Wird `restore -A partger` ausgegeben, wird die Platte *partger* mit der Methode `fdisk(1M)` (Wiederherstellen der gesamten Platte) neu partitioniert und mit Einträgen versehen. *partger* ist der Name des Geräts, das die gesamte Platte umfaßt. Für den AT&T 3B2 ist dies `/dev/rdisk/c1d?s6`.

Optionen

`-d datum` Stellt die Partition per *datum* wieder her. Dies kann, muß aber nicht das neueste Archiv sein. Die gültigen Datumsformate entnehmen Sie bitte dem Handbucheintrag `getdate(1M)`.

`-m` Wenn die Wiederherstellung nicht sofort ausgeführt werden kann, benachrichtigt diese Option den aufrufenden Benutzer (über `mail(1M)`), wenn die Anforderung abgeschlossen ist.

`-n` Zeigt eine Liste aller archivierten Versionen des Objekts im Sicherungsprotokoll an, versucht aber nicht, das Objekt wiederherzustellen.

`-o ziel` Anstatt die Wiederherstellung direkt auf dem angegebenen Objekt (*partger* oder *dsger*) vorzunehmen, wird sie für *ziel* durchgeführt. *ziel* hat die Form:

`[aname][:ager]`

Dabei ist *aname* der Name des Dateisystems, das wiederhergestellt werden soll (für `-S`-Archive), und *ager* ist der Name der Partition, in der die Wiederherstellung erfolgen soll (für `-P`- und `-A`-Archive).

`-s` Während der Wiederherstellung wird alle 100 vom Ziel-Gerät übertragenen Blöcke (zu 512 Byte) ein Punkt "." angezeigt.

restore (1M)

restore (1M)

- v Zeigt den Namen des jeweils zurückgespeicherten Objekts an. Diese Option wird nur von den Archivierungsmethoden unterstützt, die benannte Verzeichnisse und Dateien (*incfile*, *ffile*) wiederherstellen.
- A *partger* Startet die Wiederherstellung der gesamten Platte *partger*.
- P *partger* Startet die Wiederherstellung der Datenpartition *partger*.
- S *agerät* Startet die Wiederherstellung der Dateisystempartition *agerät*.

EXIT-CODES

Die Exit-Codes von `restore` haben folgende Bedeutung:

- 0 = Aufgabe erfolgreich abgeschlossen
- 1 = ein oder mehrere Parameter zu `restore` sind ungültig
- 2 = es ist ein Fehler aufgetreten, aufgrund dessen `restore` nicht alle Teile der Aufgabe ausführen konnte

BEISPIELE

Beispiel 1:

```
restore -m -S /usr
```

Setzt eine Anforderung ab, um die aktuellste archivierte Version von `/usr` wiederherzustellen. Wenn die Wiederherstellung nicht sofort ausgeführt werden kann, soll der aufrufende Benutzer benachrichtigt werden, wenn sie abgeschlossen ist.

Beispiel 2:

```
restore -o /dev/rdisk/0s8 -P /dev/rdisk/1s2
```

Setzt eine Anforderung ab, wonach die archivierte Datenpartition `/dev/rdisk/0s8` auf der Zielgerätpartition `/dev/rdisk/c1d0s8` wiederhergestellt werden soll.

Beispiel 3:

```
restore -d "december 1, 1987" -A /dev/rdisk/0s6
```

Setzt eine Anforderung ab, die gesamte Platte `/dev/rdisk/0s6` wiederherzustellen. Die Wiederherstellung soll per 1. Dezember 1987 ausgeführt werden.

Beispiel 4:

```
restore -n -P /dev/drsk/0s1
```

Fordert das System auf, das Sicherungsdatum und eine `ls -l`-Liste aus dem Sicherungsprotokoll anzuzeigen, die alle archivierten Versionen der Datenpartition `/dev/rdisk/0s1` enthält. Die Datenpartition wird nicht wiederhergestellt.

DATEIEN

- `/etc/bkup/bkhist.tab` gibt eine Liste der Etiketten aller Datenträger aus, die für Sicherungsoperationen verwendet wurden
- `/etc/bkup/rsstatus.tab` erstellt eine Liste der Stati aller von Benutzern abgesetzten Anforderungen zum Wiederherstellen
- `/etc/bkup/rsnotify.tab` erstellt eine Adreßliste für elektronische Post für den Bediener, der benachrichtigt werden soll, wenn Anforderungen zum Wiederherstellen den Eingriff eines Bedieners erfordern.

restore (1M)

restore (1M)

SIEHE AUCH

fdisk(1M), getdate(3), mail(1M), rsnotify(1M), rsoper(1M), rsstatus(1M),
urestore(1M), ursstatus(1M)

BEZEICHNUNG

rexecd – Server für Fernausführung

ÜBERSICHT

in.rexecd host .port

BESCHREIBUNG

in.rexecd ist der Server für die Routine *rexec(3N)*. Er stellt Einrichtungen zur Fernausführung zur Verfügung und führt Gültigkeitsprüfungen anhand von Benutzernamen und verschlüsselten Paßwörtern durch. Der Server wird bei Bedarf automatisch von *inetd(1M)* aufgerufen und führt dann das folgende Protokoll aus:

- 1) Der Server liest Zeichen von der Socket-Schnittstelle bis zu einem Null-`(\0)`-Byte. Die sich daraus ergebende Zeichenkette wird als ASCII-Zahl zur Basis 10 interpretiert.
- 2) Ist die aus Schritt 1 resultierende Zahl nicht Null, wird sie als Anschlußnummer eines Sekundär-Datenstroms interpretiert, der für *stderr* verwendet werden soll. Daraufhin wird eine zweite Verbindung zu dem angegebenen Anschluß am Rechner des Client hergestellt.
- 3) Von der Socket-Schnittstelle, die das Protokoll eingeleitet hat, wird ein mit Null abschließender Benutzername mit maximal 16 Zeichen abgerufen.
- 4) Von der Socket-Schnittstelle, die das Protokoll eingeleitet hat, wird ein mit Null abschließendes, verschlüsseltes Paßwort mit maximal 16 Zeichen abgerufen.
- 5) Von der Socket-Schnittstelle, die das Protokoll eingeleitet hat, wird ein mit Null abschließendes Kommando abgerufen, das an eine Shell übergeben werden soll. Das Kommando darf nur so lang sein, wie es die Obergrenze für die Größe der Argumentliste für das System zuläßt.
- 6) Nun überprüft *in.rexecd* die Berechtigung des Benutzers wie beim Laden, wechselt bei Erfolg in das Home-Verzeichnis des Benutzers und richtet die Benutzer- und Gruppenrechte für den Benutzer ein. Schlägt einer dieser Schritte fehl, wird die Verbindung abgebrochen, und es wird eine Diagnosemeldung zurückgegeben.
- 7) Über die zu *stderr* gehörende Verbindung wird ein Null-Byte zurückgegeben, und die Kommandozeile wird an die normale Anmelde-Shell des Benutzers übergeben. Die Shell übernimmt die von *in.rexecd* hergestellten Netzwerkverbindungen.

SIEHE AUCH

inetd(1M)

FEHLERMELDUNGEN

Alle Fehlermeldungen werden über die zur Standardfehlerausgabe *stderr* gehörende Verbindung zurückgegeben, danach werden alle Netzwerkverbindungen beendet. Ein Fehler wird mit einem führenden Byte mit dem Wert 1 angezeigt (0 wird in Schritt 7 bei erfolgreichem Abschluß aller Schritte vor Ausführung des Kommandos zurückgegeben).

`username too long` (Benutzername zu lang)
Der Name ist länger als 16 Zeichen.

`password too long` (Paßwort zu lang)
Das Paßwort ist länger als 16 Zeichen.

`command too long` (Kommando zu lang)
Die übergebene Kommandozeile ist länger als die für das System konfigurierte Argumentliste.

`Login incorrect.` (Anmeldung nicht korrekt)
Für den Benutzernamen ist kein Eintrag in der Paßwortdatei vorhanden.

`Password incorrect.` (Paßwort nicht korrekt)
Es wurde ein falsches Paßwort eingegeben.

`No remote directory.` (Kein fernes Verzeichnis)
Das Kommando `chdir` an das Home-Verzeichnis ist fehlgeschlagen.

`Try again.` (Kommando wiederholen)
Ein vom Server ausgegebenes Kommando `fork` ist fehlgeschlagen.

`/usr/bin/sh: ...`
Die Login-Shell des Benutzers konnte nicht gestartet werden.

HINWEIS

Wird die Meldung `Login incorrect` (Anmeldung nicht korrekt) anstelle von `Password incorrect` (Paßwort nicht korrekt) ausgegeben, deutet dies auf eine Verletzung der Datensicherheit hin. Damit kann ein System auf Benutzer überprüft werden, für die kein Paßwort eingerichtet ist.

Es sollte eine Einrichtung zum Verschlüsseln des gesamten Datenaustauschs vorhanden sein.

BEZEICHNUNG

`rlogind` in `rlogind` – Server für fernes Login

ÜBERSICHT

`in.rlogind` *host.anschluß*

BESCHREIBUNG

`in.rlogind` ist der Server für das Programm `rlogin(1)`. Er stellt eine Funktion für fernes Login mit einer Berechtigungsprüfung auf der Basis privilegierter Anschlußnummern zur Verfügung.

`in.rlogind` wird von `inetd(1M)` aufgerufen, wenn eine Verbindung zum fernen Login hergestellt wird, und führt das folgende Protokoll aus:

- 1) Der Server prüft den Ausgangsanschluß des Client. Liegt der Anschluß nicht im Bereich zwischen 0 und 1023, bricht der Server die Verbindung ab. Adresse und Anschlußnummer des Client werden als Argumente von `inetd` in der Form *host.anschluß an* `in.rlogind` übergeben. Dabei wird der Host als Hexadezimal- und der Anschluß als Dezimalwert angegeben.
- 2) Der Server prüft die Ausgangsadresse des Client. Ist für den Client sowohl in `/etc/hosts` als auch in `/etc/hosts.equiv` ein Eintrag vorhanden, wird ein Benutzer, der sich vom Client aus anmeldet, nicht zur Eingabe eines Paßworts aufgefordert. Gehört die Adresse zu einem Host, für den in `/etc/hosts` kein entsprechender Eintrag vorhanden ist, wird der Benutzer zur Eingabe eines Paßworts aufgefordert, unabhängig davon, ob für den Client in `/etc/hosts.equiv` (siehe `hosts(4)` und `hosts.equiv(4)`) ein Eintrag steht.

Nachdem Ausgangsanschluß und -adresse geprüft wurden, weist `in.rlogind` ein Pseudo-Terminal zu und bearbeitet die Dateideskriptoren, so daß der Slave-Teil des Pseudo-Terminals zur `stdin` (Standardeingabe), `stdout` (Standardausgabe) und `stderr` (Standardfehlerausgabe) für einen Login-Prozeß wird. Der Login-Prozeß ruft das Programm `login(1)` mit der Option `-r` auf. Er führt dann die Berechtigungsprüfung gemäß `rshd(1M)` durch. Schlägt die automatische Berechtigungsprüfung fehl, wird der Benutzer zum erneuten Login wie auf einer Standard-Terminal-Leitung aufgefordert.

Der Vaterprozeß des Login-Prozesses bearbeitet den Master-Teil des Pseudo-Terminals und fungiert dort als Vermittler zwischen dem Login-Prozeß und dem ablaufenden Programm `rlogin` auf dem Client. Bei Normalbetrieb wird ein Paketprotokoll aufgerufen, um Einrichtungen des Typs `Ctrl-S` / `Ctrl-Q` bereitzustellen und Interrupt-Signale an die fernen Programme zu senden. Der Login-Prozeß sendet die Übertragungsgeschwindigkeit und den Typ des Client-Terminals entsprechend der Umgebungsvariablen `TERM`; siehe `environ(4)`.

SIEHE AUCH

`inetd(1M)`, `hosts(4)`, `hosts.equiv(4)`.

FEHLERMELDUNG

Alle Fehlermeldungen werden über die zu `stderr` gehörende Verbindung zurückgegeben, anschließend werden die Verbindungen abgebrochen. Ein Fehler ist durch ein führendes Byte mit dem Wert 1 gekennzeichnet.

Hostname for your address unknown. (Host-Name für Ihre Adresse unbekannt) Für den Client-Rechner war in der Namensdatenbank des Hosts kein Eintrag vorhanden.

Try again. (Aufruf wiederholen)

Der Versuch des Servers, mit `fork` einen neuen Prozeß zu erzeugen, schlug fehl.

/usr/bin/sh: ...

Die Anmelde-Shell des Benutzers konnte nicht aktiviert werden.

HINWEIS

Die hier verwendete Berechtigungsprüfung geht davon aus, daß alle Client-Rechner sowie das Übertragungsmedium intakt sind. Dies ist unsicher, aber in einer "offenen" Umgebung nützlich.

Es muß eine Funktion zum Verschlüsseln der gesamten Datenübertragung vorhanden sein.

BEZEICHNUNG

rootcp – Root-Dateisystem kopieren

ÜBERSICHT

/usr/sbin/rootcp *newroot*

BESCHREIBUNG

rootcp kopiert entweder das aktuelle root-Dateisystem auf eine andere Slice und/oder die aktuelle Boot-Slice auf eine andere Slice. Bei Beschädigung der aktuellen Root-/Boot-Slices kann dann das System von den neuen Slices geladen werden.

Im ersten Fall führt rootcp neben dem Kopieren des root-Dateisystems nach *newroot* noch folgende Anpassungen durch: rootcp trägt den Dateisystemtyp von *newroot* in die Datei /etc/vfstab ein. Weiter erstellt rootcp eine leere Datei /etc/mnttab. In der Datei /stand/boot.alt werden in den Bootstring die Gerätenummer (minor number) von *newroot* und die Gerätenummer des neuen Swap-Bereichs eingetragen. Außerdem wird in dieser Datei der Dateityp von *newroot* hinterlegt. Schließlich paßt /etc/rc2.d/S90rootcp beim nächsten Booten das NVRAM an die neue Gerätenummer des Swap-Bereichs an.

Soll rootcp eine neue Boot-Slice anlegen, dann muß die Frage

Do you want a new bootslice (y/n)

mit **y** beantwortet werden. rootcp kopiert die Dateien /stand/unix, /stand/boot und /stand/boot.alt auf die neue Boot-Slice und führt noch folgende Anpassungen durch: Der Dateisystemtyp und die Slice von /stand werden in die Datei /etc/vfstab, die Gerätenummer der neuen Boot-Slice in den Bootstring der Datei /stand/boot.alt eingetragen.

newroot Geräte-Name der neuen Root-/Boot-Slice im Blockmodus. Eine mögliche Angabe ist z.B. /dev/dsk/c0d1s1. *newroot* muß die Geräteklassennummer (major number) 0 haben und darf keine primäre oder sekundäre VPSS-Partition (Spiegelplatten-Partition) sein.

Booten von einer alternativen Partition

Wenn hd(10) die aktuelle Boot-Partition ist, dann wird beim Hochfahren des Systems mit "CTRL DEL" in den Monitor geschaltet und folgendes Kommando eingegeben:

```
: sinixloc "hd(10,0)boot.alt"
: bootseq
```

Beim Booten von der neuen Boot-Slice /dev/dsk/c0d1s4 würde die Eingabe lauten:

```
: sinixloc "hd(1028,0)boot.alt"
: bootseq
```

Auf der neuen Boot-Partition existiert eine eigene Swap-Slice. Wenn von der alternativen Boot-Partition hochgefahren, der Dump aber von der ursprünglichen Swap-Slice gesichert werden soll, dann sind folgende Schritte durchzuführen:

1) Link zur ursprünglichen Swap-Slice (hier /dev/dsk/c0d0s2) herstellen

```
rm /dev/swap
rm /dev/rswap
ln /dev/dsk/c0d0s2 /dev/swap
ln /dev/rdisk/c0d0s2 /dev/rswap
```

2) Dump kopieren

```
/usr/sbin/savecore -s
```

3) Link zur alternativen Swap-Slice (hier /dev/dsk/c0d1s2) herstellen

```
rm /dev/swap
rm /dev/rswap
ln /dev/dsk/c0d1s2 /dev/swap
ln /dev/rdisk/c0d1s2 /dev/rswap
```

FEHLERMELDUNG

Ist die Geräteklassennummer von *newroot* nicht korrekt, dann beendet sich rootcp mit dem Exit-Code 2 und gibt folgende Fehlermeldung aus:

```
rootcp on device newroot not possible!
Wrong majornumber (n).
n = majornumber
```

Sollte eine neue Boot-Slice angelegt werden, dann lautet die Fehlermeldung:

```
New bootslice on device newroot not possible!
Wrong majornumber (n).
n = majornumber
```

HINWEIS

Standardmäßig wird auf *newroot* die Partition 2 als Swap-Partition eingetragen. Falls *newroot* selbst Partition 2 ist, dann wird Partition 1 als Swap-Partition verwendet.

BEZEICHNUNG

route – manuelle Bearbeitung der Routen-Tabelle

ÜBERSICHT

```
route [ -fn ] { add | delete } { ziel | default } [ host | net ]
      [ gateway [ metrischer_wert ] ]
```

BESCHREIBUNG

Mit `route` lassen sich die Routen-Tabellen im Netzwerk, die normalerweise vom Routen-Dämon des Systems, `routed(1M)`, oder über Standardrouten und Umleitungsmeldungen von Routern verwaltet werden, manuell bearbeiten. Mit `route` kann ein privilegierter Benutzer die Daten für den betreffenden Host bzw. das betreffende Netzwerk, *ziel*, direkt in der Routen-Tabelle bearbeiten. Auf `default` können Gateways zurückgreifen, nachdem sie versucht haben, alle anderen Routen zu benutzen. Wurde das Argument *gateway* angegeben, bezeichnet es den Netzwerk-Gateway, an den Pakete adressiert werden müssen. Das Argument *metrischer_wert* gibt die Anzahl der Knoten bis zum *ziel* an. Der *metrische_wert* wird für `add`-Kommandos (Kommandos zum Hinzufügen) benötigt; er muß Null sein, wenn es sich bei dem Ziel um ein direkt angeschlossenes Netzwerk handelt, und einen anderen Wert enthalten, wenn die Route einen oder mehrere Gateways verwendet.

Das Kommando `add` weist `route` an, eine Route zu *ziel* hinzuzufügen. Mit `delete` wird eine Route gelöscht.

Routen zu einem bestimmten Host müssen von Routen zu einem Netzwerk verschieden sein. Die optionalen Schlüsselwörter `net` und `host` erzwingen eine Interpretation als Route zu einem Netzwerk bzw. Host. Enthält das Ziel einen lokalen Adreßteil `INADDR_ANY`, wird angenommen, daß die Route zu einem Netzwerk führt; anderenfalls wird er als Route zu einem Host angesehen. Führt die Route über einen Gateway zu einem Ziel, muß der Parameter *metrischer_wert* größer als 0 sein. Wird eine Route mit dem metrischen Wert 0 hinzugefügt, ist der angegebene Gateway die Adresse dieses Hosts im allgemeinen Netzwerk, die die Schnittstelle angibt, die direkt für die Datenübertragung verwendet werden soll. Alle für ein *ziel* (außer `default`) oder *gateway* angegebenen symbolischen Namen werden mit `gethostbyname(3N)` in der Datenbank des Hosts gesucht. Schlägt diese Suche fehl, wird der Name mit `getnetbyname(3N)` in der Netzwerkdatenbank gesucht.

OPTIONEN

- f Löscht die Daten in den Routen-Tabellen aller Gateway-Einträge. Wird diese Option zusammen mit einem der oben beschriebenen Kommandos angegeben, leert `route` die Gateways, bevor das Kommando ausgeführt wird.
- n Verhindert den Versuch, beim Melden von Aktionen Host- und Netzwerknamen symbolisch auszugeben. Dies ist z.B. dann nützlich, wenn alle Name-Server im lokalen Netz deaktiviert sind, so daß eine Route erforderlich ist, um die Verbindung zum Name-Server herzustellen.

route (1M)

route (1M)

DATEIEN

/etc/hosts
/etc/networks

SIEHE AUCH

ioctl(2), gethostbyname(3N), getnetbyname(3N), routing(4N), routed(1M).

FEHLERMELDUNGEN

`add [ host | net ] ziel:gateway`

Die angegebene Route wird in die Tabellen aufgenommen. Die ausgegebenen Werte stammen aus dem Eintrag in der Routen-Tabelle, der in dem Aufruf `ioctl(2)` angegeben wurde.

`delete [ host | net ] ziel:gateway`

Die angegebene Route wird gelöscht.

`ziel done`

Wird die Option `-f` angegeben, wird nach jedem Löschen eines Eintrags aus der Routen-Tabelle eine Meldung dieses Formats ausgegeben.

`Network is unreachable` (Netzwerk nicht zu erreichen)

Der Versuch, eine Route hinzuzufügen, schlug fehl, weil der aufgeführte Gateway kein direkt angeschlossenes Netzwerk war; statt dessen den Gateway am nächsten Knoten angeben.

`not in table` (nicht in der Tabelle)

Es wurde versucht, einen Eintrag, der nicht in der Tabelle stand, zu löschen.

`routing table overflow` (Überlauf der Routen-Tabelle)

Es wurde versucht, eine Route hinzuzufügen, aber das System konnte zum Anlegen des neuen Eintrags keinen Speicherplatz zuweisen.

BEZEICHNUNG

`routed`, `in.routed` – Routen-Dämon im Netzwerk

ÜBERSICHT

`in.routed [-gstv] [protokolldatei]`

BESCHREIBUNG

`in.routed` wird beim Laden des Systems zum Verwalten der Routen-Tabellen des Netzwerks aufgerufen. Der Routen-Dämon verwendet eine Variante des Xerox NS Routing Information Protocol, um die Einträge in der Routen-Tabelle auf dem neuesten Stand zu halten.

Bei Normalbetrieb hört `in.routed` den `udp(4P)`-Socket 520 (dezimal) auf Routen-datenpakete hin ab. Handelt es sich beim Host um einen Netzverbund-Router, liefert er in regelmäßigen Abständen Kopien seiner Routen-Tabellen an alle direkt angeschlossenen Hosts und Netzwerke.

Wenn der Dämon `in.routed` gestartet wird, sucht er mit `SIOCGIFCONF ioctl(2)` die direkt angeschlossenen Schnittstellen, die in das System konfiguriert und als aktiv gekennzeichnet sind (die Software-Loopback-Schnittstelle wird ignoriert). Sind mehrere Schnittstellen vorhanden, wird angenommen, daß der Host Pakete zwischen einzelnen Netzwerken weiterleitet. In diesem Fall sendet `in.routed` ein Anforderungspaket an jede Schnittstelle (mit einem Broadcast-Paket, wenn die Schnittstelle dies unterstützt), startet eine Schleife und horcht dort nach Anforderungs- und Antwortpaketen von anderen Hosts.

Beim Empfang eines Anforderungspakets formuliert `in.routed` anhand der Daten in seinen internen Tabellen eine Antwort. Dieses Antwortpaket enthält eine Liste bekannter Routen, die jeweils mit einem metrischen Wert für die Anzahl der Knoten markiert sind (ein Wert von 16 oder höher wird als unendlich angesehen). Der jeder Route zugeordnete, zurückgegebene metrische Wert bezieht sich auf den Absender.

Anforderungspakete, die `in.routed` empfängt, werden zum Aktualisieren der Routen-Tabellen verwendet, wenn eine der folgenden Bedingungen erfüllt ist:

- (1) Für das Zielnetzwerk bzw. den Ziel-Host besteht kein Eintrag in der Routen-Tabelle, und aus dem metrischen Wert geht hervor, daß das Ziel erreichbar ist, d.h. die Anzahl der Knoten ist nicht unendlich.
- (2) Der Ursprungs-Host des Pakets ist mit dem Router in dem vorhandenen Eintrag in der Routen-Tabelle identisch. D.h. aktualisierte Daten stammen von demselben Netzverbund-Router, über den die Pakete für das Ziel geleitet werden.
- (3) Der vorhandene Eintrag in der Routen-Tabelle wurde eine Zeitlang (definitionsgemäß 90 Sekunden) nicht aktualisiert, und die Route ist mindestens so kostengünstig wie die aktuelle.
- (4) Die neue Route gibt einen kürzeren Weg zum Ziel an als die, die zur Zeit in den Routen-Tabellen gespeichert ist; um dies zu erkennen, wird der metrische Wert der neuen Route mit dem der gespeicherten verglichen.

Bei einer Aktualisierung zeichnet `in.routed` die Änderung in seinen internen Tabellen auf und erstellt ein Antwortpaket an alle direkt angeschlossenen Hosts und Netzwerke. `in.routed` wartet kurz (höchstens 30 Sekunden), bevor er die Routen-Tabellen des Kernels ändert, damit eventuell bestehende Instabilitäten bereinigt werden können.

`in.routed` verarbeitet nicht nur die ankommenden Pakete, sondern prüft auch regelmäßig die Einträge in der Routen-Tabelle. Wurde ein Eintrag 3 Minuten lang nicht aktualisiert, wird sein metrischer Wert auf unendlich gesetzt und zum Löschen freigegeben. Das Löschen selbst erfolgt erst nach einer zusätzlichen Wartezeit von 60 Sekunden, um sicherzugehen, daß die Ungültigkeitsmeldung im gesamten Netzwerkverbund bekannt ist.

Als Netzwerkverbund-Router fungierende Hosts übermitteln ihre Routen-Tabellen alle 30 Sekunden frei an alle direkt angeschlossenen Hosts und Netzwerke.

Ist die Option `-s` angegeben, wird `in.routed` gezwungen, Routen-Daten zu liefern, unabhängig davon, ob er als Netzwerkverbund-Router fungiert oder nicht. Option `-q` ist das Gegenstück zu Option `-s`. Ist die Option `-t` angegeben, werden alle gesendeten oder empfangenen Pakete auf der Standardausgabe ausgegeben. Außerdem trennt sich `in.routed` nicht von dem steuernden Terminal, so daß über Tastatur eingegebene Interrupts den Prozeß beenden können. Alle sonstigen angegebenen Argumente werden als Name der Datei interpretiert, in der die Aktionen von `in.routed` protokolliert werden sollen. Dieses Protokoll enthält Informationen zu allen Änderungen der Routen-Tabellen und ein Protokoll der jüngsten gesendeten und empfangenen Nachrichten mit Bezug auf die geänderten Routen. Mit der Option `-v` kann eine Protokolldatei erstellt werden, die die an den Routen-Tabellen vorgenommenen Änderungen mit einem Zeitstempel anzeigt.

Neben den obengenannten Einrichtungen unterstützt `in.routed` das Konzept entfernter passiver und aktiver Gateways. Wenn `in.routed` aktiviert wird, liest er die Datei `gateways` und sucht darin die Gateways, die u.U. nicht mit `SIIOIFCONF` `ioctl` zu finden sind. So bezeichnete Gateways sind als passiv zu markieren, wenn sie keine Routeninformationen austauschen sollen, während als aktiv die Gateways markiert werden müssen, die zum Austausch dieser Informationen bereit sind (d.h. auf dem Rechner muß ein `in.routed`-Prozeß laufen). Passive Gateways werden in den Routen-Tabellen zeitlich unbegrenzt geführt, und Angaben zu ihrer Existenz sind in allen übertragenen Routendaten enthalten. Aktive Gateways werden wie Netzwerkschnittstellen behandelt. Routeninformationen werden an den Gateway verteilt, und wenn eine bestimmte Zeit lang keine Routeninformationen empfangen werden, wird die entsprechende Route gelöscht.

Die Datei `gateways` besteht aus mehreren Zeilen mit dem folgenden Format:

```
< net | host > dateiname1 gateway dateiname2 metric wert < passive |
active >
```

Das Schlüsselwort `net` oder `host` gibt an, ob die Route zu einem Netzwerk oder einem bestimmten Host führt.

`dateiname1` ist der Name des Zielnetzwerks oder -Hosts. Dies kann ein symbolischer, in `networks` oder `hosts` stehender Name sein oder eine in Punktschreibweise dargestellte Internet-Adresse; siehe `inet(3N)`.

routed (1M)

routed (1M)

dateiname2 ist der Name oder die Adresse des Gateways, an den Nachrichten übermittelt werden sollen.

wert ist ein metrischer Wert, der die Anzahl der Knoten bis zum Ziel-Host oder -netzwerk angibt.

Das Schlüsselwort *passive* bzw. *active* gibt an, ob der Gateway als passiv oder aktiv anzusehen ist (siehe oben).

DATEIEN

/etc/gateways für entfernte Gateways
/etc/networks
/etc/hosts

SIEHE AUCH

ioctl(2), inet(7), udp(7).

HINWEIS

Die Routen-Tabellen des Kernels können kurzfristig von denen von *in.routed* abweichen, solange Prozesse, die vorhandene Routen benutzen, gerade beendet werden. Die einzige Gegenmaßnahme besteht darin, den Routenprozeß in den Systemkern zu verlagern.

routed sollte intelligente Schnittstellen wie IMP und Fehlerprotokolle, z.B. ICMP, abhören, um mehr Informationen zu sammeln.

BEZEICHNUNG

rshd, in.rshd – ferner Shell-Server

ÜBERSICHT

in.rshd *host.anschluß*

BESCHREIBUNG

in.rshd ist der Server für das Programm rsh(1). Der Server bietet Einrichtungen zur entfernten Kommando-Ausführung mit einer auf privilegierten Anschlußnummern basierenden Berechtigungsprüfung.

in.rshd wird von inetd(1M) immer dann aufgerufen, wenn ein Shell-Dienst angefordert wird, und führt folgende Schritte aus:

- 1) Der Server prüft den Ursprungsanschluß des Client. Liegt der Anschluß nicht im Bereich zwischen 0 und 1023, bricht der Server die Verbindung ab. Die Host-Adresse des Client (hexadezimal) und die Anschlußnummer (dezimal) werden als Argument an in.rshd übergeben.
- 2) Der Server liest Zeichen von der Socket-Schnittstelle bis zu einem Null-Byte (\0). Die sich daraus ergebende Zeichenkette wird als ASCII-Zahl zur Basis 10 interpretiert.
- 3) Ist die in Schritt 1 empfangene Zahl nicht Null, wird sie als Anschlußnummer eines Sekundärdatenstroms interpretiert, der für die Standardfehlerausgabe stderr verwendet werden soll. Daraufhin wird eine zweite Verbindung zu dem angegebenen Anschluß auf dem Rechner des Client hergestellt. Der Ursprungsanschluß dieser zweiten Verbindung liegt ebenfalls im Bereich zwischen 0 und 1023.
- 4) Der Server prüft die Ursprungsadresse des Client. Gehört die Adresse zu einem Host, für den in der Datenbank der Hostnamen kein Eintrag vorhanden ist (siehe hosts(4)), bricht der Server die Verbindung ab.
- 5) Von der Start-Socket-Schnittstelle wird ein mit Null abschließender Benutzername mit maximal 16 Zeichen übermittelt. Dieser wird als Benutzerkennung für das System von server verwendet.
- 6) Von der Start-Socket-Schnittstelle wird ein mit Null endender Benutzername mit maximal 16 Zeichen ermittelt. Dieser wird als Kennung des Benutzers auf dem System von client interpretiert.
- 7) Von der Start-Socket-Schnittstelle wird ein auf Null endendes Kommando ermittelt, das an eine Shell übergeben werden soll. Die Länge des Kommandos wird durch die Obergrenze für die Länge der Argumentliste auf dem System bestimmt.
- 8) Anschließend prüft in.rshd die Berechtigung des Benutzers mit den folgenden Schritten. Der Name des fernen Benutzers wird in der Paßwortdatei gesucht, und das Home-Verzeichnis des Benutzers wird mit chdir geprüft. Schlägt die Suche fehl, wird die Verbindung abgebrochen. Schlägt chdir fehl, wird die Prüfung für / (root) wiederholt. Handelt es sich bei dem Benutzer nicht um einen privilegierten Benutzer (Benutzernummer 0), wird in der Datei /etc/hosts.equiv nach einer Liste von Hosts gesucht, die als gleichwertig gelten. Befindet sich der Host-Name des Client in dieser Datei, gilt die Berechtigungsprüfung als erfolgreich. Schlägt die Suche fehl oder ist

der Benutzer ein privilegierter Benutzer, wird anhand der Datei `.rhosts` im Home-Verzeichnis des fernen Benutzers der Name des Rechners und die Identität des Benutzers auf dem Rechner des Client überprüft. Schlägt diese Suche fehl, wird die Verbindung abgebrochen.

- 9) Auf der zu `stderr` gehörenden Verbindung wird ein Null-Byte zurückgegeben, und die Kommandozeile wird an die normale Login-Shell des Benutzers übergeben. Die Shell übernimmt die von `in.rshd` aufgebauten Netzwerkverbindungen.

DATEIEN

`/etc/hosts.equiv`

SIEHE AUCH

`rsh(1)`

FEHLERMELDUNGEN

Die folgenden Fehlermeldungen werden auf der zu `stderr` gehörenden Verbindung zurückgegeben, danach werden alle Netzwerkverbindungen abgebrochen. Ein Fehler wird durch ein führendes Byte mit dem Wert 1 angezeigt (0 wird in Schritt 9 bei erfolgreicher Ausführung aller Schritte vor der Ausführung des Kommandos ausgegeben).

`locuser too long` (lokaler Benutzername zu lang)

Der Name des Benutzers auf dem Rechner des Client ist mehr als 16 Zeichen lang.

`remuser too long` (ferner Benutzername zu lang)

Der Name des Benutzers auf dem fernen Rechner ist mehr als 16 Zeichen lang.

`command too long` (Kommando zu lang)

Die übergebene Kommandozeile ist länger als die zulässige Argumentliste (wie im System konfiguriert).

`Hostname for your address unknown.` (Host-

Name für Ihre Adresse unbekannt)

Für den Rechner des Client existiert in der Datenbank der Hostnamen kein Eintrag.

`Login incorrect.` (Anmeldung nicht korrekt)

Für den Benutzernamen existiert kein Eintrag in der Paßwortdatei.

`Permission denied.` (Zugriffsberechtigung verweigert)

Die oben beschriebene Berechtigungsprüfung ist fehlgeschlagen.

`Can't make pipe.` (Pipe kann nicht erstellt werden)

Die für die Standardausgabe `stderr` benötigte Pipe wurde nicht erstellt.

`Try again.` (Versuch wiederholen)

Ein vom Server abgesetztes Kommando `fork` ist fehlgeschlagen.

rshd(1M)

rshd(1M)

HINWEIS

Die hier verwendete Berechtigungsprüfung geht von der Integrität aller Client-Rechner und des Übertragungsmediums aus. Dies ist zwar unsicher, in einer offenen Umgebung, aber nützlich.

Es muß eine Einrichtung zum Verschlüsseln des gesamten Datenaustauschs vorhanden sein.

BEZEICHNUNG

runacct – führt Tagesabrechnung durch

ÜBERSICHT

/usr/lib/acct/runacct [*mmtt* [*schrift*]]

BESCHREIBUNG

runacct ist die wichtigste Shell-Prozedur für die tägliche Abrechnung. In der Regel wird sie über cron aktiviert. runacct verarbeitet Dateien zur Anschluß-, Gebühren-, Platten- und Prozeßabrechnung. Die Prozedur erstellt auch Summendateien für prdaily oder zur Rechnungsstellung. runacct wird nur an Lizenznehmer des Quellcodes abgegeben.

runacct sorgt dafür, daß beim Auftreten von Fehlern keine aktiven Abrechnungs- oder Summendateien beschädigt werden. Die Prozedur trägt ihre Ergebnisse in Form von beschreibenden Diagnosemeldungen in active ein. Wird ein Fehler festgestellt, wird eine Meldung in die Datei /dev/console geschrieben, elektronische Post (siehe mail(1)) an root und adm gesendet und runacct beendet. runacct verwendet verschiedene Sperrdateien als Schutz gegen erneutes Aufrufen. Mit den Dateien lock und lock1 soll gleichzeitiges Aufrufen verhindert werden, und last-date soll dafür sorgen, daß die Prozedur nicht mehrmals an einem Tag aufgerufen wird.

runacct untergliedert die Verarbeitung in getrennte, wiederanlauffähige *schritte*, wobei in der Datei statefile der letzte abgeschlossene *schrift* verzeichnet wird. Dazu schreibt sie den Namen des *schriffs* in die Datei statefile. Anschließend sucht runacct in der Datei statefile nach dem letzten abgeschlossenen Schritt, um festzustellen, welcher Schritt danach folgen soll. Die *schritte* werden in der folgenden Reihenfolge ausgeführt:

SETUP	Aktive Abrechnungsdateien in Arbeitsdateien schreiben.
WTMPFIX	Integrität der Datei wtmp prüfen, ggf. Datumsänderungen korrigieren.
CONNECT	Sitzungssätze für Anschlüsse im tacct.h-Format erstellen.
PROCESS	Prozeßabrechnungssätze in tacct.h-Format umwandeln.
MERGE	Anschluß- und Prozeßabrechnungssätze mischen.
FEES	Ausgabe von chargefee in tacct.h-Format umwandeln und mit Anschluß- und Prozeßabrechnungssätzen mischen.
DISK	Plattenabrechnungssätze mit Anschluß-, Prozeß- und Gebührenabrechnungssätzen mischen.
MERGETACCT	Die Tagessummenabrechnungssätze in daytacct mit den Gesamtsummenabrechnungssätzen in /var/adm/acct/sum/tacct mischen.
CMS	Kommandozusammenfassungen erstellen.

USEREXIT Alle installationsabhängigen Abrechnungsprogramme können hier eingegeben werden.

CLEANUP Temporäre Dateien bereinigen und Prozedur beenden.

Um **runacct** nach einem Fehler neu zu starten muß zunächst in der Datei **active** nach Diagnosemeldungen gesucht werden, und eventuell beschädigte Datendateien wie **pacct** oder **wtmp** müssen berichtigt werden. Die Dateien **lock**, **lock1** und **lastdate** müssen entfernt werden, bevor **runacct** erneut aktiviert werden kann. Das Argument **mmitt** ist zum erneuten Aktivieren von **runacct** notwendig, denn es gibt das Datum in Monat und Tag an, für das **runacct** die Abrechnung wiederholen soll. Der Einstiegspunkt für die Verarbeitung richtet sich nach dem Inhalt der Datei **statefile** ; um sie außer Kraft zu setzen, gibt man den gewünschten *schrift* in der Kommandozeile an und bestimmt damit, wo die Verarbeitung beginnen soll.

BEISPIELE

Starten von **runacct**:

```
nohup runacct 2> /var/adm/acct/nite/fd2log &
```

Neustarten von **runacct**:

```
nohup runacct 0601 2>> /var/adm/acct/nite/fd2log &
```

Neustarten von **runacct** mit einem bestimmten *schrift*:

```
nohup runacct 0601 MERGE 2>> /var/adm/acct/nite/fd2log &
```

DATEIEN

```
/var/adm/wtmp
/var/adm/pacctincr
/usr/src/cmd/acct/tacct.h
/usr/src/cmd/acct/ctmp.h
/var/adm/acct/nite/active
/var/adm/acct/nite/daytacct
/var/adm/acct/nite/lock
/var/adm/acct/nite/lock1
/var/adm/acct/nite/lastdate
/var/adm/acct/nite/statefile
```

SIEHE AUCH

acct(1M), **acctcms(1M)**, **acctcon(1M)**, **acctmerc(1M)**, **acctprc(1M)**, **acctsh(1M)**, **cron(1M)**, **fwtmp(1M)**, **acct(4)**, **utmp(4)**
acctcom(1), **mail(1)** in den *Kommandos*.
acct(2) im *Referenzhandbuch für Programmierer*

HINWEIS

In der Regel ist es nicht sinnvoll, **runacct** mit dem *schrift* **SETUP** neu zu starten. **SETUP** sollte manuell ausgeführt und die Prozedur über **runacct mmtt WTMPFIX** neu gestartet werden.

Ist **runacct** bei dem *schrift* **PROCESS** fehlgeschlagen, muß die letzte **ptacct**-Datei entfernt werden, weil sie nicht vollständig ist.

BEZEICHNUNG

rwhod, in.rwhod – Systemstatus-Server

ÜBERSICHT

in.rwhod

BESCHREIBUNG

in.rwhod ist der Server, der die Datenbank verwaltet, die die Programme rwho(1) und ruptime(1) benutzen. Seine Arbeit basiert auf der Fähigkeit, in einem Netzwerk Nachrichten an alle Benutzer zu senden.

in.rwhod erzeugt und benötigt Statusdaten. Zum Erzeugen von Daten fragt er regelmäßig den Systemstatus ab und erstellt Statusmeldungen, die über ein Netzwerk an alle Benutzer gesendet werden. Benötigt er Daten, horcht er auf Statusmeldungen anderer in.rwhod-Server, bewertet sie und zeichnet sie in verschiedenen Dateien in dem Verzeichnis /var/spool/rwho auf.

Der Server rwho sendet und empfängt Nachrichten auf dem in der Dienstspezifikation rwho angegebenen Anschluß, siehe services(4). Die gesendeten und empfangenen Nachrichten haben die Form:

```

struct    outmp {
    char  out_line[8];      /* tty-Name */
    char  out_name[8];      /* Benutzernummer */
    long  out_time;         /* zeit ein */
};

struct    whod {
    char  wd_vers;
    char  wd_type;
    char  wd_fill[2];
    int   wd_sendtime;
    int   wd_recvttime;
    char  wd_hostname[32];
    int   wd_loadav[3];
    int   wd_boottime;
    struct whoent {
        struct    outmp we_utm;
        int        we_idle;
    } wd_we[1024 / sizeof (struct whoent)];
};

```

Alle Felder werden vor dem Senden in das Übertragungsformat des Netzwerks umgewandelt. Die durchschnittliche Auslastung entspricht dem von dem Programm w(1) berechneten Wert und stellt die durchschnittliche Auslastung der Intervalle von 5, 10 und 15 Minuten vor der Übertragung durch den Server dar. Der angegebene Host-Name ist das Ergebnis des Systemaufrufs gethostname(2). Das Feld (Array) am Ende der Nachricht enthält Angaben zu den am sendenden Rechner angemeldeten Benutzern. Diese Angaben bestehen aus dem Inhalt des utmp(4)-Eintrags für alle aktiven Terminalleitungen sowie einer Angabe des Zeitpunkts, zu dem auf dieser Leitung zum letztenmal ein Zeichen empfangen wurde.

Vom Server `rwho` empfangene Nachrichten werden gelöscht, wenn sie nicht von einem Anschluß eines Servers `rwho` ausgingen. Die Nachricht wird auch dann gelöscht, wenn der Host-Name, der in der Nachricht angegeben wurde, nicht druckbare ASCII-Zeichen enthält. Von `in.rwhod` empfangene gültige Nachrichten werden in Dateien namens `whod.hostname` im Verzeichnis `/var/spool/rwho` gestellt. Diese Dateien enthalten immer nur die neueste Nachricht im oben beschriebenen Format.

Statusnachrichten werden etwa alle 60 Sekunden erstellt. Alle 10 Minuten führt `in.rwhod` auf `/stand/unix nlist(3)` aus, um sicherzugehen, daß es sich bei dieser Datei tatsächlich um das aktuelle Systemabbild handelt.

DATEIEN

`/var/spool/rwho`

SIEHE AUCH

`rwho(1)`, `ruptime(1)`, `w(1)`, `gethostname(3)`, `nlist(3)`, `utmp(4)`.

HINWEIS

Dieser Dienst benötigt um so mehr Übertragungskapazität des Netzwerks, je höher die Anzahl der Hosts auf dem lokalen Netz wird. Bei großen Netzwerken werden daher die Kosten untragbar.

`in.rwhod` übermittelt Statusdaten zwischen verschiedenen Netzwerken. Wird er deaktiviert, nehmen viele Benutzer an, ein Rechner würde ausgeschaltet.

BEZEICHNUNG

sac – Netzwerk-Service-Steuerung

ÜBERSICHT

sac -t prüfintervall

BESCHREIBUNG

Die Netzwerk-Service-Steuerung (SAC) dient der Aufsicht über das Server-System. Sie wird aktiviert, wenn das Server-System in den Mehrbenutzer-Betrieb wechselt. SAC führt mehrere wichtige Funktionen aus, die nachstehend erläutert werden.

Anpassen der SAC-Umgebung. Wenn sac aufgerufen wird, sucht es zunächst das für jedes System angelegte Konfigurations-Skript /etc/saf/_sysconfig. sac interpretiert _sysconfig, um seine eigene Umgebung anzupassen. Die Änderungen der SAC-Umgebung anhand von _sysconfig werden von allen Sohnprozessen von SAC übernommen. Diese übernommene Umgebung kann von den Sohnprozessen geändert werden.

Aktivieren der Anschlußüberwachungsprogramme. Nach dem Interpretieren der Datei _sysconfig liest sac seine Verwaltungsdatei /etc/saf/_sactab. Diese gibt an, welche Anschlußüberwachungsprogramme aktiviert werden sollen. Für jedes dieser Programme erzeugt sac einen Sohnprozeß (fork(2)) und legt einen utmp-Eintrag an, in dem das Feld typ den Wert LOGIN_PROCESS enthält. Jeder Sohnprozeß interpretiert dann sein je Anschluß erstelltes Konfigurations-Skript /etc/saf/pmtag/_config für die Anschlußüberwachungsprogramme, sofern diese Datei existiert. Diese Änderungen der Umgebung beeinflussen das Anschlußüberwachungsprogramm und werden von allen seinen Sohnprozessen übernommen. Schließlich führt der Sohnprozeß über exec das Anschlußüberwachungsprogramm aus und verwendet dazu das Kommando aus dem Eintrag _sactab. (Siehe sacadm; dieses Kommando wird mit der Option -c gegeben, wenn das Anschlußüberwachungsprogramm in das System aufgenommen wird.)

Aufrufen der Anschlußüberwachungsprogramme zum Feststellen von Fehlern. Die Option -t definiert die Häufigkeit, mit der sac die Anschlußüberwachungsprogramme im System aufruft. Diese Zeitspanne kann als die Hälfte der maximalen Latenzzeit betrachtet werden, die benötigt wird, um festzustellen, daß ein Anschlußüberwachungsprogramm fehlgeschlagen und eine Abhilfemaßnahme erforderlich ist.

Verwaltungsfunktionen. Die Netzwerk-Service-Steuerung ist auch die administrative Steuerung für Anschlußüberwachungsprogramme. Seine Verwaltungsaufgaben werden nachstehend erläutert.

Wird die Netzwerk-Service-Steuerung abgefragt (sacadm mit der Option -l oder -L), gibt sie den Status der angegebenen Anschlußüberwachungsprogramme zurück, den sacadm auf der Standardausgabe ausgibt. Das Anschlußüberwachungsprogramm kann einen von sechs Stati haben:

ENABLED	Das Anschlußüberwachungsprogramm läuft gerade und akzeptiert Verbindungen. Siehe <code>sacadm(1M)</code> mit der Option <code>-e</code> .
DISABLED	Das Anschlußüberwachungsprogramm läuft gerade und akzeptiert keine Verbindungen. Siehe <code>sacadm</code> mit der Option <code>-d</code> sowie den Status <code>NOTRUNNING</code> .
STARTING	Das Anschlußüberwachungsprogramm wird gerade gestartet. <code>STARTING</code> ist ein Zwischenschritt auf dem Weg zum Status <code>ENABLED</code> oder <code>DISABLED</code> .
FAILED	Das Anschlußüberwachungsprogramm konnte nicht gestartet und in Betrieb gehalten werden.
STOPPING	Das Anschlußüberwachungsprogramm wurde manuell beendet, hat aber die Abschlußprozedur nicht abgeschlossen. <code>STOPPING</code> ist ein Zwischenschritt auf dem Weg zu <code>NOTRUNNING</code> .
NOTRUNNING	Das Anschlußüberwachungsprogramm läuft zur Zeit nicht. (Siehe <code>sacadm</code> mit der Option <code>-k</code> .) Dies ist der normale "inaktive" Status. Wird ein Anschlußüberwachungsprogramm abgebrochen, werden alle von ihm überwachten Anschlüsse unerreichbar. Ein externer Benutzer kann dann nicht unterscheiden, ob ein Anschluß nicht überwacht wird oder ob das System ausgeschaltet ist. Wurde das Anschlußüberwachungsprogramm nicht abgebrochen, sondern befindet es sich im Status <code>DISABLED</code> , ist es unter Umständen (je nach dem verwendeten Anschlußüberwachungsprogramm) möglich, auf dem unerreichbaren Anschluß eine Nachricht für den aufrufenden Benutzer zu hinterlassen, daß dieser Anschluß deaktiviert ist. Dies ist der Vorteil, wenn neben dem Status <code>NOTRUNNING</code> auch der Status <code>DISABLED</code> zur Verfügung steht.

Wird ein Anschlußüberwachungsprogramm beendet, löscht die Netzwerk-Service-Steuerung (SAC) den `utmp`-Eintrag für dieses Anschlußüberwachungsprogramm.

Die SAC nimmt alle Anforderungen zum Aktivieren, Deaktivieren, Starten oder Beenden von Anschlußüberwachungsprogrammen entgegen und führt die entsprechenden Schritte durch.

Die SAC sorgt dafür, daß beendete Anschlußüberwachungsprogramme wieder gestartet werden. Ob die SAC ein bestimmtes Anschlußüberwachungsprogramm neu startet, hängt von zwei Faktoren ab:

- dem Neustartzähler, der für das Anschlußüberwachungsprogramm angegeben wurde, als das Programm von `sacadm` hinzugefügt wurde; diese Informationen stehen in der Datei `/etc/saf/pmtag/_sactab`
- davon, wie oft das Anschlußüberwachungsprogramm bereits neu gestartet wurde

sac(1M)

(Wichtige Dienstprogramme)

sac(1M)

SIEHE AUCH

sacadm(1M), pmadm(1M).

DATEIEN

/etc/saf/_sactab
/etc/saf/_sysconfig
/var/adm/utmp
/var/saf/_log

BEZEICHNUNG

sacadm – Administration des Service Access Controller

ÜBERSICHT

sacadm -a -p *pmtag* -t *typ* -c *kom* -v *ver* [-f *dx*] [-n *zähler*]
 [-y *kommentar*] [-z *skript*]

sacadm -r -p *pmtag*

sacadm -s -p *pmtag*

sacadm -k -p *pmtag*

sacadm -e -p *pmtag*

sacadm -d -p *pmtag*

sacadm -l [-p *pmtag* | -t *typ*]

sacadm -L [-p *pmtag* | -t *typ*]

sacadm -g -p *pmtag* [-z *skript*]

sacadm -G [-z *skript*]

sacadm -x [-p *pmtag*]

BESCHREIBUNG

sacadm ist das Systemverwalterkommando für die höhere Ebene der Hierarchie der Service Access Facility. Es dient zur Verwaltung der Port Monitore. sacadm führt folgende Funktionen aus:

- Hinzufügen oder Entfernen eines Port Monitors
- Starten oder Beenden eines Port Monitors
- Aktivieren oder Deaktivieren eines Port Monitors
- Installieren oder Ersetzen eines pro System angelegten Konfigurations-Skripts
- Installieren oder Ersetzen eines pro Port Monitor angelegten Konfigurations-Skripts
- Ausgeben von Informationen zu einem Port Monitor

Anforderungen zum Status von Port Monitoren (die Optionen -l und -L) und Anforderungen zum Ausgeben von pro Port Monitor und pro System angelegten Konfigurations-Skripts (Optionen -g und -G ohne Option -z) können von allen Benutzern im System ausgeführt werden. Andere sacadm-Kommandos dürfen nur von privilegierten Benutzern ausgeführt werden.

Die Optionen haben folgende Bedeutung:

- a Hinzufügen eines Port Monitors. Dabei legt sacadm die zugehörige Verzeichnisstruktur in /etc/saf und /var/saf an und stellt einen Eintrag für den neuen Port Monitor in /etc/saf/_sactab. Die Datei _sactab ist auf einem neu gelieferten System bereits angelegt. Ursprünglich ist sie leer bis auf eine Zeile, die die Versionsnummer des Service Access Controller enthält.

Der neue Port Monitor wird nur gestartet, wenn die Kommandozeile, mit der er aufgenommen wird, nicht die Option -f mit dem Argument x

enthält. Weil die Optionen und Argumente, die der Option `-a` folgen, so komplex sind, ist es komfortabler, ein Kommando-Skript oder das Menüsystem zu benutzen, um Port Monitore hinzuzufügen. Beim Menüsystem gibt man `sysadm ports` ein und wählt dann die Option `port_monitors`.

- `-c kom` Führt die Kommandozeichenkette *kom* zum Starten eines Port Monitors aus. Die Option `-c` darf nur zusammen mit `-a` benutzt werden. Die Option `-a` erfordert die Option `-c`.
- `-d` Deaktiviert den Port Monitor *pmtag*.
- `-e` Aktiviert den Port Monitor *pmtag*.
- `-f dx` Die Option `-f` gibt eine oder beide der folgenden beiden Marken an, die dann in das Markenfeld des `_sactab`-Eintrags für den neuen Port Monitor aufgenommen werden. Ist die Option `-f` nicht in der Kommandozeile angegeben, werden keine Marken definiert, und die Standardbedingungen bleiben gültig. Standardmäßig wird ein Port Monitor gestartet. Die Option `-f` ohne Argumente ist unzulässig.
 - `d` Neuen Port Monitor nicht aktivieren.
 - `x` Neuen Port Monitor nicht starten.
- `-g` Die Option `-g` dient zum Anfordern der Ausgabe oder zum Installieren oder Ersetzen des pro Port Monitor angelegten Konfigurations-Skripts `/etc/saf/pmtag/_config`. `-g` erfordert die Option `-p`. Wird die Option `-g` nur mit der Option `-p` angegeben, wird das pro Port Monitor angelegte Konfigurations-Skript für den Port Monitor *pmtag* ausgegeben. Bei Option `-g` kombiniert mit den Optionen `-p` und `-z` wird die Datei *skript* als pro Port Monitor angelegtes Konfigurations-Skript für den Port Monitor *pmtag* installiert. Andere Kombinationen von Optionen mit `-g` sind unzulässig.
- `-G` Die Option `-G` dient zum Anfordern der Ausgabe oder zum Installieren oder Ersetzen des pro System angelegten Konfigurations-Skripts `/etc/saf/_sysconfig`. Steht die Option `-G` allein, wird das pro System angelegte Konfigurations-Skript ausgegeben. Bei der Kombination der Optionen `-G` und `-z` wird die Datei *skript* als pro System angelegtes Konfigurations-Skript installiert. Sonstige Kombinationen von Optionen mit `-G` sind unzulässig.
- `-k` Beenden des Port Monitors *pmtag*.
- `-l` Mit der Option `-l` werden Daten zum Port Monitor angefordert. Die Option `-l` allein erstellt eine Liste aller Port Monitore im System. Wird die Option `-l` mit der Option `-p` kombiniert, wird nur der Port Monitor *pmtag* aufgeführt. Wird die Option `-l` mit der Option `-t` kombiniert, werden alle Port Monitore des Typs *typ* aufgeführt. Alle sonstigen Kombinationen mit der Option `-l` sind unzulässig.

- L Die Option -L ist identisch mit -l, allerdings wird hier eine komprimierte Ausgabe erstellt.
- n *zähler* Setzt den Neustartzähler auf *zähler*. Fehlt ein solcher Zähler, gilt der Wert 0. Ein Zähler 0 gibt an, daß der Port Monitor nicht neu gestartet wird, wenn das Hinzufügen des Port Monitors fehlschlägt.
- p *pmtag* Gibt die zu einem Port Monitor gehörende Marke an.
- r Entfernt den Port Monitor *pmtag*. *sacadm* entfernt den Eintrag für den Port Monitor aus */etc/saf/_sactab*. Ist der entfernte Port Monitor nicht in Betrieb, werden keine weiteren Schritte durchgeführt. Arbeitet der Port Monitor gerade, sendet ihm der Service Access Controller (SAC) das Signal SIGTERM, damit er beendet wird. Bitte beachten Sie, daß die Verzeichnisstruktur des Port Monitors dabei intakt bleibt.
- s Startet einen Port Monitor. Der Service Access Controller (SAC) startet den Port Monitor *pmtag*.
- t *typ* Gibt den Typ des Port Monitors an.
- v *ver* Gibt die Versionsnummer des Port Monitors an. Diese kann folgende Form haben:

`-v `pmspec` -V`

Dabei ist *pmspec* das spezielle Systemverwalterkommando für den Port Monitor *pmtag*. Dieses spezielle Kommando lautet *ttysadm* für *ttymon* und *nlssadmin* für *listen*. Der Versionsstempel des Port Monitors ist dem Kommando bekannt und wird zurückgegeben, wenn *pmspec* mit der Option -V aufgerufen wird.
- x Die Option -x allein weist den Service Access Controller an, seine Datenbankdatei (*_sactab*) zu lesen. Bei der Option -x in Kombination mit der Option -p wird der Port Monitor *pmtag* angewiesen, seine Verwaltungsdatei zu lesen.
- y *kommentar* Nimmt *kommentar* in den *_sactab*-Eintrag für den Port Monitor *pmtag* auf.
- z *skript* Wird zusammen mit den Optionen -g und -G verwendet, um den Namen einer Datei anzugeben, die ein Konfigurations-Skript enthält. Bei Option -g ist *skript* ein pro Port Monitor angelegtes Konfigurations-Skript; bei Option -G ist es ein pro System angelegtes Konfigurations-Skript. Das Ändern eines Konfigurations-Skripts erfolgt in drei Schritten. Zunächst wird eine Kopie des vorhandenen Skripts erstellt (mit -g oder -G). Diese Kopie wird editiert. Schließlich wird das vorhandene Skript mit dem neu bearbeiteten überschrieben (-g oder -G mit -z).

EXIT-STATUS

Wird `sacadm` ordnungsgemäß ausgeführt, lautet der Exit-Status 0. Schlägt `sacadm` aus irgendeinem Grunde fehl, hat der Exit-Status einen anderen Wert. Bei Optionen, die Daten anfordern, werden diese Daten in die Standardausgabe geschrieben. Im komprimierten Format der Option `(-L)` werden Daten zum Port Monitor als eine Folge durch Doppelpunkt getrennter Felder ausgegeben; leere Felder werden durch zwei aufeinanderfolgende Doppelpunkte dargestellt. Im Standardformat `(-l)` werden die einzelnen Spalten mit einem Kopf versehen, und die Angaben zum Port Monitor werden unter den entsprechenden Überschriften eingeordnet. In diesem Format wird ein leeres Feld durch einen Bindestrich kenntlich gemacht. Das Kommentarzeichen ist `#`.

BEISPIELE

Mit der folgenden Kommandozeile wird ein Port Monitor hinzugefügt. Die Marke des Port Monitors lautet `npack`; der Typ ist `listen`; bei Bedarf wird es dreimal neu gestartet, bevor es fehlschlägt; das Systemverwalterkommando lautet `nlsadmin`; das zu lesende Konfigurations-Skript steht in der Datei *skript*:

```
sacadm -a -p npack -t listen -c /usr/lib/saf/listen npack \
-v `nlsadmin -V` -n 3 -z skript
```

Entfernen eines Port Monitors mit der Marke `pmtag`:

```
sacadm -r -p pmtag
```

Starten des Port Monitors mit der Marke `pmtag`:

```
sacadm -s -p pmtag
```

Beenden des Port Monitors mit der Marke `pmtag`:

```
sacadm -k -p pmtag
```

Aktivieren des Port Monitors mit der Marke `pmtag`:

```
sacadm -e -p pmtag
```

Deaktivieren des Port Monitors mit der Marke `pmtag`:

```
sacadm -d -p pmtag
```

Erstellt eine Liste mit Statusangaben für alle Port Monitore:

```
sacadm -l
```

Erstellt eine Liste mit Statusangaben für den Port Monitor mit der Marke `pmtag`:

```
sacadm -l -p pmtag
```

Erstellt dieselbe Liste in komprimierter Form:

```
sacadm -L -p pmtag
```

Erstellt eine Liste mit Statusangaben für alle Port Monitore des Typs `listen`:

```
sacadm -l -t listen
```

Ersetzt das Konfigurations-Skript, das zum Port Monitor `pmtag` gehört, durch den Inhalt der Datei `file.config`:

```
sacadm -g -p pmtag -z file.config
```

SIEHE AUCH

`doconfig(3N)`, `pmadm(1M)`, `sac(1M)`.

DATEIEN

```
/etc/saf/_sactab  
/etc/saf/_sysconfig  
/etc/saf/pmtag/_config
```

BEZEICHNUNG

sar: sa1, sa2, sadc – erzeugt Berichte über die Systemaktivität

ÜBERSICHT

```
/usr/lib/sa/sadc [t n] [adatei]
```

```
/usr/lib/sa/sa1 [t n]
```

```
/usr/lib/sa/sa2 [-ubdycwaqvmprkxDSAC] [-s zeit] [-e zeit] [-i sek]
```

BESCHREIBUNG

Daten über die Systemaktivität können vom Benutzer (siehe sar(1)) oder automatisch über Routinen abgefragt werden, die im folgenden beschrieben werden. Das Betriebssystem enthält eine Reihe von Zählern, die erhöht werden, sobald das System eine bestimmte Aktion ausführt. Hierzu gehören Zähler für die Belegung der Zentraleinheit (CPU), die Pufferbelegung, die E/A-Aktivität der Festplatte und des Magnetbands, die Aktivität der TTY-Geräte, Wählverbindungen und Systemaufrufe, Dateizugriffe, die Warteschlangenaktivität, Interprozeßkommunikation, Paging und Remote File Sharing.

sadc und zwei Shell-Prozeduren, sa1 und sa2, werden dazu verwendet, diese Daten abzufragen, zu speichern und zu verarbeiten.

sadc, das Datenerfassungsprogramm, fragt *n* mal alle *t* Sekunden die Systemdaten ab und schreibt sie im Binärformat in die Datei *adatei* oder in die Standardausgabe. Das Intervall *t* sollte größer als 5 Sekunden sein, da sonst die Aktivität von sadc selbst die Daten beeinflussen kann. Werden *t* und *n* ausgelassen, wird ein besonderer Datensatz geschrieben. Diese Möglichkeit wird beim Laden des Systems im Mehrbenutzermodus verwendet, um den Zeitpunkt zu markieren, zu dem die Zähler auf Null zurückgesetzt werden. Mit dem folgenden Kommando wird z.B. bewirkt, daß die Datei /etc/init.d/perf die Tagesdaten mit dieser Markierung versieht:

```
su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d` "
```

Mit dem Shell-Skript sa1, einer Variante von sadc, werden Daten in der Binärdatei /var/adm/sa/sa tt zusammengefaßt und gespeichert, wobei tt den aktuellen Tag angibt. Die Argumente *t* und *n* bewirken, daß *n* mal alle *t* Sekunden Datensätze geschrieben werden. Ohne diese Argumente wird nur ein Datensatz angelegt. Mit den folgenden Einträgen in /var/spool/cron/crontabs/sys wird bewirkt, daß Datensätze während der Arbeitszeit alle 20 Minuten und sonst stündlich angelegt werden:

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
```

Weitere Informationen sind unter crontab(1) zu finden.

Das Shell-Skript `sa2`, eine Variante von `sar`, schreibt täglich einen Bericht in die Datei `/var/adm/sa/sar $\mathit{tt}$` . Die Optionen werden unter `sar(1)` erläutert. Mit dem folgenden Eintrag in `/var/spool/cron/crontabs/sys` wird bewirkt, daß während der Arbeitszeit stündlich ein Bericht über wichtige Aktivitäten angelegt wird:

```
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

Die binäre Datei mit den Tagesdaten weist folgende Struktur auf:

```
struct sa {
    struct sysinfo si;          /* see /usr/include/sys/sysinfo.h */
    struct minfo mi;           /* defined in sys/sysinfo.h */
    struct vminfo vmi;         /* defined in /usr/include/sys/sysinfo.h */
    rf_srv_info_t rf_srv;      /* defined in /usr/include/sys/fs/rf_acct.h */
    fsinfo_t rfs_in;
    fsinfo_t rfs_out;          /* defined in /usr/include/sys/sysinfo.h */
    rfc_info_t rfc;            /* defined in /usr/include/sys/fs/rf_acct.h */
    struct kmeminfo km;        /* defined in /usr/include/sys/sysinfo.h */
    struct bpbinfo bi[4];      /* Co-processor info defined in sys/sysinfo.h */
    int hpb_utilize;           /* Co-processor utilize flag */
    int minserve, maxserve;    /* RFS server low and high water marks */
    int szinode;               /* current size of inode table */
    int szfile;                /* current size of file table */
    int szproc;                /* current size of proc table */
    int szlckf;                /* current size of file record header table */
    int szlckr;                /* current size of file record lock table */
    int mszinode;              /* size of inode table */
    int mszfile;               /* size of file table */
    int mszproc;               /* size of proc table */
    int mszlckf;               /* maximum size of file record header table */
    int mszlckr;               /* maximum size of file record lock table */
    long inodeovf;             /* cumulative overflows of inode table */
    long fileovf;              /* cumulative overflows of file table */
    long procovf;              /* cumulative overflows of proc table */
    time_t ts;                 /* time stamp, seconds */
    int apstate;               /* Co-processor flag */
    long devio[NDEVS][5];      /* device unit information */
#define IO_OPS 0               /* cumulative I/O requests */
#define IO_BCNT 1              /* cumulative blocks transferred */
#define IO_ACT 2               /* cumulative drive busy time in ticks */
#define IO_RESP 3              /* cumulative I/O resp time in ticks */
#define IO_ID 4
};
```

DATEIEN

<code>/var/adm/sa/satt</code>	Datei mit Tagesdaten
<code>/var/adm/sa/sartt</code>	Datei mit Tagesbericht
<code>/tmp/sa.adrfl</code>	Adreßdatei

SIEHE AUCH

`crontab(1)`, `sag(1G)`, `sar(1)`, `timex(1)`.

BEZEICHNUNG

savecore – Speicherabzug und Zusatzinformationen organisieren

ÜBERSICHT

```
/usr/sbin/savecore
/usr/sbin/savecore -c [ default_dump_dev ]
/usr/sbin/savecore -q [ default_dump_dev ]
/usr/sbin/savecore -d dump_gerät
/usr/sbin/savecore -g dump_gerät[ n ][ sysname ] ext_dev
/usr/sbin/savecore -s[targetdir [ unixsource ] ]
```

BESCHREIBUNG

Ohne die Angabe von Argumenten listet das Kommando `savecore` die Verwaltungsstrukturen der Plattenpartition auf, die für Hauptspeicherabzüge eingerichtet wurde und im NVRAM (non-volatile RAM) steht. Ist dort als "Dump Location" ein Exabyte- oder Streamerlaufwerk angegeben, liefert das Kommando eine Fehlermeldung und bricht ab.

`-c [ default_dump_dev ]`

Löscht das Flag `DUMP_EXIST` im NVRAM, falls es gesetzt ist. Dieses Flag zeigt an, daß beim letzten Hochfahren ein Dump geschrieben wurde. Wurde er auf die Slice geschrieben, die als Standard-Dumpgerät im NVRAM eingestellt ist oder als Parameter mitgegeben wurde, dann wird zusätzlich die Kennung "0xdeadbabe" nach dem Offset des Dumps auf dieser Slice gelöscht.

`-q [ default_dump_dev ]`

Gibt eine Meldung aus, wohin der letzte Dump geschrieben wurde, und liefert einen entsprechenden Return-Wert. `savecore` meldet z.B. ob auf ein spezielles Dumpgerät (special dump device) oder auf das Standard-Dumpgerät (default dump device) geschrieben wurde. Ist `default_dump_dev` angegeben, dann wird anstelle des im NVRAM angegebenen Standard-Dumpgeräts nach einem Dump in `default_dump_dev` gesucht.

"savecore: no dump in special dump device when booted this time or last savecore -s was successful" mit Return-Wert 2

"savecore: dump went to special dump device" mit Return-Wert 2

"savecore: dump of size <n> MB on <device> at offset <n> MB available" mit Return-Wert 0

"savecore: no dump available on default device <device>" mit Return-Wert 0

Im Fehlerfall beträgt der Return-Wert 1.

`-d dump_gerät`

Initialisiert `dump_gerät`. Erst danach kann beim Hochfahren ein Hauptspeicherabzug auf dieses Gerät geschrieben werden. Bezeichnet `dump_gerät` eine Plattenslice, dann wird berechnet, wieviele Hauptspeicherabzüge auf dieser Slice Platz haben. Die Offsets der einzelnen Speicherabzüge in der Slice werden mit anderen Daten in einer Datenstruktur gespeichert und an den

Anfang der Slice geschrieben. Bezeichnet *dump_gesät* eine Kassette, dann wird diese vor und nach dem Schreiben automatisch an den Beginn gespult.

`-g dump_gesät[ n][ sysname] ext_dev`

Sichert auf das angegebene Medium *ext_dev* das Betriebssystem *sysname*, `/usr/sbin/crash` und den Hauptspeicherabzug, den `boot` beim Hochfahren des Systems nach Plattenpartition *dump_gesät* geschrieben hat. Fehlt die Angabe *sysname*, dann wird stattdessen `/unix` genommen. Ist *n* aufgeführt, dann wird der *n*-te Speicherabzug gesichert; sonst der jüngste. Damit kann `savecore` mehrmals auf einen Speicherabzug angewendet werden. Eine Kassette wird weder vor dem Kopieren der Dateien noch danach zurückgespult. Der Anwender muß selbst positionieren. Ist *ext_dev* der Name eines Dateiverzeichnisses, dann wird der Hauptspeicherabzug, ohne den Umweg über eine Kassette, direkt in eine Datei dieses Verzeichnisses geschrieben.

`-s[targetdir [ unixsource ]]`

Wird beim Hochfahren des Systems von `/etc/rc2.d/S03savecore` aufgerufen. Das Kommando sollte nicht vom Benutzer aufgerufen werden. Wurde ein Dump auf die Standard-Slice für Dumps geschrieben, dann wird der Hauptspeicherabzug nach *targetdir/vmcore.n* (falls angegeben) oder `/var/crash` kopiert; `/unix` bzw. *unixsource* (falls angegeben) wird nach *targetdir/vmunix.n* kopiert. Und `/usr/sbin/crash` wird nach *targetdir/crash.n* geschrieben. Es wird nur dann kopiert, wenn genügend Platz für den Dump, `/unix` bzw. *unixsource* und `/usr/sbin/crash` im Dateisystem vorhanden ist. Wenn die Datei `minfree` existiert, dann wird nur kopiert, falls der Platz auf dem Dateisystem abzüglich der Zahl in `minfree` (in Kbyte) ausreicht. Wurde ein Speicherabzug auf eine Magnetbandkassette geschrieben, dann werden nach diesem die Dateien `/unix` und `/usr/sbin/crash` darauf kopiert. Liegt der Hauptspeicherabzug auf einem speziellen Dump-Gerät, dann wird nichts kopiert.

In jedem Fall wird nach erfolgreichem Ablauf des Kommandos das Flag `DUMP_EXIST` im NVRAM gelöscht. Wurde der Dump auf die Slice geschrieben, die standardmäßig dafür eingerichtet ist, dann wird zusätzlich die Kennung "0xdeadbabe" nach dem Offset des Dumps auf dieser Slice gelöscht.

Wird nicht auf das Swap-Gerät kopiert, dann läuft `savecore` im Hintergrund ab und alle Meldungen werden in die Datei `/var/adm/savecorelog` geschrieben. Wird auf Kassette gedumpt, dann muß der Systemverwalter nach dem Hochfahren des Systems die Meldungen in dieser Datei beachten, bevor er die Kassette entnimmt. `savecore` könnte sie nämlich noch benutzen, um den Dump darauf zu schreiben. Wenn auf das Swap-Gerät kopiert wird, dann erscheinen alle

Meldungen auf der Konsole.

Tritt beim Aufruf von `savecore -s` ein Fehler auf (erkenntlich an einer Meldung auf der Konsole, in `/var/adm/savecorelog` oder mit `savecore -q`), so muß nach der Fehlerbehebung `savecore -s` nochmals direkt aufgerufen werden.

RETURN-WERTE

- 0 Ordnungsgemäßes Verhalten
- 0 Aufruf von `/usr/sbin/savecore -q` mit Dump auf der Standard-Dumpslice
- 1 Fehler
- 2 Aufruf von `/usr/sbin/savecore -q` mit Dump auf einer speziellen Slice

DATEIEN

`/usr/sbin/crash`
targetdir/`minfree`
`/var/crash/minfree`
`/var/adm/savecorelog`
`/etc/rc2.d/S03savecore`

SIEHE AUCH

`boot(1M)`, `nvrnm(1M)`.

HINWEIS

Vor einer Neuinstallation von `/unix` und `/usr/sbin/crash` sollten interessante Hauptspeicherabzüge von einem speziellen Dump-Gerät mittels `savecore -g` gesichert werden.

savecore(1M)

(WX200)

savecore(1M)

BEZEICHNUNG

savecore – Speicherabzug organisieren

ÜBERSICHT

/usr/sbin/savecore

BESCHREIBUNG

Wenn das Gerät /dev/dump eingerichtet ist, dann informiert sich savecore, ob dort ein Dump existiert. Ansonsten sucht savecore unter /dev/swap nach einem Dump. Falls das Kommando einen findet, dann versucht es den Dump nach /var/crash zu schreiben. Ist zu wenig Platz vorhanden, so wird versucht, eventuell vorhandene ältere Dumps auf Band oder Diskette zu schreiben. Gelingt dies, dann wird der Dump auf die Platte gesichert und der Bereich auf /dev/dump bzw. /dev/swap, der den Dump enthalten hat, wird gelöscht. Zusätzlich zum Dump werden das aktuelle crash-Programm und der aktuelle SINIX-Kern nach /var/crash gesichert.

RETURN-WERTE

- 0 Ordnungsgemäßes Verhalten
- 1 Kein Dump vorhanden
- 2 Kein Speicherplatz
- 3 Kein Speicherplatz

DATEIEN

/usr/sbin/crash

SIEHE AUCH

boot(1M)

BEZEICHNUNG

scsign – Gerädateien und Node Files für SCSI-Geräte am NCR Host Adapter erzeugen

ÜBERSICHT

scsign *[option]*

BESCHREIBUNG

Die Datei `scsign` wird als Level 0 und Level 2 `rc`-Datei ausgeführt, um die am Host Adapter (ADP32 Controller) angeschlossenen und eingeschalteten SCSI-Geräte einzurichten.

Die Datei `/etc/rc0.d/K70scsi` wird bei Beendigung einer Sitzung als Level 0 `rc`-Datei gestartet. Wenn ein neuer Kern gebunden wurde, wird eine Datei `/etc/.scsi_node` angelegt, die während der nächsten Boot Phase anzeigt, daß für die SCSI-Geräte Gerädateien eingerichtet werden sollen.

Die Datei `/etc/rc2.d/S12scsi` wird beim Hochfahren des Systems als Level 2 `rc`-Datei gestartet. Wenn die Datei `/etc/.scsi_node` vorhanden ist, werden alle möglichen Host Adapter (`adp32`) und SCSI-Target Adressen daraufhin überprüft, ob ein SCSI-Gerät (`acc`, `cdr`(nur MX300), `lad`, `shd`) angeschlossen und eingeschaltet ist. Falls für diese Geräte noch kein Eintrag im dazugehörigen Node File existiert, werden die benötigten Gerädateien eingerichtet und das Node File erweitert.

Falls die Datei `/etc/init.d/scsign` vom Systemverwalter mit dem SCSI-Treibernamen als Option (`acc`, `cdr`(nur MX300), `exa`, `lad`, `shd`) gestartet wird, werden alle möglichen Host Adapter und SCSI-Target Adressen durchsucht, ob ein SCSI-Gerät der ausgewählten Geräteklasse angeschlossen und eingeschaltet ist. Wenn für diese Geräte noch kein Eintrag im dazugehörigen Node File existiert, werden die benötigten Gerädateien eingerichtet und das Node File erweitert. Falls die Gerädateien gelöscht wurden, aber die dazugehörigen Einträge noch im Node File stehen, werden keine neuen Gerädateien angelegt. Bei einem `idmknod`-Aufruf werden die Gerädateien wieder angelegt.

Mögliche Optionen sind:

<code>start</code>	Es werden für die von <code>acc</code> , <code>cdr</code> , <code>lad</code> und <code>shd</code> unterstützten angeschlossenen Geräte Gerädateien eingerichtet, nachdem ein neuer Kern gebaut wurde.
<code>stop</code>	Es wird die Datei <code>/etc/.scsi_node</code> angelegt, nachdem ein Kern gebunden wurde.
<code>acc</code>	Es werden für die von <code>acc</code> unterstützten angeschlossenen Geräte Gerädateien eingerichtet.
<code>exa</code>	Es werden für die von <code>exa</code> unterstützten angeschlossenen Geräte Gerädateien eingerichtet. In den Geräten muß sich ein Band befinden.
<code>lad</code>	Es werden für die von <code>lad</code> unterstützten angeschlossenen Geräte Gerädateien eingerichtet.
<code>shd</code>	Es werden für die von <code>shd</code> unterstützten angeschlossenen Geräte Gerädateien eingerichtet.

scsigen (1M)

(MX300, MX500)

scsigen (1M)

`cdr` (Nur MX300) Es werden für die von `cdr` unterstützten angeschlossenen Geräte Gerätedateien eingerichtet.

DATEIEN

`/etc/init.d/scsigen,`
`/etc/rc0.d/K70scsi,`
`/etc/rc2.d/S12scsi,`
`/etc/.scsi_node.`

BEZEICHNUNG

scsigen – Gerätedateien und Node Files für SCSI-Geräte am AHA1542B Host Adapter erzeugen

ÜBERSICHT

scsigen *[option]*

BESCHREIBUNG

Die Datei `scsigen` wird als Level 0 und Level 2 `rc`-Datei ausgeführt, um die am AHA1542B Host Adapter angeschlossenen und eingeschalteten SCSI-Geräte einzurichten.

Die Datei `/etc/rc0.d/K70scsi` wird bei Beendigung einer Sitzung als Level 0 `rc`-Datei gestartet. Wenn ein neuer Kern gebunden wurde, wird eine Datei `/etc/.scsi_node` angelegt, die während der nächsten Boot Phase anzeigt, daß für die SCSI-Geräte Gerätedateien eingerichtet werden sollen.

Die Datei `/etc/rc2.d/S12scsi` wird beim Hochfahren des Systems als Level 2 `rc`-Datei gestartet. Wenn die Datei `/etc/.scsi_node` vorhanden ist, werden alle möglichen Host Adapter (`aha`) und SCSI-Target Adressen daraufhin überprüft, ob ein SCSI-Gerät (`acc`, `cdr`, `lad`) angeschlossen und eingeschaltet ist. Falls für diese Geräte noch kein Eintrag im dazugehörigen Node File existiert, werden die benötigten Gerätedateien eingerichtet und das Node File erweitert.

Falls die Datei `/etc/init.d/scsigen` vom Systemverwalter mit dem SCSI-Treibernamen als Option (`acc`, `cdr`, `exa`, `lad`) gestartet wird, werden alle möglichen Host Adapter und SCSI Target Adressen durchsucht, ob ein SCSI-Gerät der ausgewählten Geräteklasse angeschlossen und eingeschaltet ist. Wenn für diese Geräte noch kein Eintrag im dazugehörigen Node File existiert, werden die benötigten Gerätedateien eingerichtet und das Node File erweitert. Falls die Gerätedateien gelöscht wurden, aber die dazugehörigen Einträge noch im Node File stehen, werden keine neuen Gerätedateien angelegt. Bei einem `idmknod`-Aufruf werden die Gerätedateien wieder angelegt.

Mögliche Optionen sind:

<code>start</code>	Es werden für die von <code>acc</code> , <code>cdr</code> und <code>lad</code> unterstützten angeschlossenen Geräte Gerätedateien eingerichtet, nachdem ein neuer Kern gebaut wurde.
<code>stop</code>	Es wird die Datei <code>/etc/.scsi_node</code> angelegt, nachdem ein Kern gebunden wurde.
<code>acc</code>	Es werden für die von <code>acc</code> unterstützten angeschlossenen Geräte Gerätedateien eingerichtet.
<code>exa</code>	Es werden für die von <code>exa</code> unterstützten angeschlossenen Geräte Gerätedateien eingerichtet. In den Geräten muß sich ein Band befinden.
<code>lad</code>	Es werden für die von <code>lad</code> unterstützten angeschlossenen Geräte Gerätedateien eingerichtet.
<code>cdr</code>	Es werden für die von <code>cdr</code> unterstützten angeschlossenen Geräte Gerätedateien eingerichtet.

scsigen (1M)

(WX200)

scsigen (1M)

DATEIEN

/etc/init.d/scsigen,
/etc/rc0.d/K70scsi,
/etc/rc2.d/S12scsi,
/etc/.scsi_node.

setclock (1M)

setclock (1M)

BEZEICHNUNG

setclock – setzt die Systemzeit nach der Hardware-Uhr

ÜBERSICHT

/sbin/setclock

BESCHREIBUNG

Mit setclock wird die interne Systemzeit auf die aktuelle Tageszeit der Hardware-Uhr gesetzt. Dieses Kommando kann nur von einem privilegierten Benutzer aufgerufen werden. Normalerweise wird es zum Zeitpunkt der Systeminitialisierung beim Hochfahren über einen Eintrag in /etc/inittab ausgeführt. Es ist zu beachten, daß setclock nur das Datum der Hardware-Uhr prüft. Ist es gesetzt, läuft setclock ohne Meldung ab, andernfalls fordert es den Benutzer auf, die Hardware-Uhr mittels date richtig einzustellen.

SIEHE AUCH

date(1) in den *Kommandos*.

BEZEICHNUNG

setmnt – legt die Einhängetabelle an

ÜBERSICHT

/sbin/setmnt

BESCHREIBUNG

setmnt erstellt die Tabelle `/etc/mnttab`, die für die Kommandos `mount` und `umount` benötigt wird. setmnt liest die Standardeingabe und erstellt für jede Zeile einen Eintrag in `mnttab`. Eingabezeilen weisen folgendes Format auf:

dateisys knoten

dateisys gibt den Namen der "Gerätedatei" des Dateisystems an (wie z.B. `/dev/dsk/c?d?s?`) und *knoten* den Root-Namen dieses Dateisystems. *dateisys* und *knoten* sind daher die ersten beiden Zeichenketten in den Einträgen der Einhängetabelle.

DATEIEN

`/etc/mnttab`

SIEHE AUCH

`mount(1M)`.

FEHLERMELDUNG

Probleme können auftreten, wenn *dateisys* oder *knoten* länger als 32 Zeichen sind. setmnt richtet eine Obergrenze für die Anzahl der Einträge in `mnttab` ein.

BEZEICHNUNG

setuname – ändert den System- oder Knotennamen

ÜBERSICHT

setuname [-s *name*] [-n *knoten*] [-t]

BESCHREIBUNG

setuname ändert die Parameterwerte für den Systemnamen und den Knotennamen. Jeder Parameter kann mit setuname und der entsprechenden Option geändert werden.

Für dieses Kommando stehen folgende Optionen und Argumente zur Verfügung:

- s Ändert den Systemnamen. *name* gibt den neuen Systemnamen an und kann aus alphanumerischen Zeichen und den Sonderzeichen Gedankenstrich, Unterstreichungszeichen und Dollarzeichen bestehen.
- n Ändert den Knotennamen. *knoten* gibt den neuen Knotennamen im Netzwerk an und kann aus alphanumerischen Zeichen und den Sonderzeichen Gedankenstrich, Unterstreichungszeichen und Dollarzeichen bestehen.
- t Bewirkt eine temporäre Änderung. Es wird nicht versucht, diese Änderung festzuschreiben.

Mindestens eine der Optionen -s und -n muß angegeben werden, wenn setuname aufgerufen wird.

Die Systemarchitektur stellt möglicherweise Anforderungen an die Größe des System- und Netzwerkknottennamens. Das Kommando meldet einen schweren Fehler und gibt eine Fehlermeldung aus, wenn der angegebene Name die Anforderungen des Systems nicht erfüllt.

HINWEIS

setuname versucht, die Parameterwerte an zwei Stellen zu ändern: den aktiven Kernel und, was aufgrund der Implementierung erforderlich ist, beim systemübergreifenden Neuladen. Eine temporäre Änderung beeinflusst immer nur den aktiven Kernel.

BEZEICHNUNG

`setup` – initialisiert das System für den ersten Benutzer

ÜBERSICHT

`setup`

BESCHREIBUNG

Mit dem Kommando `setup`, das unter demselben Namen auch als Login zur Verfügung steht, kann der erste Benutzer als "Eigentümer" des Systems etabliert werden.

Der Benutzer kann dann Datum, Zeit und Zeitzone des Systems festlegen.

Anschließend kann der Benutzer den Knotennamen des Systems festlegen.

Der Benutzer kann das System vor unberechtigter Modifikation der Maschinenkonfiguration und der Software schützen, indem er den Verwaltungs- und Pflegefunktionen Paßwörter zuweist. In der Regel aktiviert der erste Benutzer dieses Kommando über das `setup`-Login, für das zunächst kein Paßwort definiert ist, und weist dann den verschiedenen Funktionen im System Paßwörter zu. Funktionen, denen der Benutzer kein Paßwort zuweist, können von jedem Benutzer aufgerufen werden.

Der Benutzer kann den System-Logins wie `root`, `bin` etc. Paßwörter zuweisen, vorausgesetzt, daß für diese nicht bereits Paßwörter existieren. Ist für ein Login kein Paßwort angegeben, kann dieses nur über das entsprechende Login oder `root` geändert werden.

Schließlich darf der Benutzer die ersten Anmeldungen für das System definieren, beginnend mit seiner eigenen.

SIEHE AUCH

`passwd(1)`.

FEHLERMELDUNG

Das Kommando `passwd(1)` gibt eine entsprechende Meldung aus, wenn das angegebene Paßwort nicht den Standards entspricht.

HINWEIS

Ist für das Login `setup` kein Paßwort definiert, kann jeder beliebige Benutzer anderen Funktionen Paßwörter zuweisen.

BEZEICHNUNG

share – stellt fernen Dateisystemen lokale Ressourcen zum Einhängen zur Verfügung

ÜBERSICHT

```
share [-F dateisystem-typ] [-o spezifische_optionen] [-d beschreibung]
      [pfadname [ressourcenname]]
```

BESCHREIBUNG

Das Kommando `share` stellt eine Ressource zur Verfügung, so daß sie über ein fernes Dateisystem des Typs *dateisystem-typ* eingehängt werden kann. Wird die Option `-F dateisystem-typ` ausgelassen, wird der erste in der Datei `/etc/dfs/fstypes` aufgeführte Dateisystem-Typ als Standardwert verwendet. *spezifische_optionen* wie auch die Semantik des Arguments *ressourcenname* sind spezifisch für bestimmte verteilte Dateisysteme. Wird `share` nur mit dem Dateisystem-Typ aufgerufen, zeigt es alle Ressourcen an, die dem lokalen System von dem angegebenen Dateisystem zur gemeinsamen Benutzung überlassen werden. Wird `share` ohne Argumente aufgerufen, gibt es alle Ressourcen aus, die von dem lokalen System zur gemeinsamen Benutzung zur Verfügung gestellt werden.

Mit *spezifische_optionen* wird der Zugriff auf die gemeinsam benutzten Ressourcen gesteuert. Hierbei kann es sich um folgendes handeln:

<code>rw</code>	<i>pfadname</i> wird allen Clients für Lese-/Schreibzugriffe zur Verfügung gestellt. Dies ist das standardmäßige Systemverhalten.
<code>rw=client[:client]...</code>	<i>pfadname</i> wird nur den aufgelisteten Clients für Lese-/Schreibzugriffe zur Verfügung gestellt. Kein anderes System kann auf <i>ressourcenname</i> zugreifen.
<code>ro</code>	<i>pfadname</i> wird allen Clients schreibgeschützt zur Verfügung gestellt.
<code>ro=client[:client]...</code>	<i>pfadname</i> wird nur den aufgelisteten Clients schreibgeschützt zur Verfügung gestellt. Kein anderes System kann auf <i>pfadname</i> zugreifen.

Mit der Option `-d` kann eine Beschreibung der gemeinsam benutzten Ressource angegeben werden.

DATEIEN

```
/etc/dfs/dfstab
/etc/dfs/sharetab
/etc/dfs/fstypes
```

SIEHE AUCH

unshare(1M)

BEZEICHNUNG

share – stellt fernen Dateisystemen lokale NFS-Ressourcen zum Einhängen zur Verfügung

ÜBERSICHT

share [-F nfs] [-o *spezifische_optionen*] [-d *beschreibung*] *pfadname*

BESCHREIBUNG

Mit dem Kommando share werden lokale Ressourcen zur Verfügung gestellt, so daß sie von fernen Systemen eingehängt werden können.

Wird kein Argument angegeben, zeigt share alle Ressourcen an, die zur Zeit gemeinsam benutzt werden, einschließlich der NFS-Ressourcen und der Ressourcen, die über andere Pakete für verteilte Dateisysteme gemeinsam benutzt werden.

Die folgenden Optionen werden unterstützt:

-o *spezifische_optionen*

Optionen werden durch Komma getrennt in einer Liste von Schlüsselwörtern und Wertzuweisungen an Attribute angegeben. Diese werden von dem für den Dateisystem-Typ spezifischen Kommando interpretiert.

spezifische_optionen können eine Kombination der folgenden Angaben sein:

rw Allen Clients wird der gemeinsame Lese-/Schreibzugriff gewährt.

rw=*client[:client]*...

Den aufgeführten Clients wird der gemeinsame Lese-/Schreibzugriff gewährt. Diese Angabe setzt die für die Clients spezifizierte Unteroption ro außer Kraft.

ro Allen Clients wird nur der gemeinsame Lesezugriff gewährt.

ro=*client[:client]*...

Den aufgeführten Clients wird nur der gemeinsame Lesezugriff gewährt. Diese Angabe setzt die für die Clients spezifizierte Unteroption rw außer Kraft.

anon=*benutzernr*

Die *uid* wird auf die effektive ID (Benutzernummer) nicht berechtigter Benutzer gesetzt, wenn die Berechtigungsprüfung AUTH_DES verwendet wird, oder auf root, wenn die Berechtigungsprüfung AUTH_UNIX verwendet wird. Standardmäßig wird unbekannten Benutzern die effektive ID (Benutzernummer) UID_NOBODY zugewiesen. Wenn die *benutzernr* auf -1 gesetzt ist, wird der Zugriff verweigert.

root=*host[:host]*...

Nur root-Benutzer der angegebenen Hosts können sich als root anmelden. Standardmäßig wird keinem Host der root-Zugriff gewährt.

share (1M)

(NFS)

share (1M)

`secure`

Die Clients müssen die Berechtigungsprüfung AUTH_DES von RPC verwenden. Die Standardberechtigungsprüfung ist AUTH_UNIX.

Wenn keine *spezifischen_optionen* angegeben werden, gilt standardmäßig der Lese-/Schreibzugriff für alle Clients.

`-d` *beschreibung*

Hiermit kann ein Kommentar angegeben werden, der die gemeinsam benutzte Ressource beschreibt.

`pfadname`

Gibt den Pfadnamen der Ressource an, die gemeinsam benutzt werden soll.

DATEIEN

`/etc/dfs/fstypes`

`/etc/dfs/sharetab`

SIEHE AUCH

`unshare(1M)`

HINWEIS

Das Kommando schlägt fehl, wenn sowohl `ro` als auch `rw` angegeben werden. Wenn derselbe Client-Name in beiden Listen, `ro=` und `rw=`, vorkommt, setzt `rw` die Angabe `ro` außer Kraft, so daß dem angegebenen Client der Lese-/Schreibzugriff gewährt wird.

`ro=`, `rw=` und `root=` funktionieren sicher über UDP, u.U. aber nicht über andere Transport-Provider.

Wird eine Ressource mit einer `ro=-`-Liste und einer `root=-`-Liste gemeinsam benutzt, erhalten alle Hosts auf der `root=-`-Liste ausschließlich Nur-Lesezugriff, unabhängig davon, ob der Host in der `ro=-`-Liste enthalten ist, es sei denn, `rw` gilt als Standardwert oder der Host ist in der `rw=-`-Liste enthalten. Das gleiche gilt, wenn die Ressource mit `ro` als Standard gemeinsam benutzt wird. Die folgenden `share`-Kommandos beispielsweise gewähren `hostb` Nur-Lesezugriff:

```
share -F nfs -oro=hosta,root=hostb /var
```

```
share -F nfs -oro,root=hostb /var
```

Folgende Kommandos geben `hostb` Lese-/Schreibzugriff:

```
share -F nfs -oro=hosta,rw=hostb,root=hostb /var
```

```
share -F nfs -oroot=hostb /var
```

BEZEICHNUNG

shareall, unshareall – gemeinsame Benutzung mehrerer Ressourcen, Beenden der gemeinsamen Benutzung dieser Ressourcen

ÜBERSICHT

```
shareall [-F dateisystem-typ[,dateisystem-typ...]] [- | datei]  
unshareall [-F dateisystem-typ[,dateisystem-typ...]]
```

BESCHREIBUNG

Wird das Kommando `shareall` ohne Argumente eingegeben, können alle Ressourcen aus *datei*, die eine Liste von `share`-Kommandozeilen enthält, gemeinsam benutzt werden. Ist der Operand ein Bindestrich (`-`), werden die `share`-Kommandozeilen aus der Standardeingabe gelesen. Werden weder *datei* noch ein Bindestrich angegeben, wird die Datei `/etc/dfs/dfstab` als Standarddatei benutzt.

Ressourcen können bezogen auf spezifische Dateisysteme gemeinsam benutzt werden, indem man die Dateisysteme in Form durch Komma getrennter Listeneinträge als Argumente zu `-F` angibt.

`unshareall` beendet die gemeinsame Benutzung aller im Augenblick gemeinsam benutzter Ressourcen. Ohne die Option `-F` bezieht sich das Kommando auf alle verteilten Dateisystem-Typen.

DATEIEN

`/etc/dfs/dfstab`

SIEHE AUCH

`share(1M)`, `unshare(1M)`.

BEZEICHNUNG

`shutdown` – System herunterfahren, Systemstatus ändern

ÜBERSICHT

`/usr/sbin/shutdown [ -y ] [ -g wartezeit ] [ -i anf_status ]`

BESCHREIBUNG

Dieses Kommando wird vom Systemverwalter zur Statusänderung des Rechners aufgerufen. In den meisten Fällen dient es dazu, vom Mehrbenutzer-Betrieb (Status 2) in einen anderen Zustand zu wechseln (s. weiter unten).

Im Standardfall wird das System dadurch in einen Zustand versetzt, in dem nur noch von der Konsole aus Zugang zum Rechner möglich ist. Das System befindet sich dann im Einbenutzer-Betrieb (s. weiter unten).

Bevor mit dem eigentlichen Herunterfahren des System begonnen wird, sendet das Kommando eine Meldung zur Warnung an alle Benutzer. Danach wird eine endgültige Meldung über das Herunterfahren des Systems gesendet. Im Standardfall fordert das Kommando eine Bestätigung an, um mit dem Beenden der noch laufenden Dämonen und Prozesse beginnen zu können. Die Optionen werden wie folgt benutzt:

`-y` Nimmt die positive Bestätigung vorweg, so daß das Kommando ohne Eingriff des Benutzers ablaufen kann. Standardmäßig wird zwischen der Warnung und der endgültigen Meldung 60 Sekunden gewartet. Danach dürfen weitere 60 Sekunden verstreichen, bis die Bestätigung erfolgt.

`-g wartezeit` Damit kann der Systemverwalter die Voreinstellung von 60 Sekunden auf einen anderen Wert einstellen.

`-i anf_status` Damit wird der Status festgelegt, in den `init` zu versetzen ist. Normalerweise gilt hierfür der Systemzustand 0.

Weitere sinnvolle Systemzustände nach `shutdown` sind:

Status 0 Ordnungsgemäßes Herunterfahren des Rechners, so daß man den Strom abschalten kann. Falls der Rechner das kann, soll er selbst den Strom abstellen. Diese Funktionen werden durch die Prozedur `rc0` ausgeführt.

Status 1 Der Status 1 wird als Verwaltungs-Status bezeichnet. In diesem Zustand werden die für den Mehrbenutzer-Betrieb erforderlichen Dateisysteme eingehängt und es können diesbezügliche Login-Anmeldungen entgegengenommen werden. Wenn das System beim Hochfahren von der Firmware in den Status 1 wechselt, ist lediglich die Konsole aktiviert; die Dienste für den Mehrbenutzer-Betrieb (Status 2) sind noch nicht verfügbar. Es ist zu beachten, daß beim Übergang vom Mehrbenutzer-Betrieb in den Zustand 1 nicht alle Benutzerprozesse gestoppt werden.

Status s, S

Der Status `s` (oder `S`) ist der sogenannte Einbenutzer-Betrieb (single user). Beim Übergang in diesen Zustand werden alle Benutzerprozesse gestoppt, die Dateisysteme für den Mehrbenutzer-Betrieb ausgehängt und der Zugang zum System auf die Konsole beschränkt. Anmeldungen für den Mehrbenutzer-Betrieb sind nicht mehr möglich.

shutdown(1M)

shutdown(1M)

Status 5 Stoppen des SINIX Systems und Übergang zur Firmware.

Status 6 Stoppen des SINIX Systems und beim Neuhochfahren in den Status wechseln, der durch den Eintrag `initdefault` in `/etc/inittab` vorgegeben ist. Ein gegebenenfalls neu erzeugter Systemkern wird ladefähig gemacht, d.h. in die Datei `/stand/unix` kopiert. Der alte Kernel in `/stand/unix` wird nach `/stand/unix.old` kopiert. Diese Funktionen werden von der Prozedur `rc6` ausgeführt.

SIEHE AUCH

`init(1M)`, `rc0(1M)`, `rc2(1M)`, `rc6(1M)`, `inittab(4)`.

BEZEICHNUNG

slink – Binder für STREAMS

ÜBERSICHT

```
slink [-v] [-p] [-u] [-f] [-c datei] [funktion [arg1 arg2 ...]]
```

BESCHREIBUNG

slink dient zum Konfigurieren von STREAMS Modulen und Treibern für eine Anwendung mit dem STREAMS TCP/IP Protokoll. Gesteuert wird die Eingabe für slink durch ein Skript, das die einzelnen STREAMS Operationen definiert. Dieses Skript wird normalerweise aus der Datei `/etc/strcf` (Konfigurations-Datei für STREAMS) gelesen.

Die folgenden Optionen können beim Aufruf von slink verwendet werden:

- c *datei* Skript steht in *datei* anstelle von `/etc/strcf`.
- v Jede Operation wird in `stderr` protokolliert.
- p Es werden keine ständigen Verbindungen verwendet (d.h. slink läuft im Hintergrund).
- f Es werden keine ständigen Verbindungen verwendet und auch kein eigener Sohnprozess erzeugt (d.h. slink läuft im Vordergrund).
- u Auflösen aller ständigen Verbindungen (d.h. Netzwerk herunterfahren).

Die Konfigurationsdatei enthält eine oder mehrere *Funktionsaufrufe*, die wiederum aus einem oder mehreren *Kommandos* bestehen. Jedes Kommando repräsentiert den Aufruf einer Benutzerfunktion oder einer Standardfunktion. Zu diesen Standardfunktionen gehören die STREAMS Basisfunktionen `open`, `link`, und `push`, sowie einige TCP/IP-spezifische Funktionen.

Bei der Ausführung von slink wird zunächst das Konfigurations-Skript gelesen und dann die Benutzerfunktion `boot` ausgeführt, die üblicherweise dazu dient, die Standardkonfiguration beim Laden herzustellen. Falls beim Aufruf von slink eine Funktion angegeben wurde, wird diese anstelle von `boot` ausgeführt.

Normalerweise richtet slink STREAMS mit ständigen Verbindungen (`I_PLINK`) ein. Danach beendet es sich. Wurde die Option `-p` angegeben, baut slink eine STREAMS Verbindung mit regulären Verbindungen (`I_LINK`) auf, um anschließend passiv im Hintergrund zu bleiben. Dabei werden alle während der Konfiguration geöffneten Datei-Deskriptoren offen gehalten. Wurde die Option `-f` angegeben, baut slink eine STREAMS Verbindung mit regulären Verbindungen (`I_LINK`) auf, um dann im Vordergrund zu bleiben. Dabei werden alle während der Konfiguration geöffneten Datei-Deskriptoren offen gehalten.

Eine Benutzerfunktion wird wie folgt definiert:

```
Funktions-Name {
    Kommando1
    Kommando2
    ...
}
```

Ein Kommando hat folgende Syntax:

```
funktion arg1 arg2 arg3 ...
oder
Variable = funktion arg1 arg2 arg3 ...
```

Die Position des Zeilenvorschubs ist zu beachten: ein Zeilenvorschub muß jeweils auf die linke und rechte geschweifte Klammer, sowie jedes Kommando folgen. Es können mehrere Zeilenvorschübe hintereinander folgen (Lesbarkeit). Ein Gegenschrägstrich (\) unmittelbar hinter einem Zeilenvorschub entspricht einem Leerzeichen, d.h. ein Kommando kann in der nächsten Zeile fortgesetzt werden. Andere Trennzeichen (Leerzeichen und Tabulatoren) können beliebig gesetzt werden; es muß jedoch mindestens ein Trennzeichen zwischen dem Funktionsnamen und den Argumenten eines Kommandos stehen.

Kommentare werden durch # und Zeilenvorschub begrenzt und sind gleichbedeutend mit einem einfachen Zeilenvorschub.

Funktions- und Variablenname können aus beliebig vielen Zeichen A-Z, a-z, 0-9, und _ bestehen; das erste Zeichen darf jedoch keine Zahl sein. Funktions- und Variablennamen werden in getrennten Wertebereichen verwaltet (d.h. gleiche Namen sind zulässig). Alle Funktionen sind global definiert und können vorwärts referenziert werden. Alle Variablen sind innerhalb der definierenden Funktion lokal.

Variablen werden immer dann definiert, wenn sie auf der linken Seite eines Gleichheitszeichens (=) innerhalb eines Kommandos stehen, so wie im folgenden Beispiel:

```
tcp = open /dev/tcp
```

Die Variable nimmt den Wert an, der vom aufgerufenen Kommando zurückgeliefert wird. Im obigen Beispiel enthält die Variable `tcp` den Datei-Deskriptor, der vom Kommando `open` zurückgeliefert wird.

Als Argumente für ein Kommando können Variablen, Parameter oder Zeichenketten übergeben werden.

Eine Variable kann nur dann als Argument übergeben werden, wenn ihr vorher innerhalb derselben Funktion ein Wert zugewiesen wurde.

Parameter beginnen mit einem Dollarzeichen (\$), gefolgt von einer maximal zweistelligen Dezimalzahl, und werden durch die zugehörigen Argumente aus dem Funktionsaufruf aktualisiert. Die Angabe eines Parameters ohne entsprechendes Argument im Funktionsaufruf führt zu einem Laufzeitfehler (z.B. wenn in einem Kommando ein Parameter \$3 verwendet wird, im Funktionsaufruf aber nur 2 Argumente übergeben werden).

Zeichenketten werden als Folge von Buchstaben definiert, die zusätzlich noch durch Anführungsstriche (") begrenzt sein können. Anführungsstriche verhindern darüberhinaus, daß eine Zeichenkette als Variablenname oder Parameter interpretiert wird, beziehungsweise sie gestatten die Verwendung von Leerzeichen, Tabulatoren, und der Sonderzeichen {, }, =, und #. Der Gegenschrägstrich (\) kann dazu benutzt werden, die Sonderzeichen mit besonderer Bedeutung {, }, =, #, ", und \ einzeln zu entwerten.

Die folgenden Standardfunktionen werden durch `slink` unterstützt:

<code>open pfadname</code>	Öffnet das Gerät mit dem angegebenen <i>pfadname</i> . Liefert einen Datei-Deskriptor auf die geöffnete STREAMS Verbindung.
<code>link fd1 fd2</code>	Ordnet die durch <i>fd2</i> bezeichnete STREAMS Verbindung unterhalb der durch <i>fd1</i> bezeichneten STREAMS Verbindung zu. Liefert einen Identifikator für die neue Verbindung. Falls weder die Option <code>-f</code> noch die Option <code>-p</code> in der Befehlszeile angegeben wurde, werden alle STREAMS Verbindungen mit ständigen Verbindungen aufgebaut. Hinweis: <i>fd2</i> kann nach dieser Funktion nicht anderweitig verwendet werden.
<code>push fd modul</code>	Pusht das Modul <i>modul</i> auf den durch <i>fd</i> bezeichneten Stream.
<code>sifname fd verbindung name</code>	Sendet ein <code>SIOCSIFNAME</code> ioctl-Signal (set interface name) über die durch <i>fd</i> bezeichnete STREAMS Verbindung an die Verbindung mit dem Identifikator <i>verbindung</i> und stellt das neue Interface <i>name</i> ein.
<code>unitsel fd gerät</code>	Sendet ein <code>IF_UNITSEL</code> ioctl-Signal (unit select) über die durch <i>fd</i> bezeichnete STREAMS Verbindung und wählt das Gerät <i>gerät</i> aus.
<code>dlattach fd gerät</code>	Sendet eine <code>DL_ATTACH_REQ</code> -Meldung über die durch <i>fd</i> bezeichnete STREAMS Verbindung für das Gerät <i>gerät</i> .
<code>initqp pfadname queueName minimum maximum ...</code>	Sendet ein <code>INITQPARMS</code> (initialize queue parameters) ioctl-Signal an das Gerät mit dem Pfadnamen <i>pfadname</i> . <i>queueName</i> bezeichnet die Warteschlange, für die Minimum- und Maximum-Grenzwerte gesetzt werden: <code>hd</code> Kopf der STREAMS Verbindung <code>rq</code> Lese-Warteschlange <code>wq</code> Schreib-Warteschlange <code>muxrq</code> Multiplexer Lese-Warteschlange <code>muxwq</code> Multiplexer Schreib-Warteschlange <i>minimum</i> und <i>maximum</i> bestimmen die neuen Grenzwerte (Minimum, Maximum) für die Warteschlange. Sowohl <i>minimum</i> als auch <i>maximum</i> müssen angegeben werden; wenn nur einer der beiden Parameter geändert werden soll, kann der andere durch <code>(-)</code> ersetzt werden. Bis zu fünf Tupel der Form <i>queueName minimum maximum</i> können im Funktionsaufruf abgesetzt werden.
<code>strcat zeichenkette1 zeichenkette2</code>	Verkettet die Zeichenkette <i>zeichenkette1</i> mit der Zeichenkette <i>zeichenkette2</i> und liefert die resultierende Zeichenkette zurück.

slink(1M)**slink(1M)**

return *wert*

Setzt den Return-Code der aktuellen Funktion auf *wert*.
Hinweis: `return` beendet nicht die Abarbeitung der aktuellen Funktion.

DATEIEN

/etc/strcf

SIEHE AUCH

strcf(4)

BEZEICHNUNG

smtp – Versenden von Nachrichten an einen fernen Rechner mittels SMTP

ÜBERSICHT

smtp [-D] [-d *bereich*] [-H *hallohost*] *sender host empf ...*

BESCHREIBUNG

Das Kommando `smtp` verwendet SMTP (Simple Mail Transfer Protocol), um eine Nachricht an einen fernen Rechner zu senden. Die Nachricht wird von der Standardeingabe gelesen. In *sender* steht der Absender; die Empfänger sind in *empf* aufgelistet.

Nach dem Verbindungsaufbau wird `smtp` den ersten Transportweg verwenden, für den die Funktion `netdir_getbyname(3)` eine Adresse abliefert, die wiederum auf dem Namen des Hosts, dem von `getnetpath(3)` zurückgelieferten Transportweg und dem `smtp` Dienst basiert. Normalerweise wird dies `tcp` sein.

Die Optionen für `smtp` haben folgende Bedeutung:

- D Schaltet die Fehlerverfolgung ein. Die Ausgabe erfolgt auf die Standardfehlerausgabe.
- H *hallohost* Damit kann man den Namen des Hosts angeben, der in der SMTP HELO Meldung verwendet wird (standardmäßig ist das der Rechnername).
- d *bereich* Damit kann der Bereichsname gesetzt werden, der für diesen Host verwendet werden soll.

Normalerweise wird `smtp` von `smtpsched` aufgerufen, um Post aus der Warteschlange `/var/spool/smtpq` auszuliefern.

DATEIEN

`/var/spool/smtpq` Warteschlange für Nachrichten

SIEHE AUCH

`smtpsched(1M)`
RFC821 – Simple Mail Transfer Protocol

BEZEICHNUNG

smtpd – Empfang eingehender SMTP Nachrichten

ÜBERSICHT

```
smtpd [ -n ] [ -H hallohost ] [ -h akt_host ] [ -L lastlim ] [ -l maxproz ]
```

BESCHREIBUNG

Normalerweise läuft smtpd im Mehrbenutzer-Betrieb als Dämon und wartet auf Anfragen ferner Rechner, Post senden zu dürfen. Zu diesem Zweck horcht smtpd, ob auf irgend einem auf TLI basierenden Netzwerk (normalerweise das tcp Netzwerk), in dem der SMTP Dienst definiert ist (vgl. netdir_getbyname(3)), solche Anfragen anstehen. Sowie welche empfangen werden, erzeugt smtpd für jede Anfrage einen Sohnprozeß, in dem ebenfalls smtpd abläuft, der die SMTP-Transaktion bearbeitet.

Die Optionen für smtpd haben folgende Bedeutung:

- n Es sollen keine smtpsched Prozesse erzeugt werden, um eintreffende Post zu bearbeiten. Statt dessen soll diese Aufgabe dem stündlichen Aufruf von smtpsched durch cron(1) überlassen bleiben.
- H hallohost Mit dieser Option kann der Name für den Host angegeben werden, wie er in der SMTP HELO-Meldung verwendet wird. Fehlt diese Option, wird in der HELO-Meldung der Knotenname des Systems verwendet.
- h akt_host Legt fest, daß in der From-Zeile der Nachricht der Netzwerkname des aktuellen Hosts dem Pfadnamen des Absenders vorangestellt wird. Diese Option wird dem Programm fromsmtp durchgereicht.
- L lastlim Legt die Obergrenze der Systemlast fest, ab der smtpd keine Sohnprozesse mehr erzeugt. Die aktuelle Systemlast wird festgestellt, indem die Variable des Systemkerns avenrun gelesen wird. Ist diese Option nicht angegeben, existiert keine solche Obergrenze.
- l maxproz Mit dieser Option wird die maximale Anzahl von Sohnprozessen festgelegt, die zur gleichen Zeit ablaufen dürfen. Jeder Sohnprozeß behandelt eine SMTP-Interaktion. Ist diese Option nicht angegeben, existiert keine solche Obergrenze.

Erfolgreich empfangene Post wird über eine Pipe dem Programm fromsmtp zugeleitet, das seinerseits die Post über eine Pipe an rmail ausliefert. Die Datei /var/spool/smtpq/LOG enthält das Protokoll aller Aktivitäten von smtpd.

DATEIEN

/dev/kmem	Geräte-datei, um die aktuelle Systemlast festzustellen
/etc/services	Liste der TCP/UDP Dienste (SMTP sollte als 25/tcp eingetragen sein)
/etc/net/*/services	Liste weiterer Dienste im TLI-Netzwerk
/usr/lib/mail/smr/cmd/fromsmtp	Programm, dem eingehende Post über eine Pipe zugeleitet wird
/var/spool/smtpq/LOG	Protokoll der smtpd-Aktivitäten

SIEHE AUCH

cron(1M), fromsmtp(1M), smtp(1M)

BEZEICHNUNG

smtpqer – Nachrichten zur Auslieferung durch SMTP in Warteschlange einreihen

ÜBERSICHT

smtpqer [-nu] [-a *zieladr*] [-d *bereich*] [-H *hallohost*] *sender host empf ...*

BESCHREIBUNG

smtpqer liest Postnachrichten von der Standardeingabe ein und reiht sie zur gelegentlichen Auslieferung durch smtp in eine Warteschlange ein. Die gewählte Warteschlange korrespondiert zu dem unter *zieladr* spezifizierten Host.

Im Normalfall sollte smtpqer von mail aufgerufen werden, was durch den folgenden Eintrag in die Datei /etc/mail/mailstyp bewirkt wird:

```
‘.‘ ‘([!@]+)!(.+)’ ‘< /usr/lib/mail/surrecmd/smtpqer %R \\1 \\2’
```

Als erstes überprüft smtpqer den Namen des Hosts in *zieladr*. Ist es ein erreichbarer Host (d.h. netdir_getbyname(3) kann ihn auf mindestens einem TLI-Netzwerk finden), wird die Nachricht in die Warteschlange eingereiht und smtpqer beendet sich mit einem Exit-Code 0. Das bedeutet, daß die Operation erfolgreich ausgeführt wurde. Andernfalls wird die Nachricht nicht eingereiht und als Exit-Code 1 zurückgegeben.

Das SMTP Verzeichnis /var/spool/smtpq enthält die Warteschlangen, in die die Nachrichten eingereiht werden. Sofern nicht die Option -u gesetzt ist, werden die Nachrichten zuerst noch durch das Filterprogramm tosmtp auf das RFC822-Format konvertiert. Schließlich ruft smtpqer zur Auslieferung der Post das Programm smtpsched auf.

Mit der Option -H kann der Name für den Host angegeben werden, wie er in der SMTP HELO-Meldung verwendet wird. Diese Option wird an die beiden Programme tosmtp und smtp durchgereicht.

Mit der Option -d wird der Bereichsname festgelegt, der für den Host verwendet werden soll. Diese Option wird an das Programm tosmtp durchgereicht. Fehlt diese Option und gibt es in der Konfigurationsdatei mailcnfg eine Definition für den Bereichsnamen, wird dieser benutzt.

Die Option -a dient der Angabe von *zieladr* und wird an das Programm smtp durchgereicht. Die Angabe der Option -n schließlich verhindert, daß smtpqer einen smtpsched Prozeß zur Auslieferung der Post startet.

smtpqer (1M)

smtpqer (1M)

DATEIEN

/usr/bin/rmail	woher die Post stammt
/etc/hosts	Datenbank mit den fernen Rechnern (für TCP/IP)
/etc/mail/mailcnfg	Konfigurationsdatei für Postdienste
/etc/net/*/hosts	Datenbank mit den fernen Rechnern (für andere TLI Netzwerke)
/etc/mail/maillsurr	Steuerdatei, die den Aufruf von smtpqer regelt
/usr/lib/mail/surrcmd/smtpsched	Programm zur Bearbeitung von Warteschlangen für Nachrichten
/usr/lib/mail/surrcmd/smtp	Programm zur Auslieferung von Nachrichten an ferne Rechner
/usr/lib/mail/surrcmd/tosmtp	Filterprogramm zur Konvertierung auf RFC822-Format
/var/spool/smtpq	Verzeichnis für die Warteschlangen

SIEHE AUCH

rmail(1M), smtpsched(1M), smtp(1M), tosmtp(1M)
getdomainname(3) im *Referenzhandbuch für Programmierer*.
RFC822-Standardformat der ARPA Internet Text Messages

BEZEICHNUNG

smtpsched – Bearbeitung von Nachrichten in der SMTP-Warteschlange

ÜBERSICHT

```
smtpsched [ -c ] [ -v ] [ -t ] [ -s p_anz ] [ -r tage ] [ -w tage ] [ wsnamen ]
```

BESCHREIBUNG

Das Kommando `smtpsched` dient der Bearbeitung von Nachrichten in der SMTP-Warteschlange `/var/spool/smtpq`. Es wird von dem SMTP-Post-Ersatzdienst `smtpqer` automatisch aufgerufen, sowie Post für die Zustellung an einen fernen Rechner in die SMTP-Warteschlange eingereiht wird. Bei eingehender Post wird es von `smtpd` aufgerufen. Außerdem sollte es jede Stunde von `cron` aufgerufen werden, um Post auszuliefern, die nicht sofort zugestellt werden konnte.

Normalerweise versucht `smtpsched` alle Nachrichten zu versenden, die in den Unterverzeichnissen von `/var/spool/smtpq` eingereiht sind. Ist jedoch *wsnamen* spezifiziert, werden nur die dort aufgezählten Unterverzeichnisse nach auszuliefernder Post durchsucht. Die einzelnen Unterverzeichnisse beziehen sich auf jeweils verschiedene ferne Rechner.

Die Optionen für `smtpsched` sind:

- `-c` Leere Verzeichnisse für Warteschlangen werden gelöscht.
- `-v` Schaltet ein ausführliches Protokoll ein.
- `-t` Testmodus. Die Aktionen, die `smtpsched` durchführen würde, werden aufgezeichnet, aber nicht ausgeführt.
- `-s p_anz` Damit wird die maximale Anzahl konkurrierender `smtpsched` Prozesse festgelegt, die zu gleicher Zeit laufen dürfen. Bei Überschreitung dieser Anzahl beendet sich `smtpsched` sofort.
- `-r tage` Verursacht das Zurückschicken von Post, die älter als *tage* ist.
- `-w tage` Post, die älter als *tage* ist, löst eine Warnung aus, die dem Absender zugestellt wird.

DATEIEN

<code>/usr/lib/mail/surrcmd/smtp</code>	liefert die Post aus
<code>/usr/lib/mail/surrcmd/smtpqer</code>	reiht Post in Warteschlangen ein
<code>/var/spool/smtpq</code>	Warteschlange für Nachrichten
<code>/var/spool/smtpq/LOG*</code>	Protokolldateien
<code>/var/spool/smtpq/<i>host</i></code>	Warteschlange für Nachrichten, die für <i>host</i> bestimmt sind

SIEHE AUCH

`cron(1M)`, `smtp(1M)`, `smtpqer(1M)`

BEZEICHNUNG

strace – Ausgabe von STREAMS Überwachungsmeldungen

ÜBERSICHT

strace [*modul untermodul ebene*] ...

BESCHREIBUNG

strace ohne Argumente schreibt alle STREAMS Überwachungsmeldungen von allen Treibern und Modulen in die eigene Standardausgabe. Diese Meldungen kommen vom STREAMS Protokolltreiber (log(7)). Argumente müssen als Tupel der Form *modul*, *untermodul*, *ebene* angegeben werden, wobei *modul* die STREAMS Modulnummer bezeichnet, *untermodul* die Nummer eines Untermoduls, und *ebene* die Überwachungsebene. Jedes Tupel bestimmt, für welchen Modul/Treiber und Untermodul (üblicherweise das untergeordnete Gerät) Überwachungsmeldungen empfangen werden sollen, und die gewünschte Überwachungsebene (gleich oder unterhalb der aktuellen Ebene). Das reservierte Wort `all` innerhalb eines solchen Tupels bewirkt, daß das entsprechende Attribut nicht eingeschränkt wird.

Jede Überwachungsmeldung hat das Format:

<nr> <uhrzeit> <systemzeit> <ebene> <status> <modul> <untermodul> <text>

<nr> Laufende Nummer der Überwachungsmeldung

<uhrzeit> Uhrzeit der Meldung in *hh:mm:ss*

<systemzeit>
Systemzeit der Meldung seit dem letztem Laden

<ebene> Prioritätsebene

<status> E : Meldung wurde auch in die Fehlerdatei geschrieben
F : Schwerer Fehler
N : Eine Nachricht wurde an den Systemverwalter gesendet

<modul> Modulnummer der Sendequelle

<untermodul>
Untermodulnummer der Sendequelle

<text> Klartext der Überwachungsmeldung

Ein einmal gestarteter strace bleibt aktiv, bis er wieder vom Benutzer beendet wird.

BEISPIELE

Ausgabe aller Überwachungsmeldungen für das Modul bzw. den Treiber mit der Nummer 41:

```
strace 41 all all
```

Ausgabe aller Überwachungsmeldungen für das Modul bzw. den Treiber mit der Nummer 41 und der Untermodulnummer 0,1 oder 2:

```
strace 41 0 1 41 1 1 41 2 0
```

Meldungen von den Untermodulen 0 und 1 müssen eine Überwachungsebene von 1 oder weniger haben, diejenigen vom Untermodul 2 eine Überwachungsebene von 0.

strace (1M)

strace (1M)

SIEHE AUCH

log(7)

CES, Leitfaden und Werkzeuge für die Programmierung mit C

HINWEIS

Aufgrund von Durchsatzgründen darf nur ein einziger `strace` Prozeß im System Zugriff zum STREAMS Protokolltreiber haben. Der Protokolltreiber verwaltet die Liste der im Funktionsaufruf angegebenen Tupel und vergleicht sie mit jeder anfallenden Überwachungsmeldung. Bei Übereinstimmung wird die Meldung aufbereitet und an den `strace`-Prozeß gesendet. Das bedeutet, daß eine lange Liste von Tupeln eine größere Auswirkung auf den STREAMS Durchsatz insgesamt hat. Ein laufender `strace` hat die größte Auswirkung auf das Laufzeitverhalten derjenigen Module und Treiber, die die Überwachungsmeldungen für den `strace`-Prozeß auslösen. Immer dann, wenn Überwachungsmeldungen schneller erzeugt werden als sie der `strace`-Prozeß verarbeiten kann, können einzelne Meldungen verloren gehen. Dies kann anhand der laufenden Numerierung der Überwachungsmeldungen überprüft werden.

strclean (1M)

strclean (1M)

BEZEICHNUNG

strclean – Löschen der STREAMS Fehlerprotokolle

ÜBERSICHT

strclean [-d *verzeichnis*] [-a *alter*]

BESCHREIBUNG

strclean wird benutzt, um das Verzeichnis mit den STREAMS Fehlerprotokollen in regelmäßigen Abständen aufzuräumen (zum Beispiel mittels cron). Normalerweise werden alle Dateien gelöscht, deren Namen mit *error.** im Verzeichnis */var/adm/streams* beginnen und die innerhalb der letzten 3 Tage nicht verändert wurden. Mit der -d Option kann ein anderes Verzeichnis als das Standardverzeichnis */var/adm/streams* gelöscht werden. Das maximale Alter in Tagen für eine Protokolldatei kann mit der Option -a modifiziert werden.

BEISPIEL

strclean -d /var/adm/streams -a 3

ist gleichbedeutend mit dem Aufruf von strclean ohne Argumente.

DATEIEN

*/var/adm/streams/error.**

SIEHE AUCH

cron(1M), strerr(1M)

CES, Leitfaden und Werkzeuge für die Programmierung mit C

HINWEIS

strclean wird normalerweise einmal pro Tag oder Woche vom cron aufgerufen.

strerr (1M)

strerr (1M)

BEZEICHNUNG

`strerr` – Dämon des STREAMS Fehlerprotokolls

ÜBERSICHT

`strerr`

BESCHREIBUNG

`strerr` empfängt Fehlerprotokollmeldungen vom STREAMS Protokolltreiber (`log(7)`) und hängt sie an eine Protokolldatei an. Die Fehlerprotokolldateien werden im Verzeichnis `/var/adm/streams` mit dem Namen `error.mm-tt` angelegt, wobei `mm` den Monat und `tt` den Tag der Meldung in einer Protokolldatei darstellt.

Eine Fehlermeldung hat das Format:

`<nr> <uhrzeit> <systemzeit> <status> <modul> <untermodul> <text>`

`<nr>` Laufende Nummer der Fehlermeldung

`<uhrzeit>` Uhrzeit der Meldung in `hh:mm:ss`

`<systemzeit>`

Systemzeit der Meldung seit dem letztem Laden

`<status>` T : Meldung wurde auch an einen Überwachungsprozeß geschickt

F : Schwerer Fehler

N : Eine Nachricht wurde an den Systemverwalter gesendet

`<modul>` Modulnummer der Sendequelle

`<untermodul>`

Untermodulenummer der Sendequelle

`<text>` Klartext der Fehlermeldung

Meldungen im Fehlerprotokoll weisen auf eine außergewöhnliche Situation hin, die eine geeignete Maßnahme des Systemverwalters erfordern. Meldungen, die mit dem Status F versehen sind, bedeuten den Totalausfall eines STREAMS Treibers oder Moduls. Meldungen mit dem Status N erfordern die Aufmerksamkeit des Systemverwalters; deshalb wird vom Protokolltreiber eine entsprechende Nachricht mittels `mail` an den Systemverwalter gesendet. Die Angabe einer Protokollebene hat normalerweise keine Bedeutung für das Fehlerprotokoll, außer wenn die Meldung gleichzeitig auch zum Überwachungsprozeß geschickt wird.

Ein einmal gestarteter `strerr` bleibt aktiv, bis er vom Benutzer wieder beendet wird, und wird üblicherweise asynchron gestartet.

DATEIEN

`/var/adm/streams/error.mm-dd`

SIEHE AUCH

`log(7)`

Leitfaden für Programmierer: STREAMS

strerr (1M)

strerr (1M)

HINWEIS

Es darf nur ein einziger `strerr` Prozeß im System auf den STREAMS Treiber zugreifen.

Falls ein Modul oder Treiber eine große Anzahl von Fehlermeldungen erzeugt und der Fehlerprotokollierer aktiv ist, kann der STREAMS Durchsatz beeinträchtigt werden. Wenn viele Meldungen in einer kurzen Zeit eintreffen, kann der Protokolltreiber möglicherweise nicht alle Meldungen absetzen. Dieser Umstand wird anhand einer lückenhaften Numerierung der Fehlermeldungen in den Protokolldateien angezeigt.

BEZEICHNUNG

sttydefs – Verwaltung von Leitungseinstellung und Suchsequenzen für TTY-Anschlüsse

ÜBERSICHT

```
/usr/sbin/sttydefs -a tty_marke [-b] [-n nächste_marke] [-i init_schalter]
                        [-f end_schalter]
```

```
/usr/sbin/sttydefs -l [tty_marke]
```

```
/usr/sbin/sttydefs -r tty_marke
```

BESCHREIBUNG

Mit dem Kommando sttydefs werden die Leitungseinstellungen und die Suchsequenzen für TTY-Anschlüsse verwaltet. Zu diesem Zweck erzeugt sttydefs in der Datei /etc/ttydefs neue Einträge oder löscht welche daraus.

Zusammen mit den Schaltern -a und -r kann sttydefs nur durch privilegierte Benutzer aufgerufen werden. Bei Verwendung des Schalters -l kann sttydefs von jedem Benutzer am System aufgerufen werden.

Die Optionen haben folgende Bedeutung:

- l Ist *tty_marke* angegeben, gibt sttydefs den Eintrag aus der Datei /etc/ttydefs aus, der zu *tty_marke* paßt. Ohne die Angabe von *tty_marke* gibt sttydefs den kompletten Inhalt der Datei /etc/ttydefs aus, wobei darauf geachtet wird, daß jede auszugebende Zeile formal korrekt ist und sich der Feldinhalt von *nächste_marke* auf eine existierende *tty_marke* bezieht.
 - a *tty_marke* Erweitert die Datei ttydefs um einen neuen Eintrag, wobei *tty_marke* als Zielverweis benutzt wird. Im folgenden wird die Wirkung der Optionen -b, -n, -i oder -f im Zusammenhang mit der Option -a beschrieben:
 - b Aktiviert die automatische Einstellung der Übertragungsgeschwindigkeit. Damit wird dem Rechner die Möglichkeit gegeben, ohne Benutzeranfrage die Übertragungsgeschwindigkeit eines TTY-Anschlusses gemäß der des damit verbundenen Geräts einzustellen.
 - n *nächste_marke* Legt den Wert des Feldes *nächste_marke* in der Datei /etc/ttydefs fest. Ohne diese Angabe setzt sttydefs diesen Wert auf *tty_marke*.
 - i *init_schalter* Legt den Wert des Feldes *init_schalter* in der Datei /etc/ttydefs fest. Er muß in einem von dem Kommando stty auswertbaren Format angegeben sein. Diese Schalter werden von ttymon bei der Bestimmung der richtigen Übertragungsgeschwindigkeit verwendet und werden vor der Ausgabe der Eingabeaufforderung gesetzt.
- Ohne diese Angabe setzt sttydefs den Wert des Feldes *init_schalter* auf den Schalter 9600 von termio(7).

`-f end_schalter`

Legt den Wert des Feldes `init_schalter` in der Datei `/etc/ttydefs` fest. Er muß in einem von dem Kommando `stty` auswertbaren Format angegeben sein. In `end_schalter` stehen die `termio(7)` Einstellungen, wie sie von `ttymon` verwendet werden, nachdem eine Anfrage für Verbindungsaufbau erfolgreich war, aber noch bevor der Service für den Anschluß aufgerufen wird. Ohne diese Angabe setzt `sttydefs` die Schalter `end_schalter` genau wie die Schalter `9600` und `sane` von `termio(7)`.

`-r tty_marke`

Damit werden alle Einträge aus der Datei `sttydefs` gelöscht, auf die ein Verweis `tty_marke` existiert.

AUSGABE

Bei fehlerfreiem Ablauf gibt `sttydefs` den Endestatus 0 zurück. `sttydefs -l` stellt die angeforderte Information zusammen und gibt sie auf der Standardausgabe aus.

BEISPIELE

Das folgende Kommando bewirkt die Ausgabe aller Einträge in der Datei `sttydefs`, wobei für jeden ungültigen Eintrag eine Fehlermeldung erfolgt.

```
sttydefs -l
```

Das folgende Kommando bewirkt die Ausgabe von Information zu nur einem Eintrag:

```
# sttydefs -l 9600
```

```
-----
9600:9600 hupcl erase ^h:9600 sane ixany tab3 hupcl erase ^h::4800
-----
```

```
ttylabel: 9600
initial flags: 9600 hupcl erase ^h
final flags: 9600 sane ixany tab3 hupcl erase ^h
autobaud: no
nextlabel: 4800
```

Die folgende Kommandosequenz fügt die Marken 1200, 2400, 4800 und 9600 neu hinzu und ordnet sie in eine ringförmige Liste ein.

```
sttydefs -a 1200 -n 2400 -i 1200 -f "1200 sane"
sttydefs -a 2400 -n 4800 -i 2400 -f "2400 sane"
sttydefs -a 4800 -n 9600 -i 4800 -f "4800 sane"
sttydefs -a 9600 -n 1200 -i 9600 -f "9600 sane"
```

DATEIEN

`/etc/ttydefs`

SIEHE AUCH

Leitfaden für Systemverwalter, "Terminal Line Settings."

BEZEICHNUNG

su – temporär zu einer anderen oder privilegierten Benutzerkennung wechseln

ÜBERSICHT

su [-] [*name* [*arg* ...]]

BESCHREIBUNG

Das Kommando `su` ermöglicht es einem Benutzer, temporär als ein anderer Benutzer zu arbeiten, ohne sich abmelden zu müssen. Die Voreinstellung für *name* ist `root` (Privilegierung).

Um `su` verwenden zu können, braucht man natürlich das geeignete Paßwort (sofern man nicht bereits als `root` arbeitet). Bei richtigem Paßwort startet `su` eine neue Shell, in der die reale und effektive Benutzernummer sowie zusätzliche Gruppenmitgliedschaft gemäß der des angegebenen Benutzers gilt. Die neue Shell ist das Programm, welches in dem Feld für die Shell im Eintrag des Benutzers in der Paßwortdatei aufgeführt ist (siehe `passwd(4)`). Fehlt der optionale Eintrag dort, wird `/usr/bin/sh` gestartet (siehe `sh(1)`). Das Betätigen der Tastenkombination `CTRL-d` bewirkt die Rückkehr in die ursprüngliche Shell mit den dort geltenden Benutzerrechten.

Eventuelle weitere Argumente der Kommandozeile werden an das als Shell aufgerufene Programm durchgereicht. Bei Programmen wie `sh` als Shell bewirkt ein Argument *arg* der Form `-c string`, daß *string* durch die Shell als Kommando ausgeführt wird. Ein Argument der Form `-r` stellt dem Benutzer eine eingeschränkte Shell zur Verfügung.

Die folgenden Aussagen gelten nur dann, wenn im Feld für die Shell im Benutzereintrag der Paßwortdatei ein `sh`-ähnliches Programm steht. Ist das erste Argument von `su` ein Minuszeichen (`-`), wird die Umgebung so geändert wie sie wäre, wenn sich der angegebene Benutzer selbst angemeldet hätte. Das wird dadurch erreicht, daß das als Shell aufgerufene Programm im ersten Argument *arg0* als erstes Zeichen ein Minuszeichen (`-`) hat, welches bewirkt, daß zuerst die Systemprofildatei (`/etc/profile`) und danach die Profildatei `.profile` des spezifizierten Benutzers in dessen HOME-Verzeichnis durchlaufen wird. Ansonsten wird die aktuelle Umgebung durchgereicht, wobei als Ausnahme für `root` die Umgebungsvariable `$PATH` mit `/sbin:/usr/sbin:/usr/bin:/etc` versorgt wird. Es ist zu beachten, daß im Fall von `/usr/bin/sh` als verwendete Shell beim Durchlauf der Profildatei `.profile` des Benutzers aus dem ersten Argument abgeleitet werden kann, ob das Programm von `login` oder von `su` aufgerufen wurde, je nachdem, ob *arg0* entweder `-sh` oder `-su` ist. Ist das aufgerufene Programm nicht `/usr/bin/sh`, so ist *arg0* beim Durchlauf von `.profile` sowohl von `login` als auch von `su` auf `-program` gesetzt.

Alle Versuche von Benutzern, mit `su` in eine andere Umgebung zu wechseln, werden in der Protokolldatei `/var/adm/sulog` vermerkt.

BEISPIELE

Um temporär als Benutzer `bin` mit der gleichen (exportierten) Umgebung wie vorher arbeiten zu können, muß man das folgende Kommando eingeben:

```
su bin
```

Um temporär als Benutzer `bin` mit der gleichen Umgebung arbeiten zu können, wie sie bei ursprünglicher Anmeldung als `bin` existierte, muß man das folgende Kommando eingeben:

```
su - bin
```

Um ein Kommando *command* vorübergehend in der Umgebung und mit den Zugriffsrechten von `bin` ablaufen zu lassen, muß man folgendes eingeben:

```
su - bin -c "command args"
```

DATEIEN

`/etc/passwd` Paßwortdatei des Rechners

`/etc/profile` Profildatei des Rechners

`$HOME/.profile` Profildatei des Benutzers

`/var/adm/sulog` Protokolldatei

`/etc/default/su` enthält die folgenden Standardparameter:

SULOG: Ist die Variable mit einem Dateinamen belegt, werden alle Versuche, mit `su` zu einem andern Benutzer zu wechseln, in dieser Datei protokolliert.

CONSOLE: Existiert diese Variable, werden alle `su`-Versuche auf der Konsole ausgegeben.

PATH: Voreinstellung für Suchpfad.

SUPATH: Voreinstellung für den Suchpfad, wenn ein Benutzer `su` aufruft.

SIEHE AUCH

`env(1)`, `login(1)`, `sh(1)` in den *Kommandos*.

`passwd(4)`, `profile(4)`, `environ(5)` im *Referenzhandbuch für Programmierer*.

sulogin (1M)

sulogin (1M)

BEZEICHNUNG

sulogin – Zugang zum Einbenutzer-Betrieb

ÜBERSICHT

sulogin

BESCHREIBUNG

Das Kommando `sulogin` wird beim Erststart des Systems von `init` automatisch aufgerufen und fordert den Benutzer auf, das Paßwort für den Systemverwalter oder aber EOF (üblicherweise CTRL-d) einzugeben. Im ersten Fall geht das System in den Modus für Systemverwaltung über (Einbenutzer-Betrieb), im zweiten wird es normal für den Mehrbenutzer-Betrieb hochgefahren. Dieses Kommando sollte von einem Benutzer niemals direkt aufgerufen werden.

DATEIEN

/etc/sulogin

SIEHE AUCH

`init(1M)` im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

swap – Schnittstelle für die swap-Verwaltung

ÜBERSICHT

```
/usr/sbin/swap -a swapname swaplow swaplänge
/usr/sbin/swap -d swapname swaplow
/usr/sbin/swap -l [ -s ]
/usr/sbin/swap -s
```

BESCHREIBUNG

Mit swap ist es möglich, im System Auslagerungsbereiche für die Speicherverwaltung zu vergrößern, zu verkleinern oder zu löschen. Gültige Optionen sind:

- a Der mit *swapname* angegebene Bereich wird dem Auslagerungsbereich hinzugefügt. *swapname* bezeichnet entweder eine normale Datei oder ist der Name eines blockorientierten logischen Geräts, wie zum Beispiel */dev/dsk/c0d0s2*. *swaplow* gibt den Offset in 512 Byte Blöcken an, bei dem der Swap-Bereich auf dem logischen Gerät beginnen soll. *swaplänge* gibt die Größe (in 512 Byte Blöcken) des Swap-Bereichs an. Diese Option kann nur ein privilegierter Benutzer verwenden. Normalerweise wird das Einrichten zusätzlicher Swap-Bereiche beim Hochfahren des Systems erledigt, wenn eine Start/Stop-Routine, die der Systemverwalter im Verzeichnis */etc/rc2.d* hinterlegt hat, von */sbin/rc2* aufgerufen wird.
- d Der mit *swapname* angegebene Swap-Bereich wird gelöscht. *swapname* bezeichnet entweder eine normale Datei oder ist der Name eines blockorientierten logischen Geräts, wie zum Beispiel */dev/dsk/c0d0s2*. *swaplow* gibt den Offset in 512 Byte Blöcken an, bei dem der zu löschende Swap-Bereich auf dem logischen Gerät beginnt. Die Verwendung dieser Option markiert den Swap-Bereich als "INDEL", d. h. daß sich dieser Bereich im Zustand des Gelöschtwerdens befindet. Von nun an werden keine neuen Blöcke mehr von diesem Bereich angefordert und das System versucht, noch belegte Blöcke aus diesem Swap-Bereich freizugeben. Der Swap-Bereich bleibt solange gültig, bis alle belegten Blöcke freigegeben sind. Diese Option kann nur von einem privilegierten Benutzer verwendet werden.
- l Ausgabe des Zustands aller Swap-Bereiche. Die Ausgabe erfolgt fünfspaltig:

path	Pfadname des Swap-Bereiches.
dev	Die Geräte- und Geräteklassennummer des Blockgeräts in dezimaler Form. Bei einer normalen Datei wird dafür Null ausgegeben.
swaplo	Der Wert <i>swaplow</i> für den Bereich (in 512 Byte Blöcken).
blocks	Die Größe <i>swaplänge</i> des Bereichs (in 512 Byte Blöcken).
free	Die Anzahl der freien (512 Byte) Blöcke des Bereichs. Rechts neben der Zahl wird INDEL ausgegeben, falls dieser Swap-Bereich gerade gelöscht wird.

swap(1M)

swap(1M)

- s Informiert über die Belegung des gesamten Auslagerungsbereiches:
 - allocated Der Anteil des Swap-Bereichs, der von privaten Seiten (Pages) belegt ist (in 512 Byte Blöcken).
 - reserved Der Anteil des Swap-Bereichs (in 512 Byte Blöcken), der zwar noch nicht belegt, aber von der Speicherverwaltung für die Erzeugung privater Seiten angefordert worden ist.
 - used Der Anteil des Swap-Bereichs (in 512 Byte Blöcken), der entweder belegt oder angefordert ist.
 - available Der aktuell verfügbare Anteil des Swap-Bereichs (in 512 Byte Blöcken) für künftige Reservierung und Belegung.

HINWEIS

Es wird nicht überprüft, ob eine Erweiterung des Swap-Bereichs zu Überschneidungen mit existierenden Dateisystemen führt.

sync(1M)

sync(1M)

BEZEICHNUNG

sync – Superblock aktualisieren

ÜBERSICHT

sync

BESCHREIBUNG

sync führt die Systembasisfunktion sync aus. Muß der Rechner angehalten werden, muß diese Funktion aus Gründen der Systemintegrität aufgerufen werden. Damit erzwingt man das Schreiben aller noch nicht geschriebenen Puffer auf die Platte. Damit ist sichergestellt, daß alle bis dahin erfolgten Dateiänderungen gesichert sind. Für nähere Einzelheiten siehe sync(2).

HINWEIS

Mit sync kann man das Schreiben von Puffern nur in lokaler Umgebung erzwingen. Wenn innerhalb einer NFS-Umgebung ein Schreibaufruf in eine Datei auf einem fernen Rechner abgesetzt wurde, kann sync nicht dazu verwendet werden, die Puffer des fernen Rechners auf dessen Platte zu schreiben.

SIEHE AUCH

sync(2) im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

sysadm – menügeführte Systemverwaltung

ÜBERSICHT

sysadm [*menü* | *funktion*]

BESCHREIBUNG

Wird sysadm ohne Argumente aufgerufen, bietet es eine Reihe von Menüs zur Erleichterung von Verwaltungstätigkeiten an. Falls beim Aufruf ein Menü oder eine Funktion (task) angegeben ist, geschieht folgendes: Ist das Menü oder die Funktion eindeutig spezifiziert, wird es bzw. sie sofort am Bildschirm ausgegeben, andernfalls erscheint eine Auswahlliste.

Die Kennung sysadm kann mit einem Paßwort geschützt werden. Um ein Paßwort einzurichten oder zu ändern, wird im Menü system_setup die Funktion password angestoßen. Zum Ändern eines vergebenen Paßworts dient auch das Kommando password.

Das Hauptmenü von sysadm enthält die folgenden Untermenüs:

backup_service	Backup Scheduling, Setup and Control
file_systems	File System Creation, Checking and Mounting
machine	Machine Configuration, Display and Shutdown
network_services	Network Services Administration
ports	Port Access Services and Monitors
printers	Printer Configuration and Services
restore_service	Restore From Backup Data
schedule_task	Schedule Automatic Task
software	Software Installation and Removal
storage_devices	Storage Device Operations and Definitions
system_setup	System Name, Date/Time and Initial Password Setup
users	User Login and Group Administration

Sollten Sie zusätzlich zu den mit SINIX ausgelieferten Softwarepaketen noch andere auf Ihr System einspielen, so erscheint ein weiterer Eintrag im Menü, über den diese aufgelistet werden können.

applications	Administration for Available Applications
--------------	---

Der restliche Abschnitt beschreibt alle Menüs, die im Hauptmenü angezeigt werden.

- Backup Scheduling, Setup and Control

Dieses Menü wird nicht unterstützt.

- File System Creation, Checking and Mounting

Dieses Menü bietet Funktionen aus dem Bereich Dateisystemverwaltung. Die Funktionen umfassen Aufgaben wie das Überprüfen bestimmter Dateisysteme auf Fehler und deren Reparatur, die Überwachung der Plattenbelegung aller Dateisysteme, das Auffinden von Dateien nach Alter oder Größe, das Auflisten aller aktuell am System montierten Dateisysteme, die Erzeugung eines neuen Dateisystems sowie Montieren und Demontieren von Dateisystemen.

- check (Check a File System)

Diese Funktion erlaubt es, ein Dateisystem auf Fehler zu überprüfen und entweder im Dialog oder automatisch zu reparieren.

- defaults (Manage Defaults)

Diese Funktion erlaubt es, Einträge in der Standarddateisystemtabelle /etc/vfstab hinzufügen, zu ändern, zu entfernen oder anzuzeigen.

- diskuse (Display Disk Usage)

Diese Funktion bestimmt, zu wieviel Prozent die Festplatten aktuell durch Dateien belegt sind. Die Ausgabe erfolgt in Form einer nach den Namen der Dateisysteme geordneten Liste.

- display (Display Installed Types)

Diese Funktion gibt eine Liste aller am System verfügbaren Dateisystem-Typen aus.

- fileage (List Files by Age)

Diese Funktion erstellt eine Liste älterer Dateien in dem angegebenen Verzeichnis. Falls kein Alter spezifiziert ist, werden alle Dateien aufgelistet, die älter als 90 Tage sind.

- filesize (List Files by Size)

Diese Funktion gibt die Namen der größten Dateien im angegebenen Verzeichnis aus. Sofern keine bestimmte Anzahl aufzulistender Dateien angegeben ist, werden die Namen der zehn größten Dateien ausgegeben.

- identify (Identify File System Type)

Diese Funktion versucht, den Typ jedes nicht montierten Dateisystems zu bestimmen, ohne dabei Schaden an den Daten oder am Datenträger anzurichten.

- list (List Mounted File Systems)

Diese Funktion listet alle am Rechner montierten Dateisysteme auf.

- make (Create a File System)

Diese Funktion ermöglicht es, ein neues Dateisystem oder eine Slice auf einem wechselbaren Datenträger einzurichten, welches dann Daten aufnehmen kann, die nicht auf der Festplatte gespeichert bleiben sollen. Wird dieses Dateisystem montiert, hat es die gleichen Eigenschaften wie wenn es auf der Festplatte wäre.

- mount (Mount a File System)

Diese Funktion ermöglicht es, ein Dateisystem einzuhängen, das sich auf einem wechselbaren Datenträger befindet, und es so allen Benutzern am System verfügbar zu machen. Auch Platten-Dateisysteme können eingehängt werden. Ein Dateisystem kann mit der Funktion `unmount` wieder ausgehängt werden.

WARNUNG: Der Datenträger darf nicht entfernt werden, solange das Dateisystem eingehängt ist.

- unmount (Unmount a File System)

Diese Funktion erlaubt es, ein Dateisystem auszuhängen und somit danach den Datenträger zu entfernen, auf dem es sich befindet. Die Dateisysteme `/` und `/usr` können nicht ausgehängt werden. Auch Platten-Dateisysteme können ausgehängt werden. Sobald ein Dateisystem ausgehängt ist, kann sein Datenträger entnommen werden.

- Machine Configuration, Display and Shutdown

Dieses Menü bietet Funktionen, wie das Abschalten des Rechners, das Wieder-Hochfahren des Rechners oder das Wechseln in den Firmware-Modus.

- configuration (System Configuration Display)

Diese Funktion ermöglicht es, die aktuelle Konfiguration des Rechners zu überprüfen. Die Funktion wird zur Zeit nicht unterstützt.

- reboot (Stops All Running Programs and Reboots Machine)

Diese Funktion ermöglicht es, den Rechner neu hochzufahren, nachdem alle noch laufenden Programme gestoppt, alle offenen Dateien geschlossen und wichtige Daten (wie z.B. über die Verzeichnisstruktur) auf Platte gesichert worden sind. Damit können bestimmte Störungen am System beseitigt werden, wie zum Beispiel Prozesse, die sich nicht beenden lassen.

- shutdown (Stops All Running Programs and Turns Off Machine)

Diese Funktion ermöglicht es, alle noch laufenden Programme zu stoppen, alle offenen Dateien zu schließen, wichtige Daten (wie z.B. über die Verzeichnisstruktur) auf Platte zu sichern und danach das System herunterzufahren.

- whos on (Displays List of Users Logged onto Machine)

Diese Funktion gibt für alle aktuell am Rechner arbeitenden Benutzer die Benutzernummer (UID), die Gerätenummer des Terminals und den Zeitpunkt der Anmeldung im System aus.

- Network Services Administration

Dieses Menü stellt Untermenüs für die Netzwerkverwaltung bereit.

- `basic_networking` (Basic Networking Utilities Management)
Dieses Untermenü ermöglicht die Erstellung von Verwaltungsdateien für UUCP-Dienstprogramme.
- `selection` (Network Selection Management)
Dieses Untermenü ermöglicht die Erstellung von Verwaltungsdateien für die Netzwerkauswahl. Damit kann das Transportprotokoll dynamisch gewählt werden.
- `name_to_address` (Machine and Service Address Management)
Dieses Untermenü ermöglicht es, für die auf dem Rechner existierenden Protokolle Rechneradressen zu definieren und Information über die Anschlußdienste bereitzustellen.
- `remote_files` (Distributed File System Management)
Dieses Menü dient zur Verwaltung von fernen Dateisystemen.
- `Port Access Services and Monitors`
Dieses Menü bietet Verwaltungsfunktionen für Port Monitore.
- `port_monitors` (Port Monitor Management)
Dieses Untermenü dient der Steuerung von Port Monitoren innerhalb des Wartungspakets (Service Access Facility). Im besonderen kann man damit Port Monitoren hinzufügen und entfernen, sie an- und abschalten, auflisten und verändern und sie starten und stoppen.
- `port_services` (Port Service Management)
Dieses Untermenü dient der Steuerung von Anschlußdiensten, die von Port Monitoren bereitgestellt werden. Im besonderen kann man damit Anschlußdienste hinzufügen und entfernen, sie an- und abschalten und sie auflisten und verändern.
- `quick-terminal`
(Schnelles Einrichten eines Terminals.) Damit kann ein Benutzer in einfacher Weise ein Terminal und dessen Geschwindigkeit einrichten.
- `tty_settings` (Terminal Line Setting Management)
Dieses Untermenü dient zum Festlegen der Eigenschaften von Terminalleitungen. Im besonderen ermöglicht es, neue Terminaleigenschaften und Suchsequenzen festzulegen, sie (am Bildschirm) auszugeben und auch wieder zu löschen. Um die Eigenschaften eines existierenden Terminals zu ändern, wird der entsprechende Eintrag gelöscht und dann zusammen mit den Änderungen neu definiert.
- `Printer Configuration and Services`
Dieses Menü dient der Druckerverwaltung und der Verwaltung von Dienstprogrammen zum Drucken, die den Benutzern über den Druckerdienst LP zur Verfügung gestellt werden können. Im einzelnen erleichtert dieses

Menü die folgenden Aufgaben: den Druckerdienst LP einrichten und steuern; seinen Zustand überprüfen und ihn bei Bedarf stoppen und wieder starten; neue Drucker am System einrichten und bereits existierende neu konfigurieren; Formulare und Filterprogramme hinzufügen, ändern und bereitstellen; Druckanforderungen der Benutzer überwachen.

- classes (Manage Classes of Related Printers)

Dieses Untermenü ermöglicht es, neue Druckerklassen einzurichten und die aktuellen Klassen aufzulisten.

- filters (Manage Filters for Special Processing)

Dieses Untermenü dient der Verwaltung von Filterprogrammen für spezielle Dateibearbeitung.

- forms (Manage Pre-Printed Forms)

Dieses Untermenü dient der Verwaltung vordefinierter Formulare.

- operations (Perform Daily Printer Service Operations)

Dieses Untermenü ermöglicht es, tägliche Aufgaben für die Druckerverwaltung durchzuführen, wie z.B. Drucker zu aktivieren, die Druckerverwaltung zu starten und Formulare bereitzustellen.

- printers (Configure Printers for the Printer Service)

Dieses Untermenü dient der Konfigurierung von Druckern für den LP Druckerdienst.

- priorities (Assign Print Queue Priorities to Users)

Dieses Untermenü ermöglicht es, Druckaufträgen in der Warteschlange Prioritäten zuzuweisen.

- requests (Manage Active Print Requests)

Dieses Untermenü ermöglicht es, anstehende Druckaufträge zurückzuhalten oder freizugeben, auf andere Zielgeräte umzulenken oder sie zu löschen.

- status (Display Status of Printer Service)

Dieses Untermenü dient der Ausgabe des aktuellen Zustands des LP Druckerdienstes.

- systems (Configure Connections to Remote Systems)

Dieses Untermenü ermöglicht es, die Verbindungen zwischen Ihrem LP Druckerdienst und irgendeinem anderen LP Druckerdienst zu konfigurieren.

- Restore From Backup Data

Dieses Menü wird nicht unterstützt.

- Schedule Automatic Task

Mit diesem Menü können Benutzer Aufträge, die zu bestimmten Zeiten gestartet werden sollen, in die `cron`-Datei eintragen.

- add

Damit wird ein weiterer Auftrag in die `cron`-Datei eingetragen.

- change

Ändern eines in der `cron`-Datei eingetragenen Auftrags.

- delete

Löschen eines Auftrags aus der `cron`-Datei.

- display

Ausgabe der in der `cron`-Datei stehenden Aufträge.

- Software Installation and Removal

Dieses Menü bietet Funktionen für die Installation von Softwarepaketen, deren Löschung und die Verwaltung von zugehörigen Informationen. Dazu gehört auch die Fähigkeit, die Korrektheit der Installation zu überprüfen. Außerdem können damit Voreinstellungen für die Installation gesetzt und der Dialog bei der Installation bestimmter Softwarepakete abgespeichert werden. Es kann auch Software eingespielt werden, ohne daß sie installiert wird, und es gibt die Möglichkeit, eine komplette Liste der installierten Software auszugeben.

- check (Checks Accuracy of Installation)

Diese Funktion erlaubt Konsistenzprüfungen installierter Softwarepakete und entsprechende Korrekturen. Weiter kann das System auf versteckte Dateien geprüft werden und die Inhalte von Dateien überprüft werden, die anscheinend verändert wurden.

- defaults (Sets Installation Defaults)

Mit dieser Funktion kann man im voraus festlegen, wie das System auf Probleme bei der Installation reagieren soll.

- install (Installs Software Packages)

Mit dieser Funktion kann man Softwarepakete installieren, die sich auf Band, Festplatte oder Diskette befinden, und die Art und Weise festlegen, wie das System auf Probleme bei der Installation reagieren soll.

- interact (Stores Interactions with Package)

Diese Funktion ermöglicht es, Dialoge während der Installation abzuspeichern, um diese bei späteren vollautomatischen Installationen wieder zu verwenden.

- `list` (Displays Information about Packages)

Diese Funktion informiert über die am System installierten Softwarepakete: wie sie bezeichnet sind, wo sie abgelegt sind und in welche Kategorie sie gehören.

- `read_in` (Stores Packages Without Installing)

Mit dieser Funktion kann man Softwarepakete einlesen, ohne sie zu installieren.

- `remove` (Removes Packages)

Mit dieser Funktion kann man ein installiertes Softwarepaket aus dem System löschen.

- Storage Device Operations and Definitions

Dieses Menü macht die Beschreibung der Einstellung und der Synonyme von Geräten verfügbar und ermöglicht die Zuweisung von Gerätegruppen.

- `descriptions` (Device Alias and Attribute Management)

Dieses Untermenü dient dazu, die Einstellung und Beschreibung von Geräten zu ändern, zu löschen, hinzuzufügen und auszugeben. Es erlaubt auch den Zugang zu Dienstprogrammen für die Reservierung von Geräten.

- `groups` (Device Group Management)

Mit diesem Untermenü kann man Gerätegruppen und Listen der Gruppenzugehörigkeit ausgeben lassen und verwalten.

- `add` (Add Storage Device)

Diese Funktion dient dazu, die Software-Konfiguration eines Geräts im Dialog auszuführen. Das Gerät kann eine zweite Festplatte, ein SCSI-Gerät oder eine andere Hardware sein, die eine Software-Konfiguration erfordert. Es ist sowohl das Neukonfigurieren eines leeren Geräts als auch das Rekonfigurieren eines nicht-leeren Geräts möglich, das bereits vorhanden ist. Die Funktion verwendet Aliase aus der Datei `/etc/device.tab`, für die das Attribut `addcmd` definiert sein muß. Das Attribut muß den gültigen Pfadnamen eines Kommandos enthalten, das die Konfiguration ausführt.

- `copy` (Make Duplicate Copies of Storage Volumes)

Mit dieser Funktion erstellen Sie eine doppelte Kopie des Inhalts eines auswechselbaren Speichermediums.

- `display` (Display Information About Storage Devices)

Diese Funktion gibt Informationen zur Partitionierung der Festplatte(n) aus.

- `erase` (Erase the Contents of Storage Volumes)

Diese Funktion löscht den Inhalt eines auswechselbaren Speichermediums. Die Funktion dient hauptsächlich dazu, Daten allgemein unlesbar zu machen. Die gelöschten Daten sind unwiderruflich verloren.

- format (Formats Removable Volumes)

Mit dieser Funktion formatieren Sie ein auswechselbares Speichermedium. Auf ein neues Speichermedium können Daten erst nach dem Formatieren geschrieben werden. Beim Formatieren werden alle Daten gelöscht, die sich bereits auf dem Speichermedium befanden.

- remove (Remove Storage Device)

Mit dieser Funktion entfernen Sie die Software zu Speichergeräten, mit allen dazugehörenden Einträgen und Informationen. Auf das Gerät kann nicht mehr zugegriffen werden. Sie können das Gerät physikalisch entfernen, nachdem die Funktion ausgeführt wurde und Sie das System heruntergefahren haben.

- System Name, Date Time and Initial Password Setup

Dieses Menü dient dem Einrichten des Rechners. Es umfaßt Funktionen zum Setzen von Systemzeit und -datum und zur Festlegung des Rechnernamens. Weiter dient es der Ersteinrichtung des Rechners und der Zuweisung von Paßwörtern an Kennungen für die Systemverwaltung.

- datetime (System Date and Time Information)

Mit dieser Funktion wird am Rechner Datum, Zeit und Zeitzone eingestellt und ob Sommerzeit gilt (Daylight Savings Time: DST). Normalerweise wird das genau einmal bei der Ersteinrichtung des Rechners durchgeführt. Gilt die Sommerzeit (DST), schaltet der Rechner im Herbst automatisch zurück zur Normalzeit und im Frühling auf Sommerzeit. Um sicherzustellen, daß alle Zeitangaben richtig rückgemeldet werden, muß der Rechner aus- und wieder eingeschaltet werden. Beim nächsten Login eines Benutzers stimmen die meisten Zeitangaben.

- nodename (System Name and Network Node Name of the Machine)

Mit dieser Funktion kann man den Knoten- und Rechnernamen dieses Rechners ändern. Diese Namen dienen verschiedenen Kommunikationsnetzwerken zur Identifizierung dieses Rechners.

- password (Assigns Administrative Login Passwords)

Diese Funktion dient der Zuweisung von Paßwörtern an Kennungen für die Systemverwaltung.

- setup (Sets up System Information for First Time)

Mit dieser Funktion ist es möglich, die erste Kennung einzurichten, an Kennungen für die Systemverwaltung Paßwörter zu vergeben und die entsprechende Zeitzone einzustellen.

- User Login and Group Administration

Dieses Menü dient der Benutzerverwaltung am Rechner. Es umfaßt die entsprechenden Funktionen, um dem System neue Benutzer oder Gruppen bekannt zu machen und existierende zu ändern oder zu löschen. Die

Gruppenzugehörigkeit von Benutzern kann so organisiert werden, daß der Zugriff auf Dateien gleicher Gruppenmitglieder erlaubt ist, nicht jedoch der Zugriff auf Dateien fremder Gruppen. Außerdem können mit diesem Menü Voreinstellungen für die Neueinrichtung künftiger Benutzer festgelegt und neue oder zusätzliche Informationen zu den Benutzern in der Paßwortdatei eingetragen werden.

••add (Adds Users or Groups)

Mit dieser Funktion werden neue Benutzer oder Gruppen am Rechner definiert.

••defaults (Defines Defaults for Adding Users)

Mit dieser Funktion kann die Voreinstellung für die Einrichtung neuer Benutzer mittels des `add user` Aufrufs verändert werden. Die Änderung von Voreinstellungswerten hat keinerlei Auswirkung auf existierende Benutzer, sondern betrifft ausschließlich neue.

••list (Lists Users or Groups)

Mit dieser Funktion können die Merkmale der Benutzer und Gruppen am System untersucht werden.

••modify (Modifies Attributes of Users or Groups)

Mit dieser Funktion können am System Benutzer- oder Gruppeneinträge geändert werden.

••password ((Re-)defines User Password Information)

Mit dieser Funktion kann für Benutzer am System ein Paßwort vergeben oder geändert werden.

••remove (Removes Users or Groups)

Mit dieser Funktion können am System Benutzerberechtigungen gelöscht werden.

EXIT-CODES

Die Exit-Codes von `sysadm` haben folgende Bedeutung:

- 0 normales Ende
- 2 Kommandoaufruf fehlerhaft; es wird eine Meldung zum korrekten Aufruf von `sysadm` ausgegeben
- 4 den für ein Menü oder eine Funktion angegebenen Namen gibt es nicht
- 5 der für ein Menü angegebene Name ist lediglich ein Platzhalter ohne funktionalen Hintergrund und daher nicht verwendbar
- 7 das Kommando `sysadm` kann `fmli` nicht aufrufen, ohne das es nicht korrekt ablaufen kann; das kann daran liegen, daß das FMLI-Paket zerstört oder möglicherweise nicht installiert ist

BEISPIEL

```
sysadm nodename
```

sysadm (1M)

(MX300)

sysadm (1M)

SIEHE AUCH

backup(1M), bkexcept(1M), bkhistory(1M), bkoper(1M), bkreg(1M),
bkstatus(1M), checkfsys(1M), delsysadm(1M), edsysadm(1M), groupadd(1M),
groupdel(1M), groupmod(1M), makefsys(1M), mountfsys(1M), password(1M),
restore(1M), rsnotify(1M), rsoper(1M), rsstatus(1M), setup(1M),
urestore(1M), ursstatus(1M), useradd(1M), userdel(1M), usermod(1M).

BEZEICHNUNG

sysadm – menügeführte Systemverwaltung

ÜBERSICHT

sysadm [*menü* | *funktion*]

BESCHREIBUNG

Wird sysadm ohne Argumente aufgerufen, bietet es eine Reihe von Menüs zur Erleichterung von Verwaltungstätigkeiten an. Falls beim Aufruf ein Menü oder eine Funktion (task) angegeben ist, geschieht folgendes: Ist das Menü oder die Funktion eindeutig spezifiziert, wird es bzw. sie sofort am Bildschirm ausgegeben, andernfalls erscheint eine Auswahlliste.

Die Kennung sysadm kann mit einem Paßwort geschützt werden. Um ein Paßwort einzurichten oder zu ändern, wird im Menü system_setup die Funktion password angestoßen. Zum Ändern eines vergebenen Paßworts dient auch das Kommando password.

Das Hauptmenü von sysadm enthält die folgenden Untermenüs:

backup_service	Backup Scheduling, Setup and Control
file_systems	File System Creation, Checking and Mounting
machine	Machine Configuration, Display and Shutdown
network_services	Network Services Administration
ports	Port Access Services and Monitors
printers	Printer Configuration and Services
restore_service	Restore From Backup Data
schedule_task	Schedule Automatic Task
software	Software Installation and Removal
storage_devices	Storage Device Operations and Definitions
system_setup	System Name, Date/Time and Initial Password Setup
users	User Login and Group Administration

Sollten Sie zusätzlich zu den mit SINIX ausgelieferten Softwarepaketen noch andere auf Ihr System einspielen, so erscheint ein weiterer Eintrag im Menü, über den diese aufgelistet werden können.

applications	Administration for Available Applications
--------------	---

Es ist möglich, daß Softwarepakete, die nicht der aktuellen Version von SINIX angepaßt sind, sysadm-Menüs von früheren Versionen erfordern. Solche Menüs werden zusammen mit diesen Softwarepaketen installiert. In diesem Fall enthält das Hauptmenü noch den Eintrag

preSVR4	Peripherals Setup
---------	-------------------

Der restliche Abschnitt beschreibt alle Menüs, die im Hauptmenü angezeigt werden.

- Backup Scheduling, Setup and Control

Dieses Menü wird nicht unterstützt.

- File System Creation, Checking and Mounting

Dieses Menü bietet Funktionen aus dem Bereich Dateisystemverwaltung. Die Funktionen umfassen Aufgaben wie das Überprüfen bestimmter Dateisysteme auf Fehler und deren Reparatur, die Überwachung der Plattenbelegung aller Dateisysteme, das Auffinden von Dateien nach Alter oder Größe, das Auflisten aller aktuell am System montierten Dateisysteme, die Erzeugung eines neuen Dateisystems sowie Montieren und Demontieren von Dateisystemen.

- check (Check a File System)

Diese Funktion erlaubt es, ein Dateisystem auf Fehler zu überprüfen und entweder im Dialog oder automatisch zu reparieren.

- defaults (Manage Defaults)

Diese Funktion erlaubt es, Einträge in der Standarddateisystemtabelle `/etc/vfstab` hinzufügen, zu ändern, zu entfernen oder anzuzeigen.

- diskuse (Display Disk Usage)

Diese Funktion bestimmt, zu wieviel Prozent die Festplatten aktuell durch Dateien belegt sind. Die Ausgabe erfolgt in Form einer nach den Namen der Dateisysteme geordneten Liste.

- display (Display Installed Types)

Diese Funktion gibt eine Liste aller am System verfügbaren Dateisystem-Typen aus.

- fileage (List Files by Age)

Diese Funktion erstellt eine Liste älterer Dateien in dem angegebenen Verzeichnis. Falls kein Alter spezifiziert ist, werden alle Dateien aufgelistet, die älter als 90 Tage sind.

- filesize (List Files by Size)

Diese Funktion gibt die Namen der größten Dateien im angegebenen Verzeichnis aus. Sofern keine bestimmte Anzahl aufzulistender Dateien angegeben ist, werden die Namen der zehn größten Dateien ausgegeben.

- identify (Identify File System Type)

Diese Funktion versucht, den Typ jedes nicht montierten Dateisystems zu bestimmen, ohne dabei Schaden an den Daten oder am Datenträger anzurichten.

- list (List Mounted File Systems)

Diese Funktion listet alle am Rechner montierten Dateisysteme auf.

- make (Create a File System)

Diese Funktion ermöglicht es, ein neues Dateisystem oder eine Slice auf einem wechselbaren Datenträger einzurichten, welches dann Daten aufnehmen kann, die nicht auf der Festplatte gespeichert bleiben sollen. Wird dieses Dateisystem montiert, hat es die gleichen Eigenschaften wie wenn es auf der Festplatte wäre.

- mount (Mount a File System)

Diese Funktion ermöglicht es, ein Dateisystem einzuhängen, das sich auf einem wechselbaren Datenträger befindet, und es so allen Benutzern am System verfügbar zu machen. Auch Platten-Dateisysteme können eingehängt werden. Ein Dateisystem kann mit der Funktion `umount` wieder ausgehängt werden.

WARNUNG: Der Datenträger darf nicht entfernt werden, solange das Dateisystem eingehängt ist.

- umount (Unmount a File System)

Diese Funktion erlaubt es, ein Dateisystem auszuhängen und somit danach den Datenträger zu entfernen, auf dem es sich befindet. Die Dateisysteme `/` und `/usr` können nicht ausgehängt werden. Auch Platten-Dateisysteme können ausgehängt werden. Sobald ein Dateisystem ausgehängt ist, kann sein Datenträger entnommen werden.

- Machine Configuration, Display and Shutdown

Dieses Menü bietet Funktionen, wie das Abschalten des Rechners, das Wieder-Hochfahren des Rechners oder das Wechseln in den Firmware-Modus.

- configuration (System Configuration Display)

Diese Funktion ermöglicht es, die aktuelle Konfiguration des Rechners zu überprüfen.

- reboot (Stops All Running Programs and Reboots Machine)

Diese Funktion ermöglicht es, den Rechner neu hochzufahren, nachdem alle noch laufenden Programme gestoppt, alle offenen Dateien geschlossen und wichtige Daten (wie z.B. über die Verzeichnisstruktur) auf Platte gesichert worden sind. Damit können bestimmte Störungen am System beseitigt werden, wie zum Beispiel Prozesse, die sich nicht beenden lassen.

- shutdown (Stops All Running Programs and Turns Off Machine)

Diese Funktion ermöglicht es, alle noch laufenden Programme zu stoppen, alle offenen Dateien zu schließen, wichtige Daten (wie z.B. über die Verzeichnisstruktur) auf Platte zu sichern und danach das System herunterzufahren.

- whos on (Displays List of Users Logged onto Machine)

Diese Funktion gibt für alle aktuell am Rechner arbeitenden Benutzer die Benutzernummer (UID), die Gerätenummer des Terminals und den Zeitpunkt der Anmeldung im System aus.

- Network Services Administration

Dieses Menü stellt Untermenüs für die Netzwerkverwaltung bereit.

- basic_networking (Basic Networking Utilities Management)

Dieses Untermenü ermöglicht die Erstellung von Verwaltungsdateien für UUCP-Dienstprogramme.

- selection (Network Selection Management)

Dieses Untermenü ermöglicht die Erstellung von Verwaltungsdateien für die Netzwerkauswahl. Damit kann das Transportprotokoll dynamisch gewählt werden.

- name_to_address (Machine and Service Address Management)

Dieses Untermenü ermöglicht es, für die auf dem Rechner existierenden Protokolle Rechneradressen zu definieren und Information über die Anschlußdienste bereitzustellen.

- Port Access Services and Monitors

Dieses Menü bietet Verwaltungsfunktionen für Port Monitore.

- port_monitors (Port Monitor Management)

Dieses Untermenü dient der Steuerung von Port Monitoren innerhalb des Wartungspakets (Service Access Facility). Im besonderen kann man damit Port Monitoren hinzufügen und entfernen, sie an- und abschalten, auflisten und verändern und sie starten und stoppen.

- port_services (Port Service Management)

Dieses Untermenü dient der Steuerung von Anschlußdiensten, die von Port Monitoren bereitgestellt werden. Im besonderen kann man damit Anschlußdienste hinzufügen und entfernen, sie an- und abschalten und sie auflisten und verändern.

- quick-terminal

(Schnelles Einrichten eines Terminals.) Damit kann ein Benutzer in einfacher Weise ein Terminal und dessen Geschwindigkeit einrichten.

- tty_settings (Terminal Line Setting Management)

Dieses Untermenü dient zum Festlegen der Eigenschaften von Terminalleitungen. Im besonderen ermöglicht es, neue Terminaleigenschaften und Suchsequenzen festzulegen, sie (am Bildschirm) auszugeben und auch wieder zu löschen. Um die Eigenschaften eines existierenden Terminals zu ändern, wird der entsprechende Eintrag gelöscht und dann zusammen mit den Änderungen neu definiert.

- Peripheral Setup

Dieses Untermenü ermöglicht es, Peripheriegeräte einzurichten, die von Vorgängerversionen (pre-SVR4.0) unterstützt wurden.

- Printer Configuration and Services

Dieses Menü dient der Druckerverwaltung und der Verwaltung von Dienstprogrammen zum Drucken, die den Benutzern über den Druckerdienst LP zur Verfügung gestellt werden können. Im einzelnen erleichtert dieses Menü die folgenden Aufgaben: den Druckerdienst LP einrichten und steuern; seinen Zustand überprüfen und ihn bei Bedarf stoppen und wieder starten; neue Drucker am System einrichten und bereits existierende neu konfigurieren; Formulare und Filterprogramme hinzufügen, ändern und bereitstellen; Druckanforderungen der Benutzer überwachen.

- classes (Manage Classes of Related Printers)

Dieses Untermenü ermöglicht es, neue Druckerklassen einzurichten und die aktuellen Klassen aufzulisten.

- filters (Manage Filters for Special Processing)

Dieses Untermenü dient der Verwaltung von Filterprogrammen für spezielle Dateibearbeitung.

- forms (Manage Pre-Printed Forms)

Dieses Untermenü dient der Verwaltung vordefinierter Formulare.

- operations (Perform Daily Printer Service Operations)

Dieses Untermenü ermöglicht es, tägliche Aufgaben für die Druckerverwaltung durchzuführen, wie z.B. Drucker zu aktivieren, die Druckerverwaltung zu starten und Formulare bereitzustellen.

- printers (Configure Printers for the Printer Service)

Dieses Untermenü dient der Konfigurierung von Druckern für den LP Druckerdienst.

- priorities (Assign Print Queue Priorities to Users)

Dieses Untermenü ermöglicht es, Druckaufträgen in der Warteschlange Prioritäten zuzuweisen.

- requests (Manage Active Print Requests)

Dieses Untermenü ermöglicht es, anstehende Druckaufträge zurückzuhalten oder freizugeben, auf andere Zielgeräte umzulenken oder sie zu löschen.

- status (Display Status of Printer Service)

Dieses Untermenü dient der Ausgabe des aktuellen Zustands des LP Druckerdienstes.

- systems (Configure Connections to Remote Systems)

Dieses Untermenü ermöglicht es, die Verbindungen zwischen Ihrem LP Druckerdienst und irgendeinem anderen LP Druckerdienst zu konfigurieren.

- Restore From Backup Data

Dieses Menü wird nicht unterstützt.

- Schedule Automatic Task

Mit diesem Menü können Benutzer Aufträge, die zu bestimmten Zeiten gestartet werden sollen, in die `cron`-Datei eintragen.

- add

Damit wird ein weiterer Auftrag in die `cron`-Datei eingetragen.

- change

Ändern eines in der `cron`-Datei eingetragenen Auftrags.

- delete

Löschen eines Auftrags aus der `cron`-Datei.

- display

Ausgabe der in der `cron`-Datei stehenden Aufträge.

- Software Installation and Removal

Dieses Menü bietet Funktionen für die Installation von Softwarepaketen, deren Löschung und die Verwaltung von zugehörigen Informationen. Dazu gehört auch die Fähigkeit, die Korrektheit der Installation zu überprüfen. Außerdem können damit Voreinstellungen für die Installation gesetzt und der Dialog bei der Installation bestimmter Softwarepakete abgespeichert werden. Es kann auch Software eingespielt werden, ohne daß sie installiert wird, und es gibt die Möglichkeit, eine komplette Liste der installierten Software auszugeben.

- check (Checks Accuracy of Installation)

Diese Funktion erlaubt Konsistenzprüfungen installierter Softwarepakete und entsprechende Korrekturen. Weiter kann das System auf versteckte Dateien geprüft werden und die Inhalte von Dateien überprüft werden, die anscheinend verändert wurden.

- defaults (Sets Installation Defaults)

Mit dieser Funktion kann man im voraus festlegen, wie das System auf Probleme bei der Installation reagieren soll.

- install (Installs Software Packages)

Mit dieser Funktion kann man Softwarepakete installieren, die sich auf Band, Festplatte oder Diskette befinden, und die Art und Weise festlegen, wie das System auf Probleme bei der Installation reagieren soll.

- interact (Stores Interactions with Package)

Diese Funktion ermöglicht es, Dialoge während der Installation abzuspeichern, um diese bei späteren vollautomatischen Installationen wieder zu verwenden.

- list (Displays Information about Packages)

Diese Funktion informiert über die am System installierten Softwarepakete: wie sie bezeichnet sind, wo sie abgelegt sind und in welche Kategorie sie gehören.

- read_in (Stores Packages Without Installing)

Mit dieser Funktion kann man Softwarepakete einlesen, ohne sie zu installieren.

- remove (Removes Packages)

Mit dieser Funktion kann man ein installiertes Softwarepaket aus dem System löschen.

- Storage Device Operations and Definitions

Dieses Menü macht die Beschreibung der Einstellung und der Synonyme von Geräten verfügbar und ermöglicht die Zuweisung von Gerätegruppen.

- descriptions (Device Alias and Attribute Management)

Dieses Untermenü dient dazu, die Einstellung und Beschreibung von Geräten zu ändern, zu löschen, hinzuzufügen und auszugeben. Es erlaubt auch den Zugang zu Dienstprogrammen für die Reservierung von Geräten.

- groups (Device Group Management)

Mit diesem Untermenü kann man Gerätegruppen und Listen der Gruppenzugehörigkeit ausgeben lassen und verwalten.

- System Name, Date Time and Initial Password Setup

Dieses Menü dient dem Einrichten des Rechners. Es umfaßt Funktionen zum Setzen von Systemzeit und -datum und zur Festlegung des Rechnernamens. Weiter dient es der Ersteinrichtung des Rechners und der Zuweisung von Paßwörtern an Kennungen für die Systemverwaltung.

- `datetime` (System Date and Time Information)

Mit dieser Funktion wird am Rechner Datum, Zeit und Zeitzone eingestellt und ob Sommerzeit gilt (Daylight Savings Time: DST). Normalerweise wird das genau einmal bei der Ersteinrichtung des Rechners durchgeführt. Gilt die Sommerzeit (DST), schaltet der Rechner im Herbst automatisch zurück zur Normalzeit und im Frühling auf Sommerzeit. Um sicherzustellen, daß alle Zeitangaben richtig rückgemeldet werden, muß der Rechner aus- und wieder eingeschaltet werden. Beim nächsten Login eines Benutzers stimmen die meisten Zeitangaben.

- `nodename` (System Name and Network Node Name of the Machine)

Mit dieser Funktion kann man den Knoten- und Rechnernamen dieses Rechners ändern. Diese Namen dienen verschiedenen Kommunikationsnetzwerken zur Identifizierung dieses Rechners.

- `password` (Assigns Administrative Login Passwords)

Diese Funktion dient der Zuweisung von Paßwörtern an Kennungen für die Systemverwaltung.

- `setup` (Sets up System Information for First Time)

Mit dieser Funktion ist es möglich, die erste Kennung einzurichten, an Kennungen für die Systemverwaltung Paßwörter zu vergeben und die entsprechende Zeitzone einzustellen.

- `User Login and Group Administration`

Dieses Menü dient der Benutzerverwaltung am Rechner. Es umfaßt die entsprechenden Funktionen, um dem System neue Benutzer oder Gruppen bekannt zu machen und existierende zu ändern oder zu löschen. Die Gruppenzugehörigkeit von Benutzern kann so organisiert werden, daß der Zugriff auf Dateien gleicher Gruppenmitglieder erlaubt ist, nicht jedoch der Zugriff auf Dateien fremder Gruppen. Außerdem können mit diesem Menü Voreinstellungen für die Neueinrichtung künftiger Benutzer festgelegt und neue oder zusätzliche Informationen zu den Benutzern in der Paßwortdatei eingetragen werden.

- `add` (Adds Users or Groups)

Mit dieser Funktion werden neue Benutzer oder Gruppen am Rechner definiert.

- `defaults` (Defines Defaults for Adding Users)

Mit dieser Funktion kann die Voreinstellung für die Einrichtung neuer Benutzer mittels des `add user` Aufrufs verändert werden. Die Änderung von Voreinstellungswerten hat keinerlei Auswirkung auf existierende Benutzer, sondern betrifft ausschließlich neue.

- `list` (Lists Users or Groups)

Mit dieser Funktion können die Merkmale der Benutzer und Gruppen am System untersucht werden.

- modify (Modifies Attributes of Users or Groups)

Mit dieser Funktion können am System Benutzer- oder Gruppeneinträge geändert werden.

- password((Re-)defines User Password Information)

Mit dieser Funktion kann für Benutzer am System ein Paßwort vergeben oder geändert werden.

- remove (Removes Users or Groups)

Mit dieser Funktion können am System Benutzerberechtigungen gelöscht werden.

EXIT-CODES

Die Exit-Codes von `sysadm` haben folgende Bedeutung:

- 0 normales Ende
- 2 Kommandoaufruf fehlerhaft; es wird eine Meldung zum korrekten Aufruf von `sysadm` ausgegeben
- 4 den für ein Menü oder eine Funktion angegebenen Namen gibt es nicht
- 5 der für ein Menü angegebene Name ist lediglich ein Platzhalter ohne funktionalen Hintergrund und daher nicht verwendbar
- 7 das Kommando `sysadm` kann `fmli` nicht aufrufen, ohne das es nicht korrekt ablaufen kann; das kann daran liegen, daß das FMLI-Paket zerstört oder möglicherweise nicht installiert ist

BEISPIEL

```
sysadm nodename
```

SIEHE AUCH

`backup(1M)`, `bkexcept(1M)`, `bkhhistory(1M)`, `bkoper(1M)`, `bkreg(1M)`, `bkstatus(1M)`, `checkfsys(1M)`, `delsysadm(1M)`, `edsysadm(1M)`, `groupadd(1M)`, `groupdel(1M)`, `groupmod(1M)`, `makefsys(1M)`, `mountfsys(1M)`, `password(1M)`, `restore(1M)`, `rsnotify(1M)`, `rsoper(1M)`, `rsstatus(1M)`, `setup(1M)`, `urestore(1M)`, `ursstatus(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`.

BEZEICHNUNG

sysadm – menügeführte Systemverwaltung

ÜBERSICHT

sysadm [*menü* | *funktion*]

BESCHREIBUNG

Wird sysadm ohne Argumente aufgerufen, bietet es eine Reihe von Menüs zur Erleichterung von Verwaltungstätigkeiten an. Falls beim Aufruf ein Menü oder eine Funktion (task) angegeben ist, geschieht folgendes: Ist das Menü oder die Funktion eindeutig spezifiziert, wird es bzw. sie sofort am Bildschirm ausgegeben, andernfalls erscheint eine Auswahlliste.

Die Kennung sysadm kann mit einem Paßwort geschützt werden. Um ein Paßwort einzurichten oder zu ändern, wird im Menü system_setup die Funktion password angestoßen. Zum Ändern eines vergebenen Paßworts dient auch das Kommando password.

Das Hauptmenü von sysadm enthält die folgenden Untermenüs:

backup_service	Backup Scheduling, Setup and Control
file_systems	File System Creation, Checking and Mounting
machine	Machine Configuration, Display and Shutdown
network_services	Network Services Administration
ports	Port Access Services and Monitors
printers	Printer Configuration and Services
restore_service	Restore From Backup Data
schedule_task	Schedule Automatic Task
software	Software Installation and Removal
storage_devices	Storage Device Operations and Definitions
system_setup	System Name, Date/Time and Initial Password Setup
users	User Login and Group Administration

Sollten Sie zusätzlich zu den mit SINIX ausgelieferten Softwarepaketen noch andere auf Ihr System einspielen, so erscheint ein weiterer Eintrag im Menü, über den diese aufgelistet werden können.

applications	Administration for Available Applications
--------------	---

Haben Sie das Package SISadmex installiert, dann enthält das Menü noch zusätzliche Einträge:

performance	Monitoring System Activity and Workload
spoolV3.1	Printer Configuration for SINIX Spool
systemdata	Systemdata Backup and Restore

Der restliche Abschnitt beschreibt alle Menüs, die im Hauptmenü angezeigt werden.

- Backup Scheduling, Setup and Control

Dieses Menü wird nicht unterstützt.

- File System Creation, Checking and Mounting

Dieses Menü bietet Funktionen aus dem Bereich Dateisystemverwaltung. Die Funktionen umfassen Aufgaben wie das Überprüfen bestimmter Dateisysteme auf Fehler und deren Reparatur, die Überwachung der Plattenbelegung aller Dateisysteme, das Auffinden von Dateien nach Alter oder Größe, das Auflisten aller aktuell am System montierten Dateisysteme, die Erzeugung eines neuen Dateisystems sowie Montieren und Demontieren von Dateisystemen.

- check (Check a File System)

Diese Funktion erlaubt es, ein Dateisystem auf Fehler zu überprüfen und entweder im Dialog oder automatisch zu reparieren.

- defaults (Manage Defaults)

Diese Funktion erlaubt es, Einträge in der Standarddateisystemtabelle `/etc/vfstab` hinzufügen, zu ändern, zu entfernen oder anzuzeigen.

- diskuse (Display Disk Usage)

Diese Funktion bestimmt, zu wieviel Prozent die Festplatten aktuell durch Dateien belegt sind. Die Ausgabe erfolgt in Form einer nach den Namen der Dateisysteme geordneten Liste.

- display (Display Installed Types)

Diese Funktion gibt eine Liste aller am System verfügbaren Dateisystem-Typen aus.

- fileage (List Files by Age)

Diese Funktion erstellt eine Liste älterer Dateien in dem angegebenen Verzeichnis. Falls kein Alter spezifiziert ist, werden alle Dateien aufgelistet, die älter als 90 Tage sind.

- filesize (List Files by Size)

Diese Funktion gibt die Namen der größten Dateien im angegebenen Verzeichnis aus. Sofern keine bestimmte Anzahl aufzulistender Dateien angegeben ist, werden die Namen der zehn größten Dateien ausgegeben.

- identify (Identify File System Type)

Diese Funktion versucht, den Typ jedes nicht montierten Dateisystems zu bestimmen, ohne dabei Schaden an den Daten oder am Datenträger anzurichten.

- list (List Mounted File Systems)

Diese Funktion listet alle am Rechner montierten Dateisysteme auf.

- make (Create a File System)

Diese Funktion ermöglicht es, ein neues Dateisystem oder eine Slice auf einem wechselbaren Datenträger einzurichten, welches dann Daten aufnehmen kann, die nicht auf der Festplatte gespeichert bleiben sollen. Wird dieses Dateisystem montiert, hat es die gleichen Eigenschaften wie wenn es auf der Festplatte wäre.

- mount (Mount a File System)

Diese Funktion ermöglicht es, ein Dateisystem einzuhängen, das sich auf einem wechselbaren Datenträger befindet, und es so allen Benutzern am System verfügbar zu machen. Auch Platten-Dateisysteme können eingehängt werden. Ein Dateisystem kann mit der Funktion `unmount` wieder ausgehängt werden.

WARNUNG: Der Datenträger darf nicht entfernt werden, solange das Dateisystem eingehängt ist.

- unmount (Unmount a File System)

Diese Funktion erlaubt es, ein Dateisystem auszuhängen und somit danach den Datenträger zu entfernen, auf dem es sich befindet. Die Dateisysteme `/` und `/usr` können nicht ausgehängt werden. Auch Platten-Dateisysteme können ausgehängt werden. Sobald ein Dateisystem ausgehängt ist, kann sein Datenträger entnommen werden.

- Machine Configuration, Display and Shutdown

Dieses Menü bietet Funktionen, wie das Abschalten des Rechners, das Wieder-Hochfahren des Rechners oder das Wechseln in den Firmware-Modus.

- configuration (System Configuration Display)

Diese Funktion ermöglicht es, die aktuelle Konfiguration des Rechners zu überprüfen. Die Funktion wird zur Zeit nicht unterstützt.

- reboot (Stops All Running Programs and Reboots Machine)

Diese Funktion ermöglicht es, den Rechner neu hochzufahren, nachdem alle noch laufenden Programme gestoppt, alle offenen Dateien geschlossen und wichtige Daten (wie z.B. über die Verzeichnisstruktur) auf Platte gesichert worden sind. Damit können bestimmte Störungen am System beseitigt werden, wie zum Beispiel Prozesse, die sich nicht beenden lassen.

- shutdown (Stops All Running Programs and Turns Off Machine)

Diese Funktion ermöglicht es, alle noch laufenden Programme zu stoppen, alle offenen Dateien zu schließen, wichtige Daten (wie z.B. über die Verzeichnisstruktur) auf Platte zu sichern und danach das System herunterzufahren.

- `whos on` (Displays List of Users Logged onto Machine)

Diese Funktion gibt für alle aktuell am Rechner arbeitenden Benutzer die Benutzernummer (UID), die Gerätenummer des Terminals und den Zeitpunkt der Anmeldung im System aus.

- `Network Services Administration`

Dieses Menü stellt Untermenüs für die Netzwerkverwaltung bereit.

- `basic_networking` (Basic Networking Utilities Management)

Dieses Untermenü ermöglicht die Erstellung von Verwaltungsdateien für UUCP-Dienstprogramme.

- `selection` (Network Selection Management)

Dieses Untermenü ermöglicht die Erstellung von Verwaltungsdateien für die Netzwerkauswahl. Damit kann das Transportprotokoll dynamisch gewählt werden.

- `name_to_address` (Machine and Service Address Management)

Dieses Untermenü ermöglicht es, für die auf dem Rechner existierenden Protokolle Rechneradressen zu definieren und Information über die Anschlußdienste bereitzustellen.

- `remote_files` (Distributed File System Management)

Dieses Menü dient zur Verwaltung von fernen Dateisystemen.

- `Monitoring System Activity and Workload`

Dieses Menü bietet Funktionen an, die aktuelle Systemaktivitäten anzeigen, alle aktiven Prozesse auflisten und Signale zu ausgewählten Prozessen schicken.

- `processes` (Show Activities and Send Signals)

Dieses Untermenü bietet Funktionen an, die alle aktiven Prozesse auflisten und Signale zu ausgewählten Prozessen schicken.

- `sar` (System Activity Report)

Dieses Untermenü bietet Funktionen an, die aktuelle Systemaktivitäten anzeigen.

- `Port Access Services and Monitors`

Dieses Menü bietet Verwaltungsfunktionen für Port Monitore.

- `port_monitors` (Port Monitor Management)

Dieses Untermenü dient der Steuerung von Port Monitoren innerhalb des Wartungspakets (Service Access Facility). Im besonderen kann man damit Port Monitoren hinzufügen und entfernen, sie an- und abschalten, auflisten und verändern und sie starten und stoppen.

- port_services (Port Service Management)

Dieses Untermenü dient der Steuerung von Anschlußdiensten, die von Port Monitoren bereitgestellt werden. Im besonderen kann man damit Anschlußdienste hinzufügen und entfernen, sie an- und abschalten und sie auflisten und verändern.

- quick-terminal

(Schnelles Einrichten eines Terminals.) Damit kann ein Benutzer in einfacher Weise ein Terminal und dessen Geschwindigkeit einrichten.

- tty_settings (Terminal Line Setting Management)

Dieses Untermenü dient zum Festlegen der Eigenschaften von Terminalleitungen. Im besonderen ermöglicht es, neue Terminaleigenschaften und Suchsequenzen festzulegen, sie (am Bildschirm) auszugeben und auch wieder zu löschen. Um die Eigenschaften eines existierenden Terminals zu ändern, wird der entsprechende Eintrag gelöscht und dann zusammen mit den Änderungen neu definiert.

- monitor (Monitor Configuration for Booting)

Dieses Untermenü bietet Funktionen zum Konfigurieren einer Tastatur, einer Maus oder eines Digitalisiertablets. Weiter gibt es Funktionen zum Einstellen der Standardsprache für die sprachabhängigen Dateien der Tastatur und des Grafikbildschirms. Die Einstellungen treten in Kraft, sobald das System das nächste Mal hochgefahren wird.

- x-terminals (X-Terminal Configuration for Booting)

Dieses Untermenü bietet Funktionen zur Verwaltung des Bootens von X-Terminals. X-Terminals laden ihr Betriebssystem durch den bootp-Server entsprechend einer Konfigurationsdatei, die den Namen, die Ethernet-Adresse, die Internet-Adresse und die Bootdatei des X-Terminals enthält. Mit dem Untermenü kann diese Konfigurationsdatei verwaltet werden. Gibt es für ein X-Terminal keinen Eintrag in der Datei, dann wird automatisch die Bootdatei `Xtd` geladen.

- Printer Configuration and Services

Dieses Menü wird nicht unterstützt.

- Restore From Backup Data

Dieses Menü wird nicht unterstützt.

- Schedule Automatic Task

Mit diesem Menü können Benutzer Aufträge, die zu bestimmten Zeiten gestartet werden sollen, in die `cron`-Datei eintragen.

- add

Damit wird ein weiterer Auftrag in die `cron`-Datei eingetragen.

- change

Ändern eines in der `cron`-Datei eingetragenen Auftrags.

- delete

Löschen eines Auftrags aus der `cron`-Datei.

- display

Ausgabe der in der `cron`-Datei stehenden Aufträge.

- Software Installation and Removal

Dieses Menü bietet Funktionen für die Installation von Softwarepaketen, deren Löschung und die Verwaltung von zugehörigen Informationen. Dazu gehört auch die Fähigkeit, die Korrektheit der Installation zu überprüfen. Außerdem können damit Voreinstellungen für die Installation gesetzt und der Dialog bei der Installation bestimmter Softwarepakete abgespeichert werden. Es kann auch Software eingespielt werden, ohne daß sie installiert wird, und es gibt die Möglichkeit, eine komplette Liste der installierten Software auszugeben.

- check (Checks Accuracy of Installation)

Diese Funktion erlaubt Konsistenzprüfungen installierter Softwarepakete und entsprechende Korrekturen. Weiter kann das System auf versteckte Dateien geprüft werden und die Inhalte von Dateien überprüft werden, die anscheinend verändert wurden.

- defaults (Sets Installation Defaults)

Mit dieser Funktion kann man im voraus festlegen, wie das System auf Probleme bei der Installation reagieren soll.

- install (Installs Software Packages)

Mit dieser Funktion kann man Softwarepakete installieren, die sich auf Band, Festplatte oder Diskette befinden, und die Art und Weise festlegen, wie das System auf Probleme bei der Installation reagieren soll.

- interact (Stores Interactions with Package)

Diese Funktion ermöglicht es, Dialoge während der Installation abzuspeichern, um diese bei späteren vollautomatischen Installationen wieder zu verwenden.

- `list` (Displays Information about Packages)

Diese Funktion informiert über die am System installierten Softwarepakete: wie sie bezeichnet sind, wo sie abgelegt sind und in welche Kategorie sie gehören.

- `read_in` (Stores Packages Without Installing)

Mit dieser Funktion kann man Softwarepakete einlesen, ohne sie zu installieren.

- `remove` (Removes Packages)

Mit dieser Funktion kann man ein installiertes Softwarepaket aus dem System löschen.

- Printer Configuration for SINIX Spool

Dieses Menü dient der Druckerverwaltung des SINIX-Spoolsystems V3.1. Sie können neue Drucker konfigurieren, entfernen oder die Druckerkonfiguration ändern. Alle weiteren Verwaltungsaufgaben führt das Spoolsystem aus.

- `add` (Add Printer to Configuration)

Mit dieser Funktion konfigurieren Sie Drucker für das SINIX-Spoolsystem. Ein Drucker kann nur für ein einziges Spoolsystem konfiguriert werden.

- `list` (List Printer Configuration)

Diese Funktion zeigt die aktuelle Druckerkonfiguration an.

- `modify` (Modify Printer Configuration)

Mit dieser Funktion ändern Sie die Konfiguration eines Druckers.

- `remove` (Remove Printer from Configuration)

Mit dieser Funktion entfernen Sie die Konfiguration eines Druckers vom SINIX-Spoolsystem.

- Storage Device Operations and Definitions

Dieses Menü macht die Beschreibung der Einstellung und der Synonyme von Geräten verfügbar und ermöglicht die Zuweisung von Gerätegruppen.

- `descriptions` (Device Alias and Attribute Management)

Dieses Untermenü dient dazu, die Einstellung und Beschreibung von Geräten zu ändern, zu löschen, hinzuzufügen und auszugeben. Es erlaubt auch den Zugang zu Dienstprogrammen für die Reservierung von Geräten.

- `groups` (Device Group Management)

Mit diesem Untermenü kann man Gerätegruppen und Listen der Gruppenzugehörigkeit ausgeben lassen und verwalten.

- `add` (Add Storage Device)

Diese Funktion dient dazu, die Software-Konfiguration eines Geräts im Dialog auszuführen. Das Gerät kann eine zweite Festplatte, ein SCSI-Gerät oder eine andere Hardware sein, die eine Software-Konfiguration erfordert. Es ist sowohl das Neukonfigurieren eines leeren Geräts als auch das Rekonfigurieren eines nicht-leeren Geräts möglich, das bereits vorhanden ist. Die Funktion verwendet Aliase aus der Datei `/etc/device.tab`, für die das Attribut `addcmd` definiert sein muß. Das Attribut muß den gültigen Pfadnamen eines Kommandos enthalten, das die Konfiguration ausführt.

- `copy` (Make Duplicate Copies of Storage Volumes)

Mit dieser Funktion erstellen Sie eine doppelte Kopie des Inhalts eines auswechselbaren Speichermediums.

- `display` (Display Information About Storage Devices)

Diese Funktion gibt Informationen zur Partitionierung der Festplatte(n) aus.

- `erase` (Erase the Contents of Storage Volumes)

Diese Funktion löscht den Inhalt eines auswechselbaren Speichermediums. Die Funktion dient hauptsächlich dazu, Daten allgemein unlesbar zu machen. Die gelöschten Daten sind unwiderruflich verloren.

- `format` (Formats Removable Volumes)

Mit dieser Funktion formatieren Sie ein auswechselbares Speichermedium. Auf ein neues Speichermedium können Daten erst nach dem Formatieren geschrieben werden. Beim Formatieren werden alle Daten gelöscht, die sich bereits auf dem Speichermedium befanden.

- `remove` (Remove Storage Device)

Mit dieser Funktion entfernen Sie die Software zu Speichergeräten, mit allen dazugehörigen Einträgen und Informationen. Auf das Gerät kann nicht mehr zugegriffen werden. Sie können das Gerät physikalisch entfernen, nachdem die Funktion ausgeführt wurde und Sie das System heruntergefahren haben.

- `System Name, Date Time and Initial Password Setup`

Dieses Menü dient dem Einrichten des Rechners. Es umfaßt Funktionen zum Setzen von Systemzeit und -datum und zur Festlegung des Rechnernamens. Weiter dient es der Ersteinrichtung des Rechners und der Zuweisung von Paßwörtern an Kennungen für die Systemverwaltung.

- `datetime` (System Date and Time Information)

Mit dieser Funktion wird am Rechner Datum, Zeit und Zeitzone eingestellt und ob Sommerzeit gilt (Daylight Savings Time: DST). Normalerweise wird das genau einmal bei der Ersteinrichtung des Rechners durchgeführt. Gilt die Sommerzeit (DST), schaltet der Rechner im Herbst automatisch zurück zur Normalzeit und im Frühling auf Sommerzeit. Um sicherzustellen, daß alle Zeitangaben richtig rückgemeldet werden, muß der Rechner aus- und

wieder eingeschaltet werden. Beim nächsten Login eines Benutzers stimmen die meisten Zeitangaben.

- nodename (System Name and Network Node Name of the Machine)

Mit dieser Funktion kann man den Knoten- und Rechnernamen dieses Rechners ändern. Diese Namen dienen verschiedenen Kommunikationsnetzwerken zur Identifizierung dieses Rechners.

- password (Assigns Administrative Login Passwords)

Diese Funktion dient der Zuweisung von Paßwörtern an Kennungen für die Systemverwaltung.

- setup (Sets up System Information for First Time)

Mit dieser Funktion ist es möglich, die erste Kennung einzurichten, an Kennungen für die Systemverwaltung Paßwörter zu vergeben und die entsprechende Zeitzone einzustellen.

- Systemdata Backup and Restore

Dieses Menü dient zum Sichern und Wiederherstellen von wichtigen Konfigurationsdaten des Systems, die beim Installieren eines neuen Systems oder von neuen Packages überschrieben werden könnten.

- list (List Systemdata)

Mit dieser Funktion listen Sie die Konfigurationsdaten auf, die gesichert worden sind.

- restore (Restore Systemdata)

Mit dieser Funktion stellen Sie Konfigurationsdaten wieder ein, die vorher gesichert wurden.

- save (Save Systemdata)

Mit dieser Funktion sichern Sie Konfigurationsdaten.

- User Login and Group Administration

Dieses Menü dient der Benutzerverwaltung am Rechner. Es umfaßt die entsprechenden Funktionen, um dem System neue Benutzer oder Gruppen bekannt zu machen und existierende zu ändern oder zu löschen. Die Gruppenzugehörigkeit von Benutzern kann so organisiert werden, daß der Zugriff auf Dateien gleicher Gruppenmitglieder erlaubt ist, nicht jedoch der Zugriff auf Dateien fremder Gruppen. Außerdem können mit diesem Menü Voreinstellungen für die Neueinrichtung künftiger Benutzer festgelegt und neue oder zusätzliche Informationen zu den Benutzern in der Paßwortdatei eingetragen werden.

- add (Adds Users or Groups)

Mit dieser Funktion werden neue Benutzer oder Gruppen am Rechner definiert.

- defaults(Defines Defaults for Adding Users)

Mit dieser Funktion kann die Voreinstellung für die Einrichtung neuer Benutzer mittels des `add user` Aufrufs verändert werden. Die Änderung von Voreinstellungswerten hat keinerlei Auswirkung auf existierende Benutzer, sondern betrifft ausschließlich neue.

- list(Lists Users or Groups)

Mit dieser Funktion können die Merkmale der Benutzer und Gruppen am System untersucht werden.

- modify(Modifies Attributes of Users or Groups)

Mit dieser Funktion können am System Benutzer- oder Gruppeneinträge geändert werden.

- password((Re-)defines User Password Information)

Mit dieser Funktion kann für Benutzer am System ein Paßwort vergeben oder geändert werden.

- remove(Removes Users or Groups)

Mit dieser Funktion können am System Benutzerberechtigungen gelöscht werden.

EXIT-CODES

Die Exit-Codes von `sysadm` haben folgende Bedeutung:

- 0 normales Ende
- 2 Kommandoaufruf fehlerhaft; es wird eine Meldung zum korrekten Aufruf von `sysadm` ausgegeben
- 4 den für ein Menü oder eine Funktion angegebenen Namen gibt es nicht
- 5 der für ein Menü angegebene Name ist lediglich ein Platzhalter ohne funktionalen Hintergrund und daher nicht verwendbar
- 7 das Kommando `sysadm` kann `fmli` nicht aufrufen, ohne das es nicht korrekt ablaufen kann; das kann daran liegen, daß das FMLI-Paket zerstört oder möglicherweise nicht installiert ist

BEISPIELE

`sysadm nodename`

SIEHE AUCH

`backup(1M)`, `bkexcept(1M)`, `bkhhistory(1M)`, `bkoper(1M)`, `bkgreg(1M)`, `bkstatus(1M)`, `checkfsys(1M)`, `delsysadm(1M)`, `edsysadm(1M)`, `groupadd(1M)`, `groupdel(1M)`, `groupmod(1M)`, `makefsys(1M)`, `mountfsys(1M)`, `password(1M)`, `restore(1M)`, `rsnotify(1M)`, `rsoper(1M)`, `rsstatus(1M)`, `setup(1M)`, `urestore(1M)`, `ursstatus(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`.

BEZEICHNUNG

sysrestore – System wiederherstellen

ÜBERSICHT

/sbin/sysrestore [option]

BESCHREIBUNG

Das Kommando `sysrestore` ermöglicht das Rückkopieren von Sicherungen, die mit dem Kommando `sys save` erstellt wurden.

Mögliche Optionen sind:

- f *tape* *tape* ist der Gerätename des Bandlaufwerks, von dem gelesen werden soll.
 -f nicht angegeben: /dev/tape.
- B *boot* *boot* ist der Name des Boot-Programmes, das beim Initialisieren eines Laufwerks auf die Platte kopiert werden soll.
 -B nicht angegeben: /etc/boot.

Sysrestore-Menü

Nach dem Start fordert `sysrestore` das letzte Folgeband der Sicherung an und gibt folgendes Menü aus:

Sysrestore MENÜ

- (0) Exit
- (1) List the slices saved on the backup tape(s)
- (2) List the VTOCs saved on the backup tape(s)
- (4) Restore one slice (w/o VTOC)
- (5) Restore some (more than one) slices (w/o VTOC)
- (6) Restore all saved slices (w/o VTOC)
- (7) Add a slice from the tape to another drive
- (8) Initialize (clear) a disk drive
- (9) Restore one VTOC from the backup tape
- (10) Restore all VTOCs from the backup tape
- (11) Restore one drive (VTOC and slices)
- (12) Restore everything on the backup tape

Bedeutung der einzelnen Menüpunkte:

- (0) Beenden von `sysrestore`.
- (1) Anzeigen aller Slices, die auf dem Sicherungsband bzw. den Sicherungsbändern vorhanden sind.
- (2) Auflisten aller Laufwerke oder Partitionen, deren VTOC auf dem Sicherungsband/-bändern gesichert wurden. Das sind alle Laufwerke, von denen sich mindestens eine Slice auf einem Sicherungsband befindet.
- (4) Rückkopieren einer Slice vom Sicherungsband. Das VTOC des zugehörigen Laufwerks wird nur dann rückkopiert, wenn das Plattenlaufwerk, auf das kopiert werden soll, keine gültige `fdisk`-Partitions-Tabelle oder kein gültiges VTOC enthält. Das ist z.B. bei einer neu

formatierten Platte der Fall; sie wird implizit initialisiert und auch das VTOC wird vom Band rückkopiert.

- (5) Rückkopieren mehrerer Slices vom Sicherungsband. In einem Untermenü werden die einzelnen Slices aufgelistet, die sich auf dem Sicherungsband befinden. Nachdem der Anwender die Restore-Liste entsprechend seinen Wünschen aufgebaut hat, kann er von diesem Untermenü aus alle selektierten Slices rückkopieren. Jede ausgewählte Slice wird wie unter (4) beschrieben rückkopiert.
- (6) Rückkopieren aller Slices vom Sicherungsband. Eine Platte wird implizit nur dann initialisiert und die VTOCs werden nur dann rückkopiert, wenn ein Plattenlaufwerk, auf das kopiert werden soll, keine gültige `fdisk`-Partitions-Tabelle oder kein gültiges VTOC enthält.
- (7) Rückkopieren einer Slice auf eine andere Partition. Der Anwender wählt eine Plattenpartition aus, auf der eine neue Slice eingerichtet werden soll. Danach wird eine Slice vom Sicherungsband auf diese neue Slice rückkopiert. Dabei wird eine Platte nur dann initialisiert, wenn die Platte, auf die kopiert werden soll, keine gültige `fdisk`-Partitions-Tabelle oder kein gültiges VTOC enthält.
- (8) Initialisieren einer Platte. Der Anwender wählt aus, welches Plattenlaufwerk initialisiert werden soll. Es wird eine einzige Partition - die aktive UNIX Partition - eingerichtet, die sich über die ganze Platte erstreckt. Eine entsprechende `fdisk`-Partitions-Tabelle wird im Block 0 installiert, sofern sie noch nicht vorhanden ist. Weiter wird ein leeres VTOC angelegt, das nur die `V_BACKUP`-Slice und die `V_BOOT`-Slice enthält (siehe `sysrestore(1M)`).
- (9) Rückkopieren des VTOC eines Laufwerks. Alle bereits vorhandenen Daten auf dem Laufwerk gehen dabei verloren. Enthält das Laufwerk, auf das rückkopiert werden soll, keine gültige `fdisk`-Partitions-Tabelle oder kein gültiges VTOC, dann wird die Platte implizit initialisiert.
- (10) Rückkopieren der VTOCs aller gesicherten Laufwerke. Alle Daten von Laufwerken, deren VTOC auf dem Band gesichert ist, werden dabei zerstört. Platten, die keine gültige `fdisk`-Partitions-Tabelle oder kein gültiges VTOC enthalten (z.B. neue Platten) werden vor dem Rückkopieren implizit initialisiert.
- (11) Rückkopieren des VTOC und der Slices eines Laufwerks. Alle bereits vorhandenen Daten auf dem Laufwerk gehen dabei verloren. Enthält ein Laufwerk, auf das rückkopiert werden soll, keine gültige `fdisk`-Partitions-Tabelle oder kein gültiges VTOC, dann wird die Platte implizit initialisiert.
- (12) Rückkopieren der VTOCs und der Slices aller gesicherten Laufwerke. Alle bereits vorhandenen Daten, die sich auf diesen Laufwerken befinden, werden durch den Rückkopiervorgang zerstört. Alle Platten, die keine gültige `fdisk`-Partitions-Tabelle oder kein gültiges VTOC enthalten (z. B. neue Platten) werden vor dem Rückkopieren implizit initialisiert.

Slicetypen, die nicht gesichert werden

V_BOOT Enthält das VTOC und den pdinfo-Block.

V_BACKUP

Erstreckt sich über die gesamte UNIX-Partition einer Platte. Slice 0 ist von diesem Typ.

V_ALTS Wird z.B. bei AT386-Systemen für die Zuweisung von Ersatzsektoren verwendet. Bei der MX300 wird diese Slice nicht verwendet, da dort die Zuweisung über den Controller erfolgt.

V_ALTRK

Wird z.B. bei AT386-Systemen für die Zuweisung von Ersatzspuren verwendet. Bei der MX300 wird diese Slice nicht verwendet, da dort die Zuweisung über den Controller erfolgt.

V_OTHER

Nicht-UNIX-Slice. Eine eventuell vorhandene DOS-Partition ist ebenfalls im VTOC eingetragen. Diese Slice hat dann den Typ V_OTHER.

DATEIEN

/var/syssave/savetab,
/var/syssave/LOGFILE.

SIEHE AUCH

syssave(1M)
savetab(4)

HINWEIS

Alle vom Kommando `sysrestore` ausgegebenen Meldungen werden in der Datei `/var/syssave/LOGFILE` protokolliert, wenn `sysrestore` nicht vom Syssave-Hilfssystem aus gestartet wurde.

Bei den Menüpunkten 4, 6, 7 und 9 bis 12 werden alle Platten initialisiert, die keine gültige `fdisk`-Partitions-Tabelle oder kein gültiges VTOC enthalten.

BEZEICHNUNG

syssave – System sichern

ÜBERSICHT

/sbin/syssave [option]

BESCHREIBUNG

syssave sichert physikalisch einen konsistenten Stand des Systems. Nach einem Plattencrash kann diese Sicherung mit /sbin/sysrestore zurückkopiert und das System wieder restauriert werden. syssave sichert alle Plattenslices, die in einer Datei aufgelistet sind (standardmäßig: /var/syssave/savetab). Die Slices werden zusammen mit ihrer Aufteilung (VTOC) auf ein oder mehrere Bänder geschrieben. Für die Konsistenz der zu sichernden Daten während der Sicherung hat der Anwender selbst zu sorgen. Die Sicherung eines konsistenten Standes ist nur im Single-User-Modus oder bei Einsatz des Syssave-Hilfssystems gegeben. Das Kommando darf nur von einem privilegiertem Benutzer aufgerufen werden.

Mögliche Optionen sind:

- f *tape* *tape* ist der Gerätename des Bandlaufwerks, auf das geschrieben werden soll. Mögliche Angaben sind z.B. /dev/tape, /dev/xa0 oder /dev/rmt/c0s0.
-f nicht angegeben: /dev/tape.
- i Startet das Syssave-Menüsystem.
- t *datei* In *datei* sind die Namen der Slices aufgelistet, die syssave sichern soll.
-t nicht angegeben: /var/syssave/savetab
- y Batch-Betrieb von syssave. syssave beantwortet alle ausgegebenen Fragen automatisch mit "y". Außerdem wird das Sicherungsband nicht angefordert. Es muß bereits eingelegt sein und alle zu sichernden Slices müssen darauf Platz haben.
- M *mbytes* Ermöglicht die Angabe der Netto-Bandkapazität in Megabytes. syssave schreibt maximal die hier angegebene Datenmenge (und die Datei-Marken zwischen den Banddateien) auf ein Sicherungsband. Die Netto-Bandkapazität eines Bandes ist in der Regel geringer als die spezifizierte Bandkapazität. Auf Magnetbandkassetten ist z.B. der Wert 155MB aufgedruckt. Man darf jedoch nur mit einer Netto-Kapazität von ca. 130-150MB (je nach Anzahl der defekten Stellen des Bandes) rechnen.
-M nicht angegeben: 150

Syssave-Hilfssystem

Das Syssave-Hilfssystem ist ein eigenständiges Betriebssystem. Es dient dazu, einen konsistenten Stand des Systems zu sichern oder nach Reparatur einer defekten Systemplatte ein System wiederherzustellen. Um alle Platten und Slices einschließlich des VTOC wiederherstellen zu können, darf das Hilfssystem keinerlei Platten zum Betrieb benötigen. Es besteht aus drei 3½" Disketten SYSSAVE-0, SYSSAVE-1 und SYSSAVE-2, die der Systemverwalter selbst erstellen muß. SYSSAVE-0 enthält einen minimalen Systemkern und ein Bootprogramm, SYSSAVE-1 ein minimales Root-Dateisystem (/) und SYSSAVE-2 ein minimales /usr-Dateisystem. Das Syssave-Hilfssystem wird, z.B. nach einem Plattencrash, von Diskette hochgefahren und sein Root-Dateisystem wird als RAM-Disk im Speicher abgelegt. Ein zweites benötigtes Dateisystem (/usr) wird eingehängt.

Um das Syssave-Hilfssystem zu erstellen, muß der Systemverwalter so vorgehen:

- Anmelden unter der Kennung `root`.
- Wechseln ins Dateiverzeichnis `/opt/syssave`.
- Starten des Installations-Scripts mit `./install`

Das Script fordert nacheinander die drei Disketten an, formatiert und erstellt sie.

Booten des Syssave-Hilfssystems

Um das Syssave-Hilfssystem hochzufahren, muß so vorgegangen werden:

- Einlegen der Diskette `SYSSAVE-0` ins Laufwerk
- Entweder Neustarten des Systems oder Einschalten der Maschine
- Einlegen der Diskette `SYSSAVE-1` statt der geforderten Diskette `SINIX-1`
- Einlegen der geforderten Diskette `SYSSAVE-2`
- Angeben des Typs der Konsolen-Tastatur (international oder national)

Daraufhin erscheint das Syssave-Menüsystem.

Syssave-Menüsystem

Das Syssave-Menüsystem kann auf zwei Arten aktiviert werden: Entweder durch den Aufruf von `syssave -i` als privilegierter Benutzer oder automatisch nach dem Hochfahren, wenn das Syssave-Hilfssystem verwendet wird.

Das Hauptmenü sieht so aus:

- (0) Exit
- (1) Option management
- (2) Add/remove/list slices to be saved
- (3) Save system to (syssave)
- (4) Restore system from ... (sysrestore)

Bedeutung der einzelnen Menüpunkte:

- (0) Menü verlassen. Wurde das Hilfssystem verwendet, dann schaltet sich die Maschine selbstständig ab.
- (1) Es folgt ein Untermenü, in dem verschiedene Parameter für die Sicherung eingestellt werden können. Die Menüpunkte sind selbsterklärend.
- (2) Es folgt ein Untermenü, mit dem die Liste der zu sichernden Slices verwaltet werden kann (siehe unten).
- (3) Physikalisches Sichern aller im Menüpunkt (2) festgelegten Slices auf Band mit dem Programm `syssave`. Dabei werden die im Punkt (1) festgelegten Optionen verwendet. `syssave` fordert den Anwender bei Bedarf zum Einlegen des (der) Sicherungsbandes (-bänder) auf.
- (4) Startet das Programm `sysrestore`, um eine Sicherungskopie zurückzukopieren, die mit `syssave` angelegt wurde. Dabei werden die im Punkt (1) festgelegten Optionen berücksichtigt. Das Programm `sysrestore` startet wiederum mit einem Menü (siehe `sysrestore(1M)`).

Untermenü: Add/remove/list slices to be saved

Das Untermenü sieht so aus:

- (0) Return to previous menu
- (1) List the slices that are to be saved
- (2) Add a slice to the list of saved slices
- (3) Remove a slice from the list of saved slices
- (4) Clear the list of saved slices

Bedeutung der einzelnen Menüpunkte:

- (0) Kehrt zum Hauptmenü zurück.
- (1) Ausgabe der Liste der Slices, die bei Auswahl des Menüpunktes (3) des Hauptmenüs gesichert werden sollen. Pro Zeile werden der Name der Slice im Format `shd1t5s6` oder `c0d1s3` und ein der Slice zugeordneter Kommentar ausgegeben. Nach jeweils 15 Zeilen wird die Meldung
Press <END> to stop, <return> to continue listing
ausgegeben. Der Anwender kann die Ausgabe der Liste abbrechen oder fortsetzen.
- (2) Dieser Menüpunkt dient dazu, eine Slice in die Liste der zu sichernden Slices aufzunehmen. Der Anwender wird nach dem Namen der Slice und einem eventuellen Kommentar gefragt. Mögliche Namen sind z.B. `shd1t5s6` oder `c0d1`.
- (3) Bei diesem Menüpunkt kann eine Slice aus der Liste der zu sichernden Slices entfernt werden. Der Anwender wird nach dem Namen der zu entfernenden Slice gefragt. Gibt er statt eines Namens "?" ein, dann wird ähnlich wie bei Punkt (1) die Liste der zu sichernden Slices ausgegeben, aus der der Anwender die zu löschenden Slices auswählen kann.
- (4) Es werden alle Slices aus der Liste der zu sichernden Slices gelöscht.

Aufbau von /var/syssave/savetab

syssave erwartet die Liste der zu sichernden Slices in einer Datei, die bei der Option `-t` angegeben werden kann. Standardmäßig ist dies `/var/syssave/savetab`.

Die Datei ist eine editierbare ASCII-Datei. Pro Zeile steht ein Slice- oder Platten-Name (z.B. `shd1t5s6` oder `c0d1`). Die Slices können auf Platten am Interphase Storage oder am NCR Host Adapter liegen. Leerzeilen in der Datei werden überlesen. Kommentare beginnen mit dem Zeichen `"#"` und reichen bis zum Zeilenende.

Wird Slice 0 angegeben (z.B. `c0d0s0`), dann werden alle Slices dieses Laufwerks gesichert, bis auf Slices vom Typ `V_BACKUP`, `V_BOOT`, `V_OTHER`, `V_ALTRK` oder `V_ALTS`. syssave sichert nur die aktive UNIX-Partition, eventuelle DOS-Partitionen oder andere werden nicht gesichert.

Anstatt eines Slice-Namens kann auch ein Laufwerk-Name ohne Sliceangabe (z.B. `c0d0`) angegeben werden. Dieser wird wie `c0d0s0` behandelt, d.h. alle Slices des Laufwerks werden gesichert.

Beispiel:

```
#
# Beispiel für /var/syssave/savetab
#

c0d1s0      # whole drive 1
c1d0s3      # data base slice
shd0t5s6    # some other slice
shd1t5      # whole external drive
```

In diesem Beispiel werden alle Slices des Laufwerks `c0d1` gesichert. Weiter werden die Slices `c1d0s3` und `shd0t5s6` sowie alle Slices auf dem Laufwerk `shd1t5` gesichert.

Slicetypen, die nicht gesichert werden

V_BOOT Enthält das VTOC und den `pdinfo`-Block.

V_BACKUP

Erstreckt sich über die gesamte UNIX-Partition einer Platte. Slice 0 ist von diesem Typ.

V_ALTS Wird z.B. bei AT386-Systemen für die Zuweisung von Ersatzsektoren verwendet. Bei der MX300 wird diese Slice nicht verwendet, da dort die Zuweisung über den Controller erfolgt.

V_ALTRK

Wird z.B. bei AT386-Systemen für die Zuweisung von Ersatzspuren verwendet. Bei der MX300 wird diese Slice nicht verwendet, da dort die Zuweisung über den Controller erfolgt.

V_OTHER

Nicht-UNIX-Slice. Eine eventuell vorhandene DOS-Partition ist ebenfalls im VTOC eingetragen. Diese Slice hat dann den Typ `V_OTHER`.

syssave (1M)

(MX300)

syssave (1M)

DATEIEN

/var/syssave/savetab, /var/syssave/LOGFILE.

SIEHE AUCH

sysrestore(1M)

savetab(4)

HINWEIS

Alle vom Kommando `syssave` ausgegebenen Meldungen werden in der Datei `/var/syssave/LOGFILE` protokolliert, wenn `syssave` nicht vom Syssave-Hilfssystem aus gestartet wurde.

talkd (1M)

talkd (1M)

BEZEICHNUNG

talkd, in.talkd – Server für das Talk-Programm

ÜBERSICHT

in.talkd

BESCHREIBUNG

talkd ist ein Server, der vom Talk-Programm gestartet wird. Dieser Server hört den UDP Port ab, der in der "talk" Servicebeschreibung angegeben wurde; siehe auch services(4). Die tatsächliche Verbindung findet über ein TCP-Protokoll statt, das vorher zwischen beiden beteiligten Systemen vereinbart wurde.

SIEHE AUCH

talk(1), inetd(1M), services(4).

HINWEIS

Das Protokoll ist abhängig von der gewählten Architektur.

BEZEICHNUNG

telnetd – Protokoll-Server für DARPA TELNET

ÜBERSICHT

in.telnetd

BESCHREIBUNG

telnetd ist ein Server für das virtuelle Terminalprotokoll TELNET nach dem DARPA Standard. telnetd wird vom Internet-Server aufgerufen (siehe inetd(1M)), um eine Verbindung zum TELNET-Port aufzubauen, so wie sie in der Datei /etc/services beschrieben ist (siehe services(4)).

Prinzipiell funktioniert telnetd wie folgt: zunächst wird ein virtuelles Terminal einem Client zugeordnet; anschließend wird ein Login-Prozeß erzeugt, dessen Standardeingabe (stdin), Standardausgabe (stdout) und Standardfehlerausgabe (stderr) mit der passiven Seite (Slave) des virtuellen Terminals verbunden wird. telnetd versorgt die aktive Seite (Master) dieses virtuellen Terminals; hierbei werden Zeichen mittels des TELNET-Protokolls zwischen Client (im fernen System) und dem lokalen Login-Prozeß übertragen.

Wenn eine neue TELNET-Sitzung gestartet wird, sendet telnetd die folgenden TELNET-Optionen an den Client, um die Bereitschaft für remote echo (Zeichenwiederholung), suppress go ahead und terminal type information (Mitteilung des Terminaltyps des Client) mitzuteilen. Falls der Client im fernen System dazu bereit ist, wird der Terminaltyp der gesamten Umgebung des neuen Login-Prozesses bekanntgemacht. Das dem Client zugeordnete virtuelle Terminal wird im cooked-Modus (Zeilenverarbeitung), sowie mit den gesetzten Optionen XTABS, ICRNL, und ONLCR konfiguriert (siehe termio(4)).

telnetd selbst ist in der Lage, Optionen zu unterstützen, die in der Datei include/arpa/telnet.h aufgelistet sind.

telnetd kann wiederum den Client veranlassen, die folgenden Optionen zu unterstützen: binary , terminal type und suppress go ahead.

SIEHE AUCH

telnet(1)

Postel, Jon, and Joyce Reynolds, "Telnet Protocol Specification," RFC 854, Network Information Center, SRI International, Menlo Park, Calif., Mai 1983.

HINWEIS

Einige der definierten TELNET-Kommandos sind nur teilweise implementiert.

So sieht beispielsweise das TELNET-Protokoll die Übertragung der Zeilen- und Spaltenanzahl auf dem Benutzerterminal vor, jedoch macht `telnetd` hiervon keinen Gebrauch.

Der Binärmodus (binary mode) wird nur zwischen ähnlichen Betriebssystemen ausgewertet.

Der vom Client angeforderte Terminaltyp wird in Kleinschreibung umgewandelt.

Das *Paket-Interface* des virtuellen Terminals sollte für ein optimaleres Leeren der Warteschlangen für Ein-/Ausgabe genutzt werden.

`telnetd` sendet zu keinem Zeitpunkt `go ahead` Kommandos an TELNET.

`telnetd` unterstützt nur bis zu 64 virtuelle Terminals.

BEZEICHNUNG

tftpd, in.tftpd – DARPA-Server für Trivial File Transfer Protokoll

ÜBERSICHT

in.tftpd [-s] [*Home-Verzeichnis*]

BESCHREIBUNG

tftpd ist ein Server, der das DARPA Trivial File Transfer Protokoll (TFTP) für die Übertragung von Dateien unterstützt. Üblicherweise wird dieser Server vom inetd(1M) Internet-Dämon für den Anschluß gestartet, der in der tftp Internet-Servicebeschreibung in der Datei /etc/inetd.conf angegeben ist. Der zugehörige Eintrag für tftpd in der Datei etc/inetd.conf ist normalerweise auskommentiert. Um den tftpd Server einsatzbereit zu machen, müssen daher die Kommentarzeichen entfernt werden. Siehe inetd.conf(4) für weitere Erläuterungen.

Bevor ein Auftrag beantwortet wird, versucht der Server, vom aktuellen Verzeichnis in das angegebene *Home-Verzeichnis* zu wechseln; als Standardverzeichnis wird /tftpboot angenommen.

OPTIONEN

-s (secure) Mit dieser Option wird sichergestellt, daß der oben beschriebene Wechsel des Verzeichnisses auch tatsächlich erfolgt ist; in diesem Fall wird das angegebene *Home-Verzeichnis* für den Dämon zum neuen Root-Verzeichnis.

Die Benutzung von tftp setzt keinen eigenen Benutzereintrag oder Paßwort auf dem fernen System voraus. Da die Überprüfung von Zugriffsrechten vom tftpd nicht unterstützt wird, kann nur auf allgemein lesbare Dateien zugegriffen werden. Genauso können nur solche Dateien geschrieben werden, die bereits existieren und allgemein schreibbar sind. Bitte beachten Sie, daß diese Definition den allgemeinen (public) Zugriff auf alle Benutzer ausweitet, die über das Netz erreicht werden können; dies mag nicht für alle Systeme geeignet sein, daher sollten die Auswirkungen vor Bereitstellung dieser Funktion beachtet werden.

Für den laufenden tftpd Server wird die Benutzernummer (UID) und Gruppennummer (GID) auf den Wert [GU]ID_NOBODY gesetzt. -2, unter der Annahme, daß für diesen Benutzer bzw. Gruppe keine Dateien existieren. Es wird jedoch diese Annahme nicht überprüft, noch wird diese Einschränkung erzwungen.

SIEHE AUCH

tftp(1), inetd(1M), ipallocald(1M), netconfig(4).

Sollins, K.R., *The TFTP Protocol* (2. Auflage), RFC 783, Network Information Center, SRI International, Menlo Park, Calif., Juni 1981.

BEZEICHNUNG

tic – Übersetzer von *terminfo*

ÜBERSICHT

tic [-v[n]] [-c] *datei*

BESCHREIBUNG

Das Kommando *tic* übersetzt Quelldateien für *terminfo* in Binärformat und legt die übersetzten Dateien in dem Verzeichnis */usr/share/lib/terminfo* ab. Bei der Verwendung der Bibliotheksroutinen aus *curses(3X)* braucht man das Binärformat von *terminfo*.

-vn (verbose) Diese Option legt fest, daß *tic* während seiner Arbeit ausführliche Information zur Ablaufüberwachung auf die Standardausgabe ausgibt. Die Angabe der natürlichen Zahl *n* aus dem Bereich von 1 bis 10 einschließlich ist optional; sie bestimmt den gewünschten Detailgrad der Information. Die Voreinstellung dafür ist 1; bei Angabe eines höheren Wertes wird der Umfang der ausgegebenen Information erhöht.

-c (check) Dient der Überprüfung von Fehlern in der Datei *datei*. Durch Verweise der Form *use=* auftretende Fehler werden nicht entdeckt.

datei Die Datei *datei* enthält die Beschreibung eines oder mehrerer Terminals im Quellformat (vgl. *terminfo(4)*). Jede dieser Beschreibungen erklärt die funktionalen Fähigkeiten des jeweiligen Terminals. Wird in der gerade bearbeiteten Terminalbeschreibung ein Eintrag der Form *use=eintrag* vorgefunden, liest *tic* zur Vervollständigung dieses Eintrags die Binärinformation dazu aus */usr/share/lib/terminfo* ein. (Dabei gilt, daß die für die Datei *datei* bereits erzeugten Einträge zuerst verwendet werden. Außerdem wird für den Fall, daß die Umgebungsvariable *TERMINFO* belegt ist, statt des Verzeichnisses */usr/share/lib/terminfo* das in der Variablen stehende Verzeichnis durchsucht.) *tic* kopiert die in dem Eintrag *eintrag* aufgeführten Terminaleigenschaften in den aktuellen Eintrag, wobei diejenigen ausgelassen werden, die im aktuellen Eintrag bereits definiert sind.

Ist die Umgebungsvariable *TERMINFO* belegt, werden die übersetzten Terminalbeschreibungen in dem dortigen Verzeichnis abgelegt statt in */usr/share/lib/terminfo*.

Komplett übersetzte Einträge dürfen nicht länger als 4096 Bytes sein. Das Feld für den Namen darf 128 Bytes nicht überschreiten. Bezeichnungen für Terminals, die länger als 14 Zeichen sind, werden auf 14 Zeichen verkürzt und es erfolgt eine diesbezügliche Warnmeldung.

DATEIEN

*/usr/share/lib/terminfo/?/**

Datenbank mit den übersetzten Terminalbeschreibungen

SIEHE AUCH

curses(3X), *captainfo(1M)*, *infocmp(1M)*, *terminfo(4)*.

tic(1M)

tic(1M)

HINWEIS

Enthält beispielsweise ein Eintrag `eintrag_1` ein Feld der Form `use=eintrag_2` und in `eintrag_2` sind bestimmte Terminaleigenschaften außer Kraft gesetzt, dann müssen alle diese Eigenschaften vorher bereits in `eintrag_1` aufgeführt sein, bevor sie durch die `use=-`Anweisung in dem Eintrag `eintrag_1` aufgehoben werden können.

tnamed (1M)

tnamed (1M)

BEZEICHNUNG

tnamed, in.tnamed – Einfacher DARPA Name-Server

ÜBERSICHT

in.tnamed [-v]

BESCHREIBUNG

tnamed ist ein Server, der das DARPA-Name-Server-Protokoll unterstützt. Der Name-Server wird dem Anschluß zugewiesen, der in der Name-Servicebeschreibung angegeben ist (siehe *services(4)*), und wird vom *inetd(1M)* Dämon aktiviert, wenn ein Auftrag für den Name-Server vorliegt.

OPTIONEN

-v Startet den Dämon im Protokoll-Modus.

SIEHE AUCH

uucp(1C), *inetd(1M)*, *services(4)*.

Postel, Jon, *Internet Name Server*, IEN 116, SRI International, Menlo Park, California, August 1979.

BEZEICHNUNG

tosmtp – Nachrichten an SMTP schicken

ÜBERSICHT

tosmtp [-f] [-n] [-u] [-d *bereich*] [-H *hallohost*] *sender host empf ...*

BESCHREIBUNG

Das Programm tosmtp konvertiert eine Nachricht, die aus einem SINIX-System stammt (und von der Standardeingabe gelesen wurde), in eine RFC822-Nachricht, die über SMTP ausgeliefert werden kann. Normalerweise wird tosmtp von smtpger aufgerufen, während dieses die Auslieferung von Nachrichten vorbereitet.

Die Optionen für tosmtp haben folgende Bedeutung:

- d *bereich* Der angegebene *bereich* soll direkt an das Programm smtp durchgereicht werden.
- f tosmtp soll als Filter wirken und die RFC822-Nachricht auf die Standardausgabe schreiben.
- H *hallohost* Damit kann man den Namen des Hosts angeben, der in der SMTP HELO-Meldung verwendet wird. Diese Option wird an das Programm smtp durchgereicht.
- n Unterdrückung von To:-Zeilen im RFC822-Kopf.
- u Keine Konvertierung. Die Standardeingabe wird unverändert an die Standardausgabe weitergereicht.

DATEIEN

/usr/lib/mail/surrcmd/smtp

Dorthin wird die Nachricht über eine Pipe weitergeleitet

SIEHE AUCH

smtp(1M), smtpger(1M)

RFC822 – Standardformat der ARPA Internet Text Messages

BEZEICHNUNG

trpt – Überwachung einer TCP/IP-Verbindung

ÜBERSICHT

trpt [-afjst] [-p *hex-adresse*] [system [*speicherabzugsdatei*]]

BESCHREIBUNG

Um eine TCP/IP-Verbindung zu überwachen, kann das trpt Kommando verwendet werden; wurde für einen TCP/IP-Socket die *debugging*-Option für Überwachung eingeschaltet (siehe *getsockopt(3N)*), wertet trpt den Puffer mit den TCP Überwachungs-Datensätzen aus und erzeugt eine ausführliche Beschreibung dieser Datensätze. Falls keine Optionen angegeben wurden, liefert trpt alle im System vorliegenden Überwachungs-Datensätze, geordnet nach TCP Protokoll-Kontrollblöcken (PCB). Mit den folgenden Optionen kann die Standardausgabe modifiziert werden.

OPTIONEN

- a (address) Druckt zusätzlich zur Standardausgabe die Quell- und Ziela-dresse jedes aufgezeichneten Pakets.
- f (follow) Die Datensätze werden in der Reihenfolge ihres Eintreffens ausgegeben; nach Erreichen des letzten Datensatzes wird noch eine kurze Zeit auf weitere eintreffende Datensätze gewartet.
- j (just) Es wird lediglich eine Liste der zu den Überwachungs-Datensätzen gehörenden Protokoll-Kontrollblöcken ausgegeben.
- s (sequence) Gibt zusätzlich zur normalen Ausgabe eine detaillierte Beschreibung der Reihenfolge der Pakete aus.
- t (time) Gibt zusätzlich zur normalen Ausgabe die Werte der Zeitstempel während des gesamten Protokolls aus.
- p *hex-address* (pcb) Gibt lediglich die Überwachungs-Datensätze für den Protokoll-Kontrollblock aus, dessen Adresse folgt.

trpt sollte wie folgt verwendet werden: Isolieren Sie die fehlerhafte Verbindung und schalten Sie die Überwachung für den/die betroffenen Socket(s) ein. Finden Sie die Adresse der zu den Sockets gehörenden Protokoll-Kontrollblöcke mit dem *netstat(1M)* Kommando und der Option -A. Dann rufen Sie das trpt Kommando mit der Option -p für jede der so erhaltenden PCB-Adressen auf. Die Option -f kann darüberhinaus dazu verwendet werden, die in Frage kommenden Datensätze in der Reihenfolge ihres Eintreffens auszugeben. Falls die Überwachung für mehrere Sockets eingeschaltet ist, kann mit der Option -j leicht geprüft werden, ob bereits Datensätze für den bestimmten Socket vorhanden sind.

Mit den letzten zwei Argumenten können die eingestellten Standardwerte überschrieben werden, falls die Überwachung für ein anderes System bzw. eine Dump-Datei vorgenommen werden soll.

trpt (1M)

trpt (1M)

DATEIEN

/stand/unix
/dev/kmem

SIEHE AUCH

netstat(1M), getsockopt(3N).

FEHLERMELDUNGEN

no namelist

Der Überwachungspuffer konnte anhand der vorhandenen Symboldefinitionen im Dump nicht gefunden werden. Andere Fehlermeldungen sind selbsterklärend.

HINWEIS

Dieser Befehl sollte auch die jeweiligen Ein- und Ausgabedaten ausgeben, aber diese Angaben werden nicht in den Überwachungs-Datensatz geschrieben.

BEZEICHNUNG

ttyadm – Formatierung und Ausgabe Anschlußüberwachungsspezifischer Informationen

ÜBERSICHT

```
/usr/sbin/ttyadm [ -b ] [ -c ] [ -r anzahl ] [ -h ] [ -i meldung ] [ -m moduln ]
                  -p prompt ] [ -t zeitlim ] -d gerät -l ttyzeiger -s service

/usr/sbin/ttyadm -V
```

BESCHREIBUNG

Das Systemverwalterkommando ttyadm bereitet ttymon-spezifische Information auf und gibt sie auf der Standardausgabe aus. Innerhalb der Service Access Facility (SAF) muß jede Anschlußüberwachung ein solches Kommando bereitstellen. Es ist zu beachten, daß die Verwaltungsdatei der Anschlußüberwachung durch Verwaltungskommandos sacadm und pmadm von SAC (Service Access Controller) aktualisiert wird. ttyadm sorgt dafür, daß diesen Kommandos die formatierten Daten zur Verfügung gestellt werden, welche die spezifische (d.h. spezifisch für ttymon) Information für die Anschlußüberwachung enthalten.

- b Setzt den Schalter "bidirektionaler Anschluß". In diesem Fall kann die Leitung in beiden Richtungen benutzt werden. Benutzer können dann die diesem Anschluß zugeordneten Dienste in Anspruch nehmen. Doch für den Fall, daß diese Schnittstelle gerade frei ist, erlaubt es ttymon den Kommandos uucico, cu oder ct, sie für den Aufbau von Wahlverbindungen zu nutzen.
- c Setzt den Schalter "sofortige Verbindung" für diesen Anschluß. In diesem Fall startet ttymon den diesem Anschluß zugeordneten Service, sowie das Zustandekommen einer Verbindung ersichtlich ist; es wird also keine Eingabeaufforderung ausgegeben und die Suche nach der richtigen Übertragungsgeschwindigkeit entfällt.
- d *gerät* *gerät* ist der volle Pfadname der Gerätedatei zu diesem TTY-Anschluß.
- h Setzt den Hangup-Schalter für diesen Anschluß. Ist der Schalter -h nicht gesetzt, wird ttymon zuerst einen Hängezustand für diese Leitung simulieren, indem er die Baudrate auf Null setzt, bevor er die Übertragungsgeschwindigkeit entweder auf den angegebenen Wert oder aber gemäß Voreinstellung setzt.
- i *meldung* Damit wird eine Meldung bezüglich des Zustands "inaktiv" definiert. Diese Meldung wird dann an den TTY-Anschluß geschickt, wenn entweder die Schnittstelle oder aber das Anschlußüberwachungsprogramm ttymon deaktiviert wird.
- l *ttyzeiger* *ttyzeiger* ist ein Verweis auf einen Eintrag in der Datei ttydefs, welchem ttymon entnimmt, welche anfängliche Übertragungsgeschwindigkeit bei der Suche nach der richtigen Baudrate gewählt werden soll.

ttyadm (1M)

ttyadm (1M)

- m *moduln*** Damit wird eine Liste stapelverträglicher STREAMS-Moduln spezifiziert. *moduln* enthält eine ohne jeden Zwischenraum aufgezählte Reihe von durch Kommata getrennten Moduln, die in der angegebenen Reihenfolge auf den Stream-Stapel gepackt werden. Vorher allerdings wird der aktuelle Modul-Stapel von allen Moduln, die sich gerade auf dem Stream befinden, entleert.
- r *anzahl*** Wird **ttymon** mit der Option **-r** aufgerufen, wartet es mit der Ausgabe der Eingabeaufforderung solange, bis Daten von der Schnittstelle empfangen werden. Ist der Wert von *anzahl* gleich Null, wartet **ttymon** bis zum Eintreffen eines beliebigen Zeichens. Ist er größer Null, wird solange gewartet, bis *anzahl* Neue-Zeile-Zeichen empfangen wurden.
- p *prompt*** Damit kann die Zeichenkette für die Eingabeaufforderung definiert werden, beispielsweise "login:".
- s *service*** *service* ist der volle Pfadname des Service-Programms, welches beim Empfang einer Verbindungsaufforderung aufgerufen wird. Falls das Programm Argumente benötigt, müssen diese zusammen mit dem Programmnamen in Anführungszeichen stehen.
- t *zeitlim*** Legt die Zeitdauer in Sekunden fest, in welcher nach erfolgreichem Öffnen der Schnittstelle eine Eingabe erfolgen soll. Verstreicht diese Zeit, schließt **ttymon** die Schnittstelle wieder.
- V** Gibt die aktuelle Versionsnummer des Kommandos `/usr/lib/saf/ttymon` aus.

Exit-Status

Bei fehlerfreiem Ablauf bereitet **ttyadm** die gewünschte Information auf und gibt sie auf die Standardausgabe aus. Für diesen Fall wird ein Exit-Status 0 zurückgegeben. Im Fehlerfall ist der Exit-Status ungleich Null und **ttyadm** gibt eine entsprechende Fehlermeldung auf die Standardfehlerausgabe aus. Fehlerfälle sind beispielsweise der Aufruf von **ttyadm** mit ungültigen Argumenten, mit einer falschen Anzahl von Argumenten oder die Angabe von unvollständigen Optionen.

DATEIEN

`/etc/ttydefs`

SIEHE AUCH

pmadm(1M), **sacadm(1M)**, **ttymon(1M)**.

Leitfaden für Systemverwalter, "The Port Monitor **ttymon**."

BEZEICHNUNG

`ttymon` – Monitor für Terminalanschlüsse

ÜBERSICHT

`/usr/lib/saf/ttymon`

```
/usr/lib/saf/ttymon -g [ -d gerät ] [ -h ] [ -t zeitschranke ] [ -l tty_marke ]
[ -p prompt ] [ -m moduln ]
```

BESCHREIBUNG

`ttymon` ist ein Monitor für Terminalanschlüsse und setzt auf dem Konzept der STREAMS auf. Neben der Überwachung von Terminalanschlüssen dient er dazu, die Übertragungsprozedur und -geschwindigkeit für die Anschlüsse und die Modi der Terminals festzulegen. Außerdem dient er dazu, Benutzer oder Applikationen mit Diensten zu verbinden, die diesen Anschlüssen zugeordnet sind. Normalerweise ist `ttymon` so konfiguriert, daß er unter SAC abläuft und das Kommando `sacadm` benutzt (SAC steht für Service Access Controller und ist ein Teil der Service Access Facility (SAF)). Jeder `ttymon`-Prozeß kann mehrere Anschlüsse überwachen; diese sind in der Verwaltungsdatei für die Anschlußüberwachung angegeben. Die Verwaltungsdatei wird mit Hilfe der Kommandos `pmadm` und `tyadm` erstellt. Sowie ein `ttymon`-Prozeß durch das Kommando `sac` gestartet wird, beginnt er mit der Anschlußüberwachung. Pro Anschluß initialisiert `ttymon` zunächst die Übertragungsprozedur, sofern eine definiert ist, sowie die Übertragungsgeschwindigkeit und den Terminalmodus. Die Werte dafür werden den passenden Einträgen der Datei für die TTY-Einstellungen entnommen, die mit Hilfe des Kommandos `sttydefs` verwaltet wird. Die Übertragungsprozeduren gemäß Voreinstellung werden gewöhnlich durch das Kommando `autopush` der Autopush Facility vorgenommen.

Danach wartet `ttymon` nach einer Eingabeaufforderung auf Anweisungen des Benutzers. Drückt dieser als Hinweis auf fehlerhafte Übertragungsgeschwindigkeit auf die PAUSE-Taste (BREAK), probiert es `ttymon` mit der nächsten Übertragungsgeschwindigkeit und gibt die Eingabeaufforderung erneut aus. Nach dem Empfang einer richtigen Eingabe wertet `ttymon` eine (eventuell vorhandene) Konfigurationsdatei zu diesem Service aus, erzeugt bei Bedarf einen Eintrag `utmp`, richtet die Service-Umgebung ein und ruft dann den zu diesem Anschluß gehörigen Service auf. Zeichenketten, die gültige Eingaben darstellen, bestehen im Mindestfall aus einem Neue-Zeile-Zeichen und sind mit einem Wagenrücklauf abgeschlossen. Zum Zeitpunkt der Beendigung des Service wird der Eintrag in `utmp` von `ttymon` wieder bereinigt und der ursprüngliche Zustand des Anschlusses wieder hergestellt.

Ist für den Anschluß `autobaud` aktiviert, versucht `ttymon`, die Übertragungsrate des Anschlusses selbsttätig zu bestimmen. Damit `ttymon` die richtige Geschwindigkeit erkennen und danach die Eingabeaufforderung ausgeben kann, muß ein Benutzer die Wagenrücklaftaste betätigen. Zur Zeit sind die durch `autobaud` bestimmbaren Übertragungsgeschwindigkeiten 110, 1200, 2400, 4800 und 9600.

Ist ein Anschluß bidirektional konfiguriert, können Benutzer über diese Schnittstelle Dienste in Anspruch nehmen und für den Fall, daß sie frei ist, erlaubt es `ttymon` den Kommandos `uucico`, `cu` oder `ct`, sie für den Aufbau von Wählverbindungen zu nutzen. Bei bidirektionalen Anschlüssen wartet `ttymon` auf ein Eingabezeichen, bevor er die Eingabeaufforderung ausgibt.

Bei Anschlüssen mit gesetztem *connect-on-carrier* Schalter ruft ttymon beim Eintreffen einer Verbindungsanfrage sofort den zugehörigen Service auf und gibt keine Eingabeaufforderung aus.

Im Fall eines deaktivierten Anschlusses startet ttymon für diesen keinerlei Service. Ist festgelegt, daß für diesen Fall eine entsprechende Meldung ausgegeben werden soll, so gibt ttymon sie beim Eintreffen einer Verbindungsanfrage aus. Wird ttymon deaktiviert, werden damit gleichzeitig auch alle Anschlüsse deaktiviert, die dieser Prozeß bedient.

AUFRUF VON DIENSTEN

Welchen Dienst ttymon für den jeweiligen Anschluß aufruft, ist in der Verwaltungsdatei von ttymon angegeben. Dazu durchsucht ttymon die Zeichenkette, die den Namen des aufzurufenden Dienstes enthält, nach einem Zeichenpaar der Form %d oder %%. Das Zeichenpaar %% wird durch ein einfaches % ersetzt. Im andern Fall ersetzt es ttymon durch den vollen Pfadnamen des Anschlusses (das ist sein Gerätenamen) und verwendet das auf diese Weise veränderte Servicekommando.

Beim Aufruf des Dienstes werden für das angeschlossene Gerät die Dateideskriptoren 0, 1 und 2 zum Lesen und Schreiben geöffnet. Der aufgerufene Service wird von ttymon mit dem aktuellen Verzeichnis und der Benutzer- und Gruppennummer desjenigen Benutzers versorgt, der den Service von ttymon angefordert hat. Außerdem werden die beiden Umgebungsvariablen HOME und TTYPROMPT der Umgebung des Service hinzugefügt. Dabei wird HOME mit dem Home-Verzeichnis des anfordernden Benutzers versorgt und TTYPROMPT mit der Eingabeaufforderung, wie sie für den Service an diesem Anschluß konfiguriert ist. Damit hat ein von ttymon aufgerufener Service die Möglichkeit, herauszufinden, ob eine Eingabeaufforderung von ttymon stammt, und für diesen Fall ihre tatsächliche Form zu bestimmen.

Die Optionen, die für einen unter sac (Service Access Controller) ablaufenden ttymon möglich sind, sind bei der Beschreibung von ttyadm(1M) aufgeführt.

AUFRUF EINES UNABHÄNGIGEN TTYMON-PROZESSES

Die Option -g dient einer besonderen Anwendung von ttymon für Anschlüsse, die nicht unter SAC vorkonfiguriert werden können. Diese Aufrufform sollte ausschließlich von Anwendungsprogrammen benutzt werden, die die Übertragungsrate und die Terminaleinstellungen an einem Anschluß selbst vornehmen müssen und dann Verbindung mit dem login-Prozeß aufnehmen. Folgende Kombinationen mit der Option -g sind erlaubt:

- d *gerät* *gerät* ist der volle Pfadname des Anschlusses, den ttymon überwachen soll. Ist diese Option nicht angegeben, muß der aufrufende Prozeß dafür sorgen, daß der Dateideskriptor 0 auf einen TTY-Anschluß eingestellt wird.
- h Ist der Schalter -h nicht gesetzt, wird ttymon zuerst einen Hängezustand für diese Leitung simulieren, indem er die Baudrate auf Null setzt, bevor er die Übertragungsgeschwindigkeit entweder auf den angegebenen Wert oder aber gemäß Voreinstellung setzt.

ttymon (1M)

ttymon (1M)

- t *zeitschranke*** Legt die Zeitdauer in Sekunden fest, in welcher nach der Ausgabe der Eingabeaufforderung eine Eingabe erfolgen soll. Verstreicht diese Zeit, beendet sich **ttymon**.
- l *tty_marke*** *tty_marke* ist ein Verweis auf einen Eintrag in der Datei **ttydefs**, welchem **ttymon** entnimmt, wie die anfänglichen TTY-Einstellungen sind, welche anfängliche Übertragungsgeschwindigkeit gewählt und welche als nächste ausprobiert werden soll, falls der Benutzer durch Betätigung der PAUSE-Taste (BREAK) signalisiert, daß die Baudrate nicht stimmt. Als Voreinstellung für die Übertragungsgeschwindigkeit gilt 9600 Baud.
- p *prompt*** Damit kann ein Benutzer die Zeichenkette für die Eingabeaufforderung definieren. Die Voreinstellung hierfür ist "Login: ".
- m *moduln*** Bei der Initialisierung des Anschlusses entleert **ttymon** zunächst den Modul-Stapel dieses Anschlusses und packt dann die in *moduln* aufgezählten, durch Kommata getrennten, stapelverträglichen Moduln in der angegebenen Reihenfolge auf den Stapel. Die Standardmoduln eines Anschlusses werden normalerweise durch die Autopush Facility eingestellt.

SIEHE AUCH

pmadm(1M), **sac(1M)**, **sacadm(1M)**, **ttyadm(1M)**.
Leitfaden für Systemverwalter, "The Port Monitor **ttymon**."

HINWEIS

Wird für einen Anschluß mehr als ein **ttymon** aufgerufen, kann das dazu führen, daß mehrere Eingabeaufforderungen ausgegeben werden und es dann zu Überschneidungen beim Abrufen der Eingabe kommt.

BEZEICHNUNG

tunefs – optimiert ein vorhandenes Dateisystem

ÜBERSICHT

tunefs [-a *maxfortl*] [-d *rotzeit*] [-e *maxbzyl*] [-m *minfrei*] [-o [*s|t*]]
gerät | *dateisystem*

BESCHREIBUNG

Mit tunefs können die dynamischen Parameter eines Dateisystems geändert werden, mit denen bestimmt wird, wie das Dateisystem organisiert wird. Das Dateisystem darf nicht eingehängt sein, bzw. muß ausgehängt werden, bevor das Kommando tunefs benutzt wird. Die Parameter, die geändert werden können, werden im Zusammenhang mit den Optionen unten erläutert.

Folgende Optionen stehen zur Verfügung:

- a *maxfortl* Gibt die Höchstzahl der fortlaufenden Blöcke an, die angelegt werden können, bevor eine Verzögerung erzwungen wird (siehe -d unten). Standardwert ist 1, da die meisten Gerätetreiber einen Interrupt pro Plattenzugriff erfordern. Für Gerätetreiber, die mehrere Puffer für eine einzige Übertragung verketteten können, muß dieser Wert auf die Höchstlänge der Verkettung gesetzt werden.
- d *rotzeit* Gibt die erwartete Zeit (in Millisekunden) zur Abarbeitung eines Interrupts an, der die Beendigung einer Übertragung anzeigt, und initiiert eine neue Übertragung auf dieselbe Platte. Der Wert wird benutzt, um anzugeben, wieviel Blockzwischenraum zwischen den aufeinanderfolgenden Blöcken einer Datei erforderlich ist.
- e *maxbzyl* Gibt die Höchstzahl der Blöcke an, die einer Datei aus einer Zylindergruppe zugewiesen werden können, bevor weitere Blöcke aus einer anderen Zylindergruppe zugewiesen werden müssen. In der Regel wird dieser Wert auf etwa ein Viertel der Gesamtzahl der Blöcke in einer Zylindergruppe gesetzt. Der Zweck dieser Funktion liegt darin zu verhindern, daß eine Datei alle Blöcke in einer Zylindergruppe belegt und damit die Zugriffszeit für alle Dateien verlängert, denen später Blöcke in der betreffenden Zylindergruppe zugewiesen werden. Diese Beschränkung bewirkt, daß bei langen Dateien öfter längere Suchoperationen durchgeführt werden müssen, als wenn die Dateien alle Blöcke in einer Zylindergruppe belegen könnten. Bei Dateisystemen, die nur aus langen Dateien bestehen, sollte dieser Parameter auf einen höheren Wert gesetzt werden.
- m *minfrei* Gibt den Prozentsatz von Speicherplatz an, auf den normale Benutzer keinen Zugriff erhalten, d.h. den Schwellenwert für freien Speicherplatz. Als Standardwert gilt 10%. Dieser Wert kann auf Null gesetzt werden. Dann jedoch kann sich der Durchsatz gegenüber dem Wert, der bei einem Schwellenwert von 10% zu erzielen wäre, um den Faktor drei verringern. Anmerkung: Wird ein höherer Wert als der aktuelle Wert festgesetzt, können Benutzer erst dann wieder Dateien zuweisen, wenn so viele Dateien gelöscht wurden, daß der höhere Schwellenwert entsprechend unterschritten wird.

tunefs (1M)

(UFS)

tunefs (1M)

-o {s | t} Ändert die Optimierungsstrategie für das Dateisystem.
s - (space) Speicherplatz (Speicherplatz soll möglichst gering gehalten werden)
t - (time) Zeit (Datei soll so organisiert werden, daß Zugriffszeit minimiert wird)

In der Regel sollte man im Hinblick auf kürzere Zugriffszeiten optimieren, es sei denn, das Dateisystem ist zu über 90% gefüllt.

SIEHE AUCH

mkfs(1M), ufs(4). fork(2), terminfo(4) im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

`uadmin` – Verwaltungssteuerung

ÜBERSICHT

`/sbin/uadmin kmd funk`

BESCHREIBUNG

Das Kommando `uadmin` ermöglicht die Steuerung grundlegender Verwaltungsfunktionen. Dieses Kommando ist eng mit den Systemverwaltungsprozeduren (System Administration Procedures) verbunden und darf nicht von normalen Benutzern aktiviert werden. Nur ein privilegierter Benutzer darf dieses Kommando aufrufen.

Die Argumente *kmd* (Kommando) und *funk* (Funktion) werden in Ganzzahlen umgewandelt und an den Systemaufruf `uadmin` übergeben.

SIEHE AUCH

`uadmin(2)` im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

ufsdump – teilweise Sicherung des Dateisystems

ÜBERSICHT

```
/usr/sbin/ufsdump [ [0-9] [-b faktor] [-c] [-d bpi] [-f dump-datei] [-n]
                  [-s grösse] [-t spuren] [-u] [-w] [-W] ] dateisystem
```

BESCHREIBUNG

ufsdump sichert entweder alle Dateien aus *dateisystem*, oder die, die nach einem bestimmten Datum verändert wurden, auf Band. *optionen* ist eine Zeichenreihe, die die im folgenden beschriebenen Optionen für ufsdump spezifiziert.

Fehlen die Optionen, gilt als Voreinstellung 9u.

OPTIONEN

- | | |
|-----------|---|
| 0-9 | Sicherungsstufe. Es werden alle Dateien des <i>dateisystems</i> auf Band gesichert, die seit der letzten Sicherung mit ufsdump unter Angabe einer geringeren Stufe verändert wurden. Wenn beispielsweise montags eine Sicherung der Stufe 2 durchgeführt wurde, der am Dienstag eine Sicherung auf Stufe 4 folgte, dann würde eine Sicherung auf Stufe 3 am Mittwoch alle Dateien erfassen, die seit der Sicherung auf Stufe 2 vom Montag neu hinzugekommen oder verändert worden wären. Eine Sicherung auf Stufe 0 spielt das gesamte Dateisystem auf das Sicherungsband. |
| -b faktor | Blockungsfaktor. Gibt die Anzahl der Blöcke an, die pro Schreibaufruf auf das Band geschrieben werden. Die Voreinstellung ist 20. Beachten Sie: Aus Kompatibilitätsgründen zu tar wird der Blockungsfaktor in Einheiten von 512 Byte Blöcken angegeben. Bei Bändern mit einer Schreibdichte von mindestens 6250BPI ist die Voreinstellung für den Blockungsfaktor 64. Bei Bandkassetten (-c Schalter angeben) ist die Voreinstellung für den Blockungsfaktor 126. |
| -c | Bandkassette. Gibt an, daß statt eines Halbzoll-Bands eine Bandkassette verwendet wird. Damit wird die Schreibdichte auf 1000BPI und der Blockungsfaktor auf 126 gesetzt. Die Bandlänge wird auf 425 Fuß eingestellt. Dieser Schalter ist mit der Option -d unverträglich, es sei denn, daß dort eine Schreibdichte von 1000BPI spezifiziert ist. |
| -d bpi | Schreibdichte. Mit <i>bpi</i> wird die Schreibdichte des Bands in Einheiten von BPI spezifiziert. Damit wird laufend buchgeführt, wieviel Platz pro Band zur Verfügung steht. Außer für Bandkassetten ist die Voreinstellung für die Schreibdichte 1600. Sofern nicht eine höhere Schreibdichte angegeben ist, verwendet ufsdump immer die Voreinstellung, auch wenn das Bandlaufwerk zu einer höheren Schreibdichte fähig ist (z. B. 6250BPI). Achtung: Die spezifizierte Schreibdichte sollte mit jener des verwendeten Bandlaufwerks übereinstimmen, denn sonst kann ufsdump die Bandende-Bedingung nicht korrekt behandeln. |

- f *dump-datei*** Sicherungsdatei. Statt auf die Gerätedatei `/dev/rmt8` soll die Sicherung in die Datei *dump-datei* geschrieben werden. Wenn *dump-datei* als `-` spezifiziert ist, wird die Sicherung auf die Standardausgabe gelenkt.
- n** Benachrichtigen aller in der Gruppe der Operatoren, deren Aufmerksamkeit von `ufsdump` benötigt wird, in ähnlicher Form wie dies mittels des Kommandos `wall` erfolgt.
- s *grösse*** Gibt die *grösse* des Sicherungsmediums an. Sowie die angegebene Kapazität erreicht ist, wartet `ufsdump`, bis ein neues Band zur Verfügung steht. `ufsdump` interpretiert die angegebene Kapazität bei Bändern und Bandkassetten als Länge, gemessen in Fuß, bei Disketten hingegen als Anzahl der 1 KB - Blöcke. Die jeweilige Voreinstellung ist wie folgt:
- | | |
|--------------|--|
| Bandkassette | 425 Fuß |
| Diskette | 1422 Blöcke (Das entspricht einer 1.44 MB Diskette, wobei ein Zylinder für defekte Blöcke reserviert ist.) |
- t *spuren*** Gibt bei Bandkassetten die Anzahl der Spuren an, wobei die Voreinstellung 9 Spuren ist. Dieser Schalter (`-t`) darf nicht zusammen mit der Option `-D` angegeben werden.
- u** Aktualisierung des Sicherungsprotokolls. Für jedes erfolgreich gesicherte Dateisystem wird ein Eintrag in die Datei `/etc/dumpdates` vorgenommen, welcher auch den Namen des Dateisystems, das Datum und die Sicherungsstufe enthält. Diese Datei kann vom Superuser verändert werden.
- w** Auflisten aller Dateien, die gesichert werden sollen. Die diesbezügliche Information wird den Dateien `/etc/dumpdates` und `/etc/vfstab` entnommen. Wird der Schalter `-w` angegeben, werden alle anderen Optionen verworfen. `ufsdump` gibt die Liste aus und beendet sich danach sofort.
- W** Ähnelt der `-w` Option. Allerdings werden bei Angabe des Schalters `-W` alle Dateisysteme aufgelistet, die in der Datei `/etc/dumpdates` enthalten sind. Zusätzlich wird jeweils Datum und Stufe der letzten Sicherung mit aufgelistet. Dateisysteme, die gesichert werden sollen, sind hervorgehoben.

ufsdump (1M)

(UFS)

ufsdump (1M)

HINWEIS

Wenn beim Lesen des Dateisystems die Anzahl der Lesefehler unter 32 bleibt, werden sie ignoriert.

Für jedes neue Band wird ein eigener Prozeß erzeugt. Das hat zur Folge, daß Vaterprozesse bis zur Beendigung der gesamten Sicherung weiter existieren.

Außerdem ist es ratsam, teilweise Sicherungen im Einbenutzer-Betrieb durchzuführen.

DATEIEN

/dev/dump

Voreinstellung des Zielgeräts für die Sicherung

/etc/dumpdates

Liste mit dem Datum der jeweiligen Sicherungen

/etc/group

Information über den Operator der Gruppe

/etc/hosts

SIEHE AUCH

tar(1), wall(1), shutdown(1M), ufsrestore(1M).

BEZEICHNUNG

ufsrestore – Teilweise Wiederherstellung des Dateisystems

ÜBERSICHT

/usr/sbin/ufsrestore *optionen* [*dateiname...*]

BESCHREIBUNG

ufsrestore restauriert Dateien von Sicherungsbändern, die mit dem Kommando `ufsdump` erzeugt worden sind. *optionen* ist eine Zeichenreihe, die mindestens eine der unten angegebenen Optionen enthält. Die Zeichenreihe kann mit `-` beginnen (`-xy`) oder nur aus den Optionen bestehen (`xy`). Die Angabe `-x -y` ist nicht erlaubt. Nach der Zeichenreihe können Argumente zu den Optionen spezifiziert werden. Weitere Parameter für `ufsrestore` werden als Namen der Dateien (bzw. der Verzeichnisse, deren Dateien) interpretiert, die auf die Platte zurückzuspielen sind. Ist ein Verzeichnis spezifiziert, ohne daß die Option `h` angegeben ist, bezieht sich das auf alle darin enthaltenen Dateien sowie (auf rekursive Weise) auf alle darin enthaltenen Verzeichnisse.

Gültige Optionen sind:

- i (Interaktiv) Nachdem die Verzeichnisstruktur von Band eingelesen ist, wechselt `ufsrestore` in den Dialog-Modus, der dem Benutzer erlaubt, sich quer durch den Dateibaum auf Band zu bewegen und einzelne Dateien für das Rückspielen auf Platte auszuwählen. Die dafür zur Verfügung stehenden Kommandos sind weiter unten unter *Interaktive Kommandos* aufgeführt.
- r (restore) Wiederherstellung aller auf dem Band gespeicherten Dateien. Der Bandinhalt wird komplett in das aktuelle Verzeichnis eingespielt. Dieser Schalter sollte ausschließlich dafür verwendet werden, um entweder eine komplette Sicherung in ein neues Dateisystem wieder einzuspielen oder um nach einer vollständigen Wiederherstellung auf Stufe 0 eine teilweise Sicherung durchzuführen.
- R (resume) Neuaufsetzen bei der Wiederherstellung. `ufsrestore` fordert ein bestimmtes Band einer aus mehreren Bändern bestehenden Sicherung an, mit dem bei einer vollständigen Wiederherstellung neu aufgesetzt werden soll (vgl. die Option `r` weiter oben). Damit hat `ufsrestore` die Möglichkeit, an einem Kontrollpunkt neu aufzusetzen, falls während einer vollständigen Wiederherstellung eine Unterbrechung erfolgte.
- t (table of contents) Inhaltsverzeichnis. Führt alle *Dateinamen* auf, die auch auf Band vorkommen. Falls *dateiname* nicht spezifiziert ist, wird das Root-Verzeichnis aufgelistet. Sofern die Option `h` nicht benutzt wird, hat das zur Folge, daß eine Liste aller Dateien auf Band erstellt wird.
- x (extract) Einspielen der angegebenen Dateien von Band. Falls einer der spezifizierten Dateinamen ein auf dem Band enthaltenes Verzeichnis bezeichnet, wird dessen Inhalt rekursiv wieder eingepielt, sofern die Option `h` nicht angegeben ist. Eigentümer, Datum der letzten Änderung und Zugriffsrechte werden übernommen (falls möglich). Falls *dateiname* nicht spezifiziert ist, wird das

- Root-Verzeichnis, also das gesamte Band eingespielt, sofern die Option `h` nicht benutzt wird.
- `c` (convert) Konvertiert den Inhalt des Sicherungsbands auf das neue Dateiformat.
- `d` (Debug). Einschalten der Ausgabe von Informationen für die Fehlersuche.
- `h` Es werden lediglich die Dateien des aktuellen Verzeichnisses und keine darin enthaltenen Unterverzeichnisse eingespielt. Damit wird die Übernahme kompletter Teilbäume verhindert.
- `m` Um die Neugenerierung kompletter Pfadnamen zu vermeiden, soll der Zugriff auf die einzuspielenden Dateien über den Indexeintrag erfolgen. Dies ist recht nützlich, wenn nur wenige Dateien eingespielt werden sollen.
- `v` (verbose). `ufsrestore` gibt Dateityp und Name jeder Datei aus, die wieder eingespielt wird.
- `y` Kein Nachfragen, ob bei auftretenden Fehlern während des Einlesens des Bandes abgebrochen werden soll. `ufsrestore` versucht, nicht lesbare Blöcke auf dem Band zu überspringen und nach bestem Vermögen neu aufzusetzen.
- `b faktor` Blockungsfaktor. Legt den Blockungsfaktor für das Lesen von Band fest. Gemäß Voreinstellung versucht `ufsrestore`, die Blockgröße des Bandes selbst herauszufinden.
Beachten Sie: auf Bändern hat ein Block 512 Byte.
- `f sich_datei` Statt der Datei `/dev/rmt8` soll die Datei `sich_datei` als Quelle für die Wiederherstellung benutzt werden. Falls für `sich_datei` `'-'` angegeben ist, liest `ufsrestore` von der Standard-Eingabe. Damit ist es möglich, `ufsdump(1M)` und `ufsrestore` in einer Pipe zur Sicherung und Wiederherstellung zu verwenden:

```
#Beispiel
ufsdump 0f - /dev/rdisk/0s3 | (cd /mnt; ufsre-
store xf -)
```

`s n` Bei Bändern, die mehrere Sicherungen enthalten, Vorlauf zur Sicherung Nummer *n*. Das folgende Kommando

```
ufsrestore xfs /dev/rmt/c0s0n 5
```

würde beispielsweise zur fünften Sicherung auf dem Band positionieren.

Wenn `ufsrestore` mit der `i` Option aufgerufen wird, wechselt es in den Dialog-Modus. Die Kommandos in diesem Modus erinnern an Shell-Kommandos. Als Voreinstellung für alle Kommandos, die ein Argument haben kann, gilt das aktuelle Verzeichnis.

`ls[verzeichnis]` Erstellen einer Liste der Dateien in dem Verzeichnis *verzeichnis* oder im aktuellen Verzeichnis, welches durch einen `.` (Punkt) dargestellt wird. Weitere Verzeichnisse werden mit einem führenden `/` (Schrägstrich) angehängt. Für ein Wiedereinspielen vorgesehene Einträge sind mit einem führenden `*` (Stern) gekennzeichnet. Falls die Option für Anzeige (`v`) benutzt wird, werden auch die I-Nodes mit aufgeführt.

`cd verzeichnis` Wechsel in das Verzeichnis *verzeichnis* (auf dem Sicherungsband).

`pwd` Ausgabe des vollen Pfadnamens des aktuellen Verzeichnisses.

`add[dateiname]` Die Liste der wieder einzuspielenden Dateien wird erweitert entweder um das aktuelle Verzeichnis, um die Datei *dateiname* oder um das Verzeichnis *dateiname*. Handelt es sich bei dem angegebenen Dateinamen um ein Verzeichnis, wird dieses samt aller Unterverzeichnisse (in rekursiver Weise) der oben genannten Liste hinzugefügt (sofern die Option `h` nicht benutzt wird).

`delete[dateiname]` Streichen des aktuellen Verzeichnisses, der spezifizierten Datei oder des spezifizierten Verzeichnisses aus der Liste der wieder einzuspielenden Dateien. Handelt es sich bei der Angabe um ein Verzeichnis, so wird dieses samt aller Unterverzeichnisse (rekursiv) aus der Liste entfernt (sofern die `h` Option nicht benutzt wird). Die eleganteste Art, die Mehrzahl der Dateien in einem Verzeichnis zum Wiedereinspielen auszuwählen, besteht darin, das Verzeichnis der Liste hinzuzufügen und daraus dann die nicht gewünschten Dateien zu streichen.

`extract` Einspielen aller Dateien, die in der Liste der wieder einzuspielenden Dateien enthalten sind. `ufsrestore` erfragt das vom Benutzer einzulegende Band. Der schnellste Weg, eine kleine Anzahl von Dateien wieder einzuspielen, besteht darin, mit dem letzten Band zu beginnen und sich zum ersten hin vorzuarbeiten.

`verbose` Ein-/Ausschalter für die Option `v`. Wenn die Option `v` eingeschaltet ist, werden beim `ls` Kommando auch die Indexeinträge ausgegeben und `ufsrestore` listet alle eingespielten Dateien auf.

help	Ausgabe aller verfügbaren Kommandos.
quit	Sofortige Beendigung von <code>ufsrestore</code> , auch wenn die Liste der einzuspielenden Dateien nicht leer ist.

HINWEIS

`ufsrestore` kann durcheinanderkommen, wenn es eine teilweise Wiederherstellung von Sicherungsbändern durchführt, die bei in Betrieb befindlichen Dateisystemen erzeugt wurden.

Nach einer vollständigen Wiederherstellung muß eine Sicherung auf Stufe 0 durchgeführt werden. Da `ufsrestore` im Benutzermodus abläuft, hat es keinerlei Kontrolle über die Zuteilung von Indexeinträgen. Deswegen werden die Dateien neu angeordnet, obgleich ihre Inhalte von `ufsrestore` nicht verändert werden. Um also eine neue Verzeichnisstruktur zu erhalten, die die Neuordnung der Dateien widerspiegelt, muß eine vollständige Sicherung durchgeführt werden, so daß spätere teilweise Sicherungen korrekt ablaufen.

FEHLERMELDUNGEN

`ufsrestore` moniert falsche Optionen.

Lesefehler werden ebenfalls moniert. Falls der Benutzer mit `y` antwortet oder die Option `y` spezifiziert ist, versucht `ufsrestore` fortzufahren.

Falls die Sicherung mehr als ein Band belegt, fordert `ufsrestore` den Benutzer auf, die Bänder zu wechseln. Sind die Optionen `x` oder `i` gesetzt, erfragt `ufsrestore` zusätzlich, welches Band der Benutzer einlegen möchte.

`ufsrestore` kann verschiedene Meldungen über Konsistenzprüfungen ausgeben. Die meisten sind selbsterklärend. Bestimmte Prüfungen sind oft hinfällig. Im folgenden sind häufig auftretende Fehler erläutert.

Converting to new file system format.

Es ist ein Band eingelegt worden, das eine Sicherung des alten Dateisystems enthält. Es wird automatisch auf das Format des neuen Dateisystems konvertiert.

dateiname: not found on tape

Die Datei *dateiname* war im Verzeichnis des Bandes aufgeführt, konnte aber auf dem Band nicht gefunden werden. Das kann einerseits von Bandlesefehlern während der Suche nach dieser Datei herrühren und auch davon, daß die Sicherung von einem in Betrieb befindlichen Dateisystem erfolgte.

expected next file *inumber*, got *inumber*

Es wurde eine Datei gefunden, die nicht im Verzeichnis enthalten war. Dies kann dann auftreten, wenn die Sicherung von einem in Betrieb befindlichen Dateisystem erfolgte.

Incremental tape too low

Tritt während einer teilweisen Wiederherstellung auf und bedeutet: Das aktuelle Band ist zeitlich vor dem zuletzt eingelegten beschrieben worden oder es hat eine zu niedrige Sicherungsstufe.

Incremental tape too high

Tritt während einer teilweisen Wiederherstellung auf und bedeutet: Das aktuelle Band schließt nicht an die Sicherungsüberdeckung des vorhergehenden Bandes an oder hat eine zu hohe Sicherungsstufe.

Tape read error while restoring *dateiname*

Tape read error while skipping over inode *inumber*

Tape read error while trying to resynchronize

A tape read error has occurred.

Ist ein Dateiname *dateiname* angegeben, so ist wahrscheinlich der Inhalt dieser Datei teilweise zerstört. Falls ein Indexeintrag (inode) übersprungen wird oder versucht wird, das Band neu zu synchronisieren, kann es zwar sein, daß manche Dateien auf dem Band nicht gefunden werden, die eingespielten Dateien sind jedoch alle in Ordnung.

resync ufsrestore, skipped *num*

Nach einem Lesefehler muß ufsrestore möglicherweise neu aufsetzen. Diese Meldung gibt Auskunft über die Anzahl der übersprungenen Blöcke.

DATEIEN

/dev/rmt8

Voreinstellung der Bandgerätedatei, wenn der Aufruf als ufsrestore erfolgt

/tmp/rstdir*

Datei, die die Verzeichnisse auf Band enthält

/tmp/rstmode*

Eigentümer, Zugriffsrechte und Zeitstempel für Verzeichnisse

./restoresymtable

Information, die zwischen teilweisen Wiederherstellungen ausgetauscht wird.

SIEHE AUCH

ufsdump(1M), mkfs(1M), mount(1M).

unshare(1M)

(DFS)

unshare(1M)

BEZEICHNUNG

`unshare` – Sperren lokaler Ressourcen gegen Einhängerversuche ferner Systeme

ÜBERSICHT

`unshare [-F dateisystem-typ] [-o spezifische_optionen] [pfadname | ressource]`

BESCHREIBUNG

Das Kommando `unshare` sorgt dafür, daß eine gemeinsam nutzbare lokale Ressource für das Dateisystem vom Typ *dateisystem-typ* nicht mehr verfügbar ist. Ist die Option `-F dateisystem-typ` für den Dateisystem-Typ nicht angegeben, wird der erste Dateisystemeintrag in der Datei `/etc/dfs/fstypes` als Voreinstellung benutzt. Die Optionen *spezifische_optionen* und die Schreibweise des Namens der *ressource* sind dateisystemspezifisch und hängen vom ausgelieferten Dateisystem ab.

DATEIEN

`/etc/dfs/fstypes`
`/etc/dfs/sharetab`

SIEHE AUCH

`share(1M)`, `shareall(1M)`.

FEHLERMELDUNGEN

Wird in der Information zu gemeinsamer Datennutzung einer der beiden Begriffe *pfadname* oder *ressource* nicht gefunden, erfolgt eine Fehlermeldung auf die Standardausgabe.

unshare(1M)

(NFS)

unshare(1M)

BEZEICHNUNG

unshare - Sperren lokaler NFS-Ressourcen gegen Einhängerversuche ferner Systeme

ÜBERSICHT

unshare [-F nfs] *pfadname*

BESCHREIBUNG

Das Kommando unshare sperrt lokale Ressourcen gegen Einhängerversuche ferner Systeme. Für die gemeinsame Ressource muß es in der Datei `/etc/dfs/sharetab` eine entsprechende Zeile geben, in der, als Typ für das Dateisystem, NFS eingetragen ist. Falls der erste Dateisystem-Eintrag in der Datei `/etc/dfs/fstypes` den Typ NFS bezeichnet, kann die Option -F weggelassen werden.

DATEIEN

`/etc/dfs/fstypes`
`/etc/dfs/sharetab`

SIEHE AUCH

share(1M)

BEZEICHNUNG

useradd – Einrichtung einer neuen Benutzerkennung im System

ÜBERSICHT

```
/usr/sbin/useradd [-u uid [-o] [-g gruppe] [-G gruppe[,gruppe...]] [-d verz] [-s shell]
                  [-c kommentar] [-m [-k skel_verz]] [-f inaktiv] [-e verfall] login
```

```
/usr/sbin/useradd -D [-g gruppe] [-b basis_verz] [-f inaktiv] [-e verfall]
```

BESCHREIBUNG

Der Aufruf von useradd ohne die Option -D bewirkt den Eintrag eines neuen Benutzers in die Dateien /etc/passwd und /etc/shadow. Im Bedarfsfall werden außerdem zusätzliche Gruppenrechte (-G Option) eingerichtet und das Home-Verzeichnis (-m Option) erzeugt. Die neue Kennung bleibt gesperrt, bis das passwd(1M)-Kommando ausgeführt wird.

Der Aufruf useradd -D ohne weitere Optionen bewirkt die Ausgabe der Voreinstellungen für *gruppe*, *basis_verz*, *skel_verz*, *shell*, *inaktiv* und *verfall*. Diese Werte für *gruppe*, *basis_verz*, *inaktiv*, *verfall* und *shell* werden beim Aufruf ohne die Option -D verwendet.

Der Aufruf useradd -D zusammen mit -g, -b, -f oder -e (oder einer beliebigen Kombination davon) weist den entsprechenden Feldern ihre Voreinstellungen zu. (Nach der Installation ist die Voreinstellung für Gruppe *other* (Gruppen-ID 1) und die Voreinstellung für *basis_verz* ist /home). Weitere Aufrufe von useradd ohne die Option -D verwenden dann diese Werte.

Die Systemdatei-Einträge, die mit diesem Kommando erzeugt werden können, sind auf 512 Byte beschränkt. Werden den einzelnen Optionen sehr lange Argumente zugeordnet, kann dieses Limit möglicherweise überschritten werden.

Gültige Optionen sind:

- u *uid* Die Benutzernummer UID des neuen Benutzers. Diese UID muß eine positive ganze Zahl sein und kleiner sein als die Obergrenze MAXUID, die in <param.h> definiert ist. Per Voreinstellung wird der neuen UID der nächsthöhere (eindeutige) Wert der aktuell höchsten UID zugewiesen. Sind zum Beispiel die UIDs 100, 105 und 200 bereits vergeben, dann ist die neue UID per Voreinstellung 201. (Die UIDs von 0 bis 99 sind reserviert.)
- o Mit dieser Option ist es möglich, eine UID doppelt zu vergeben (die dann nicht eindeutig ist).
- g *gruppe* Eine bereits existierende Gruppennummer als ganze Zahl oder ein Name in Form einer Zeichenkette. Fehlt die Option -D, so wird damit die Hauptgruppe des neuen Benutzers definiert. Ist die Option -g ganz weggelassen, gilt die Voreinstellung für Gruppe. Diese Voreinstellung kann durch den Aufruf useradd -D -g *gruppe* aufgehoben werden.
- G *gruppe* Eine bereits existierende Gruppennummer als ganze Zahl oder ein Name in Form einer Zeichenkette. Damit wird eine zusätzliche Gruppenzugehörigkeit des neuen Benutzers definiert. Wird die gleiche Gruppennummer *gruppe* sowohl bei der Option -g als auch

- bei `-G` angegeben, bleibt dies unberücksichtigt. Die Obergrenze für Gruppennummern liegt bei `NGROUPS_MAX`.
- `-d verz` Das Home-Verzeichnis des neuen Benutzers. Als Voreinstellung gilt *basis_verz/login*, wobei *basis_verz* das Basis-Verzeichnis für Home-Verzeichnisse neuer Benutzer darstellt und *login* die neue Kennung ist.
 - `-s shell` Der volle Pfadname des Programms, das nach dem Login des Benutzers als Shell gestartet werden soll. Als Voreinstellung bleibt das entsprechende Feld leer und das System startet `/sbin/sh`. Der Wert von *shell* muß ein gültiges ausführbares Programm sein.
 - `-c kommentar` Beliebiger Text. Im allgemeinen handelt es sich hier um eine kurze Beschreibung der Kennung und das entsprechende Feld wird zur Zeit für den kompletten Benutzernamen verwendet. All diese Informationen werden in dem zu diesem Benutzer gehörenden Eintrag in der Datei `/etc/passwd` abgelegt.
 - `-m` Legt das Home-Verzeichnis des neuen Benutzers an, sofern es nicht bereits existiert. Im Fall der Existenz muß es für *gruppe* die Lese-, Schreib- und Ausführberechtigung aufweisen, wobei *gruppe* die Nummer der Hauptgruppe des Benutzers ist.
 - `-k skel_verz` Verzeichnis, welches provisorisch Informationen bereitstellt, die in das Home-Verzeichnis des neuen Benutzers kopiert werden können (wie z.B. `.profile`). Dieses Verzeichnis muß existieren. Standardmäßig wird vom System ein solches Verzeichnis `/etc/skel` für diesen Zweck zur Verfügung gestellt.
 - `-e verfall` Das Verfallsdatum dieser Login-Kennung. Nach diesem Zeitpunkt ist es nicht mehr möglich, mit dieser Kennung in das System zu gelangen. (Diese Option ist für temporäre Kennungen nützlich.) Der Wert für das Verfallsdatum (*verfall*) kann in jedem beliebigen Format (außer dem Julianischen) angegeben werden. Beispielsweise kann `10/6/90` oder `October 8, 1990` eingegeben werden. Eine Eingabe von `" "` hebt die Wirkung des Verfallsdatums auf.
 - `-f inaktiv` Maximale Anzahl von Tagen, bis die Login-Kennung *ID* für gültig erklärt wird. Normalwerte sind ganzzahlige Angaben, wogegen ein Wert von `-1` diese Wirkung aufhebt.
 - login* Eine Zeichenkette von abdruckbaren Zeichen, die die Login-Kennung des neuen Benutzers darstellt. Sie darf weder einen Doppelpunkt (`:`) noch das Zeichen für Zeilenvorschub (`\n`) enthalten.
 - `-b basis_verz` Voreinstellung für das Basis-Verzeichnis im System. Falls `-d verz` nicht angegeben ist, wird an *basis_verz* die Login-Kennung des Benutzers angehängt und so das Home-Verzeichnis des Benutzers definiert. Ist `-m` nicht angegeben, muß *basis_verz* bereits existieren.

useradd (1M)

useradd (1M)

DATEIEN

/etc/passwd
/etc/shadow
/etc/group
/etc/skel

SIEHE AUCH

groupadd(1M), groupdel(1M), groupmod(1M), logins(1M), passwd(1),
passwd(1M), userdel(1M), usermod(1M), users(1).

EXIT-CODES

Die Exit-Codes von `useradd` haben folgende Bedeutung:

- 0 = Kommando ist fehlerfrei abgelaufen
- 2 = ungültige Kommandozeile; es wird ein Hinweis auf die korrekte Aufrufform von `useradd` ausgegeben
- 3 = Argument für eine Option war fehlerhaft
- 4 = *uid* als Argument für die `-u` Option ist bereits vergeben
- 6 = Gruppe *gruppe* als Argument für die `-g` Option existiert nicht
- 9 = Kennung *login* existiert bereits
- 10 = es ist nicht möglich, die Datei `/etc/group` zu aktualisieren; die neue Kennung ist in `/etc/passwd` eingetragen worden, nicht aber in `/etc/group`
- 12 = entweder konnte das Home-Verzeichnis (mit der `-m` Option) nicht angelegt werden oder das Verzeichnis *skel_verz* konnte nicht in das Home-Verzeichnis kopiert werden

BEZEICHNUNG

`userdel` – Löschen einer Benutzerkennung auf dem Rechner

ÜBERSICHT

`/usr/sbin/userdel [-r] login`

BESCHREIBUNG

Das Kommando `userdel` löscht eine Benutzerkennung auf dem Rechnersystem. Es entfernt die Zugangsberechtigung für die angegebene Login-Kennung und führt die entsprechenden Änderungen in den Systemdateien und im Dateisystem durch.

Gültige Optionen sind:

- `-r` Entfernung des Home-Verzeichnisses des Benutzers aus dem System. Nach erfolgreicher Ausführung des Kommandos ist kein Zugriff auf die Dateien und Unterverzeichnisse des Home-Verzeichnisses mehr möglich.
- `login` Eine Zeichenkette aus abdruckbaren Zeichen, die eine existierende Login-Kennung am Rechner darstellt. Sie darf weder einen Doppelpunkt (:) noch das Zeichen für Zeilenvorschub enthalten (`\n`).

DATEIEN

`/etc/passwd`
`/etc/shadow`
`/etc/group`

SIEHE AUCH

`groupadd(1M)`, `groupdel(1M)`, `groupmod(1M)`, `logins(1M)`, `passwd(1)`, `passwd(1M)`, `useradd(1M)`, `usermod(1M)`, `users(1)`.

EXIT-CODES

Die Exit-Codes von `userdel` haben folgende Bedeutung:

- 0 = fehlerfreie Ausführung
- 2 = unzulässige Syntax; es wird ein Hinweis auf die korrekte Aufrufform von `userdel` ausgegeben
- 6 = spezifizierte Login-Kennung gibt es nicht
- 8 = unter der zu löschenden Login-Kennung wird gerade gearbeitet
- 10 = es ist nicht möglich, die Datei `/etc/group` zu aktualisieren, die Kennung ist jedoch aus `/etc/passwd` entfernt worden
- 12 = Home-Verzeichnis kann nicht verändert oder gelöscht werden

BEZEICHNUNG

`usermod` – Änderung der Login-Kennung eines Benutzers im System

ÜBERSICHT

```
/usr/sbin/usermod [-u uid [-o] [-g gruppe] [-G gruppe [, gruppe...]] [-d verz [-m]]
                  [-s shell] [-c kommentar] [-l neue_kennung] [-f inaktiv] [-e verfall] login
```

BESCHREIBUNG

Das Kommando `usermod` ändert die Parameter einer Login-Kennung am System und nimmt die entsprechenden Änderungen am Dateisystem und den zu dieser Kennung gehörenden Systemdateien vor.

Die Systemdatei-Einträge, die mit diesem Kommando erzeugt werden können, sind auf 512 Bytes beschränkt. Werden den einzelnen Optionen sehr lange Argumente zugeordnet, kann dieses Limit möglicherweise überschritten werden.

Gültige Optionen sind:

- `-u uid` Neue Benutzernummer UID für den Benutzer. Sie muß eine positive ganze Zahl sein und kleiner sein als die Obergrenze MAXUID, die in `<param.h>` definiert ist.
- `-o` Mit dieser Option ist es möglich, eine UID doppelt zu vergeben (die dann nicht eindeutig ist).
- `-g gruppe` Eine bereits existierende Gruppennummer als ganze Zahl oder ein Name in Form einer Zeichenkette. Damit wird die vorrangige Gruppenzugehörigkeit des Benutzers neu definiert.
- `-G gruppe` Eine bereits existierende Gruppennummer als ganze Zahl oder ein Name in Form einer Zeichenkette. Damit wird die zusätzliche Gruppenzugehörigkeit des Benutzers neu definiert. Wird die gleiche Gruppennummer *gruppe* sowohl bei der Option `-g` als auch bei `-G` angegeben, bleibt dies unberücksichtigt. Die Obergrenze für Gruppennummern liegt bei NGROUPS_MAX, wie sie in `<param.h>` definiert ist.
- `-d verz` Das neue Home-Verzeichnis des neuen Benutzers. Als Voreinstellung gilt `basis_verz/login`, wobei *basis_verz* das Basis-Verzeichnis für neue Home-Verzeichnisse darstellt und *login* die neue Kennung ist.
- `-m` Verlagert das Home-Verzeichnis des Benutzers in das mittels der Option `-d` spezifizierte Verzeichnis. Falls dieses bereits existiert, muß es für *gruppe* die Lese-, Schreib- und Ausführberechtigung aufweisen, wobei *gruppe* die Nummer der Hauptgruppe des Benutzers ist.
- `-s shell` Der volle Pfadname des Programms, das nach dem Login des Benutzers als Shell gestartet werden soll. Der Wert von *shell* muß ein gültiges ausführbares Programm sein.

usermod(1M)

usermod(1M)

<code>-c <i>kommentar</i></code>	Beliebiger Text. Im allgemeinen handelt es sich hier um eine kurze Beschreibung der Kennung und das entsprechende Feld wird zur Zeit für den kompletten Benutzernamen verwendet. All diese Informationen werden in dem zu diesem Benutzer gehörenden Eintrag in der Datei <code>/etc/passwd</code> abgelegt.
<code>-l <i>neue_kennung</i></code>	Eine Zeichenkette von abdruckbaren Zeichen, die die neue Login-Kennung des Benutzers darstellt. Sie darf weder einen Doppelpunkt (:) noch das Zeichen für Zeilenvorschub (\n) enthalten.
<code>-e <i>verfall</i></code>	Das Verfallsdatum dieser Login-Kennung. Nach diesem Zeitpunkt ist es nicht mehr möglich, mit dieser Kennung in das System zu gelangen. (Diese Option ist für temporäre Kennungen nützlich.) Der Wert für das Verfallsdatum (<i>verfall</i>) kann in jedem beliebigen Format (außer dem Julianischen) angegeben werden. Beispielsweise kann <code>10/6/90</code> oder <code>October 8, 1990</code> eingegeben werden. Eine Eingabe von <code>"</code> hebt die Wirkung des Verfallsdatums auf.
<code>-f <i>inaktiv</i></code>	Maximale Anzahl von Tagen, bis die Login-Kennung <code>ID</code> für gültig erklärt wird. Normalwerte sind ganzzahlige Angaben, wogegen ein Wert von <code>-1</code> diese Wirkung aufhebt.
<code><i>login</i></code>	Eine Zeichenkette von abdruckbaren Zeichen, die die Login-Kennung des neuen Benutzers darstellt. Sie darf weder einen Doppelpunkt (:) noch das Zeichen für Zeilenvorschub (\n) enthalten.

DATEIEN

`/etc/passwd`
`/etc/shadow`
`/etc/group`

SIEHE AUCH

`groupadd(1M)`, `groupdel(1M)`, `groupmod(1M)`, `logins(1M)`, `passwd(1)`, `passwd(1M)`, `useradd(1M)`, `userdel(1M)`, `users(1)`.

EXIT-CODES

Die Exit-Codes von `useradd` haben folgende Bedeutung:

- 0 = Kommando ist fehlerfrei abgelaufen
- 2 = ungültige Kommandozeile; es wird ein Hinweis auf die korrekte Aufrufform von `useradd` ausgegeben
- 3 = Argument für eine Option war fehlerhaft
- 4 = *uid* als Argument für die Option `-u` ist bereits vergeben
- 6 = Gruppe *gruppe* oder die Login-Kennung, die verändert werden soll, existiert nicht
- 8 = unter der Login-Kennung, die verändert werden soll, wird gerade gearbeitet

usermod (1M)

usermod (1M)

- 9 = Kennung *neue_kennung* existiert bereits
- 10 = es ist nicht möglich, die Datei */etc/group* zu aktualisieren; die anderen Änderungen werden eingetragen
- 11 = zu wenig Platz, um das Home-Verzeichnis zu verlagern (*-m* Option); die anderen Änderungen werden eingetragen
- 12 = es ist nicht möglich, das alte Home-Verzeichnis vollständig in das neue Verzeichnis zu verlagern

BEZEICHNUNG

uucheck – Überprüfen der Verzeichnisse und Zugriffsrechte für UUCP

ÜBERSICHT

/usr/lib/uucp/uucheck [-v] [-x *überwachungsstufe*]

BESCHREIBUNG

uucheck prüft die Vollständigkeit der für das uucp System erforderlichen Dateien und Verzeichnisse. Von uucheck werden die Berechtigungen in der *Permissions*-Datei (/etc/uucp/Permissions) auf Fehler untersucht.

Die Optionen haben folgende Bedeutung:

-v Erzeugt eine ausführliche Beschreibung der Berechtigungen in der *Permissions*-Datei, so wie UUCP sie interpretieren würde.

-x *überwachungsstufe*

überwachungsstufe ist eine Zahl von 0 bis 9; dabei bedeutet eine höhere Zahl eine ausführlichere Überwachungsinformation.

uucheck wird bei der Installation von UUCP ausgeführt und kann nur von einem privilegierten Benutzer oder von UUCP selbst benutzt werden.

DATEIEN

/etc/uucp/Systems
/etc/uucp/Permissions
/etc/uucp/Devices
/etc/uucp/Limits
/var/spool/uucp/*
/var/spool/locks/*
/var/spool/uucppublic/*

SIEHE AUCH

uucico(1M), uusched(1M).
uucp(1C), uustat(1C), uux(1C) in den *Kommandos*.

HINWEIS

Das Programm überprüft nicht die gesetzten Zugriffsrechte auf Verzeichnisse und Dateien; ebenso werden auch logische Fehler in der *Permissions*-Datei (doppelte Benutzer- oder System-Namen) nicht entdeckt.

BEZEICHNUNG

uucico – Datenübertragungsprogramm für UUCP

ÜBERSICHT

/usr/lib/uucp/uucico *[optionen]*

BESCHREIBUNG

uucico ist für die Übertragung von uucp-Dateien verantwortlich. Die folgenden Optionen werden unterstützt:

- c *typ* Der erste Eintrag in der Devices-Datei ist der Typeintrag. Mit der Option -c wird uucico veranlaßt, nur diejenigen Typeneinträge zu verwenden, die mit dem vorgegebenen Typ *typ* übereinstimmen. Üblicherweise gibt dieser Typ den Namen eines lokalen Netzes an.
- d *spool_verzeichnis* Mit dieser Option wird das Verzeichnis *spool_verzeichnis* angegeben, das die zu übertragenden Dateien für UUCP enthält. Als Standardverzeichnis wird /var/spool/uucp angenommen.
- f Diese Option "erzwingt" die Ausführung von uucico, indem eine in der Datei /etc/uucp/Limits festgelegte Höchstzahl von "uucicos" ignoriert wird.
- i *schnittstelle* Die Definition einer Schnittstelle *schnittstelle* betrifft nur die Verwendung von uucico als Slave. Mögliche Schnittstellen sind UNIX (Standard), TLI (Transport Layer Interface), und TLIS (Transport Layer Interface mit STREAMS Lese- und Schreib-Modulen).
- r *rolle_nr* Die *rolle_nr* 1 legt die Verwendung von uucico als Master, die *rolle_nr* 0 die Verwendung als Slave (Standard) fest. Beim Start durch ein Programm oder cron sollte mit *rolle_nr* 1 der Master-Modus eingestellt werden.
- s *system_name* Die Option -s bestimmt das ferne System (*system_name*), zu dem uucico eine Verbindung aufbauen soll. Diese Option muß bei Verwendung (Rolle) als Master angegeben werden; *system_name* muß vorher in der Systems-Datei definiert werden.
- x *überwachungsstufe* Sowohl uux als auch uucp reihen Aufträge für uucico in eine Warteschlange ein. Diese Aufträge werden normalerweise von uusched für die Fehlersuche gestartet, aber können auch manuell gestartet werden. So startet beispielsweise das Uutry Kommando uucico mit eingeschaltetem Debugging. Die *überwachungsstufe* ist eine Zahl zwischen 0 und 9. Eine höhere Zahl bedeutet eine ausführlichere Überwachungsinformation.

DATEIEN

/etc/uucp/Systems
/etc/uucp/Permissions
/etc/uucp/Devices
/etc/uucp/Devconfig
/etc/uucp/Sysfiles
/etc/uucp/Limits
/var/spool/uucp/*
/var/spool/locks/*
/var/spool/uucppublic/*

SIEHE AUCH

cron(1M), uused(1M), Uutry(1M).
uucp(1C), uustat(1C), uux(1C) in den *Kommandos*.

BEZEICHNUNG

uucleanup – Aufräumen des UUCP Spool-Verzeichnisses

ÜBERSICHT

/usr/lib/uucp/uucleanup [*optionen*]

BESCHREIBUNG

uucleanup durchsucht das Spool-Verzeichnis nach alten Dateien und ergreift geeignete Maßnahmen, um diese Dateien zu entfernen:

Informieren des Auftraggebers über Sende- und Empfangs-Aufträge mit fern Systemen, zu denen keine Verbindung hergestellt werden konnte.

Schickt liegengeliebene Nachrichten (Mail) an den Absender zurück.

Bearbeitet rnews-Dateien, die für das lokale System bestimmt sind.

Löscht alle anderen Dateien.

Zusätzlich gibt es die Möglichkeit, Benutzer auf Aufträge hinzuweisen, die länger als eine bestimmte Anzahl von Tagen (Standard ist 1 Tag) auf Bearbeitung warten. Hinweis: uucleanup wird so ausgeführt, als ob alle *anzahl* Optionen (siehe unten) Standardwerte angenommen hätten, mit Ausnahme der Fälle, in denen *anzahl* explizit gesetzt ist.

Die folgenden Optionen werden unterstützt:

- C *anzahl* Alle C. Dateien, die mindestens *anzahl* Tage (Standard ist 7 Tage) oder mehr alt sind, werden, mit einer entsprechenden Meldung an den Auftraggeber, gelöscht.
- D *anzahl* Alle D. Dateien, die mindestens *anzahl* Tage (Standard ist 7 Tage) oder mehr alt sind, werden gelöscht. Falls erforderlich, wird vorher versucht, Nachrichten mittels mail zu senden bzw. das rnews Kommando auszuführen.
- W *anzahl* Alle C. Dateien, die *anzahl* Tage (Standard ist 1 Tag) alt sind, führen dazu, daß eine Nachricht mit mail an den Auftraggeber geschickt wird, um auf die Verzögerung des Auftrags hinzuweisen. Die Nachricht enthält die JOBID und im Falle eines mail Auftrags die zu übermittelnde Nachricht. Der Systemverwalter kann weiterhin eine Nachrichtenzeile vorsehen, um die für das Problem zuständige Person mitzuteilen (-m Option).
- X *anzahl* Alle X. Dateien, die mindestens *anzahl* Tage (Standard ist 2 Tage) oder mehr alt sind, werden gelöscht. Die Ursache liegt wahrscheinlich im Fehlen der D. Dateien, denn sonst könnten die X. Dateien ausgeführt werden. Falls hingegen die Dateien vorhanden sind, werden sie wie oben beschrieben behandelt.
- m *text* Fügt den *text* in die Nachricht ein, die mit der Option -W erzeugt wird (siehe oben).

uucleanup(1M)

uucleanup(1M)

`-o anzahl` Alle anderen Dateien, die älter als *anzahl* Tage (Standard 2 Tage) sind, werden gelöscht.

Als Standardtext wird der Hinweis "See your local administrator to locate the problem" ausgegeben.

`-s system` Ausführung lediglich für das Spool-Verzeichnis des Systems *system*.

`-x überwachungsstufe`

Die *überwachungsstufe* ist eine Zahl zwischen 0 und 9; eine höhere Zahl bedeutet eine ausführlichere Überwachungsinformation (diese Option ist nicht auf allen Systemen verfügbar).

Dieses Programm wird gewöhnlich durch das `uudemon.cleanup`-Kommando gestartet, das wiederum von `cron(1M)` aufgerufen wird.

DATEIEN

`/usr/lib/uucp` Verzeichnis mit den Kommandos, die von `uucleanup` intern verwendet werden

`/var/spool/uucp` Spool-Verzeichnis

SIEHE AUCH

`cron(1M)`.

`uucp(1C)`, `uux(1C)` in den *Kommandos*.

BEZEICHNUNG

uusched – Auftragsverwaltung für die UUCP-Datenübertragung

ÜBERSICHT

/usr/lib/uucp/uusched [*optionen*]

BESCHREIBUNG

uusched ist die Auftragsverwaltung für die uucp(1C) Datenübertragung und wird normalerweise durch den Dämon uudemond.hour gestartet, der wiederum von cron(1M) durch einen entsprechenden Eintrag in der Datei /var/spool/cron/crontab aktiviert wird:

```
41,11 **** /usr/bin/su uucp -c "/usr/lib/uucp/uudemond.hour >
/dev/null"
```

Die Optionen dienen ausschließlich der Fehlersuche.

-u *überwachungsstufe*

Die Option -u *überwachungsstufe* wird dem uucico(1M) Kommando als -x *überwachungsstufe* übermittelt.

-x *überwachungsstufe*

Liefert Fehlermeldungen von uusched(1M). *überwachungsstufe* ist eine Zahl zwischen 0 und 9. Eine höhere Zahl bedeutet eine ausführlichere Fehlerinformation.

DATEIEN

/etc/uucp/Systems
/etc/uucp/Permissions
/etc/uucp/Devices
/var/spool/uucp/*
/var/spool/locks/*
/var/spool/uucppublic/*

SIEHE AUCH

cron(1M), uucico(1M).
uucp(1C), uustat(1C), uux(1C) in den *Kommandos*.

BEZEICHNUNG

Uutry - Ansprechversuch eines fernen Systems mit eingeschalteter Fehlerüberwachung

ÜBERSICHT

/usr/lib/uucp/Uutry [-c *typ*] [-r] [-x *überwachungsstufe*] *system_name*

BESCHREIBUNG

Uutry stellt eine eigene Shell dar, in der uucico aufgerufen wird, um einen fernen Rechner anzusprechen. Gleich zu Beginn wird die Fehlerüberwachung mit dem Standardwert 5 eingeschaltet. Die Ausgabe der Fehlerüberwachung wird in die Datei /tmp/*system_name* geleitet. Die Optionen haben folgende Bedeutung:

- c *typ* In der Datei Devices ist das erste Feld das "Typ"-Feld. Die Option -c erzwingt, daß uucico ausschließlich solche Einträge im "Typ"-Feld benutzt, die mit *typ* übereinstimmen, wie es der Benutzer spezifiziert hat. Gewöhnlich ist der angegebene *typ* der Name eines lokalen Netzwerks (LAN).
- r Mit dieser Option wird die in der Datei /var/uucp/.status/*system_name* angegebene Zeitspanne für Wiederholungsversuche außer Kraft gesetzt.
- x *überwachungsstufe* Stellt den Umfang der Fehlerüberwachung von 0 bis 9 ein. Je höher der Wert, desto umfangreicher ist die Information zur Fehlerüberwachung.

DATEIEN

/etc/uucp/Systems
 /etc/uucp/Permissions
 /etc/uucp/Devices
 /etc/uucp/Limits
 /var/spool/uucp/*
 /var/spool/locks/*
 /var/spool/uucppublic/*
 /tmp/*system_name*

SIEHE AUCH

uucico(1M).
 uucp(1C), uux(1C) in den *Kommandos*.

BEZEICHNUNG

uuxqt – Ausführen von Aufträgen eines fernen Systems

ÜBERSICHT

/usr/lib/uucp/uuxqt [-s *system*] [-x *überwachungsstufe*]

BESCHREIBUNG

uuxqt ist das Programm, das Aufträge eines fernen Systems ausführt, die zuvor mit dem uux-Kommandos generiert wurden (*Mail* benutzt uux, um Nachrichten in einem fernen System zu verarbeiten). uuxqt durchsucht die Spool-Verzeichnisse nach Aufträgen, die von einem fernen System hinterlegt worden sind. Für jeden vorliegenden Auftrag überprüft uuxqt, ob alle erforderlichen Dateien vorhanden und zugreifbar sind, und ob die gewünschten Kommandos für das zugehörige ferne System zulässig sind. Die Datei `Permissions` dient dabei zur Überprüfung von Zugriffsrechten auf Dateien und Benutzungsrechten von Kommandos.

Es werden zwei Umgebungsvariablen vor der Ausführung des uuxqt-Kommandos gesetzt:

UU_MACHINE bestimmt das ferne System, das den letzten Auftrag gesendet hat.

UU_USER bestimmt den Benutzer (Absender) des Auftrags im fernen System.

Beide Variablen können dazu verwendet werden, Befehle für Fehlersuche, Protokollierung und besondere Einschränkungen an ein fernes System zu übermitteln. Die Optionen haben folgende Bedeutung:

-s *system* Name (*system*) des fernen Systems.

-x *überwachungsstufe*

Die *überwachungsstufe* wird als Zahl von 0 bis 9 angegeben. Eine höhere Zahl bedeutet eine ausführlichere Überwachungsinformation.

DATEIEN

/etc/uucp/Permissions

/etc/uucp/Limits

/var/spool/uucp/*

/var/spool/locks/*

SIEHE AUCH

uucico(1M).

uucp(1C), uustat(1C), uux(1C), mail(1) in den *Kommandos*.

BEZEICHNUNG

`volcopy` (generisch) – physische Kopie des Dateisystems erzeugen

ÜBERSICHT

`volcopy [-F dstyp] [-V] [aktuelle_optionen] [-o spezifische_optionen] operanden`

BESCHREIBUNG

Das Kommando `volcopy` erzeugt eine physische Kopie des Dateisystems.

aktuelle_optionen sind die vom `s5`-spezifischen Modul von `volcopy` unterstützten Optionen. In anderen Dateisystemen werden diese Optionen nicht notwendigerweise unterstützt. Bei *spezifischen_optionen* werden Optionen angegeben, die von den dateisystemtyp-spezifischen Modulen dieses Kommandos ausgewertet werden; sie sind in einer Aufzählung, durch Komma getrennt, angegeben und sind entweder einfache Optionen oder ein Paar aus Schlüsselwort und Parameter.

Die *operanden* sind dateisystemspezifisch und geben im allgemeinen den Namen von Gerät und Datenträger mit an. In den dateisystemtyp-spezifischen Handbuchseiten zu `volcopy` sind die *operanden* detailliert beschrieben.

Die Optionen haben folgende Bedeutung:

- F Legt den Typ des Dateisystems fest, auf dem gearbeitet werden soll; er sollte entweder hier angegeben sein oder aber aus der Datei `/etc/vfstab` bestimmbar sein, indem ein Eintrag in der Tabelle zu *operanden* paßt.
- V Die gesamte Kommandozeile wird auf dem Bildschirm ausgegeben, jedoch nicht ausgeführt. Sie wird mit den vom Benutzer angegebenen Optionen und Argumenten aufbereitet, wobei noch die Information aus `/etc/vfstab` hinzugefügt wird. Diese Option dient der Überprüfung der Kommandozeile auf Richtigkeit.
- o Dient der Angabe dateisystemtyp-spezifischer Optionen.

HIMWEIS

Dieses Kommando wird nicht notwendigerweise von allen Dateisystemen unterstützt.

DATEIEN

`/etc/vfstab`

Liste mit den Standardparametern für die Dateisysteme

SIEHE AUCH

`vfstab(4)`.

Beschreibung der dateisystemtyp-spezifischen Handbuchseiten zu `volcopy`.

BEZEICHNUNG

volcopy (s5) – Erzeugen einer physischen Kopie eines s5-Dateisystems

ÜBERSICHT

volcopy [-F s5] [*gen_optionen*] [-a] *dsname qu_gesät bandname1 ziel_gesät bandname2*

BESCHREIBUNG

gen_optionen sind Optionen, die von dem generischen Kommando volcopy unterstützt werden.

volcopy erzeugt eine physische Kopie eines s5-Dateisystems. Dabei wird eine Blockgröße verwendet, die zum angegebenen Gerät paßt.

Gültige Optionen sind:

-F s5 Legt s5 als Dateisystem-Typ fest.

-a Bevor die Kopie erzeugt wird, wird statt der üblichen Verzögerung von 10 Sekunden eine Bestätigungsaufforderung ausgegeben, die von einem Operator positiv quittiert werden muß.

Das Argument *dsname* steht für den Namen des eingehängten Dateisystems, das kopiert werden soll (wie z.B. root, u1, usw.).

Die Namen *qu_gesät* oder *ziel_gesät* sollten ein logisches Dateisystem bezeichnen, wie z.B. /dev/rdisk/0s3.

bandname ist die Bezeichnung des physischen Datenträgers. Solche Namen sind auf sechs oder weniger Zeichen beschränkt. Für *bandname* kann ' - ' angegeben werden. In diesem Fall wird der bereits existierende Name des Datenträgers verwendet.

qu_gesät und *bandname1* sind Gerätenamen und Namen des Datenträgers, von wo das Dateisystem kopiert wird. *ziel_gesät* und *bandname2* sind die Namen für Gerät und Datenträger für das Ziel der Kopie.

dsname und *bandname* sind im Superblock aufgezeichnet (char *fsname*[6], *volname*[6];).

DATEIEN

/var/adm/filesave.log

Protokolldatei mit den kopierten Dateisystemen und Datenträgern

SIEHE AUCH

dd(1M), labelit(1M), generisches volcopy(1M), cpio(4), fs(4).
cpio(1), sh(1) in den *Kommandos*.

BEZEICHNUNG

volcopy (ufs) – Erzeugen einer physischen Kopie eines ufs-Dateisystems

ÜBERSICHT

```
volcopy [-F ufs] [gen_optionen] [-a] dsname qu _gerät bandname1
        ziel _gerät bandname2
```

BESCHREIBUNG

gen_optionen sind Optionen, die von dem generischen Kommando volcopy unterstützt werden.

volcopy erzeugt eine physische Kopie eines ufs-Dateisystems. Dabei wird eine Blockgröße verwendet, die zum angegebenen Gerät paßt.

Das Argument *dsname* steht für den Namen des eingehängten Dateisystems, das kopiert werden soll (wie z.B. root, u1, usw.).

Die Namen *qu _gerät* oder *ziel _gerät* sollten ein logisches Dateisystem bezeichnen, wie z.B. /dev/rdisk/0s3.

bandname ist die Bezeichnung des physischen Datenträgers. Solche Namen sind auf sechs oder weniger Zeichen beschränkt. Für *bandname* kann ' - ' angegeben werden. In diesem Fall wird der bereits existierende Name des Datenträgers verwendet.

qu _gerät und *bandname1* sind Gerätenamen und Name des Datenträgers, von wo das Dateisystem kopiert wird. *ziel _gerät* und *bandname2* sind die Namen für Gerät und Datenträger für das Ziel der Kopie.

dsname und *bandname* sind im Superblock aufgezeichnet (char fsname[6], volname[6];).

Gültige Optionen sind:

- F ufs Legt ufs als Dateisystem-Typ fest.
- a Bevor die Kopie erzeugt wird, wird statt der üblichen Verzögerung von 10 Sekunden eine Bestätigungsaufforderung ausgegeben, die von einem Operator positiv quittiert werden muß.

DATEIEN

/var/adm/filesave.log

Protokolldatei mit den kopierten Dateisystemen und Datenträgern

SIEHE AUCH

dd(1M), labelit(1M), generisches volcopy(1M).

cpio(1) in den Kommandos.

cpio(4), ufs(4) im Referenzhandbuch für Programmierer.

wall(1M)

wall(1M)

BEZEICHNUNG

wall – Meldung an alle Benutzer schicken

ÜBERSICHT

/usr/sbin/wall [-g group] [dateiname ...]

BESCHREIBUNG

wall liest die angegebenen Dateien oder, falls keine spezifiziert sind, von Standardeingabe bis zum Dateiendezeichen und sendet dann diese Meldung an alle Benutzer, die momentan am Rechner arbeiten. Der Meldung wird der folgende Text vorangestellt:

Broadcast Message from ...

Das Kommando wird dazu benutzt, um die Benutzer zu warnen, z.B. davor, daß gleich das System heruntergefahren wird.

Ist die Option -g spezifiziert, dann wird die Meldung statt an alle Benutzer nur an die Mitglieder der angegebenen Gruppe geschickt.

Der Absender muß privilegiert sein, um eventuelle Sperren gegen Nachrichten seitens der Benutzer zu umgehen (vgl. `msg(1)`).

wall wendet `setgid()` (siehe `setuid(2)`) auf das tty zu der Gruppennummer an, um die Schreiberlaubnis auf die Terminals anderer Benutzer zu erhalten.

wall überprüft den Text auf nicht abdruckbare Zeichen, bevor es ihn an ein Benutzerterminal schickt. Steuerzeichen werden in der Form '^', gefolgt von dem entsprechenden ASCII-Zeichen, dargestellt. Zeichen mit gesetztem höchstwertigen Bit werden in Meta-Notation dargestellt. Beispielsweise wird \003 als '^C' ausgegeben und '\372' als 'M-z'.

DATEIEN

/dev/tty*

SIEHE AUCH

`msg(1)`, `write(1)`.

HINWEIS

Für den Fall, daß das tty eines Benutzers nicht zum Schreiben geöffnet werden kann, erfolgt die Meldung 'Cannot send to ...'

BEZEICHNUNG

whodo – wer tut was

ÜBERSICHT

/usr/sbin/whodo [-h] [-l] [*benutzer*]

BESCHREIBUNG

whodo erzeugt aus der Information in den Dateien /var/adm/utmp, /etc/ps_data und /proc/pid eine datierte und formatierte Ausgabe.

Die Ausgabeüberschrift enthält Datum, Zeit und Rechnernamen. Für jeden angemeldeten Benutzer wird der Gerätenamen, die Benutzernummer UID und der Zeitpunkt der Anmeldung angezeigt, gefolgt von einer Liste der zu dieser UID gehörenden aktiven Prozesse. Die Liste enthält den Gerätenamen, die Prozeßnummer PID, die verbrauchte CPU-Zeit in Minuten und Sekunden sowie den Prozeßnamen.

Ist *benutzer* angegeben, ist die Ausgabe auf alle zu diesem Benutzer gehörenden Sitzungen beschränkt.

Gültige Optionen sind:

- h Keine Ausgabe der Überschrift.
- l Ausgabe in ausführlicher Form. Die einzelnen Ausgabefelder sind der Login-Name des Benutzers, die tty-Bezeichnung des Terminals, an dem der Benutzer arbeitet, die Tageszeit (in *Stunde:Minuten*), zu der sich der Benutzer angemeldet hat, die Leerlaufzeit, d.h. die verstrichene Zeit, seit der der Benutzer zum letzten Mal eine Eingabe machte (in *Stunde:Minuten*), die verbrauchte CPU-Zeit aller Prozesse samt Sohnprozessen an diesem Terminal (in *Minuten:Sekunden*), die verbrauchte CPU-Zeit des momentan aktiven Prozesses (in *Minuten:Sekunden*) und der Name samt den Argumenten des aktuellen Prozesses.

BEISPIEL

Das Kommando

```
whodo
```

erzeugt eine Ausgabe in der folgenden Form:

```
Tue Mar 12 15:48:03 1985
bailey

tty09mcn  8:51
          tty09 28158 0:29 sh

tty52bdr  15:23
          tty52 21688 0:05 sh
          tty52 22788 0:01 whodo
          tty52 22017 0:03 vi
          tty52 22549 0:01 sh
```

whodo(1M)

whodo(1M)

```
xt162 lee 10:20
tty08 6748 0:01 layers
xt162 6751 0:01 sh
xt163 6761 0:05 sh
tty08 6536 0:05 sh
```

DATEIEN

```
/etc/passwd
/etc/ps_data
/var/adm/utmp
/proc/pid
```

EXIT-CODES

whodo kann nicht korrekt ablaufen, wenn der PROC-Treiber nicht installiert oder konfiguriert ist, oder wenn `/proc` nicht eingehängt ist. Es wird dann eine diesbezügliche Meldung ausgegeben.

Der Exit-Code ist bei fehlerfreiem Ablauf gleich Null, andernfalls ungleich Null.

SIEHE AUCH

`ps(1)`, `who(1)` in den *Kommandos*.

BEZEICHNUNG

xfscck – Prüfung und Reparatur von XENIX-Dateisystemen

ÜBERSICHT

/bin/xfscck [optionen] [dateisystem] ...

BESCHREIBUNG

Das Kommando *xfscck* prüft XENIX System V Dateisysteme und repariert Inkonsistenzen im Dialog mit dem Benutzer. Ist das Dateisystem in Ordnung, so gibt *xfscck* die Anzahl der Dateien, die Anzahl der belegten und der freien Blöcke aus. Ist das Dateisystem inkonsistent, fragt *xfscck* den Benutzer bei jeder möglichen Reparatur, ob diese durchgeführt werden soll. Es sollte betont werden, daß die meisten Korrekturmaßnahmen einen gewissen Datenverlust zur Folge haben, dessen Umfang und Bedeutung den Fehlermeldungen entnommen werden können. Hat der Benutzer keine Schreiberlaubnis, verhält sich *xfscck* so, als sei die Option *-n* spezifiziert.

xfscck kennt folgende Optionen:

- y* Geht davon aus, daß alle Rückfragen von *xfscck* mit *yes* beantwortet werden.
- n* Geht davon aus, daß alle Rückfragen von *xfscck* mit *no* beantwortet werden. Bei dieser Option wird das Dateisystem nicht zum Schreiben geöffnet.
- s b:c* Ignoriert die aktuelle Liste der freien Blöcke und erzwingt die Erstellung einer neuen, indem der Superblock des Dateisystems zurückgeschrieben wird. Dabei muß das Dateisystem unbedingt ausgehängt sein.

Mit dieser Option ist es möglich, eine optimale Organisation der Liste der freien Blöcke zu erzeugen. Es werden folgende Formate unterstützt:

- s*
 - sBlöcke-pro-Zylinder:Lücken-Blöcke* (Interleave des Dateisystems)
- Ist *b:c* nicht angegeben, so werden die gleichen Werte wie bei der Erzeugung des Dateisystems benutzt. Wurden bei der Erzeugung keine Werte spezifiziert, werden die Werte gemäß Voreinstellung verwendet.
- S* Bedingte Neuerzeugung der Liste der freien Blöcke. Diese Option ähnelt der obigen *-s b:c* Option. Im Unterschied dazu wird die Liste der freien Blöcke nur dann neu aufgebaut, wenn das Dateisystem keine Inkonsistenzen aufweist. Die *-S* Option erzwingt die Antwort "no" auf alle Rückfragen seitens *xfscck*. Diese Option dient der Reorganisation der Liste der freien Blöcke in einwandfreien Dateisystemen.
- t* *xfscck* benutzt das nachfolgende Argument als temporäre Datei, sofern es eine braucht. Das ist dann der Fall, wenn *xfscck* nicht genügend Speicher für seine Tabellen bekommen kann. Ist die Option *-t* nicht angegeben, fordert *xfscck* den Namen der temporären Datei vom Benutzer an. Diese Datei sollte nicht im Dateisystem liegen, das gerade überprüft wird. Außerdem wird die temporäre Datei nach Beendigung von *xfscck* gelöscht, sofern sie nicht bereits existiert hat oder eine Gerätedatei ist. Hat das System eine große Festplatte, kann es allerdings sein, daß der Platz für die temporäre Datei auf einem anderen Dateisystem nicht ausreicht. In

solchen Fällen kann man, sofern am System ein Disketten-Laufwerk vorhanden ist, eine leere, formatierte Disketten dafür verwenden und z.B. `/dev/fd0` als temporäre Datei angeben.

- q `xfscck` führt die Prüfung ohne Meldungen durch. In Phase 1 werden keine Meldungen bezüglich der Prüfung der Größe ausgegeben. Nicht referenzierte `fifo5` Dateien werden selektiv gelöscht. Sofern es `xfscck` erfordert, werden im Superblock Zählereinträge automatisch korrigiert und die Liste der freien Blöcke wird bereinigt.
- D Überprüft Verzeichnisse auf zerstörte Blöcke. Diese Option sollte nach Systemabstürzen verwendet werden.
- f `xfscck` führt eine schnelle Prüfung durch. `xfscck` prüft nur Blöcke und Größen (Phase 1) und die Liste der freien Blöcke (Phase 5). Falls nötig, wird die Liste der freien Blöcke rekonstruiert (Phase 6).
- rr Stellt das root-Dateisystem wieder her. Das erforderliche Argument *dateisystem* muß sich auf das root-Dateisystem beziehen und am besten auf das Blockgerät (normalerweise `/dev/root`). Diese Option impliziert `-y` (yes) und setzt `-n` (no) außer Kraft. Müßten irgendwelche Änderungen am Dateisystem vorgenommen werden, wird das System automatisch heruntergefahren, um die Integrität des Dateisystems zu gewährleisten.
- c Konvertiert alle Dateisysteme, die unterstützt werden, auf den aktuellen Dateisystem-Typ. Für jedes Dateisystem wird der Benutzer aufgefordert, die Konvertierung zu bestätigen, sofern nicht die Option `-y` gesetzt ist. Es wird empfohlen, jedes Dateisystem mit dieser Option zu überprüfen, während es ausgehängt ist, wenn es mit der aktuellen Version von XENIX verwendet werden soll. Um das im Betrieb befindliche root-Dateisystem zu aktualisieren, kann es mit der folgenden Kommandozeile überprüft werden:

```
xfscck -c -rr /dev/root
```

Sind keine *dateisysteme* angegeben, liest `xfscck` gemäß Voreinstellung eine Liste von Dateisystemen aus der Datei `/etc/checklist` ein.

Im folgenden sind einige der Inkonsistenzen aufgezählt, die `xfscck` überprüft:

- Blöcke, die von mehr als einem Indexeintrag oder von der Liste der freien Blöcke beansprucht werden
- Blöcke außerhalb des Dateisystems, die von einem Indexeintrag oder von der Liste der freien Blöcke beansprucht werden
- Fehlerhafte Dateiverweiszähler
- Prüfung der Größe:
 - Fehlerhafte Anzahl von Blöcken
 - Die Verzeichnisgröße ist nicht auf 16 Bytes ausgerichtet
- Fehlerhaftes Format eines Indexeintrags
- Nirgends zugeordnete Blöcke

- Prüfung von Verzeichnissen:
Eine Datei verweist auf einen nicht zugewiesenen Indexeintrag
Unzulässige Nummer eines Indexeintrags
- Prüfung des Superblocks:
Mehr als 65536 Indexeinträge
Die Anzahl der Blöcke für Indexeinträge überschreitet die der im Dateisystem vorhandenen
- Fehlerhaftes Format der Liste der freien Blöcke
- Fehlerhafte Summe der freien Blöcke oder der freien Indexeinträge

Mit der Erlaubnis des Benutzers macht `xfscck` verwaiste Dateien (das sind angelegte Dateien ohne Verweis darauf) und Verzeichnisse wieder zugänglich, indem es sie in dem Verzeichnis `lost+found` ablegt. Dabei wird ihnen die Nummer ihres Indexeintrags als Name zugeordnet. Beachten Sie, daß das Verzeichnis `lost+found` bereits angelegt sein muß und dort freie Einträge existieren. Um das zu bewerkstelligen, kopieren Sie einfach einige Dateien in das Verzeichnis `lost+found` und löschen sie anschließend wieder (bevor Sie `xfscck` aufrufen).

DATEIEN

`/etc/checklist` Liste der zu prüfenden Dateisysteme gemäß Voreinstellung
`/etc/default/boot` Parameter, die das automatische Hochfahren steuern

SIEHE AUCH

`fsck(1M)`

HINWEIS

`xfscck` läuft nicht, wenn das Dateisystem eingehängt und als blockorientiert definiert ist, es sei denn, es handelt sich um das root-Dateisystem oder die Option `-n` ist gesetzt und es werden keine Schreibzugriffe auf das Dateisystem stattfinden. Jeder Versuch in dieser Richtung wird von `xfscck` mit einer Warnung quittiert und das Dateisystem auf der angegebenen Gerätedatei wird nicht weiter bearbeitet.

`xfscck` unterstützt keine Dateisysteme, die unter XENIX-86 Version 3.0 erzeugt worden sind. Der Grund dafür ist die geänderte Wortreihenfolge in Variablen vom Typ *long*. Unter der Voraussetzung, daß die Wortreihenfolge stimmt, ist `xfscck` allerdings in der Lage, ein Dateisystem der XENIX Version 3.0 zu prüfen und zu reparieren.

Für das root-Dateisystem wird `xfscck -rr /dev/root` aufgerufen. Für alle anderen *ausgehängten* Dateisysteme auf einem blockorientierten Gerät wird `xfscck /dev/??` aufgerufen.

Es ist nicht zu empfehlen, `xfscck` auf Geräte anzusetzen, die im Raw-Modus betrieben werden. Auch wenn die Prüfung eines Geräts im Raw-Modus fast immer schneller abläuft, ist es dort nicht möglich festzustellen, ob das Dateisystem eingehängt ist. Bereinigungen in einem eingehängten Dateisystem werden nahezu mit Sicherheit Inkonsistenzen im Superblock erzeugen.

BEZEICHNUNG

xinstall – Shell-Skript für XENIX-Installationen

ÜBERSICHT

/etc/xinstall [gerät]

BESCHREIBUNG

/etc/xinstall dient der Installierung von XENIX-Vertriebsprodukten (oder Anwenderprogrammen) auf Disketten. Es ist als sh(1) Prozedur ausgelegt und erfüllt die folgenden Aufgaben:

- Es fordert die Disketten an.
- Es spielt die Dateien mittels des Dienstprogramms tar(1) ein.
- Es führt die Programme /once/init.* aus, nachdem sie von der Diskette eingelesen wurden.
- Es löscht die Programme /once/init.*, nachdem die Installation abgeschlossen ist.

Das optionale Argument bezeichnet das verwendete Gerät. Per Voreinstellung ist es die Gerätedatei /dev/xinstall, die normalerweise über einen Dateiverweis auf /dev/rdsk/f0q15dt zeigt.

DATEIEN

/etc/xinstall

/once/init.*

SIEHE AUCH

custom(1M), fixperm(1M), installpkg(1).

HINWEIS

xinstall ist dafür gedacht, beliebige XENIX-Pakete, die Sie bereits besitzen, unter dem Betriebssystem SINIX zu installieren. xinstall funktioniert nicht für SINIX-Anwendungen (um SINIX-Anwendungen zu installieren, wird installpkg(1) verwendet).

BEZEICHNUNG

`xrestore`, `xrestor` – teilweise Wiederherstellung von XENIX-Dateisystemen

ÜBERSICHT

`xrestore schl [argumente]`

`xrestor schl [argumente]`

BESCHREIBUNG

`xrestore` dient dem Lesen von Archivierungen, die mit dem XENIX-Sicherungskommando `backup(C)` erzeugt worden sind. Der Schlüssel `schl` gibt an, was zu tun ist. Er ist einer der Buchstaben `rRxT` und kann mit `f` zusammen spezifiziert werden. `xrestor` ist eine andere Schreibweise für dasselbe Kommando.

`f` Verwendet das nachfolgende *argument* als Archivnamen statt der Voreinstellung.

`F anz` Spezifiziert die Anzahl der wiederherzustellenden Dateien auf dem ersten Band.

`k gröÙe` Spezifiziert die Größe des Sicherungsbandes.

`r , R` Die Sicherung wird in das Dateisystem eingelesen, das durch *argument* spezifiziert ist. Dies sollte nicht leichtfertig getan werden (s. unten). Ist für den Schlüssel `schl` der Buchstabe `R` angegeben, so erfragt `xrestore`, mit welchem Band innerhalb einer mehrbändigen Sicherung begonnen werden soll. Damit ist es möglich, mit `xrestore` nach einer Unterbrechung neu aufzusetzen (wobei vorher noch `fsck` aufgerufen werden muß).

`x` Es wird jede Datei aus der Liste *argumente* vom Archivierungsband eingelesen. Dabei werden aus den Dateinamen die Namensanteile entfernt, die sich auf das eingehängte Dateisystem beziehen. Ist beispielsweise das Dateisystem `/usr` eingehängt, so lautet der Name von `/usr/bin/lpr` auf dem Archivierungsband `/bin/lpr`. Der eingelesenen Datei wird als Name eine Zahl zugeordnet, die von `xrestore` vergeben wird (und nichts anderes als die Nummer des Indexeintrags ist). Um den Aufwand für das Einlesen vom Sicherungsmedium auf ein Minimum zu reduzieren, wird folgende Methode vorgeschlagen:

1. Legen Sie das erste Band einer mehrbändigen Sicherung ein.
2. Rufen Sie `xrestore` auf.
3. `xrestore` meldet zurück, ob es die angegebenen Dateien finden konnte, welche Nummer es der jeweiligen Datei zuordnen wird und spult für den Fall, daß das Sicherungsmedium ein Band ist, das Band zurück.
4. Danach werden Sie aufgefordert, "das gewünschte Band einzulegen", woraufhin Sie die Bandnummer des eingelegten Bandes eingeben. Bei einer mehrbändigen Sicherung sollten Sie so vorgehen, daß Sie die Bänder in umgekehrter Reihenfolge einlegen, also mit dem letzten beginnen und das erste zuletzt. `restore` prüft, ob sich welche der verlangten Dateien auf dem eingelegten Band befinden (oder einem aktuelleren – daher die umgekehrte Reihenfolge). Wenn nicht, wird das Band gar nicht erst gelesen. Besteht die Archivierung ohnehin nur

aus einem Band oder aber, wenn die Anzahl der zu restaurierenden Dateien sehr groß ist, antworten Sie mit 1, woraufhin `xrestore` die Archivierungsbänder sequentiell durcharbeiten wird.

X dateien

Legt die Dateien in dem mit *argumente* spezifizierten Verzeichnis ab.

t Gibt das Datum aus, wann das Archivierungsmedium geschrieben wurde und wann das Dateisystem gesichert wurde.

T Damit verhält sich `xrestore` wie `dumpdir(C)`, mit dem Unterschied, daß Verzeichnisse nicht aufgelistet werden.

Die Option **r** sollte ausschließlich dafür verwendet werden, eine komplette Sicherung in ein bereinigtes Dateisystem einzuspielen oder um nach einer solchen Aktion eine teilweise Wiederherstellung durchzuführen. Eine typische Kommando-sequenz zum vollständigen Einspielen einer Sicherung ist die folgende:

```
/etc/mkfs /dev/dsk/0s3 10000
xrestore r /dev/dsk/0s3
```

Darauf aufbauend könnte ein nachfolgender Aufruf von `xrestore` eine teilweise Wiederherstellung durchführen.

Um ein Dateisystem zu vergrößern, ruft man üblicherweise zuerst `backup` auf, dann `mkfs` und abschließend `xrestore`.

DATEIEN

`rst*` temporäre Dateien

`/etc/default/xrestore` Name des Standardgeräts für die Archivierung

Das Standardgerät für die Archivierung ist je nach Installation verschieden.

HINWEIS

`xrestore` steht aus Kompatibilitätsgründen für XENIX zur Verfügung und sollte lediglich für das Einspielen von Archiven, die unter XENIX erzeugt wurden, verwendet werden.

Es ist nicht möglich, ein in Betrieb befindliches `root`-Dateisystem erfolgreich zu restaurieren.

FEHLERMELDUNG

Es gibt verschiedene Fehlermeldungen bezüglich des Lesens vom Archiv und des Schreibens auf Platte. Weitere Meldungen dienen der Mitteilung, daß die Liste freier Indexeinträge oder der freien Blöcke auf dem Dateisystem für das Rückspielen der Sicherung nicht ausreicht.

Erstreckt sich die Sicherung über mehr als eine Platte oder ein Band, werden die Aufforderungen, das Medium zu wechseln, nach dem Einlegen der Folgeinheit mit ENTER quittiert.

BEZEICHNUNG

`xts` – Ausgabe statistischer Daten von xt-Treibern

ÜBERSICHT

`xts [-f]`

BESCHREIBUNG

Das Kommando `xts` ist ein Werkzeug zur Fehlerüberwachung für den `xt(7)` Treiber. Um die aufgelaufenen statistischen Daten der angeschlossenen Kanalgruppe abzurufen, setzt `xts` auf seine Standardeingabe einen `ioctl(2)` Aufruf mit `XTIOCSSTATS` ab. Falls die Standardeingabe nicht mit einem aktiven `xts`-Kanal verbunden ist, läuft der Aufruf dieser Funktion fehlerhaft ab. Die statistischen Daten werden pro Eintrag zeilenweise auf der Standardausgabe ausgegeben.

`-f` Erzeugt zur Unterstützung von seitenorientierten Ausgabeprogrammen am Ende der Ausgabe ein Zeichen für Formularvorschub.

EXIT-CODES

Bei erfolgreicher Ausführung wird der Exit-Code 0 zurückgegeben, sonst 1.

SIEHE AUCH

`layers(1)` in den *Kommandos*.

`xtt(1M)`, `ioctl(2)`, `xtproto(5)`.

`xt(7)` in *CES, Leitfaden und Werkzeuge für die Programmierung mit C*.

BEZEICHNUNG

`xtt` – Ausgabe von Meldungen der Paketüberwachung für `xt`-Treiber

ÜBERSICHT

`xtt [-f] [-o]`

BESCHREIBUNG

Das Kommando `xtt` ist ein Werkzeug zur Fehlerüberwachung für den `xt(7)` Treiber. Zum Einschalten der Überwachung ruft es zunächst `ioctl(2)` mit `XTIOCSSTATS` für seine Standardeingabe auf und holt die Überwachungsinformation für die angeschlossene Kanalgruppe aus dem ringförmigen Paketpuffer. Falls die Standardeingabe nicht mit einem aktiven `xts`-Kanal verbunden ist, läuft der Aufruf dieser Funktion fehlerhaft ab. Die Ausgabe erfolgt über die Standardausgabe.

Die folgenden Optionen stehen zur Verfügung:

- `-f` Erzeugt zur Unterstützung von seitenorientierten Ausgabeprogrammen am Ende der Ausgabe ein Zeichen für Formularvorschub.
- `-o` Schaltet die Treiberüberwachung ab.

EXIT-CODES

Bei erfolgreicher Ausführung wird der Exit-Code 0 zurückgegeben, sonst 1.

HINWEIS

Falls die Treiberüberwachung nicht durch die Angabe der Option `-t` beim Aufruf von `layers(1)` für die Terminalsitzung eingeschaltet wurde, wird `xtt` beim ersten Aufruf keinerlei Ausgabe erzeugen.

SIEHE AUCH

`layers(1)` in den *Kommandos*.
`xts(1M)`, `ioctl(2)`, `xtproto(5)`.
`xt(7)` in *CES, Leitfaden und Werkzeuge für die Programmierung mit C*.

zdump(1M)

zdump(1M)

BEZEICHNUNG

`zdump` – Ausgabe der Zeitzonen-Zeit

ÜBERSICHT

`zdump [-v] [-c bis_jahr] [zonename ...]`

BESCHREIBUNG

Das Kommando `zdump` gibt die aktuelle Zeit jeder in der Kommandozeile aufgeführten Zeitzone *zonename* aus.

Gültige Optionen sind:

`-v` Für jede Zeitzone *zonename* in der Kommandozeile werden neben der aktuellen Zeit folgende Zeiten ausgegeben. Die Zeit, die dem kleinstmöglichen Wert für die darstellbare Zeit entspricht; die Zeit einen Tag danach; die beiden Zeitpunkte, zu denen sich die Regeln zur Berechnung der örtlichen Zeit ändern, sowie die beiden Zeitpunkte unmittelbar eine Sekunde davor; die Zeit, die dem größtmöglichen Wert für die darstellbare Zeit entspricht, und die Zeit unmittelbar einen Tag davor. Jede Zeile hört mit `isdst=1` auf, falls die angegebene Zeit Sommerzeit ist. Andernfalls endet jede mit `isdst=0`.

`-c bis_jahr`

In der Umgebung des Jahres ab *bis_jahr* soll die ausführliche Ausgabe unterdrückt werden.

DATEIEN

`/usr/lib/locale/TZ` Verzeichnis mit Information für die Standardzeitzone

SIEHE AUCH

`zic(1M)`, `ctime(3C)`.

BEZEICHNUNG

zic – Zeitzonen-Compiler

ÜBERSICHTzic [-v] [-d *dateiverz*] [-l *zeitzone*] [*dateiname 1...*]**BESCHREIBUNG**

Das Kommando **zic** liest den Text der in der Kommandozeile spezifizierten Datei(en) und erzeugt daraus Dateien, die Informationen zur Zeitkonvertierung enthalten. Ist ein *dateiname* als *'-'* spezifiziert, wird hierfür von der Standardeingabe gelesen.

Die Eingabezeilen bestehen aus Feldern, die durch eine beliebige Anzahl von Leerzeichen voneinander getrennt sind. Leerzeichen am Zeilenanfang und Zeilenende werden überlesen. Das Zeichen *#* leitet einen Kommentar ein, der bis zum Ende der aktuellen Zeile reicht. Sollen Leerzeichen oder das Kommentarzeichen inhaltlicher Teil eines Feldes sein, müssen sie in Anführungszeichen *"* gesetzt werden. Zeilen, die nur Kommentar enthalten oder leer sind, werden ignoriert. Für alle nicht leeren Zeilen wird davon ausgegangen, daß sie entweder eine Regel-, eine Zonen- oder eine Verweis-Zeile sind.

Eine Zeile vom Typ "Regel" hat folgende Form:

```
Rule NAME FROM TO TYPE IN      ON      AT SAVE LETTER/S
```

Beispiel:

```
Rule USA 1969 1973 - Apr lastSun 2:00 1:00 D
```

Die einzelnen Felder, aus denen eine Regel-Zeile besteht, sind:

NAME	Ein (beliebig wählbarer) Name der Regelmenge, zu der diese Regel gehört.
FROM	Das erste Jahr, für das die Regel gilt. Der Begriff <i>minimum</i> oder <i>maximum</i> (auch in abgekürzter Form) besagt hier, daß es sich entweder um die kleinste bzw. um die größte darstellbare Jahreszahl handelt.
TO	Das letzte Jahr, für das die Regel gilt. Die Begriffe <i>minimum</i> und <i>maximum</i> haben die gleiche Bedeutung wie oben. Zusätzlich kann hier der (abgekürzte) Begriff <i>only</i> dazu verwendet werden, den Wert des Feldes FROM hier zu wiederholen.
TYPE	Der Typ des Jahres, für den die Regel gilt. Ist hier <i>'-'</i> angegeben, gilt die Regel für alle Jahre von FROM bis einschließlich TO. Ist TYPE als <i>uspres</i> spezifiziert, gilt die Regel für die Jahre, in denen US-Präsidenten gewählt werden. Die Angabe <i>nonpres</i> hingegen bedeutet, daß die Regel für die Jahre gilt, in denen keine Präsidentenwahlen stattfinden. Steht hier irgendetwas anderes, führt zic das folgende Kommando aus:

```
yearistype year type ,
```

um den Typ des Jahres zu prüfen. Dieses Kommando gibt den Exit-Code 0 zurück, wenn das Jahr vom angegebenen Typ ist, andernfalls gibt es den Exit-Code 1 zurück.

zic(1M)**zic(1M)**

IN Gibt den Monatsnamen an, in dem die Regel wirksam wird. Monatsnamen dürfen abgekürzt werden.

ON Gibt den Tag an, an dem die Regel wirksam wird. Die folgenden Beispiele zeigen, welche Formate erkannt werden:

5 der fünfte Tag des Monats
lastSun der letzte Sonntag des Monats
lastMon der letzte Montag des Monats
Sun>=8 der erste Sonntag nach dem siebten Tag des Monats
Sun<=25 der letzte Sonntag vor dem 26. Tag des Monats

Die Namen der Wochentage dürfen abgekürzt oder voll ausgeschrieben werden. Allerdings ist zu beachten, daß in dem Feld ON keine Leerzeichen stehen dürfen.

AT Gibt die Tageszeit an, zu welcher die Regel in Kraft tritt. Die folgenden Beispiele zeigen, welche Formate erkannt werden:

2 Zeitangabe in Stunden
2:00 Zeitangabe in Stunden und Minuten
15:00 Zeitangabe in Stunden (im 24-Stunden-Format)
1:28:14 Zeitangabe in Stunden, Minuten und Sekunden

Jedem dieser Formate kann der Buchstabe *w* folgen, falls die angegebene Zeit die lokale aktuelle Zeit (Ortszeit) darstellt, oder der Buchstabe *s*, falls die lokale Standardzeit gemeint ist. Folgt dem Format keiner der beiden Buchstaben, wird die lokale aktuelle Zeit genommen.

SAVE Der Betrag an Zeit, der zu der örtlichen Standardzeit addiert werden muß, solange die Regel in Kraft ist. Dieses Feld hat dasselbe Format wie das Feld AT, wobei hier natürlich die Suffixe *w* und *s* nicht gelten.

LETTER/S

Gibt den variablen Anteil der während der Gültigkeit der Regel zu verwendenden Zeitzone an (beispielsweise das "S" oder "D" in "EST" oder "EDT"). Ist hier '-' angegeben, ist der variable Anteil leer.

Eine Zeile vom Typ "Zone" hat folgende Form:

Zone **NAME** **GMTOFF** **RULES/SAVE** **FORMAT** **[UNTIL]**

Beispiel:

Zone Australia/South-west GMTOFF RULES/SAVE FORMAT

Die einzelnen Felder, aus denen eine Zonen-Zeile besteht, sind:

- NAME** Name der Zeitzone. Die Datei, die `zic` zur Zeitkonvertierung für diese Zeitzone erzeugt, erhält dann diesen Namen.
- GMTOFF** Der Betrag an Zeit, der zur GMT-Zeit addiert werden muß, um die Standardzeit für diese Zone zu erhalten. Dieses Feld hat dasselbe Format wie die Felder `AT` und `SAVE` in der Zeile vom Typ "Regel". Soll statt der Addition eine Subtraktion stattfinden, hat in diesem Feld als erstes Zeichen ein Minuszeichen zu stehen.
- RULES/SAVE** Der Name der auf diese Zeitzone anzuwendenden Regel(n) oder als Alternative der Betrag an Zeit, der zur örtlichen Standardzeit zu addieren ist. Steht hier '-', wird in dieser Zeitzone stets die Standardzeit genommen.
- FORMAT** Das Format der Abkürzungen für Bezeichnungen von Zeitzonen in dieser Zone. Das Zeichenpaar `%s` zeigt an, an welcher Stelle im Zeitzonennamen der variable Anteil steht.
- UNTIL** Der Zeitpunkt, zu dem sich für die Region die Regel(n) bzw. die Differenz zur GMT -Zeit wieder ändern. Die Angabe erfolgt in Jahr, Monat, Tag und Tageszeit. Ist dieses Feld belegt, wird die Information für die Zeitzone aus diesen Angaben abgeleitet und bleibt für die Dauer bis zum Erreichen des angegebenen Zeitpunktes bestehen.
- Die nächste Zeile muß eine Art Fortsetzungszeile sein; sie hat das gleiche Format wie eine Zonen-Zeile, aber mit dem Unterschied, daß hier die beiden Felder "Zone" und "Name" weggelassen sind. Diese Fortsetzungszeile bezieht sich auf den `UNTIL`-Eintrag der vorhergehenden und besagt, daß die Werte dieser Zeile ab dem unter `UNTIL` genannten Zeitpunkt wirksam werden sollen. Auch Fortsetzungszeilen können ein Feld `UNTIL` enthalten, um eine weitere Fortsetzungszeile anzukündigen.

Eine Zeile vom Typ "Verweis" hat folgende Form:

Link `LINK-FROM` `LINK-TO`

Beispiel:

Link US/Eastern EST5EDT

Das Feld `LINK-FROM` sollte in irgendeiner Zonen-Zeile an der Stelle von `NAME` auftauchen. Das Feld `LINK-TO` dient als alternative Bezeichnung für diese Zone.

Mit der Ausnahme von Fortsetzungszeilen ist die Reihenfolge der Zeilen in der Eingabe völlig beliebig.

OPTIONEN

-v Bewirkt die Ausgabe einer entsprechenden Fehlermeldung, falls in den Eingabedaten eine Jahresangabe vorkommt, die für den Rechner nicht aus dem darstellbaren Bereich für Zeitangaben stammt. Der darstellbare Bereich erstreckt sich von 0:00:00 GMT, 1. Januar 1970 bis 3:14:07 GMT, 19. Januar 2038.

-d *dateiverz*
Die Dateien mit der Information für die Zeitkonvertierung sollen in dem Verzeichnis *dateiverz* angelegt werden, statt im Standardverzeichnis /usr/lib/locale/TZ.

-l *zeitzone*
Es soll die Zeitzone *zeitzone* für die Ortszeit verwendet werden. *zic* verhält sich so, als enthielte die Eingabedatei eine Zeile der Form

Link *zeitzone* localtime

DATEIEN

/usr/lib/locale/TZ Standardverzeichnis für die Aufnahme der erzeugten Dateien

SIEHE AUCH

time(1), ctime(3)

HINWEIS

Bei Gebieten mit mehr als zwei Zeitzonen müssen Sie für die dazugehörenden Regel-Zeilen folgendes beachten: die lokale Standardzeit muß im Feld AT der Regel stehen, die den Zeitpunkt für den zeitlich frühesten Zeitwechsel festlegt. Nur dann ist auch in der von *zic* erzeugten Datei dieser Zeitwechsel korrekt angegeben.

4

Dateiformate

Dateiformate

4-1

BEZEICHNUNG

intro – Einführung zum Kapitel Dateiformate

BESCHREIBUNG

In diesem Kapitel werden die Formate verschiedener Dateien kurz beschrieben. Wo es möglich ist, werden die Deklarationen der C-Strukturen für die jeweiligen Dateiformate angegeben. Die Include-Dateien, die diese Strukturdeklarationen enthalten, sind üblicherweise in den Verzeichnissen `/usr/include` und `/usr/include/sys` abgelegt. Um sie in C-Programme einzuschleusen, wird je nach Herkunft die Schreibweise `#include <filename.h>` oder `#include <sys/filename.h>` verwendet.

Aufgrund der Tatsache, daß nunmehr unter dem Betriebssystem SINIX mehrere Dateisysteme zulässig sind, gibt es im Handbuch zu manchen Begriffen mehrere Einträge gleicher Bezeichnung. Auf diesen Seiten ist stets der Typ des betreffenden Dateisystems genannt, und zwar am Seitenanfang in Klammern eingeschlossen und zentriert.

BEZEICHNUNG

acct – Dateiformat der prozeßspezifischen Abrechnung

ÜBERSICHT

```
#include <sys/types.h>
#include <sys/acct.h>
```

BESCHREIBUNG

Durch den Aufruf von acct erzeugte Dateien haben Einträge der Form, wie sie in <sys/acct.h> definiert sind. Sie haben folgenden Inhalt:

```
typedef ushort comp_t; /* "Gleitkomma" */
                        /* 13-Bit Mantisse, 3-Bit Exponent */

struct      acct {
    char      ac_flag;          /* Abrechnungs-Schalter */
    char      ac_stat;         /* Exit-Status */
    uid_t     ac_uid;          /* Benutzernummer für Abrechnung */
    gid_t     ac_gid;          /* Gruppennummer für Abrechnung */
    dev_t     ac_tty;          /* Protokolldrucker */
    time_t    ac_btime;         /* Startzeit */
    comp_t    ac_utime;         /* berechnet Benutzerzeit (in Zeittakten) */
    comp_t    ac_stime;         /* berechnet Systemzeit (in Zeittakten) */
    comp_t    ac_etime;         /* berechnet verstrichene Zeit (in Zeittakten) */
    comp_t    ac_mem;           /* Speicherbedarf in Zugriffseinheiten */
    comp_t    ac_io;            /* übertragene Zeichen für Lesen/Schreiben */
    comp_t    ac_rw;            /* Zahl der Blöcke für Lesen/Schreiben */
    char      ac_comm[8];       /* angegebenes Kommando */
};

extern      struct      acct acctbuf;
extern      struct      vnode *acctp; /* vnode der Abrechnungsdatei */

#define AFORK 01                /* fork ausgeführt, aber kein exec */
#define ASU 02                  /* benutzte Privilegien */
#define ACCTF 0300              /* Satztyp: 00 = acct */
#define AEXPND 040              /* Erweiterter Satztyp */
```

Im Feld ac_flag wird bei jedem Aufruf von fork die Marke AFORK gesetzt und durch einen Aufruf von exec rückgesetzt. Der Inhalt von ac_comm wird vom Vaterprozeß geerbt und durch exec rückgesetzt. Jedesmal, wenn das System den Prozeß mit einem Zeittakt versorgt, wird auch die aktuelle Größe des Prozesses zu ac_mem addiert. Sie wird folgendermaßen berechnet:

$$\frac{(\text{Datenbereich}) + (\text{Codegröße})}{(\text{Anzahl der Prozesse mit Codesegment im Hauptspeicher})}$$

Der Wert $\text{ac_mem} / (\text{ac_stime} + \text{ac_utime})$ kann als Näherung für die mittlere Prozeßgröße betrachtet werden, da die gemeinsame Nutzung von Code dies beeinflussen kann.

Die Struktur `tacct` gibt das Format der Gesamtabrechnung wieder und ist in den Quelldateien der verschiedenen Kommandos zur Abrechnung definiert.

```
/*
 *          Gesamtabrechnung (für den Abrechnungszeitraum), auch pro Tag
 */
struct tacct {
    uid_t      ta_uid;      /* Benutzernummer */
    char       ta_name[8]; /* Login-Kennung */
    float      ta_cpu[2];  /* kum. Prozessorzeit, p/np (min) */
    float      ta_kcore[2]; /* kum. Zeit im Speicher, p/np */
    float      ta_con[2];  /* kum. Verbindungszeit, p/np, min */
    float      ta_du;      /* kum. Plattenbedarf */
    long       ta_pc;      /* Anzahl der Prozesse */
    unsigned short ta_sc;  /* Zahl der Login-Sitzungen */
    unsigned short ta_dc;  /* Zahl der Plattenzugriffe */
    unsigned short ta_fee; /* Gebühr für Spezialdienste */
};
```

SIEHE AUCH

`acct(2)`, `exec(2)`, `fork(2)`.
`acct(1M)` im *Referenzhandbuch für Systemverwalter*.
`acctcom(1)` in den *Kommandos*.

HINWEIS

Der Wert von `ac_mem` ist für kurzlebige Kommandos wenig aussagekräftig bezüglich der tatsächlichen Größe des Kommandos, da `ac_mem` zu einem Zeitpunkt erhöht werden kann, zu dem der Prozeß gerade ein anderes Programm (z.B. die Shell) ausführt.

BEZEICHNUNG

`admin` – Datei mit den Voreinstellungen für die Installation

BESCHREIBUNG

`admin` ist ein generischer Name für eine ASCII-Datei, in der Installationsparametern Werte zugewiesen werden, die den Installationsvorgang steuern. Beispielsweise können Systemverwalter damit festlegen, wie weiter vorgegangen werden soll, falls sich ein Paket, das gerade installiert wird, bereits auf dem Rechner befindet.

Die Standarddatei für `admin` ist die Datei `/var/sadm/install/admin/default`. Um davon abweichende Werte zu definieren, sollte eine neue `admin`-Datei angelegt werden. Für `admin`-Dateien gibt es keine Namensbeschränkungen. Bei der Installation wird der entsprechende Dateiname dem Kommando `pkgadd` mit der Option `-a` mitgeteilt. Ohne die Option `-a` wird die Datei gemäß Voreinstellung benutzt.

Jeder Eintrag in der Datei `admin` besteht aus einer Zeile der folgenden Form und versorgt einen Parameter mit einem Wert:

param=wert

In dieser Datei können elf Parameter definiert werden. Dabei brauchen nicht alle Parameter mit Werten versorgt zu werden. Ist einem Parameter kein Wert zugeordnet, erfragt `pkgadd` die weitere Vorgehensweise vom Installierenden.

Die elf Parameter und ihre möglichen Werte sind weiter unten aufgeführt, wobei auf Ausnahmen hingewiesen wird. Sie können in beliebiger Reihenfolge angegeben werden. Jedem der Parameter kann der Wert `ask` zugewiesen werden. Wird bei der Installation dieser Parameter benötigt, muß der Benutzer entsprechende Werte eingeben.

<code>basedir</code>	Gibt das Basis-Verzeichnis an, in dem verlagerbare Softwarepakete zu installieren sind. Der zugewiesene Wert kann auch <code>\$PKGINST</code> enthalten. Dies ist ein Hinweis darauf, daß ein Basis-Verzeichnis als Funktion für dieses Paket dienen soll.				
<code>mail</code>	Definiert eine Liste von Benutzern, denen nach der Installation eine Nachricht geschickt werden soll. Bei einer leeren Liste wird keine Post versandt. Ist dieser Parameter in <code>admin</code> nicht aufgeführt, wird die Voreinstellung von <code>root</code> verwendet. Diesem Parameter darf der Wert <code>ask</code> nicht zugewiesen werden.				
<code>runlevel</code>	Bestimmt die Vorgehensweise für den Fall, daß die Betriebsart bei der Installation oder beim Löschen eines Softwarepakets nicht stimmt. Optionen hierfür: <table data-bbox="551 1288 1222 1383"> <tr> <td><code>nocheck</code></td><td>Die Betriebsart wird nicht geprüft.</td></tr> <tr> <td><code>quit</code></td><td>Bei falscher Betriebsart erfolgt Abbruch der Installation.</td></tr> </table>	<code>nocheck</code>	Die Betriebsart wird nicht geprüft.	<code>quit</code>	Bei falscher Betriebsart erfolgt Abbruch der Installation.
<code>nocheck</code>	Die Betriebsart wird nicht geprüft.				
<code>quit</code>	Bei falscher Betriebsart erfolgt Abbruch der Installation.				
<code>conflict</code>	Damit wird festgelegt, was zu tun ist, wenn die Installation das Überschreiben einer bereits installierten Datei erfordert, dies aber zum Konflikt zwischen Softwarepaketen führen würde. Optionen hierfür:				

	nocheck	Es wird nicht auf Konfliktfälle geachtet und Dateien werden gegebenenfalls einfach überschrieben.
	quit	Im Konfliktfall wird die Installation abgebrochen.
	nochange	Die für einen Konfliktfall verantwortlichen Dateien werden nicht installiert.
setuid	Damit wird überprüft, welche ausführbaren Dateien nach der Installation das s-Bit für Benutzer oder Gruppe gesetzt haben. Optionen hierfür:	
	nocheck	Die ausführbaren Dateien werden nicht auf gesetzte s-Bits überprüft.
	quit	Die Installation wird abgebrochen, wenn Programme mit gesetztem s-Bit entdeckt werden.
	nochange	Programme mit gesetztem s-Bit werden zwar installiert, allerdings ohne gesetzte s-Bits.
action	Damit wird festgestellt, ob Installationsprozeduren als Teil der Softwarepakete möglicherweise Sicherheitsaspekte verletzen. Optionen hierfür:	
	nocheck	Keine Prüfung der Prozeduren auf Verletzung von Sicherheitsaspekten.
	quit	Die Installation wird abgebrochen, falls die Prozeduren mögliche Verletzungen der Sicherheit bewirken können.
partial	Prüft, ob eine Version dieser Software bereits teilweise installiert ist. Optionen hierfür:	
	nocheck	Keine Prüfung auf teilweise Installation.
	quit	Abbruch der Installation, falls eine teilweise Installation bereits erfolgt ist.
instance	Bestimmt die Vorgehensweise bei der Installation von Software im Fall der Existenz einer Vorgängerversion (oder einer partiell installierten Version). Optionen hierfür:	
	quit	Sofortiges Beenden der Installation, falls irgendeine Version dieser Software schon installiert ist (sie wird dadurch nicht überschrieben).
	overwrite	Überschreiben von bestehender Software, vorausgesetzt, es gibt genau eine Version davon. Existieren mehrere, aber nur eine mit gleicher Architektur, so wird diese ebenfalls überschrieben. Andernfalls werden die existierenden Versionen ausgegeben und der Benutzer wird gefragt, welche überschrieben werden soll.

	unique	Existierende Software soll unverändert bleiben und stattdessen eine neue Installationsversion erzeugt werden. Zur Kennzeichnung wird der nächste freie Versionsidentifikator verwendet.
idepend	Bestimmt die Vorgehensweise für den Fall, daß andere Softwarepakete von dem gerade zu installierenden abhängen. Optionen hierfür:	
	nocheck	Keine Überprüfung auf Abhängigkeiten.
	quit	Werden keinerlei Abhängigkeiten festgestellt, soll die Installation abgebrochen werden.
rdepend	Bestimmt die Vorgehensweise für den Fall, daß andere Softwarepakete von dem gerade zu löschenden Paket abhängen. Optionen hierfür:	
	nocheck	Keine Überprüfung auf Abhängigkeiten.
	quit	Werden keinerlei Abhängigkeiten festgestellt, soll das Löschen abgebrochen werden.
space	Bestimmt die Vorgehensweise für den Fall, daß der erforderliche Plattenspeicher für dieses Softwarepaket nicht zur Verfügung steht. Optionen hierfür:	
	nocheck	Keine Überprüfung des Plattenbedarfs (die Installation wird nur teilweise durchgeführt, falls der Platz auf der Platte nicht reicht).
	quit	Die Installation wird abgebrochen, falls der Platz nicht reicht.

HINWEIS

In einer admin-Datei, für die kein Dialog mit dem Benutzer vorgesehen ist, sollte der Wert `ask` keinem Parameter zugewiesen werden. Dies würde die ordnungsgemäße Installation dann gefährden, wenn eine Eingabe vom Benutzer erwartet wird.

BEISPIEL

```
basedir=default
runlevel=quit
conflict=quit
setuid=quit
action=quit
partial=quit
instance=unique
idepend=quit
rdepend=quit
space=quit
```

BEZEICHNUNG

ar – Format von Archivdateien

ÜBERSICHT

```
#include <ar.h>
```

BESCHREIBUNG

Das Archivkommando ar bildet aus mehreren Dateien eine Archivdatei. Archive werden vor allem als Bibliotheken verwendet, die der Binder ld durchläuft.

Jedes Archiv beginnt mit der Magic-Zeichenkette (magic string) für das Archiv.

```
#define ARMAG    "<arch>\n"    /* magic-Zeichenkette */
#define SARMAG    8            /* Länge der magic-Zeichenkette */
```

Im Anschluß an die Magic-Zeichenkette folgen die Dateien des Archivs. Jeder Datei ist ein Kopfsatz im folgenden Format vorangestellt:

```
#define ARFMAG    "`\n"    /* Zeichenkette für Dateikopf-Anhang */

struct    ar_hdr    /* Dateikopf */
{
    char    ar_name[16]; /* mit '/' abgeschlossener Dateiname */
    char    ar_date[12]; /* Zeitstempel der Datei */
    char    ar_uid[6];   /* Benutzernummer der Datei */
    char    ar_gid[6];   /* Gruppennummer der Datei */
    char    ar_mode[8];  /* Dateimodus (oktal) */
    char    ar_size[10]; /* Dateigröße */
    char    ar_fmag[2];  /* Dateikopf-Anhang */
};
```

Sämtliche im Kopfsatz enthaltenen Informationen sind in abdruckbarem ASCII-Code dargestellt. Die im Kopfsatz enthaltenen Nummern sind Dezimalzahlen (mit Ausnahme der in *ar_mode* abgelegten Oktalzahlen). Das Archiv kann also ausgedruckt werden, sofern es abdruckbare Dateien enthält.

Paßt der Dateiname in das Feld *ar_name*, wird er dort abgelegt, mit einem Schrägstrich / abgeschlossen und verbleibender Platz mit Leerzeichen aufgefüllt. Paßt er nicht, enthält *ar_name* einen Schrägstrich / gefolgt von einer Dezimalzahl, die die Distanzadresse (offset) des Namens in der weiter unten beschriebenen Zeichenkettentabelle des Archivs darstellt.

Das Feld *ar_date* enthält das Änderungsdatum, das die Datei zum Zeitpunkt des Eintrags in das Archiv trug. Bei Verwendung des portierbaren Archivkommandos ar können Archive, die dem gemeinsamen Format entsprechen, von einem System auf ein anderes übertragen werden.

Jede Datei in einem Archiv beginnt an einer geraden Bytegrenze; falls erforderlich, wird ein Neue-Zeile-Zeichen zwischen Dateien eingefügt. Die angegebene Dateigröße ist jedoch die tatsächliche Länge der Datei ohne eventuelle Füllzeichen.

Man beachte, daß es keine Möglichkeit gibt, leere Bereiche in einem Archiv zu erzeugen.

Zu jedem Archiv, das Objektdateien enthält (siehe `a.out(4)`), gehört auch eine archivspezifische Symboltabelle. Anhand dieser Tabelle stellt der Binder fest, welche Dateien aus dem Archiv während des Binderlaufs geladen werden müssen. Die Symboltabelle ist immer die erste Datei in einem Archiv (sofern es sie gibt), wird aber nie ausgegeben. Sie wird von `ar` automatisch erzeugt bzw. aktualisiert.

Der Name der Symboltabelle in einem Archiv hat die Länge Null (d.h. `ar_name[0]` ist `''`), `ar_name[1]` ist `' '`, usw.). Jeder Eintrag in der Symboltabelle belegt vier Bytes. Dabei wird eine rechnerunabhängige Codierung verwendet, wie sie unten erläutert ist. Die hier beschriebene Codierung für Symboltabellen wird von allen Rechnern verwendet, auch wenn sich deren interne Byte-Reihenfolge davon unterscheidet.

0x01020304

0	1	2	3
01	02	03	04

Diese Datei hat folgenden Inhalt:

1. Anzahl der Symbole. Länge: 4 Byte.
2. Das Feld (array) mit den Distanzadressen (offset) im Archiv. Länge: 4 Byte * "Anzahl der Symbole".
3. Namentabelle. Länge: `ar_size - 4 Byte * ("Anzahl der Symbole" + 1)`.

Zum Beispiel sind in der folgenden Symboltabelle vier Symbole definiert. In der Datei mit der Distanzadresse 114 sind `name` und `object` definiert und in der Datei mit der Distanzadresse 426 ist `function` und ein weiteres Vorkommen von `name` definiert.

Distanz	+0	+1	+2	+3	
0	4				4 Distanzadressen
4	114				name
8	114				object
12	426				function
16	426				name
20	n	a	m	e	
24	\0	o	b	j	
28	e	c	t	\0	
32	f	u	n	c	
36	t	i	o	n	
40	\0	n	a	m	
44	e	\0			

Die Anzahl der Symbole und das Distanzadressen-Feld werden von `sget1` und `sput1` verwaltet. Die Anzahl der in der Namentabelle enthaltenen Zeichenketten, die mit dem Nullzeichen abgeschlossen werden, ist identisch mit der Anzahl der Elemente in dem Distanzadressen-Feld. Jeder Distanzadresse entspricht also genau ein Name in der Namentabelle (in derselben Reihenfolge). Die Namen in der

ar(4)

ar(4)

Namentabelle sind alle globale Symbole, die in den gemeinsamen Objektdateien des Archivs definiert sind. Jede Distanzadresse gibt die Adresse des Dateikopfs der Datei im Archiv an, die das entsprechende Symbol enthält.

Für Bibliotheksdateien, deren Name länger als 15 Zeichen ist, gibt es eine spezielle Archivdatei, die eine Liste von Dateinamen enthält, welche alle mit einem Schrägstrich und einem nachfolgenden Neue-Zeile-Zeichen abgeschlossen sind. Diese spezielle Datei mit der Dateinamenliste gilt nicht als "normale" Datei und ist, falls es sie gibt, stets die erste Datei vor den "normalen" im Archiv. Der Eintrag *ar_name* im Dateikopf dieser speziellen Archivdatei enthält einen Namen der Länge Null. Das erste Zeichen in diesem Eintrag ist der abschließende Schrägstrich, dem ein zweiter folgt, und der Rest ist mit Leerzeichen aufgefüllt (d.h., *ar_name*[0]=='/', *ar_name*[1]=='/', *ar_name*[2]==' ', *ar_name*[3]==' ', usw.). Die Distanzadresse in der Namentabelle beginnt mit Null. Es folgt ein Beispiel für den Inhalt von *ar_name* bei kurzen und langen Dateinamen.

Distanz	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	f	i	l	e	_	n	a	m	e	_
10	s	a	m	p	l	e	/	\n	l	o
20	n	g	e	r	f	i	l	e	n	a
30	m	e	x	a	m	p	l	e	/	\n

Dateiname	<i>ar_name</i>	Bemerkung
short-name	short-name/	Nicht in der Namentabelle
file_name_sample	/0	Distanz 0 in der Namentabelle
longerfilenameexample	/18	Distanz 18 in der Namentabelle

SIEHE AUCH

ar(1), ld(1), strip(1), sputl(3X), a.out(4).

HINWEIS

Mit *strip* werden alle Einträge in der Symboltabelle des Archivs gelöscht. Diese müssen mit Hilfe des Kommandos *ar* unter Angabe der Option *-ts* wiederhergestellt werden, bevor das Archiv vom Binder *ld* verwendet werden kann.

BEZEICHNUNG

archives - Include-Datei für Geräte

ÜBERSICHT

/usr/include/archives.h

BESCHREIBUNG

```

/* Dateikennungsformate */

#define CMN_ASC 0x070701 /* Dateiformatkennung für den -c-Kopf (cpio) */
#define CMN_BIN 070707 /* Dateiformatkennung für den Binär-Kopf (cpio) */
#define CMN_BBS 0143561 /* Dateiformatkennung für den Byte-Swap-Kopf (cpio) */
#define CMN_CRC 0x070702 /* Dateiformatkennung für den CRC-Kopf (cpio) */
#define CMS_ASC "070701" /* Magic-Zeichenkette für den -c-Kopf (cpio) */
#define CMS_CHR "070707" /* Magic-Zeichenkette für den odc-Kopf (cpio) */
#define CMS_CRC "070702" /* Magic-Zeichenkette für den CRC-Kopf (cpio) */
#define CMS_LEN 6 /* Länge der Magic-Zeichenkette (cpio) */

/* Feld- und Kopflängen */

#define CHRSZ 76 /* Größe des -H Feldes ohne Dateinamen-Feld */
#define ASCSZ 110 /* Größe von -c and CRC-Kopf ohne Dateinamen-Feld */
#define TAR SZ 512 /* Größe des tar-Kopfs */

#define HNAMLN 256 /* maximale Länge des Dateinamens in Binär- und odc-Köpfen */
#define EXPNLN 1024 /* maximale Länge des Dateinamens in -c- und CRC-Köpfen */
#define HTIMLN 2 /* Länge des Feldes mit dem Änderungsdatum */
#define HSIZLN 2 /* Länge des Feldes mit der Dateigröße */

/* Definition von Binär-Köpfen bei cpio */

struct hdr_cpio {
    short h_magic, /* Feld mit der Dateiformatkennung */
          h_dev; /* Dateisystem der Datei */
    ushort h_ino, /* I-Node der Datei */
           h_mode, /* Zugriffsrechte der Datei */
           h_uid, /* Benutzernummer der Datei */
           h_gid; /* Gruppennummer der Datei */
    short h_nlink, /* Anzahl der Verweise auf die Datei */
          h_rdev, /* Geräteklasse und -nummer bei Gerätedateien */
          h_mtime[HTIMLN], /* Datum der letzten Änderung der Datei */
          h_namesize, /* Länge des Dateinamens */
          h_filesize[HSIZLN]; /* Dateigröße */
    char h_name[HNAMLN]; /* Dateiname */
};

```

```

/* Format des -H odc-Kopfs bei cpio */
struct c_hdr {
    char    c_magic[CMS_LEN],
           c_dev[6],
           c_ino[6],
           c_mode[6],
           c_uid[6],
           c_gid[6],
           c_nlink[6],
           c_rdev[6],
           c_mtime[11],
           c_namesz[6],
           c_filesz[11],
           c_name[HNAMLEN];
} ;

/* Format des -c- und CRC-Kopfs */

struct Exp_cpio_hdr {
    char    E_magic[CMS_LEN],
           E_ino[8],
           E_mode[8],
           E_uid[8],
           E_gid[8],
           E_nlink[8],
           E_mtime[8],
           E_filesize[8],
           E_maj[8],
           E_min[8],
           E_rmaj[8],
           E_rmin[8],
           E_namesize[8],
           E_chksum[8],
           E_name[EXP_NLEN];
} ;

/* Struktur und Format des tar-Kopfs */

#define TBLOCK 512 /* Länge des tar-Kopfs und der Datenblöcke */
#define TNAMLEN 100 /* maximale Länge der Dateinamen bei tar */
#define TMODLEN 8 /* Länge des Feldes mit den Zugriffsrechten */
#define TUIDLEN 8 /* Länge des Feldes mit der Benutzernummer */
#define TGIDLEN 8 /* Länge des Feldes mit der Gruppennummer */
#define TSIZLEN 12 /* Länge des Feldes mit der Dateigröße */
#define TTIMLEN 12 /* Länge des Feldes mit dem Datum der letzten Änderung */
#define TCRLEN 8 /* Länge des Feldes mit der Prüfsumme des Kopfes */

```

```

/* Definition des tar-Kopfs */

union tblock {
    char dummy[TBLOCK];
    struct header {
        char t_name[TNAMLEN];          /* Name der Datei */
        char t_mode[TMODLEN];          /* Zugriffsrechte der Datei */
        char t_uid[TUIDLEN];           /* Benutzernummer der Datei */
        char t_gid[TGIDLEN];           /* Gruppennummer der Datei */
        char t_size[TSIZLEN];           /* Größe der Datei in Bytes */
        char t_mtime[TTIMLEN];          /* Datum der letzten Änderung der Datei */
        char t_chksum[TCRCLEN];         /* Prüfsumme des Kopfs */
        char t_typeflag;                /* Anzeige für den Typ der Datei */
        char t_linkname[TNAMLEN];       /* Datei, auf die ein Verweis existiert */
        char t_magic[6];                /* Magic-Zeichenkette (immer "ustar") */
        char t_version[2];              /* Versionszeichenkette (immer "00") */
        char t_uname[32];               /* Eigentümer der Datei (in ASCII) */
        char t_gname[32];               /* Gruppe der Datei (in ASCII) */
        char t_devmajor[8];             /* Geräteklasse bei Gerätedateien */
        char t_devminor[8];             /* Gerätenummer bei Gerätedateien */
        char t_prefix[155];             /* Präfix für den Pfadnamen */
    } tbuf;
};

/* Format und Struktur der Bandetiketten bei volcopy */

#define VMAGLEN 8
#define VVOLLEN 6
#define VFILLEN 464

struct volcopy_label {
    char    v_magic[VMAGLEN],
           v_volume[VVOLLEN],
           v_reels,
           v_reel;
    long    v_time,
           v_length,
           v_dens,
           v_reelblks,
           v_blksize,
           v_nblocks;
    char    v_fill[VFILLEN];
    long    v_offset;
    int     v_type;
};
/* zusätzliches Feld bei u370 */
/* zusätzliches Feld bei u370 */
/* zusätzliches Feld bei u370 */
/* zusammen mit den -e und -reel Optionen */
/* Band mit nblocks-Feld ? */

```

BEZEICHNUNG

binarsys – Information über ferne Systeme für das Kommando ckbinarsys

ÜBERSICHT

/etc/mail/binarsys

BESCHREIBUNG

binarsys enthält Zeilen der folgenden Form:

remote_system_name:wert

Dabei ist *wert* entweder Y oder N. Diese Zeile gibt an, ob das jeweilige ferne System mit Meldungen, die auch Binär-Information enthalten, vernünftig umgehen kann. Ist für ein bestimmtes System kein Eintrag vorhanden oder fehlt überhaupt die Datei binarsys, wird stets No angenommen.

Leere Zeilen oder solche, die mit # beginnen, werden als Kommentar betrachtet und nicht interpretiert. Für den Fall, daß eine Zeile mit dem Inhalt `Default=y` auftaucht, wird die im vorigen Absatz beschriebene Voreinstellung für fehlende Einträge auf Yes gesetzt. Eine Zeile mit dem Inhalt `Default=n` stellt die ursprüngliche Voreinstellung wieder her.

mail wird mit der Datei binarsys ausgeliefert, die in diesem Fall als einzige Zeile `Default=y` enthält.

DATEIEN

/etc/mail/binarsys

SIEHE AUCH

ckbinarsys(1M), mailsurr(4)

mail(1) in den *Kommandos*.

BEZEICHNUNG

`cron` – Optionen für `cron`

ÜBERSICHT

`/etc/default/cron`

BESCHREIBUNG

Die Schlüsselwörter in `/etc/default/cron` dienen dazu, Optionen für `cron` zu ändern oder zu setzen. `cron` kennt die folgenden Schlüsselwörter:

`CRONLOG=YES` oder `NO`

Wenn `CRONLOG` auf `YES` gesetzt ist, so werden alle `CRON`-Aufträge in der Datei `/usr/lib/cron/log` protokolliert. Die Voreinstellung ist `NO`.

DATEIEN

`/etc/default/cron`

SIEHE AUCH

`cron`

BEZEICHNUNG

`dfstab` – Datei mit Kommandos für die Nutzung gemeinsamer Ressourcen

ÜBERSICHT

`/etc/dfs/dfstab`

BESCHREIBUNG

Die Datei `dfstab` ist im Verzeichnis `/etc/dfs` abgelegt und enthält Kommandos zur netzwerkweiten gemeinsamen Nutzung von Ressourcen. Mittels `dfstab` kann der Systemverwalter in einheitlicher Weise die automatische gemeinsame Nutzung lokaler Ressourcen steuern.

Die Datei `dfstab` enthält in jeder Zeile einen Aufruf des Kommandos `share(1M)`. Die Datei kann direkt von der Shell eingelesen werden, um alle Ressourcen gemeinsam zu nutzen. Oder der Systemverwalter erstellt eigene Shell-Prozeduren, um gezielt bestimmte Zeilen der Datei ausführen zu lassen.

Die Kommandos in `dfstab` werden automatisch ausgeführt, sobald das System in die Betriebsstufe 3 wechselt.

SIEHE AUCH

`share(1M)`, `shareall(1M)`.

dir (4)

(s5)

dir (4)

BEZEICHNUNG

dir(s5) – Format von s5-Verzeichnissen

ÜBERSICHT

```
#include <sys/types.h>
#include <sys/fs/s5dir.h>
```

BESCHREIBUNG

Verzeichnisse erscheinen nach außen wie gewöhnliche Dateien, außer daß Benutzer in Verzeichnisse nicht schreiben können. Daß es sich bei einer Datei um ein Verzeichnis handelt, ist im zugehörigen Indexeintrag (i-node-Eintrag) in einem Bit der Zugriffsrechte verschlüsselt (vgl. s5-spezifischer inode(4)). Ein Verzeichniseintrag hat die folgende Struktur:

```
#ifndef DIRSIZ
#define DIRSIZ 14
#endif
struct direct
{
    o_ino_t    d_ino;    /* Indexeintrag vom Typ s5 */
    char  d_name[DIRSIZ];
};
```

Gemäß Konvention sind die ersten beiden Einträge in einem Verzeichnis immer . als Verweis auf sich selbst und .. als Verweis auf das Vater-Verzeichnis. Im root-Verzeichnis des Master-Dateisystems hat .. eine andere Bedeutung; da es hier kein Vater-Verzeichnis gibt, sind . . und . gleichbedeutend.

SIEHE AUCH

s5-spezifischer inode(4)

BEZEICHNUNG

dir (ufs) – Format von ufs-Verzeichnissen

ÜBERSICHT

```
#include <sys/param.h>
#include <sys/types.h>
#include <sys/fs/ufs_fsdir.h>
```

BESCHREIBUNG

Ein Verzeichnis besteht aus einer Anzahl von Blöcken zu je DIRBLKSIZ Bytes. Dabei ist DIRBLKSIZ so gewählt, daß ein Verzeichnis mit einer einzigen Operation auf Platte verlagert werden kann (das sind bei den meisten Rechnern 512 Byte).

Jeder Block der Größe DIRBLKSIZ enthält eine gewisse Anzahl von Verzeichniseinträgen unterschiedlicher Länge. Die Einträge sind Strukturen, wobei in jeder als erstes eine weitere Struktur struct direct eingebettet ist. Diese enthält die Indexnummer und die Länge des Eintrags sowie die Länge des darin enthaltenen Namens. Danach folgt der Name selbst, der am Ende so weit mit Nullzeichen aufgefüllt wird, daß er auf einer 4 Byte Grenze zu liegen kommt. Jeder Name hört garantiert mit (mindestens) einem Nullzeichen auf. Die maximale Länge eines Namens beträgt MAXNAMLEN Bytes.

```
#define DIRBLKSIZ DEV_BSIZE
#define MAXNAMLEN 256
struct direct {
    u_long    d_ino;           /* Indexnummer des Eintrags */
    u_short   d_reclen;       /* Gesamtlänge des Eintrags */
    u_short   d_namlen;       /* Länge des Namens im Eintrag */
    char      d_name[MAXNAMLEN + 1]; /* Name darf nicht länger sein */
};
```

SIEHE AUCH

ufs-spezifisches fs(4)

BEZEICHNUNG

`dirent` – dateisystemunabhängiger Verzeichniseintrag

ÜBERSICHT

```
#include <dirent.h>
```

BESCHREIBUNG

Bei verschiedenen Dateisystemen können die Einträge im Verzeichnis unterschiedlich sein. Mit Hilfe der Struktur `dirent` wird ein dateisystemunabhängiger Verzeichniseintrag definiert, der alle Informationen enthält, die den Verzeichniseinträgen in den verschiedenen Dateisystemen gemeinsam ist. Der Systemaufruf `getdents` liefert solche Strukturen zurück.

Es folgt die Definition der Struktur `dirent`.

```
struct    dirent {
    ino_t      d_ino;
    off_t      d_off;
    unsigned short d_reclen;
    char       d_name[1];
};
```

Die Indexnummer `d_ino` ist in jedem Dateisystem eindeutig. Das Feld `d_off` enthält die Distanzadresse (offset) des Verzeichniseintrags im eigentlichen Verzeichnis des betreffenden Dateisystems. Das Feld (array) `d_name` enthält den Namen dieses Eintrags als Zeichenkette. Der Name ist mit einem Nullzeichen abgeschlossen und hat nicht mehr als `MAXNAMLEN` Zeichen. Damit hat man eine dateisystemunabhängige Darstellung für Einträge unterschiedlicher Länge. Diese Länge ist in dem Feld `d_reclen` gespeichert und stellt die Anzahl der Bytes des aktuellen Eintrags bis zum nächsten dar; damit kann die nachfolgende Struktur passend ausgerichtet werden.

SIEHE AUCH

`getdents(2)`

BEZEICHNUNG

.environ, .pref, .variables – Benutzerpräferenzen für FACE

ÜBERSICHT

\$HOME/pref/.environ
\$HOME/pref/.variables
\$HOME/FILECABINET/.pref
\$HOME/WASTEBASKET/.pref

BESCHREIBUNG

Die Dateien .environ, .pref und .variables enthalten Variable, die die Präferenzen eines Benutzers bei einer Vielzahl von Operationen angeben. Die beiden Dateien .environ und .variables sind im Verzeichnis \$HOME/pref des jeweiligen Benutzers abgelegt. Die Dateien .pref befinden sich in den Verzeichnissen \$HOME/FILECABINET und \$HOME/WASTEBASKET oder beliebigen anderen, in denen mit Hilfe des Kommandos organize Präferenzen gesetzt worden sind. Die Namen und Beschreibungen zu den einzelnen Variablen folgen weiter unten. Die Variablen werden in der Form *variable=wert* definiert und stehen jeweils in einer eigenen Zeile.

In der Datei .environ stehen u.a. die folgenden Variablen :

LOGINWIN[1-4]	Bezeichnet die Fenster, die bei der Initialisierung von FACE geöffnet werden.
SORTMODE	Sortiermodus für Ausgabe von Dateien in einem Ordner. Die Wertangaben erfolgen hexadezimal und haben folgende Bedeutung: 1 Alphabetisch nach dem Dateinamen sortiert. 2 Dateien mit dem jüngsten Änderungsdatum zuerst. 800 Alphabetisch nach dem Objekttyp sortiert. Die obigen Werte, durch logisches ODER mit 1000 verknüpft, haben umgekehrte Funktion: 1000 Ausgabe in umgekehrter Reihenfolge. Der Wert von 1002 bewirkt beispielsweise, daß die Dateien eines Ordners in der Reihenfolge ausgegeben werden, daß die mit dem jüngsten Änderungsdatum zuerst kommen. Der Wert 1001 bewirkt die Ausgabe in absteigend alphabetisch, nach den Namen geordneter Form.
DISPLAYMODE	Ausgabemodus der Ordnerdateien. Die Wertangaben erfolgen hexadezimal und haben folgende Bedeutung: 0 Nur Dateinamen ausgeben. 4 Dateinamen samt Kurzbeschreibung ausgeben. 8 Dateinamen, Beschreibung und Zusatzinformation ausgeben.

environ (4)

environ (4)

WASTEPROMPT	Rückfrage vor dem Leeren des Papierkorbs (yes/no)?
WASTEDAYS	Anzahl der Tage bis zur Leerung des Papierkorbs.
PRINCMD[1-3]	Vordefiniertes Kommando zum Ausdruck von Dateien.
UMASK	Enthält die Voreinstellung für die Zugriffsrechte beim Neuanlegen von Dateien.

In der Datei `.pref` stehen u.a. die folgenden Variablen :

`SORTMODE`
Mit der gleichen Bedeutung wie die bei `.environ` (s. oben) beschriebene Variable `SORTMODE`.

`DISPMODE`
Mit der gleichen Bedeutung wie die bei `.environ` (s. oben) beschriebene Variable `DISPLAYMODE`.

In der Datei `.variables` stehen u.a. die folgenden Variablen :

`EDITOR`
Definiert den Standardeditor.

`PS1` Definiert das Prompt der Shell.

DATEIEN

`$HOME/pref/.environ`
`$HOME/pref/.variables`
`$HOME/FILECABINET/.pref`
`$HOME/WASTERASKET/.pref`

BEZEICHNUNG

ethers – Ethernet-Adreßdatenbank für Hostnamen

ÜBERSICHT

/etc/ethers

BESCHREIBUNG

Die Datei `ethers` enthält Information bezüglich der im Internet bekannten (48 Bit langen) Ethernet-Adressen von Hosts. Für jeden Host im Ethernet sollte eine eigene Zeile mit dem folgenden Inhalt vorhanden sein:

ethernet-adresse offizieller-host-name

Die Begriffe sind durch eine beliebige Anzahl von Leerzeichen und/oder Tabulatoren voneinander getrennt. Das Zeichen '#' leitet einen Kommentar ein, der bis zum Zeilenende reicht.

Das Standardformat für Ethernet-Adressen ist `x:x:x:x:x:x`. Dabei steht `x` für eine Hexadezimalzahl zwischen 0 und FF, die ein Byte belegt. Die Reihenfolge der Bytes in den Adressen folgt den Konventionen für Netzwerke. Hostnamen können beliebige abdruckbare Zeichen mit Ausnahme von Leerzeichen, Tabulatoren, Neue-Zeile-Zeichen oder dem Kommentarzeichen enthalten. Die Hostnamen in der Datei `ethers` entsprechen absichtlich denen in der Datei `hosts`.

Um Zeilen der Datei `ethers` abzuarbeiten, kann die Routine `ether_line` aus der Bibliothek für Adreßmanipulation `ethers(3N)` verwendet werden.

DATEIEN

/etc/ethers

SIEHE AUCH

`ethers(3N)`, `hosts(4)`.

BEZEICHNUNG

/dev/fd – Dateideskriptor-Dateien

BESCHREIBUNG

Dateideskriptor-Dateien werden /dev/fd/0, /dev/fd/1, /dev/fd/2 usw. genannt. Sie beziehen sich auf bereits geöffnete Dateien, die über Dateideskriptoren ansprechbar sind. Unter der Voraussetzung, daß der Dateideskriptor *n* auf eine offene Datei verweist, haben die beiden folgenden Systemaufrufe dieselbe Wirkung:

```
fd = open("/dev/fd/n", mode);  
fd = dup(n);
```

Auf diese Dateien ist die Wirkung von creat(2) äquivalent zu open, wobei mode unbeachtet bleibt. Genau wie bei dup laufen nachfolgende Lese- oder Schreibaufrufe nur dann korrekt ab, wenn auch der ursprüngliche Dateideskriptor diese Operationen zugelassen hat.

Aus praktischen Gründen sind zusätzlich noch drei weitere Namen eingeführt, die auf die Standardeingabe, -ausgabe und -fehlerausgabe verweisen: /dev/fd/0 ist ein Synonym für die Standardeingabe, /dev/fd/1 für die Standardausgabe und /dev/fd/2 für die Standardfehlerausgabe.

SIEHE AUCH

open(2), dup(2)

EXIT-CODES

Wenn der entsprechende Dateideskriptor keine offene Datei bezeichnet, gibt open(2) -1 und EBADF zurück.

BEZEICHNUNG

filehdr – Dateikopf für COFF Dateien

ÜBERSICHT

```
#include <filehdr.h>
```

BESCHREIBUNG

Jede COFF Datei beginnt mit einem 20 Byte langen Dateikopf der folgenden C-Struktur struct filehdr:

```
struct    filehdr
{
    unsigned short  f_magic ;    /* Dateiformatkennung */
    unsigned short  f_nscns ;    /* Anzahl der Objektteile */
    long            f_timdat ;    /* Zeitstempel */
    long            f_symptr ;    /* Zeiger auf die Symboltabelle */
    long            f_nsyms ;    /* Anzahl der Einträge in der
                                   Symboltabelle */
    unsigned short  f_opthdr ;    /* Größe von opt und header */
    unsigned short  f_flags ;    /* Marken */
};
```

Der Zeiger `f_symptr` enthält die Distanzadresse (offset), ab der in der Datei die Symboltabelle beginnt. Dieser Wert kann von `fseek(3S)` für die Positionierung des Dateizeigers auf die Symboltabelle verwendet werden. Der optionale Dateikopf des SINIX-Systems hat 28 Byte. Es gelten die folgenden Dateiformatkennungen (magic numbers):

```
#define I386MAGIC 0514 /* i386 Rechner */
#define WE32MAGIC 0560 /* 3B2, 3B5 und 3B15 Rechner */
#define N3BMAGIC 0550 /* 3B20 Rechner */
#define NITVMAGIC 0551 /* 3B20 Rechner */

#define VAXWRMAGIC 0570 /* VAX schreibbare Textsegmente */
#define VAXROMAGIC 0575 /* VAX schreibgeschützte Textsegmente */
```

Den in `f_timdat` abgelegte Wert liefert der Systemaufruf `time(2)`. Die zur Zeit definierten Werte der Marken (flags) sind:

```
#define F_RELFLG 0000001 /* Relokationseinträge entfernt */
#define F_EXEC 0000002 /* ausführbare Datei */
#define F_LNNO 0000004 /* Zeilennummern entfernt */
#define F_LSYMS 0000010 /* lokale Symbole entfernt */
#define F_AR16WR 0000200 /* 16-bit DEC Host */
#define F_AR32WR 0000400 /* 32-bit DEC Host */
#define F_AR32W 0001000 /* kein DEC Host */
#define F_BM32ID 0160000 /* ID-Feld der WE32000 Familie */
#define F_BM32B 0020000 /* Datei enthält WE 32100 Code */
#define F_BM32MAU 0040000 /* Datei braucht MAU für Ausführung */
#define F_BM32RST 0010000 /* diese Datei führt Restaurations-
                               arbeiten durch [3B5/3B2 only] */
```

filehdr (4)

filehdr (4)

SIEHE AUCH

`time(2)`, `fseek(3S)`, `pa.out(4)`

Kapitel "Common Object File Format (COFF)" in *CES, Leitfaden und Werkzeuge für die Programmierung mit C*.

BEZEICHNUNG

fs (bfs) – Format des bfs-Superblocks

ÜBERSICHT

```
#include <sys/types.h>
#include <sys/fs/bfs.h>
```

BESCHREIBUNG

Der bfs-Superblock steht in Sektor 0 und hat das folgende Format:

```
struct bdsuper
{
    long bdsup_bfsmagic; /* "magic number" */
    off_t bdsup_start; /* Distanzadresse zum Anfang der Daten
                       im Dateisystem */
    off_t bdsup_end; /* Distanzadresse zum Ende der Daten im
                     Dateisystem */

    /*
     * Sicherheitsfelder
     */
    daddr_t bdcop_fromblock; /* Herkunftblock des aktuellen
                             Datentransfers */
    daddr_t bdcop_toblock; /* Zielblock des aktuellen
                           Datentransfers */
    daddr_t bdcopb_fromblock; /* Sicherungskopie des Herkunftblocks */
    daddr_t bdcopb_toblock; /* Sicherungskopie des Zielblocks */
};

#define BFS_MAGIC 0xBADFACE /* bfs "magic number" */
```

Der bfs-Superblock wird ausschließlich vom System verwendet. Der Benutzer kann nicht damit arbeiten.

SIEHE AUCH

bfs-spezifischer inode(4)

BEZEICHNUNG

fs (s5) – Format des s5-Superblocks

ÜBERSICHT

```
#include <sys/types.h>
#include <sys/param.h>
#include <sys/fs/s5filsys.h>
```

BESCHREIBUNG

Der erste Sektor (Sektor 0) eines s5-Dateisystems bleibt ungenutzt. Der zweite Sektor (Sektor 1) enthält den Superblock mit folgendem Format:

```
struct    filsys
{
    ushort  s_isize;           /* Größe der I-Node Liste in Blöcken */
    daddr_t s_fsize;          /* Größe des Datenträgers in Blöcken */
    short   s_nfree;          /* Belegtheitsgrad von s_free */
    daddr_t s_free[NICFREE];   /* Liste freier Blöcke */
    short   s_ninode;         /* Belegtheitsgrad von s_inode */
    o_ino_t s_inode[NICINOD];  /* Liste freier I-Nodes */
    char    s_flock;          /* Sperre während der Änderung der */
                                /* Liste freier Blöcke */
    char    s_ilock;          /* Sperre während der Änderung der */
                                /* Liste freier I-Nodes */
    char    s_fmod;           /* Anzeige für Superblock-Änderung */
    char    s_ronly;          /* Anzeige für Schreibschutz */
    time_t  s_time;           /* letzte Superblock-Aktualisierung */
    short   s_dinfo[4];       /* Geräteinformation */
    daddr_t s_tfree;          /* Gesamtzahl freier Blöcke */
    o_ino_t s_tinode;         /* Gesamtzahl freier I-Nodes */
    char    s_fname[6];       /* Dateisystem */
    char    s_fpack[6];       /* Bezeichnung der Packungsart */
    long    s_fill[12];       /* Füllfelder, damit die Größe */
                                /* des Superblocks 512 Bytes beträgt */
    long    s_state;          /* Status des Dateisystems */
    long    s_magic;          /* bei neuem Dateisystem: */
                                /* Dateisystemkennung */
    long    s_type;           /* Typ des neuen Dateisystems */
};

#define FsMAGIC    0xfd187e20 /* s5 magic number */
#define Fs1b      1          /* 512 Byte Block */
#define Fs2b      2          /* 1024 Byte Block */
#define Fs4b      3          /* 2048 Byte Block */
#define FsOKAY    0x7c269d38 /* s_state: in Ordnung */
#define FsACTIVE  0x5e72d81a /* s_state: in Betrieb */
#define FsBAD     0xcb096f43 /* s_state: root fehlerhaft */
#define FsBADBLK  0xbadbc14b /* s_state: gestört */
                                /* durch fehlerhaften Block */
```

Das Feld `s_type` enthält den Typ des Dateisystems. Zur Zeit werden drei Typen unterstützt, die sich in der Größe ihrer logischen Blöcke unterscheiden (512 Byte, 1024 Byte und 2048 Byte). Das Feld `s_magic` dient der Unterscheidung eines s5-Dateisystems von dem eines anderen Typs. Das Feld `s_type` bestimmt die Größe der logischen Blöcke (512 Byte, 1K oder 2K). Die Umsetzung von logischen Blocknummern in physikalische Sektornummern wird durch das Betriebssystem erledigt.

Das Feld `s_state` zeigt den Zustand des Dateisystems an. Bei einem regulär ausgehängten Dateisystem ist der Wert von `FSOKAY` gleich `s_state + s_time`. Sowie ein Dateisystem zur Bearbeitung eingehängt ist, ändert sich der Status zu `FSACTIVE`. Das root-Dateisystem stellt einen Spezialfall dar. Wird es zur Ladezeit als fehlerhaft erkannt, wird es dennoch eingehängt, aber als `FSBAD` markiert. Beim Aushängen eines Dateisystems wird der Status wieder auf `FSOKAY` zurückgesetzt.

Das Feld `s_isize` zeigt auf den ersten Datenblock, der direkt hinter der Liste der I-Nodes steht. Diese Liste fängt hinter dem Superblock an, also in Sektor 2. Demzufolge belegt die Liste der I-Nodes genau `s_isize-2` Blöcke. In `s_fsize` steht die Nummer des ersten nicht mehr zum Dateisystem gehörenden Blocks. Sie wird vom System zur Überprüfung auf fehlerhafte Blocknummern verwendet. Wird ein solcher "unmöglicher" Block aus der Liste freier Blöcke reserviert oder dahin zurückgegeben, erfolgt eine entsprechende Fehlermeldung auf der Konsole. Außerdem wird die anscheinend zerstörte Liste freier Blöcke bereinigt, um weiteren Blockzuteilungen daraus vorzubeugen.

Die Liste freier Blöcke eines Datenträgers wird wie folgt verwaltet. Das Feld (array) `s_free` enthält in den Feldelementen `s_free[1], ..., s_free[s_nfree-1]` maximal 49 Blocknummern freier Blöcke. Das erste Feldelement `s_free[0]` zeigt auf den Anfang einer verketteten Liste von Blöcken, die die Liste freier Blöcke darstellt. In jedem dieser Blöcke enthält der erste Eintrag vom Datentyp `long`, wieviele Nummern freier Blöcke in den (bis zu 50) nachfolgenden Einträgen vom Datentyp `long` abgelegt sind. Der jeweils erste dieser Blöcke verweist auf das nächste Element der verketteten Liste. Um einen Block zu reservieren, geht man folgendermaßen vor: zuerst wird `s_nfree` dekrementiert und als neue Blocknummer erhält man `s_free[s_nfree]`. Ist sie Null, sind keine freien Blöcke mehr da und man sollte eine entsprechende Fehlermeldung zurückgeben. Falls durch das Dekrementieren `s_nfree` Null geworden ist, wird der Block mit der Blocknummer `s_free[0]` eingelesen, sein erster Eintrag wird der neue Wert von `s_nfree` und die folgenden 50 Einträge vom Datentyp `long` werden in das Feld (array) `s_free` übertragen. Um umgekehrt einen Block freizugeben, wird zuerst geprüft, ob `s_nfree` seinen Maximalwert 50 hat. In diesem Fall wird die Länge des Feldes `s_free` und dahinter das Feld `s_free` selbst in den Block übertragen und dieser geschrieben. Danach wird `s_nfree` auf 0 gesetzt. In jedem Fall wird die Blocknummer des freizugebenden Blocks in `s_free[s_nfree]` eingetragen und die aktuelle Feldlänge `s_nfree` inkrementiert.

Die Gesamtzahl der freien Blöcke des Dateisystems steht in `s_tfree`.

Mit `s_ninode` ist die Anzahl der freien I-Nodes in dem Feld (array) `s_inode` gegeben. Um einen I-Node zu reservieren, geht man so vor: ist `s_ninode` noch größer 0, wird dekrementiert und die Indexnummer `s_inode[s_ninode]` zurückgegeben. Andernfalls werden zuerst aus der Liste der freien I-Nodes maximal 100 Indexnummern besorgt und das Feld `s_inode` damit aufgefüllt und dann daraus ein I-Node reserviert. Die Freigabe eines I-Node erfolgt dadurch, daß dessen Nummer in das Feldelement `s_inode[s_ninode]` eingetragen und `s_ninode` inkrementiert wird, vorausgesetzt `s_ninode` war kleiner als 100. Ist die Obergrenze (`s_ninode` gleich 100) bereits erreicht, kann der freie I-Node ohne Besorgnis in eine beliebige Liste eingetragen werden. Diese Teilliste freier I-Nodes dient lediglich der Beschleunigung bei der Reservierung. Die Information darüber, ob ein I-Node tatsächlich freigegeben ist oder nicht, wird im I-Node selbst verwaltet.

Die Gesamtzahl der freien I-Nodes des Dateisystems steht in `s_tinode`.

Die beiden Schalter `s_flock` und `s_ilock` dienen in der Speicherkopie des Dateisystems der Verwaltung, solange dieses eingehängt ist. Ihr Wert auf der Platte ist bedeutungslos, ähnlich wie der von `s_fmod`, welcher anzeigt, ob der Superblock geändert wurde und bei der nächsten Aktualisierungsphase des Dateisystems auf Platte gesichert werden muß.

Der Schalter `s_ronly` ist bei einem schreibgeschützten Dateisystem gesetzt.

Mit `s_time` wird der Zeitpunkt der letzten Änderung des Superblocks angegeben, und zwar in Sekunden, die seit 1. Januar 1970, 00:00 Uhr, (UTC) verstrichen sind. Während des Neuladens entnimmt das System den Wert für die Systemzeit dem Eintrag `s_time` im Superblock.

Der Eintrag `s_fname` enthält den Namen des Dateisystems und `s_fpack` die Bezeichnung der Packungsart.

I-Nodes sind 64 Bytes groß und ihre Numerierung beginnt bei 1. Sie sind ab Block 2 auf der Platte gespeichert. Die Indexnummer 1 ist für zukünftige Verwendung reserviert und Indexnummer 2 ist dem Root-Verzeichnis des Dateisystems vorbehalten. Ansonsten gelten keine impliziten Bedeutungen für I-Nodes. Jeder I-Node steht für eine Datei. Zum Format eines I-Node und der Bedeutung seiner Schalter siehe `inode`.

SIEHE AUCH

`mount(2)`.

`fsck(1M)`, `fsdb(1M)`, `mkfs(1M)`, s5-spezifischer `inode(4)`

BEZEICHNUNG

fs (ufs) – Format des UFS-Superblocks und der Zylindergruppen-Blöcke

ÜBERSICHT

```
#include <sys/param.h>
#include <sys/types.h>
#include <sys/fs/ufs_fs.h>
```

BESCHREIBUNG

Auf jeder Platte befindet sich eine bestimmte Anzahl von Dateisystemen. Jedes davon besteht aus einer Anzahl von Zylindergruppen. Jede Zylindergruppe enthält I-Nodes und Daten.

Ein Dateisystem wird durch seinen Superblock und die Information in den Blöcken mit den Zylindergruppen beschrieben. Der Superblock enthält äußerst wichtige Daten und wird daher vor jede Zylindergruppe kopiert, um vor katastrophalem Datenverlust zu schützen. Dies wird beim Ablauf von `mkfs` erledigt. Da die kritischen Daten im Superblock nicht geändert werden, braucht man im Normalfall nicht mehr auf die Kopien zuzugreifen.

```
/*
 * Superblock eines Dateisystems.
 */
#define FS_MAGIC    0x011954
#define FSACTIVE    0x5e72d81a    /* fs_state: eingehängt */
#define FSOKAY      0x7c269d38    /* fs_state: in Ordnung */
#define FSBAD       0xcb096f43    /* fs_state: root fehlerhaft */

struct    fs {
    struct  fs *fs_link;           /* verkettete liste von Dateisystemen (DS) */
    struct  fs *fs_rlink;          /* für Superblöcke im Speicher */
    daddr_t fs_sbknk;              /* Adr. des Superblocks im DS */
    daddr_t fs_cblkno;             /* Distanzadr. des Zyl.blocks im DS */
    daddr_t fs_iblkno;            /* Distanzadr. der I-Node Blöcke im DS */
    daddr_t fs_dblkno;            /* Distanzadr. zum ersten Datenblock */
    long    fs_cgoffset;           /* Distanzadr. der Zyl.grp. im Zylinder */
    long    fs_cgmask;             /* für Berechnung modulo fs_ntrak */
    time_t  fs_time;              /* letzte Sicherung */
    long    fs_size;              /* Gesamtzahl der Blöcke im DS */
    long    fs_dsize;             /* Anzahl der Datenblöcke im DS */
    long    fs_ncg;               /* Anzahl der Zyl.grp. */
    long    fs_bsize;             /* Größe der normalen Blöcke im DS */
    long    fs_fsize;             /* Größe der Fragmentblöcke im DS */
    long    fs_frag;              /* Anzahl der Fragmente pro Block */
    /* es folgen Konfigurationsparameter */
    long    fs_minfree;           /* Minimum der freien Blöcke in Prozent */
    long    fs_rotdelay;          /* Anzahl in ms für optimalen Blockzugriff */
    long    fs_rps;               /* Plattenumdrehungen pro Sek. */
    /* die folgenden Felder sind aus den anderen berechenbar */
    long    fs_bmask;             /* Blockdistanzen */
    long    fs_fmask;             /* Fragmentdistanzen */
    long    fs_bshift;           /* logische Blocknummern */
    long    fs_fshift;           /* Anzahl der Fragmente */
};
```



```

/* weitere Konfigurationsparameter */
    long    fs_maxcontig;        /* max. Anzahl sequentieller Blöcke */
    long    fs_maxbpg;          /* max. Anzahl von Blöcken pro Zyl.grp. */
/* die folgenden Felder sind wieder berechenbar */
    long    fs_fragshift;       /* Block / Fragment - shift */
    long    fs_fsbtodb;         /* shift-Konstante f. fsbtodb und dbtofsb */
    long    fs_sbsize;          /* Größe des Superblocks */
    long    fs_csmask;          /* Blockdistanz f. Prüfsumme */
    long    fs_csshift;         /* Blocknummer f. Prüfsumme */
    long    fs_nindir;          /* NINDIR */
    long    fs_inopb;           /* INOPB */
    long    fs_nspf;            /* NSPF */
    long    fs_optim;           /* bevorzugte Optimierung, s. unten */
    long    fs_state;           /* Status des Dateisystems */
    long    fs_sparecon[2];     /* reserviert für zukünftige Konstanten */
/* eindeutige Kennung des Dateisystems (z.Z. nicht verwendet u. verwaltet */
    long    fs_id[2];           /* Kennung des Dateisystems */
/* von Anzahl und Größe der Zyl.gruppen abgeleitete Größenangaben */
    daddr_t fs_csaddr;          /* Blockadr. der Gesamtinfo f. Zyl.gruppen */
    long    fs_cssize;          /* Größe der Gesamtinfo f. Zyl.gruppen */
    long    fs_cgsize;          /* Größe der Zyl.gruppe */
/* die folgenden Werte sollten aus der Hardware abgeleitet sein */
    long    fs_ntrak;           /* Spuren pro Zyl. */
    long    fs_nsect;           /* Sektoren pro Spur */
    long    fs_spc;             /* Sektoren pro Zyl. */
/* folgender Wert ist Ergebnis der Plattenpartitionierung */
    long    fs_ncyl;            /* Anzahl d. Zyl. des Dateisystems */
/* aus den vorherigen berechenbare Werte */
    long    fs_cpg;             /* Zyl. pro Gruppe */
    long    fs_ipg;             /* I-Nodes pro Gruppe */
    long    fs_fpg;             /* Blöcke pro Gruppe * fs_frag */
/* nach Systemabsturz neu zu berechnen */
    struct  csum fs_cstotal;     /* Gesamtinfo f. Zylinder */
/* die folgenden Felder werden beim Einhängen des DS bereinigt */
    char    fs_fmod;            /* Kennung f. Superblock-Änderung */
    char    fs_clean;           /* Kennung f. Dateisystem ok. */
    char    fs_ronly;           /* Schreibschutzkennung */
    char    fs_flags;           /* zur Zeit nicht benutzt */
    char    fs_fsmnt[MAXMNTLEN]; /* Name des eingehängten DS */
/* die folgenden Felder enthalten Info über die aktuelle Blockreservierung */
    long    fs_cgrotor;         /* letzte gesuchte Zyl.grp. */
    struct  csum *fs_csp[MAXCSBUFS]; /* Liste v. Pufferzeigern auf fs_cs Info */
    long    fs_cpc;             /* Zyl. pro Zyklus in postbl */
    short   fs_postbl[MAXCPG][NRPOS]; /* Blockvorlauf pro Rotation */
    long    fs_magic;           /* UFS magic number */
    u_char  fs_rotbl[1];        /* Blockliste pro Rotation */
};

```

fs(4)

(UFS)

fs(4)

```
/*
 * Zylindergruppenblock eines Dateisystems.
 */

#define CG_MAGIC 0x090255
struct      cg {
    struct   cg *cg_link;          /* verkettete Liste der Zyl.grp. */
    struct   cg *cg_rlink;         /* f. Speicherabbild der Zyl.grp. */
    time_t   cg_time;              /* letzte Sicherung */
    long      cg_cgx;               /* es handelt sich hier um die cgx-te Zyl.grp */
    short     cg_ncyl;              /* Anzahl der Zyl. dieser Grp. */
    short     cg_niblk;             /* Anzahl der I-Nodes Blöcke dieser Grp. */
    long      cg_ndblk;             /* Anzahl der Datenblöcke dieser Grp. */
    struct     csum cg_cs;          /* Zylinder-Gesamtinfo */
    long      cg_rotor;             /* Position des zuletzt benutzten Blocks */
    long      cg_frotor;            /* Position des zul. benutzten Fragments */
    long      cg_irotor;            /* Position des zuletzt benutzten I-Node */
    long      cg_frsum[MAXFRAG];    /* zählt verfügb. Fragmente */
    long      cg_btot[MAXCPG];      /* Summe der Blöcke pro Zyl. */
    short     cg_b[MAXCPG][NRPOS]; /* Positionen freier Blöcke */
    char      cg_iused[MAXIPG/NBBY]; /* verwendete I-Node Abbildung */
    long      cg_magic;             /* magic number für Zyl.-Gruppeninformation */
    u_char     cg_free[1];          /* Freiblock-Abbildung */
};
```

SIEHE AUCH

ufs-spezifischer inode(4)

BEZEICHNUNG

fspec – Formatangaben für Textdateien

BESCHREIBUNG

Manchmal ist es recht nützlich, in Textdateien Tabulatoren beibehalten zu können, die vom Standard abweichen, also nicht in jeder achten Spalte gesetzt sind. Solche Textdateien müssen normalerweise in das Standardformat umgewandelt werden, bevor sie von Systemkommandos unter SINIX verarbeitet werden können. Sehr oft geschieht das durch Ersetzen der Tabulatoren durch die entsprechende Anzahl von Leerzeichen. Mit einer Formatspezifikation als erste Zeile einer Textdatei läßt sich angeben, wie die Tabulatoren im Rest der Datei aufzulösen sind.

Eine Formatangabe besteht aus einer Reihe von Parametern, die durch Leerzeichen voneinander getrennt sind, und mit den beiden Zeichenketten <: und :> geklammert sind. Ein Parameter besteht aus einem Buchstaben als Schlüsselzeichen, dem unmittelbar ein Wert folgen kann.

Folgende Parameterangaben sind möglich:

t *tabs* Mit dem Parameter werden die Tabulatorpositionen in der Datei festgelegt. Als Wert *tabs* für *t* muß einer der folgenden angegeben werden:

1. Eine Aufzählung von Spaltennummern, die als Tabulatorpositionen gelten. Die Spaltennummern sind durch ein Komma voneinander getrennt.
2. Ein Minuszeichen (-), dem unmittelbar eine ganze Zahl *n* folgt, welche die Sprungweite der Tabulatoren festlegt.
3. Ein Minuszeichen (-), dem die Angabe einer "vorgefertigten" Tabulatorspezifikation folgt.

Die Standardtabulatoren können mit *t-8* oder wahlweise mit *t1,9,17,25,* usw. definiert werden. Vorgefertigte Tabulatorspezifikationen werden mit Hilfe des Kommandos *tabs* erzeugt.

s *größe* Mit dem Parameter *s* wird die maximale Zeilenlänge definiert. Der Wert *größe* muß ganzzahlig sein. Die Prüfung der Zeilenlänge wird nach der Ersetzung der Tabulatoren durchgeführt, jedoch bevor ein eventueller linker Rand vorangestellt wird.

mr *rand* Mit dem Parameter wird die Anzahl von Leerzeichen festgelegt, die in jeder Zeile am Anfang eingefügt werden sollen. Der Wert *rand* muß ganzzahlig sein.

d Zu diesem Parameter gibt es keine Wertangaben. Ist er angegeben, wird die Zeile mit der Formatangabe nicht in die umgewandelte Datei übernommen.

e Zu diesem Parameter gibt es keine Wertangaben. Ist er angegeben, wird die aktuelle Formatspezifikation nur solange beibehalten, bis in der Datei eine neue Formatangabe gefunden wird.

Als Voreinstellung für nicht angegebene Parameter gilt `t=8` und `m0`. Ist der Parameter `s` nicht angegeben, wird keine Prüfung bezüglich der Zeilenlängen durchgeführt. Enthält die erste Zeile der Datei keine Formatspezifikation, gelten die oben genannten Voreinstellungen in der gesamten Datei. Als Beispiel folgt eine Zeile mit einer Formatspezifikation:

```
* <:t5,10,15 s72:> *
```

Wenn es möglich ist, die Formatzeile als Kommentar anzugeben, kann der Parameter `d` entfallen.

SIEHE AUCH

`ed(1)`, `newform(1)`, `tabs(1)` in den *Kommandos*.

BEZEICHNUNG

fstypes – Protokolldatei für verteilte Dateisysteme

ÜBERSICHT

/etc/dfs/fstypes

BESCHREIBUNG

Die Datei `fstypes` ist in dem Verzeichnis `/etc/dfs` abgelegt und enthält eine Liste der am System installierten Utility-Pakete für verteilte Dateisysteme. Als Voreinstellung für das Dateisystem gilt das in der ersten Zeile dieser Datei aufgeführte Dateisystem. Werden Kommandos aus dem DFS-Paket (Distributed File System Administration) ohne die Option `-F dateisystem-typ` gestartet, verwendet das System die Voreinstellung aus der ersten Zeile von `fstypes`.

Die Voreinstellung kann durch einfaches Editieren der Datei `fstypes` mit einem beliebigen Texteditor verändert werden.

SIEHE AUCH

`dfmounts(1M)`, `dfshares(1M)`, `share(1M)`, `shareall(1M)`, `unshare(1M)`.

BEZEICHNUNG

gettydefs – Steuerdatei für getty

ÜBERSICHT

/etc/gettydefs

BESCHREIBUNG

Die Datei /etc/gettydefs enthält Einträge, die getty steuern. Die einzelnen Einträge sind durch Leerzeilen voneinander getrennt. Jeder Eintrag muß in einer einzigen Zeile stehen. Ein Eintrag umfaßt fünf Felder, die durch das Zeichen # voneinander abgegrenzt werden. Ein Eintrag darf aus maximal 255 Zeichen bestehen. Zu lange Einträge werden abgeschnitten. getty ignoriert Einträge, die mit dem Zeichen # beginnen.

Jeder Eintrag hat folgendes Format:

name#*terminal1*#*terminal2*#*prompt*#*next_name*

name Name des Eintrags. Der Name darf maximal 15 Zeichen lang sein. Dieser Name kann an getty als Argument übergeben werden.

terminal1

Terminaleigenschaften, die getty beim Warten auf die Eingabe eines Login-Namens einstellen soll (siehe *stty(1)*).

terminal2

Terminaleigenschaften, die getty für die Login-Sitzung einstellen soll (siehe *stty(1)*).

prompt

Text, den getty als Login-Aufforderung ausgeben soll (z.B. *login:*). Der Text darf aus maximal 79 Zeichen bestehen.

next_name

Name des Folgeeintrags, den getty als nächstes verwenden soll, wenn der Benutzer die Taste BREAK drückt.

Die Felder *terminal1* und *terminal2* dürfen folgende Werte enthalten (siehe *stty(1)*):

```
IGNBRK BRKINT IGNPAR PARMRK INPCK ISTRIP INLCR IGNCR ICRNL IUCLC
IXON IXANY IXOFF OPOST OLCUC ONLCR OCRNL ONOCR ONLRET OFILL OFDEL
NLDLY NL0 NL1 CRDLY CR0 CR1 CR2 CR3 TABDLY TAB0 TAB1 TAB2 TAB3
BSDLY BS0 BS1 VTDLY VT0 VT1 FFDLY FF0 FF1 B0 B50 B75 B110 B134 B150
B200 B300 B600 B1200 B1800 B2400 B4800 B9600 B19200 B38400 EXTA EXTB
CS5 CS6 CS7 CS8 CSTOPB CREAD PARENB PARODD HUPCL CLOCAL ISIG ICANON
XCASE ECHO ECHOE ECHOK ECHONL NOFLSH
```

BEISPIEL

Beispiel einer Datei /etc/gettydefs mit drei Einträgen, die jeweils in einer Zeile stehen:

```
S# B38400 OPOST ONLCR TAB3 IGNPAR IXON ISTRIP ECHO ECHOE ECHOK
ICANON ISIG CS7 CREAD PARENB PARODD # B38400 OPOST ONLCR TAB3
IGNPAR IXON ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD PARENB
PARODD #Benutzerkennung: #S
```

```
19200# B19200 OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB
ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B19200 OPOST ONLCR
TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP ECHO ECHOE ECHOK ICANON
ISIG CS7 CREAD #login: #9600
```

```
9600# B9600 OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP
ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B9600 OPOST ONLCR TAB3
BRKINT IGNPAR IXON IXANY PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG
CS7 CREAD #login: #19200
```

DATEI

/etc/gettydefs

SIEHE AUCH

stty(1), getty(1M)

BEZEICHNUNG

group – Gruppendatei

ÜBERSICHT

/etc/group

BESCHREIBUNG

Die Datei /etc/group enthält für jede Gruppe die folgenden Informationen:

- Gruppenname
- verschlüsseltes Paßwort
- Gruppennummer
- Liste aller Gruppenmitglieder, getrennt durch Kommata

group ist eine ASCII-Datei. Die einzelnen Felder pro Gruppe sind durch Doppelpunkte voneinander getrennt. Jede Gruppensdefinition steht in einer eigenen Zeile.

Da die Paßwörter verschlüsselt sind, stört es nicht, daß auf die Gruppendatei allgemeine Lese-Erlaubnis erteilt ist, und so kann sie z.B. dazu benutzt werden, zu einer Gruppennummer den zugehörigen Namen herauszufinden.

Bei der Identifizierung und Prüfung der Zugangsrechte eines Benutzers wird die Liste der zusätzlichen Gruppenzugehörigkeit sequentiell mit der Information aus der Gruppendatei initialisiert. Wird dabei festgestellt, daß der Benutzer mehr Gruppen angehört, als dies gemäß Konfiguration die Systemgrenze NGROUPS_MAX zuläßt, wird eine Warnung ausgegeben und es werden keine weiteren zusätzlichen Gruppenzugehörigkeiten zugeteilt.

SIEHE AUCH

groups(1), passwd(1) in den *Kommandos*.

newgrp(1M), setgroups(2), initgroups(3C), unistd(4)

BEZEICHNUNG

`hosts` – Datenbank für Host-Namen

ÜBERSICHT

`/etc/hosts`

BESCHREIBUNG

Die Datei `hosts` enthält Informationen über die im DARPA Internet bekannten Hosts. Zu jedem Host sollte eine eigene Zeile mit der folgenden Information existieren:

internet-adresse offizieller-host-name alias-bezeichnungen

Die einzelnen Teile der Zeile sind durch eine beliebige Anzahl von Leerzeichen oder Tabulatoren voneinander getrennt. Das Zeichen '#' leitet einen Kommentar ein, der bis zum Zeilenende reicht. Routinen, die diese Datei durchsuchen, ignorieren Kommentare. Normalerweise wird diese Datei mit Hilfe der Informationen aus der offiziellen Host-Datenbank erzeugt, die im Network Information Control Center (NIC) gewartet wird. Allerdings können es inoffizielle Alias-Bezeichnungen oder nicht bekannte Hosts nötig machen, zur Aktualisierung lokale Änderungen vorzunehmen.

Netzwerkadressen werden in der üblichen Punktschreibweise angegeben, wie sie die `inet_addr` Routine aus der Internet-Bibliothek zur Adreßmanipulation, `inet(3N)`, erwartet. Mit Ausnahme von Kommentarzeichen, Neue-Zeile-Zeichen und Feldtrennzeichen dürfen Host-Namen beliebige abdruckbare Zeichen enthalten.

BEISPIEL

Eine typische Zeile in der `/etc/hosts` Datei ist die folgende:

```
192.9.1.20    gaia                                # John Smith
```

DATEIEN

`/etc/hosts`

SIEHE AUCH

`gethostent(3N)`, `inet(3N)`.

BEZEICHNUNG

hosts.equiv, .rhosts – vertrauenswürdige Hosts für System und Benutzer

ÜBERSICHT

/etc/hosts.equiv
\$HOME/.rhosts

BESCHREIBUNG

Die Datei /etc/hosts.equiv enthält eine Liste von Hosts, die als vertrauenswürdig gelten. Wenn von einem Host aus dieser Liste eine rlogin(1) oder rsh(1) Anforderung empfangen wird und zusätzlich der Benutzer, der sie stellt, in der Datei /etc/passwd aufgeführt ist, wird das Fern-Login ohne weitere Prüfung akzeptiert. Diese Überprüfung führt die Bibliotheksroutine ruserok durch (siehe rcmd(3N)). In diesem Fall fragt rlogin nicht nach einem Paßwort und die von rsh durchgereichten Kommandos werden ausgeführt. Hat also ein ferner Benutzer eine lokale Benutzernummer, heißt es, daß er vom fernen Rechner aus äquivalenten Zugang hat, wenn der ferne Rechner in dieser Datei aufgeführt ist.

In der Datei hosts.equiv existiert für jeden Host eine eigene Zeile folgender Form:

host-name [*benutzername*]

Normalerweise enthält das Feld *host-name* den Namen des vertrauenswürdigen Hosts, von dem aus ein Fern-Login erlaubt sein soll. Gibt es jedoch eine Zeile, deren einziger Eintrag aus einem Pluszeichen '+' besteht, heißt das, daß alle bekannten Hosts als vertrauenswürdig gelten. Als Host-Name muß der offizielle Name angegeben sein, wie er in der Datenbank hosts(4) steht; es ist im dortigen Eintrag der erste angegebene Name. Alias-Bezeichnungen für Hosts werden nicht erkannt.

Die Benutzerdatei .rhosts

Sowie ein Fern-Login versucht wird, sucht der ferne Login-Dämon nach einer Datei .rhosts im Home-Verzeichnis des Benutzers, der sich anmelden will. Die Benutzerdatei .rhosts ist vom gleichen Aufbau wie die Datei hosts.equiv, dient allerdings dazu, nur einem ganz bestimmten Benutzer auf einem fernen System den Zugang zu gestatten oder zu verwehren. Während ein Eintrag in der Datei hosts.equiv allen Benutzern des Hosts den Zugang zum fernen System erlaubt, gestattet ein Eintrag in .rhosts nur dem Benutzer den Zugang, dessen Home-Verzeichnis die Datei .rhosts enthält. Bei der Verwendung dieser Datei sollte im Home-Verzeichnis des betreffenden Benutzers die Erlaubnis für Lesen und Durchsuchen gesetzt sein, sodaß diese Datei auch gefunden und gelesen werden kann. Im Endeffekt wird beim Versuch des Fern-Login eines Benutzers dessen Datei .rhosts der Datei hosts.equiv vorangefügt. Damit ist die Anmeldung möglich, wenn in der Datei .rhosts des Benutzers ein entsprechender Host-Eintrag existiert.

DATEIEN

/etc/hosts.equiv
/etc/passwd
~/.rhosts
/etc

SIEHE AUCH

rlogin(1), rsh(1), rcmd(3N), hosts(4), passwd(4).

BEZEICHNUNG

inetd.conf – Datenbank für Internet-Server

ÜBERSICHT

/etc/inet/inetd.conf

BESCHREIBUNG

Die Datei `inetd.conf` enthält die Liste der Server, die von `inetd(1M)` aufgerufen werden, wenn eine Internet-Anforderung über die Socket-Schnittstelle eintrifft. Jeder Eintrag für einen Server besteht aus einer Zeile der folgenden Form:

service-name socket-typ protokoll wartezustand ben_nummer server-programm server-argumente

Die einzelnen Felder des Eintrags sind durch Leerzeichen oder Tabulatoren voneinander getrennt. Das Zeichen '#' leitet einen Kommentar ein, der bis zum Zeilenende reicht. Routinen, die diese Datei durchsuchen, ignorieren Kommentare.

<i>service-name</i>	Der Name eines gültigen Service, wie er in der Datei <code>/etc/inet/services</code> steht. Bei RPC-Services steht in diesem Feld der RPC Service-Name, gefolgt von einem Schrägstrich und einer Versionsnummer oder einem Bereich von Versionsnummern (z.B. <code>mountd/1</code>).
<i>socket-typ</i>	Kann von folgendem Typ sein: <code>stream</code> für STREAM-Socket, <code>dgram</code> für Datagram-Socket, <code>raw</code> für Raw-Socket, <code>seqpacket</code> für Sequenced Packet Socket
<i>protokoll</i>	Hier steht eine Protokollspezifikation, wie sie in der Datei <code>/etc/inet/protocols</code> aufgeführt ist. Bei RPC-Services enthält das Feld die Zeichenkette <i>rpc</i> , gefolgt von einem Schrägstrich und den Protokollnamen (z.B. <code>rpc/udp</code> für einen RPC-Service, der das UDP-Protokoll als Transportmechanismus verwendet).
<i>wartezustand</i>	Wird für alle auf <code>nowait</code> gesetzt, außer für single-threaded Datagram Server. Dabei handelt es sich um solche, die die Socket-Schnittstelle nicht freigeben, bis eine Zeitgrenze erreicht ist (wie z.B. <code>comsat(1M)</code> und <code>talkd(1M)</code>). Diese brauchen den Status <code>wait</code> . Obwohl <code>tftpd(1M)</code> eigene Pseudoverbindungen aufbaut, kann doch seine Art, Sohnprozesse zu erzeugen, zu einem Wettlaufzustand führen, wenn sein Status nicht auf <code>wait</code> gesetzt wird.
<i>ben_nummer</i>	Gibt die Benutzernummer an, unter der der Server ablaufen soll. Damit kann man Server so starten, daß sie zur Ablaufzeit andere Zugriffsrechte haben als unter der Kennung <code>root</code> .
<i>server-programm</i>	Hier steht entweder der Pfadname des Serverprogramms, das von <code>inetd</code> zur Durchführung des angeforderten Dienstes aufgerufen werden soll, oder der Begriff <code>internal</code> , falls <code>inetd</code> selbst den Dienst zur Verfügung stellt.

server-argumente

Dieses Feld umfaßt alle restlichen Elemente dieses Eintrags. Für den Fall, daß ein Serverprogramm Argumente erwartet, wird hier die komplette Kommandozeile einschließlich des Kommandonamens angegeben. Falls der Server (aus Kompatibilitätsgründen zu Dämonen, die unter 4.2BSD ablaufen) von `inetd` erwartet, daß die Adresse seines Peers durchgereicht wird, muß als erstes Argument `'%A'` angegeben sein.

DATEIEN

`/etc/inet/inetd.conf`
`/etc/inet/services`
`/etc/inet/protocols`

SIEHE AUCH

`rlogin(1)`, `rsh(1)`, `comsat(1M)`, `inetd(1M)`, `talkd(1M)`, `tftpd(1M)`, `services(4)`.

BEZEICHNUNG

inittab – Steuerdatei für init

ÜBERSICHT

/etc/inittab

BESCHREIBUNG

Die Datei /etc/inittab steuert die von init vorgenommene Prozeßgenerierung. Normalerweise erzeugt init Dämon-Prozesse.

Die Einträge in inittab sind positionsabhängig und haben folgendes Format:

id:bstufe:aktion:kommando

Jeder Eintrag ist durch ein Neue-Zeile-Zeichen abgeschlossen; dieses kann durch einen vorangestellten Gegenschrägstrich (\) entwertet werden, um eine Fortsetzungszeile zu ermöglichen. Ein Eintrag darf bis zu 512 Zeichen lang sein. Kommentare können in dem Feld *kommando* eingefügt werden. Dabei sind die Konventionen für Kommentare zu beachten, wie sie in sh(1) beschrieben sind. Die Anzahl der Einträge in der Datei inittab ist beliebig, wobei die maximale Größe pro Eintrag zu beachten ist. Die einzelnen Felder pro Eintrag sind:

- id* Dieses Feld dient der eindeutigen Kennzeichnung eines Eintrags und besteht aus ein bis vier Zeichen.
- bstufe* Damit wird die Betriebsstufe festgelegt, auf der dieser Eintrag abgearbeitet werden soll. Betriebsstufen entsprechen der Prozeßkonfiguration im Rechner, d.h. jedem von init erzeugten Prozeß werden eine oder mehrere Betriebsstufen zugeordnet, in der er ablaufen darf. Betriebsstufen werden durch eine Zahl von 0 bis 6 dargestellt. Wenn sich z.B. der Rechner in Betriebsstufe 1 befindet, dürfen nur Prozesse ablaufen, deren Eintrag im Feld *bstufe* eine 1 enthält. Wenn init den Auftrag erhält, Betriebsstufen zu ändern, dann wird allen Prozessen, deren Eintrag im Feld *bstufe* die neue Betriebsstufe nicht enthält, das Signal SIGTERM zugestellt und eine Zeitspanne von fünf Sekunden gewährt, um sich zu beenden, bevor sie durch das Signal SIGKILL gewaltsam beendet werden. In dem Feld *bstufe* kann eine beliebige Kombination der Betriebsstufen von 0 bis 6 aufgeführt sein, so daß der zugehörige Prozeß in jeder davon ablaufen kann. Fehlt die Angabe einer Betriebsstufe, wird die Gültigkeit aller Stufen von 0 bis 6 für diesen Prozeß angenommen. Drei weitere Angaben a, b und c können ebenfalls in dem Feld *bstufe* vorkommen, obwohl sie eigentlich keine echten Betriebsstufen, sondern Pseudo-Betriebsstufen darstellen. Prozesse mit diesen Angaben werden ausschließlich dann gestartet, wenn sie von einem init oder telinit Prozeß angefordert werden (dabei ist die aktuelle Betriebsstufe des Rechners unerheblich). Siehe dazu init(1M). Weiter gilt, daß eine Anforderung für die Ausführung solcher Prozesse die aktuelle Betriebsstufe nicht ändert. Außerdem werden diese Prozesse nicht gewaltsam beendet, wenn init in eine andere Betriebsstufe wechselt, es sei denn, daß ihr Eintrag in inittab komplett gelöscht ist oder in dem Feld *aktion* die Zeichenkette off enthalten ist oder aber, daß init in den Einbenutzer-Betrieb wechselt.

<i>aktion</i>	Den Schlüsselwörtern in diesem Feld entnimmt <code>init</code> , wie der in dem Feld <i>kommando</i> angegebene Prozeß behandelt werden soll. <code>init</code> kennt hierfür die folgenden Aktionen:
<code>respawn</code>	Gibt es den Prozeß nicht, wird er gestartet. Es wird nicht auf dessen Beendigung gewartet, sondern die Datei <code>inittab</code> weiter abgearbeitet. Stirbt der Prozeß, wird er neu gestartet. Gibt es den Prozeß, wird ohne weitere Aktion die Datei <code>inittab</code> weiter abgearbeitet.
<code>wait</code>	Zu dem Zeitpunkt, da <code>init</code> in die Betriebsstufe wechselt, die in dem Feld <i>bstufe</i> des zugehörigen Prozesseintrags enthalten ist, wird dieser Prozeß gestartet und auf dessen Beendigung gewartet. Solange <code>init</code> in der gleichen Betriebsart verbleibt, wird dieser Prozess nicht mehr gestartet.
<code>once</code>	Zu dem Zeitpunkt, da <code>init</code> in die Betriebsstufe wechselt, die in dem Feld <i>bstufe</i> des zugehörigen Prozesseintrags enthalten ist, wird dieser Prozeß gestartet und nicht auf dessen Beendigung gewartet. Stirbt der Prozeß, wird er nicht neu gestartet. Wechselt <code>init</code> in eine neue Betriebsstufe und der Prozeß läuft noch aufgrund eines früheren Betriebsartwechsels, wird er ebenfalls nicht neu gestartet.
<code>boot</code>	Dieser Eintrag wird nur ausgewertet, wenn <code>init</code> zum Zeitpunkt des Systemstarts die Datei <code>inittab</code> liest. <code>init</code> startet den zugehörigen Prozeß und wartet nicht auf dessen Beendigung. Stirbt der Prozeß, wird er nicht neu gestartet. Damit diese Anweisung auch sinnvoll ist, sollte das Feld <i>bstufe</i> entweder mit der Voreinstellung belegt sein oder aber mit der Betriebsstufe von <code>init</code> zur Ladezeit übereinstimmen. Diese Einstellung ist für Initialisierungsfunktionen nach einem hardwaremäßigen Neustart des Systems angebracht.
<code>bootwait</code>	Dieser Eintrag wird dann ausgewertet, wenn <code>init</code> zum ersten Mal vom Einbenutzer- in den Mehrbenutzerbetrieb wechselt, nachdem das System geladen ist. (Ist <code>initdefault</code> auf 2 gesetzt, läuft der Prozeß unmittelbar nach dem Ladevorgang.) <code>init</code> startet den Prozeß und wartet auf dessen Beendigung. Stirbt er, wird er nicht neu gestartet.
<code>powerfail</code>	Der zu diesem Eintrag gehörende Prozeß wird nur dann gestartet, wenn <code>init</code> ein Stromausfall-Signal SIGPWR erhält (siehe <code>signal(2)</code>).

<code>powerwait</code>	Der zu diesem Eintrag gehörende Prozeß wird nur dann gestartet, wenn <code>init</code> ein Stromausfall-Signal <code>SIGPWR</code> erhält. <code>init</code> wartet auf die Beendigung dieses Prozesses, bevor es <code>inittab</code> weiter abarbeitet.
<code>off</code>	Gibt es den zu diesem Eintrag gehörenden Prozeß nicht, bleibt der Eintrag unberücksichtigt. Andernfalls wird dem Prozeß zur Warnung das Signal <code>SIGTERM</code> zugestellt und eine Zeitspanne von fünf Sekunden gewährt, um sich zu beenden, bevor er gewaltsam mit dem Signal <code>SIGKILL</code> beendet wird.
<code>ondemand</code>	Diese Anweisung ist ein Synonym für <code>respawn</code> . Sie hat dieselbe Funktionalität wie <code>respawn</code> , trägt aber einen anderen Namen, um sie deutlich von einer Verbindung mit Betriebsarten abzusetzen, da sie ausschließlich in Verbindung mit den Werten <code>a</code> , <code>b</code> und <code>c</code> des Feldes <i>bstufe</i> (s. <i>bstufe</i>) verwendet wird.
<code>initdefault</code>	Ein Eintrag mit dieser Anweisung wird nur beim Erstaufruf von <code>init</code> ausgewertet. Er bestimmt die Betriebsstufe, in der <code>init</code> anfangs abläuft. Dazu entnimmt <code>init</code> aus dem Feld <i>bstufe</i> den höchsten Wert und verwendet ihn als Betriebsstufe, in der es beginnt. Ist das Feld <i>bstufe</i> leer, wird es als mit <code>0123456</code> belegt betrachtet. Demzufolge startet <code>init</code> auf Betriebsstufe 6 mit der Auswirkung, daß das System in einer Endlosschleife landet, nämlich auf der Firmware aufgesetzt und den Ladevorgang fortwährend neu startet. Findet <code>init</code> keinen Eintrag <code>initdefault</code> in der <code>inittab</code> -Datei, erfragt es die anfängliche Betriebsstufe zum Zeitpunkt des Neustarts vom Benutzer.
<code>sysinit</code>	Zu Einträgen mit dieser Art Anweisung gehörende Prozesse werden ausgeführt, bevor <code>init</code> auf die Konsole zuzugreifen versucht (um z.B. die Eingabeaufforderung auf der Konsole <code>Console Login:</code> auszugeben). Einträge dieser Art sollten nur dafür verwendet werden, um Geräte zu initialisieren, die <code>init</code> möglicherweise für die Rückfrage bezüglich der Anfangs-Betriebsart benutzen will. <code>init</code> wartet auf die Beendigung solcher Prozesse, bevor es mit der Initialisierung fortfährt.
<i>kommando</i>	Dieses Feld enthält ein auszuführendes Kommando. Dem gesamten Inhalt des Feldes wird <code>exec</code> vorangestellt und das Ganze über einen <code>fork</code> -Aufruf an die Shell <code>sh</code> in der Form <code>sh -c exec kommando</code> übergeben. Daher kann in dem Feld <i>kommando</i> ein beliebiges Kommando in syntaktisch korrekter <code>sh</code> -Notation stehen.

inittab (4)

inittab (4)

SIEHE AUCH

`init(1M)`, `ttymon(1M)`, `exec(2)`, `open(2)`, `signal(2)`
`sh(1)`, `who(1)` in den *Kommandos*.

BEZEICHNUNG

inode (bfs) – Format eines I-Node in einem bfs-Dateisystem

ÜBERSICHT

```
#include <sys/types.h>
#include <sys/fs/bfs.h>
```

BESCHREIBUNG

```
struct bfs_dirent
{
    ushort  d_ino; /* Indexnummer */
    daddr_t d_sblock; /* Startblock */
    daddr_t d_eblock; /* Endblock */
    daddr_t d_eoffset; /* EOF-Distanz (absolut) */
    struct  bfsvattr d_fattr; /* Dateiattribute */
};
```

Zur Definition des Datentyps `daddr_t` siehe `types(5)`. Die Struktur `bfsvattr` ist in der include-Datei `<sys/fs/bfs.h>` definiert.

SIEHE AUCH

bfs-spezifisches `fs(4)`, `types(5)`.

BEZEICHNUNG

inode (s5) – Format eines I-Node in einem s5- Dateisystem

ÜBERSICHT

```
#include <sys/types.h>
#include <sys/fs/s5ino.h>
```

BESCHREIBUNG

In einem s5-Dateisystem ist die Struktur eines I-Node für eine Normaldatei oder ein Verzeichnis folgendermaßen definiert (die Definition der Struktur steht in <sys/fs/s5ino.h>):

```
/* Struktur eines I-Node, wie sie auf Platte steht. */
struct dinode
{
    o_mode_t di_mode; /* Zugriffsrechte und Dateityp */
    o_nlink_t di_nlink; /* Anzahl der Verweise auf die Datei */
    o_uid_t di_uid; /* Benutzernummer des Eigentümers */
    o_gid_t di_gid; /* Gruppennummer des Eigentümers */
    off_t di_size; /* Dateilänge in Bytes */
    char di_chardir[39]; /* Verweise auf Datenblöcke */
    unsigned char di_gen; /* Dateierzeugungsnummer */
    time_t di_atime; /* Zeitstempel des letzten Zugriffs */
    time_t di_mtime; /* Zeitstempel der letzten Änderung */
    time_t di_ctime; /* Zeitstempel der letzten
                        Status-Änderung */
};
/*
 * Von den 40 Bytes für Adressen
 * sind 39 Byte als 13 Verweise
 * zu je 3 Byte auf Datenblöcke
 * und das 40. Byte als
 * Dateierzeugungsnummer genutzt
 */
```

Zur Definition der Datentypen off_t und time_t siehe types(5).

SIEHE AUCH

stat(2), l3tot(3C), s5-spezifisches fs(4), types(5)

BEZEICHNUNG

inode (ufs) – Format eines I-Node in einem ufs- Dateisystem

ÜBERSICHT

```
#include <sys/param.h>
#include <sys/types.h>
#include <sys/vnode.h>
#include <sys/fs/ufs_inode.h>
```

BESCHREIBUNG

Der I-Node ist der zentrale Punkt aller Dateimanipulationen unter SINIX. Für jede angelegte Datei, jedes aktuelle Verzeichnis, jedes eingehängte Dateisystem, jeden Abbildeintrag und das Root- Verzeichnis existiert ein eindeutiger I-Node. Er ist durch das Zahlenpaar Gerätenummer/I-Node-Nummer bestimmt. Die Werte in `icommon` werden aus dem permanenten I-Node des aktuellen Datenträgers übernommen.

```
#define EFT_MAGIC 0x90909090 /* "Magic Cookie" für EFT */
#define NDADDR12 /* Direkte Adressen im I-Node */
#define NIADDR3 /* Indirekte Adressen im I-Node */

struct inode {
    struct inode *i_chain[2]; /* muß erster Eintrag sein */
    struct vnode i_vnode; /* mit diesem I-Node verbundener V-Node */
    struct vnode *i_devvp; /* V-Node für Block E/A */
    u_short i_flag;
    dev_t i_dev; /* Gerätenummer des I-Node */
    ino_t i_number; /* I-Node-Nummer, 1-zu-1 Geräteadresse */
    off_t i_diroff; /* Distanz des letzten Eintrags in dir */
    struct fs *i_fs; /* zu diesem I-Node gehöriges Dateisystem */
    struct dqot *i_dquot; /* Steuerstruktur zu dieser Datei */
    short i_owner; /* proc-Index des sperrenden Prozesses */
    short i_count; /* Anzahl der I-Node Sperren für Eigentümer */
    short i_rwowner; /* proc-Index des Prozesses mit rw-Sperre */
    daddr_t i_nextr; /* Distanz für nächsten Lesezugriff
                     (Vorlesen; read-ahead) */
    struct inode *i_freef; /* Vorwärtsverweis in die Liste der
                           freien Blöcke */
    struct inode **i_freeb; /* Rückwärtsverweis in die Liste der
                           freien Blöcke */
    ulong i_vcode; /* Versionsattribut */
    ulong i_mapcnt; /* Abbildung auf Dateiseiten */
    int *i_map; /* Blockliste der entsprechenden Datei */
};
```

```

    struct icommon {
        o_mode_t ic_smode; /* 0: Dateityp und Zugriffsrechte */
        short ic_nlink; /* 2: Anzahl der Dateiverweise */
        o_uid_t ic_suid; /* 4: Benutzernummer des Eigentümers */
        o_gid_t ic_sgid; /* 6: Gruppennummer des Eigentümers */
        quad ic_size; /* 8: Dateigröße in Byte */
#ifdef _KERNEL
        struct timeval ic_atime; /* 16: Zeitpunkt des letzten Zugriffs */
        struct timeval ic_mtime; /* 24: Zeitpunkt der letzten Änderung */
        struct timeval ic_ctime; /* 32: Zeitpunkt der letzten
                                Status-Änderung */
#else
        time_t ic_atime; /* 16: Zeitpunkt des letzten Zugriffs */
        long ic_at spare;
        time_t ic_mtime; /* 24: Zeitpunkt der letzten Änderung */
        long ic_mt spare;
        time_t ic_ctime; /* 32: Zeitpunkt der letzten Status-Änderung */
        long ic_ct spare;
#endif
        daddr_t ic_db[NDADDR]; /* 40: Adressen von Datenblöcken */
        daddr_t ic_ib[NIADDR]; /* 88: Indirekte Blöcke */
        long ic_flags; /* 100: Status, zur Zeit nicht benutzt */
        long ic_blocks; /* 104: Anzahl der tatsächlich
                        reservierten Blöcke */
        long ic_gen; /* 108: Dateierzeugungsnummer */
        mode_t ic_mode; /* 112: EFT Version von mode */
        uid_t ic_uid; /* 116: EFT Version von uid */
        gid_t ic_gid; /* 120: EFT Version von gid */
        ulong ic_eftflag; /* 124: EFT Version */
    } i_ic;
};

struct dinode {
    union {
        struct icommon di_icom;
        char di_size[128];
    } di_un;
};

```

SIEHE AUCH

ufs-spezifisches fs(4)

BEZEICHNUNG

`issue` – Datei für den Login-Bildschirm

ÜBERSICHT

`/etc/issue`

BESCHREIBUNG

Die Datei `/etc/issue` enthält den Login-Bildschirm mit dem Login-Prompt. `issue` ist eine ASCII Datei, die von dem Programm `getty` eingelesen wird und auf allen Terminals ausgegeben wird, für die gemäß den Einträgen in der Datei `lines` ein Kindprozeß (neu) erzeugt worden ist.

DATEIEN

`/etc/issue`

SIEHE AUCH

`login(1)` in den *Kommandos*.

BEZEICHNUNG

limits – Include-Datei für implementierungsspezifische Konstanten

ÜBERSICHT

```
#include <limits.h>
```

BESCHREIBUNG

Die Include-Datei limits.h ist eine Liste von minimalen Größen-Beschränkungen, die durch eine spezifische Implementierung des Betriebssystems auferlegt werden.

```
ARG_MAX      5120      /* max. Länge der Argumente von exec */
CHAR_BIT     8         /* max. # von Bits in einem "char" */
CHAR_MAX     127       /* max. Wert eines "char" */
CHAR_MIN     -128      /* min. Wert eines "char" */
CHILD_MAX    25        /* max. # von Prozessen je Benutzer */
CLK_TCK      _sysconf(3) /* Zeittakte pro Sekunde */
DBL_DIG      15        /* Rechengenauigkeit einer "Gleitkommazahl mit
                        doppelter Genauigkeit" */
DBL_MAX      1.7976931348623157E+308
                        /* max. dezimaler Wert einer "Gleitkommazahl
                        mit doppelter Genauigkeit" */
DBL_MIN      2.2250738585072014E-308
                        /* min. dezimaler Wert einer "Gleitkommazahl
                        mit doppelter Genauigkeit" */
FCHR_MAX     1048576   /* max. Größe einer Datei in Bytes */
FLT_DIG      6         /* Rechengenauigkeit einer "Gleitzahl" */
FLT_MAX      3.40282347e+38 /* max. dezimaler Wert einer "Gleitzahl" */
FLT_MIN      1.17549435E-38 /* min. dezimaler Wert einer "Gleitzahl" */
INT_MAX      2147483647 /* max. Wert von "int" */
INT_MIN      (-2147483647-1) /* min. Wert von "int" */
LINK_MAX     1000      /* max. # von Verweisen auf eine einzelne Datei */
LOGNAME_MAX  8         /* max. # von Zeichen in einer Benutzerkennung */
LONG_BIT     32        /* # von Bits einer Variablen vom Typ "long" */
LONG_MAX     2147483647 /* max. Wert einer "long int" Variablen */
LONG_MIN     (-2147483647-1) /* min. Wert einer "long int" Variablen */
MAX_CANON    256       /* max. Bytes in einer Zeile für
                        kanonische Verarbeitung */
MAX_INPUT    512       /* max. Größe eines Zeichen-Eingabe-Puffers */
MB_LEN_MAX   5         /* max. # von Bytes in einem mehrere Byte
                        langen Zeichen */
NAME_MAX     14        /* max. # von Zeichen in einem Dateinamen */
NGROUPS_MAX  16        /* max. # von Gruppen für einen Benutzer */
NL_ARGMAX    9         /* max. Wert von "digit" in Aufrufen für die
                        NLS Funktionen printf() und scanf() */
NL_LANGMAX   14        /* max. # von Bytes in einem LANG Namen */
NL_MSGMAX    32767     /* max. Mitteilungsnummer */
NL_NMAX      1         /* max. # von Bytes in N-to-1 Abbildungs-Zeichen */
NL_SETMAX    255       /* max. Satz Nummer */
NL_TEXTMAX   255       /* max. # von Bytes in einer Nachrichtenzeile */
NZERO        20        /* Standard-Prozesspriorität */
OPEN_MAX     60        /* max. # von Dateien, die ein Prozeß geöffnet
                        halten kann */
PASS_MAX     8         /* max. # von Zeichen in einem Paßwort */
PATH_MAX     1024      /* max. # von Zeichen in einem Pfadnamen */
PID_MAX      30000     /* max. Wert für eine Prozeßnummer */
```

limits (4)

PIPE_BUF	5120	/* max. # Bytes einzeln beim Schreibvorgang in eine Pipe */
PIPE_MAX	5120	/* max. # Bytes geschrieben in eine Pipe in einem Schreibvorgang */
SCHAR_MAX	127	/* max. Wert von "signed char" */
SCHAR_MIN	(-128)	/* min. Wert von "signed char" */
SHRT_MAX	32767	/* max. Wert von "short int" */
SHRT_MIN	(-32768)	/* min. Wert von "short int" */
STD_BLK	1024	/* # Bytes in einem physikalischen E/A Block */
SYS_NMLN	257	/* 4.0 Größe von utsname Elementen */ /* ebenfalls definiert in sys/utsname.h */
SYSPID_MAX	1	/* max. pid von System Prozessen */
TMP_MAX	17576	/* max. # von eindeutigen Namen, die von tmpnam generiert wurden */
UCHAR_MAX	255	/* max. Wert von "unsigned char" */
UID_MAX	60000	/* max. Wert für eine Benutzer- oder Gruppenkennung */
UINT_MAX	4294967295	/* max. Wert von "unsigned int" */
ULONG_MAX	4294967295	/* max. Wert von "unsigned long int" */
USHRT_MAX	65535	/* max. Wert von "unsigned short int" */
USI_MAX	4294967295	/* max. dezimaler Wert von "unsigned" */
WORD_BIT	32	/* # von Bits in einem "word" oder "int" */

Die folgenden POSIX Definitionen sind die Mindestwerte, die als Voraussetzung für Anwendungen gelten, die dem POSIX Standard entsprechen. POSIX konforme Implementierungen müssen diese Untergrenzen einhalten.

_POSIX_ARG_MAX	4096	/* max. Länge von auszuführenden Argumenten */
_POSIX_CHILD_MAX	6	/* max. # von Prozessen je Benutzer-Kennung */
_POSIX_LINK_MAX	8	/* max. # von Verbindungen zu einer einzelnen Datei */
_POSIX_MAX_CANON	255	/* max. # von Bytes in einer Eingabezeile */
_POSIX_MAX_INPUT	255	/* max. # von Bytes in der Terminal- Eingabewarteschlange */
_POSIX_NAME_MAX	14	/* # von Bytes in einem Dateinamen */
_POSIX_NGROUPS_MAX	0	/* max. # von Gruppen in einem Prozeß */
_POSIX_OPEN_MAX	16	/* max. # von Dateien, die ein Prozeß geöffnet halten kann */
_POSIX_PATH_MAX	255	/* max. # von Zeichen in einem Pfadnamen */
_POSIX_PIPE_BUF	512	/* max. # von Bytes atomar beim Schreibvorgang in eine Pipe */

BEZEICHNUNG

login – Datei für Login-Standardwerte

ÜBERSICHT

/etc/default/login

BESCHREIBUNG

Optionen für das login-Programm können mit Schlüsselwörtern in /etc/default/login eingestellt oder geändert werden. Die folgenden Schlüsselwörter werden von login erkannt.

CONSOLE Ist es gesetzt, darf sich ausschließlich der Systemverwalter auf dem Terminal anmelden, das als Konsole definiert wurde. Zum Beispiel bedeutet:

```
CONSOLE= /dev/console
```

daß sich ausschließlich root auf der Konsole anmelden darf. Ist CONSOLE nicht in /etc/default/login enthalten, darf sich der Systemverwalter auf jedem beliebigen Terminal anmelden.

ALTSHELL Ist es auf YES gesetzt, wird die Umgebungsvariable SHELL auf die Benutzer-Shell eingestellt, sofern diese Shell nicht /bin/sh ist. Ist es mit NO belegt, werden keine Namen von Nicht-Standard-Shells in die Umgebungsvariable SHELL eingetragen. Der Standardwert für diese Variable ist NO, allerdings sollte sie zur höheren Sicherheit mit YES vorbelegt sein.

PASSREQ Ist es auf YES gesetzt, müssen alle Benutzer ein Paßwort besitzen. Jeder Benutzer ohne Paßwort wird bei der ersten Gelegenheit danach gefragt. Die erste Gelegenheit ist durch die zeitlich begrenzbare Gültigkeit von Paßwörtern pro Benutzer vorgegeben (z.B. dürfen Benutzer ohne Paßwort keines anfordern, wenn die zeitliche Begrenzung der Gültigkeit von Paßwörtern für sie eingeschaltet ist, und die Mindestzeit, bevor ein Paßwort geändert werden kann, noch nicht verstrichen ist).

TIMEZONE Dieses Schlüsselwort setzt die Variable TZ in der Umgebung des Benutzers. Sie muß mit der Zeitzone übereinstimmen, die in /etc/TIMEZONE eingestellt ist.

HZ Hiermit wird die Umgebungsvariable HZ des Benutzers mit der Taktrate des Systems versorgt.

PATH Diese Variable definiert den Standardsuchpfad für Benutzer, die nicht die Benutzernummer 0 haben.

SUPATH Diese Variable setzt den Standardsuchpfad für den Systemverwalter. Ein weiterer Standardpfad für den Systemverwalter steht in /etc/default/su. Dieser wird für Benutzer eingestellt, die mittels su in die privilegierte Benutzerkennung gewechselt sind.

ULIMIT Diese Variable legt die maximale Dateigröße für einen Benutzer fest. Sie wird in Einheiten von 512-Byte Blöcken angegeben.

login (4)

login (4)

TIMEOUT	Diese Variable gibt die Zeitdauer an, welche login nach der Eingabe eines Benutzernamens auf ein Paßwort wartet. Diese Zeitspanne wird in Sekunden angegeben.
UMASK	Diese Variable ist die Standard-umask für Benutzer.
IDLEWEEKS	Diese Variable gibt die Anzahl von Wochen an, welche eine Kennung unbenutzt bleiben darf, ehe sie deaktiviert wird.

DATEIEN

/etc/default/login

BEZEICHNUNG

loginlog – Protokoll von fehlgeschlagenen Login-Versuchen

ÜBERSICHT

/var/adm/loginlog

BESCHREIBUNG

Nach fünf erfolglosen Login-Versuchen, werden alle diese Versuche in der Datei /var/adm/loginlog protokolliert. Diese Datei enthält einen Eintrag für jeden fehlgeschlagenen Versuch. Jeder Eintrag beinhaltet den Benutzernamen, die Terminal-Bezeichnung und den jeweiligen Zeitpunkt.

Dies ist eine ASCII-Datei. Jedes Feld innerhalb eines Eintrags wird vom nächsten durch einen Doppelpunkt getrennt. Jeder Eintrag steht in einer eigenen Zeile.

Standardmäßig ist loginlog nicht vorhanden. Es existiert also keine Aufzeichnung. Wenn ein Protokoll geführt werden soll, muß die Protokolldatei angelegt sein. Eigentümer muß root und Gruppe muß sys sein. Die Lese- und Schreibberechtigung wird nur für den Eigentümer erteilt.

DATEIEN

/var/adm/loginlog

SIEHE AUCH

login(1), passwd(1) in den *Kommandos*.

BEZEICHNUNG

mailcnfg – Initialisierungs-Information für mail und rmail

ÜBERSICHT

/etc/mail/mailcnfg

BESCHREIBUNG

Die Datei /etc/mail/mailcnfg enthält Initialisierungs-Informationen für die mail und rmail Befehle. Jeder Eintrag in mailcnfg besteht aus einer Zeile der Form

schlüsselwort = *wert*

Leerzeichen am Anfang und Ende der Zeile, sowie links und rechts vom Gleichheitszeichen werden ignoriert. In *schlüsselwort* darf kein Leerzeichen vorkommen, in *wert* ist dies zulässig. Ungültige Schlüsselwörter oder fehlerhaftes Format wird stillschweigend übergangen.

Definitionen der Schlüsselwörter

DEBUG	Hat dieselben Werte wie die Aufrufoption -x von mail. Dies ermöglicht die Einstellung einer systemweiten Überwachungs-Ebene. Gewöhnlich wird DEBUG auf 2 gesetzt. Damit wird eine minimale Fehlersuche ermöglicht, was bei Ausfällen von mail und rmail für die Fehlersuche nützlich ist. Der Wert der Option -x beim Aufruf von mail setzt jede Spezifikation von DEBUG in mailcnfg außer Kraft.
CLUSTER	Belegt man <i>wert</i> mit dem Namen des Clusters, hat man die Möglichkeit, eine Anzahl eng verbundener Systeme über nur einen Namen allen anderen Systemen bekannt zu machen. Dieser Name wird dann statt des Knotennamens, wie ihn uname liefert, in der Kopfzeile der ...remote von... Information ausgegeben.
FAILSAFE	Für den Fall, daß über NFS auf das /var/mail Verzeichnis zugegriffen wird, müssen für dieses Verzeichnis bestimmte Vorkehrungen getroffen werden, damit beim Zustellungsversuch lokaler Post der Zugriff darauf unterbunden werden kann (Absturz des fernen Systems, NFS Probleme, usw.). <i>wert</i> gibt an, wohin die aktuelle Nachricht ausgeliefert werden soll. Normalerweise ist dies das ferne System, dem /var/mail tatsächlich gehört. Die Nachricht wird also in eine Warteschlange eingereiht und diesem System zugestellt, sowie es verfügbar ist. Es sei beispielsweise angenommen, daß in einem Cluster (mit den Systemen sysa, sysb, sysc) /var/mail physisch in sysc eingehängt ist und über NFS anderen Systemen zugänglich ist. Angenommen, sysc würde nun abstürzen. Dann wäre auf /var/mail kein Zugriff mehr möglich und auszuliefernde lokale Post ginge an /var/mail des lokalen Systems. Wird nun /var/mail über NFS neu eingehängt, wären alle Nachrichten in dem lokalen Verzeichnis gewissermaßen versteckt und de facto verloren. Der FAILSAFE-Eintrag in mailcnfg dient dazu, dies zu verhindern. Ist er definiert, überprüfen mail und

`rmail` das Vorhandensein einer Datei `/var/mail/:saved`. Ist dieses Unterverzeichnis nicht vorhanden, nimmt `mail` an, daß die NFS-zugängliche `/var/mail` nicht verfügbar ist, und ruft den FAILSAFE-Mechanismus auf. Dadurch wird die Nachricht automatisch an das Verzeichnis weitergeleitet, das in *wert* angegeben ist. In diesem Beispiel wäre *wert* `sysc!%n`. Das Schlüsselwort `%n` wird durch den Namen des Empfängers ersetzt (siehe hierzu `mail(1)`) und die Nachricht an `sysc!Empfängername` weitergereicht. Die Nachricht verbleibt solange auf dem lokalen System, bis `sysc` verfügbar wird, um dann dorthin geschickt zu werden.

`DEL_EMPTY_MFILE` Wenn dieser Eintrag nicht vorhanden ist, gilt die Voreinstellung. In diesem Fall löschen `mail` und `rmail` leere Maildateien mit den Zugriffsrechten `0660` und lassen leere Maildateien mit anderen Zugriffsrechten unangetastet. Ist *wert* auf `yes` gesetzt, werden leere Maildateien ohne Rücksicht auf die Zugriffsrechte immer gelöscht. Wenn *wert* `no` ist, werden leere Maildateien nie gelöscht.

`DOMAIN` Diese Zeichenkette wird als Bereichsname des Systems benutzt statt des Bereichsnamens, wie ihn `getdomainname` zurückliefert.

`SMARTERHOST` Diese Zeichenfolge bezeichnet einen intelligenten Host. Auf diesen kann dann innerhalb der Mail-Kommandoerweiterung durch `%X` zugegriffen werden.

`%mailsurr_schlüsselwort`

Wie in `mailsurr(4)` beschrieben, werden bestimmte vordefinierte Schlüsselwörter, die aus Einzelbuchstaben bestehen, textlich in Ersatzbefehlsfeldern ersetzt, ehe sie ausgeführt werden. Während keines der vorgegebenen Schlüsselwörter in seiner Bedeutung geändert werden darf, dürfen neue definiert werden, um eine Kurzbezeichnung für lange Zeichenketten einzuführen (so wie `/usr/lib/mail/surrcmd`). Diese können dann wiederholt innerhalb der Datei `mailsurr` erscheinen. Großbuchstaben sind für zukünftige Anwendungen reserviert und werden hier ignoriert.

DATEIEN

`/etc/mail/mailcnfg`
`/etc/mail/mailsurr`
`/var/mail/:saved`
`/usr/lib/mail/surrcmd`

SIEHE AUCH

`mailsurr(4)`
`mail(1)` in den *Kommandos*.
`uname(2)`, `getdomainname(3)` im Referenzhandbuch für Programmierer

mailcnfg (4)

mailcnfg (4)

HINWEIS

Wenn auf `/var/mail` durch NFS zugegriffen wird und das Unterverzeichnis `/var/mail/:saved` nicht auf dem lokalen System entfernt ist, wird der `FAILSAFE`-Mechanismus unterlaufen.

BEZEICHNUNG

mailsur – Erweiterungs-Befehle für Routing und Transport von Post

ÜBERSICHT

/etc/mail/mailsur

BESCHREIBUNG

Die Datei mailsur enthält Erweiterungs-Befehle für das Routing und den Transport von Post, die vom mail Dienstprogramm benutzt werden. Jeder Eintrag in mailsur hat drei, durch Leerzeichen getrennte und durch Hochkommata begrenzte, Felder:

'Absender' 'Empfänger' 'Befehl'

oder eine Zeile, die mit dem Schlüsselwort

Defaults:

beginnt.

Einträge und Felder dürfen mehrere Zeilen umfassen, jedoch werden führende Leerzeichen in Fortsetzungszeilen ignoriert. Felder dürfen nach der Ersetzung von Befehlskurzformen (siehe unten) nicht mehr als 1024 Zeichen lang sein.

Die Felder für Absender und Empfänger bestehen aus regulären Ausdrücken. Wenn die Absender- und Empfänger-Felder mit denjenigen der Nachricht, die gerade verarbeitet wird, übereinstimmen, wird der dazugehörige Befehl aufgerufen.

Das Feld *Befehl* kann eines der folgenden fünf Formate haben:

```
A[cept]
D[eny]
T[ranslate] R=[|]Zeichenkette
< S=...;C=...;F=...;Befehl
> Befehl
```

Reguläre Ausdrücke

Die Absender- und Empfänger-Felder bestehen aus regulären Ausdrücken, welche durch die regexp(5), compile und advance Prozeduren in der C-Bibliothek ausgewertet werden. Die regulären Ausdrücke entsprechen denen von ed(1), mit einfachen runden Klammern (), die die Funktion von \(\) übernehmen, einschließlich der Operatoren + und ? von egrep(1). Jedes Hochkomma, das innerhalb der regulären Ausdrücke eingeschlossen ist, muß durch das Voranstellen eines Gegenschrägstrichs entwertet werden oder der reguläre Ausdruck wird nicht ordnungsgemäß interpretiert.

Der mail Befehl stellt ein Zirkumflex (^) an den Anfang und hängt ein Dollar-Zeichen (\$) an das Ende jedes regulären Ausdrucks, sodaß er einer vollständigen Zeichenfolge entspricht. Deswegen ist es falsch, die Folge *^regulärer Ausdruck\$* im Absender- bzw. Empfänger-Feld zu benutzen. Um Unabhängigkeit von Groß- oder Kleinschreibung zu erreichen, werden alle regulären Ausdrücke vor Kompilierung in Kleinbuchstaben umgewandelt, und alle Absender- und Empfänger-Informationen werden vor dem Vergleich in Kleinbuchstaben umgewandelt. Diese Konvertierung wird für den Mustervergleich von regulären Ausdrücken ausgeführt; der Kopfteil der Nachricht selbst wird nicht verändert.

Die Möglichkeiten von `regex` für den Mustervergleich eines Unterausdrucks können im Befehlsfeld ausgenutzt werden. Bis zu 9 durchnummerierte Klammernungen können verwendet werden. Jedes Vorkommen von `\\n` in der Ersatzzeichenfolge wird durch die entsprechende (...) Teilzeichenkette in dem zu vergleichenden Muster ersetzt. Die Unterausdrücke von Absender- und Empfänger-Feld sind einzeln ansprechbar, wobei die Felder von links nach rechts von 1 bis 9 nummeriert sind.

Accept- und Deny-Befehle

`Accept` weist `rmail` an, seine Verarbeitung mit der Datei `mailsur` fortzusetzen, jedoch jedes darauffolgende passende `Deny` zu ignorieren. Das heißt, daß diese Nachricht zur Zustellungs-Verarbeitung auf jeden Fall angenommen wird. `Deny` weist `rmail` an, die Verarbeitung der `mailsur` Datei zu beenden und eine negative Zustellungsmitteilung an den Absender der Nachricht zu senden. Der jeweils zuerst angetroffene Befehl hat Vorrang.

Translate-Befehl

`Translate` erlaubt wahlweise vollautomatische Übersetzung von Empfänger-Adressen-Informationen. Die *Empfänger* Ersatzzeichenfolge ist als `R=Zeichenkette` angegeben.

Zum Beispiel, bei einer gegebenen Befehlszeile in der Form

```
'.' ' ([!]+)@(.+)\.EUO\.ATT\.com' 'Translate R=attmail!\\2!\\1'
```

und einer Empfänger-Adresse von `rob@sysa.EUO.att.COM`, wäre die resultierende Empfänger-Adresse `attmail!sysa!rob`.

Ist das erste Zeichen nach dem Gleichheitszeichen ein `'`, wird der Rest der Zeichenfolge als Befehlszeile interpretiert und direkt durch `rmail` ausgeführt. Wenn eine `sh(1)` Syntax erforderlich ist (Metazeichen, Umleitung usw.), dann muß der Erweiterungs-Befehl die folgende Form haben:

```
sh -c "shell Kommandozeile..."
```

Es ist besonders darauf zu achten, in der Kommandozeile vorhandene Gegenschrägstriche, Hochkommas oder Anführungszeichen ordnungsgemäß zu ersetzen, da `rmail` Anführungszeichen benutzt, um durch Leerzeichen begrenzte Felder zu gruppieren, die als ein einzelnes Argument für `execl(2)` gedacht sind. Es wird angenommen, daß der ausgeführte Befehl eine oder mehrere Ersatz-Zeichenketten auf `stdout` schreiben wird (je einen pro Zeile). Wird mehr als eine Zeile zurückgeliefert, wird jede Zeile als weiterer Empfänger für die Nachricht angenommen. Dieser Mechanismus ist nützlich für Erweiterungen der Verteilerliste. Wie oben angegeben, wird jedes Vorkommen von `\\n` durch die geeignete Teilkette ersetzt, bevor der Befehl ausgeführt wird. Wenn der aufgerufene Befehl nicht mindestens eine Ersatzzeichenfolge zurückgibt (keine Ausgabe oder nur eine neue Zeile), wird die originale Zeichenfolge nicht verändert. Zum Beispiel erlaubt die Befehlszeile

```
'.' ' (\\.)' 'Translate R=|/usr/bin/Suchpfad \\1'
```

lokale Routing-Entscheidungen zu treffen.

Wurde die Zeichenkette für die Empfänger-Adressen verändert, wird mailsur von Anfang an mit der/den neuen Adresse(n) neu gelesen, wobei ein zuvor festgestelltes Accept (siehe oben) ignoriert wird.

< Befehl

Der < Befehl wird als Teil des Transport- und Zustellungs-Mechanismus aufgerufen. Auf der Standard-Eingabe des Befehls liegt eine fertig vorbereitete Nachricht bereits vor. Es sind drei Zustände nach der Ausführung des Befehls möglich:

- Success Der Befehl hat die Nachricht erfolgreich zugestellt. Was im Einzelfall die erfolgreiche Zustellung ausmacht, hängt vom Kontext des jeweiligen Erweiterungsbefehls ab. Der rmail Prozeß nimmt an, daß keine weitere Verarbeitung dieser Nachricht für den gegenwärtigen Empfänger erforderlich ist.
- Continue Der Befehl führte zwar einige Funktionen aus (z.B. Protokollierung des fernen Nachrichten-Verkehrs), hat jedoch nicht erledigt, was als erfolgreiche Nachrichten-Zustellung definiert wurde. Der rmail Prozeß arbeitet die mailsur Datei weiter ab, um nach einem alternativen Zustellungs-Mechanismus zu suchen.
- Failure Es sind nicht behebbare Fehler aufgetreten. Der rmail Prozeß stoppt die Verarbeitung der Nachricht und sendet an den Absender eine Meldung, daß die Nachricht nicht zugestellt wurde. Diese Meldung enthält alle stdout und stderr Ausgaben, die durch den Befehl generiert wurden.

Die Semantik des < Befehlsfelds in der mailsur Datei erlaubt die Definition von eigenen Exit-Codes, die Success, Continue und Failure für jedes Erweiterungskommando individuell beschreiben. Die Spezifikation eines Exit-Codes erfolgt nach folgender Syntax:

< LZ [zustands_id=ec[,ec...]] ... LZ erweiterungsbefehl

LZ ist Leerzeichen. zustands_id kann den Wert S, C oder F haben. zustands_id kann in jeder Reihenfolge angegeben sein. ec kann sein:

jede Ganzzahl $0 \leq n \leq 255$ (Negative Exit-Codes sind nicht möglich. Siehe exit(2) und wait(2).)

ein Bereich von ganzen Zahlen in der Form *Untergrenze–Obergrenze*, in dem die Grenzen ≥ 0 und ≤ 255 sind,

*: jeder Wert ist möglich

Zum Beispiel zeigt ein Befehlsfeld in der Form:

'< S=1-5,99;C=0,12;F=*; Befehl %R'

an, daß Exit-Codes von 1 bis 5 und 99, als Success zu betrachten sind, 0 (Null) und 12 Continue anzeigen, und daß alle anderen Werte Failure bedeuten. Wenn ein Feld nicht ausdrücklich angegeben ist, gelten die Standard-Einstellungen S=0;C=*;.

Es ist möglich, daß mehrdeutige Einträge vorhanden sind, wenn zwei Ende-Zustände den gleichen Wert haben, zum Beispiel: `S=12,23;C=*;F=23,52;` oder `S=*;C=9;F=*`. Um dem gerecht zu werden, sucht `rmail` nach expliziten Exit-Codes (das heißt, kein `"*"`), und zwar in der Reihenfolge von `Success`, `Continue`, `Failure`. Wenn keine explizite Übereinstimmung gefunden wird, sucht `rmail` in derselben Reihenfolge nach `"*"`.

Es ist möglich, einen Ende-Zustand durch Wählen eines ungültigen Zustandswerts vollständig auszuschließen. Da Exit-Codes zwischen 0 und 255 (einschließlich) liegen müssen, bietet sich beispielsweise ein Wert von 256 an. Handelte es sich beispielsweise um einen Erweiterungs-Befehl, der alle Nachrichtenübermittlungen protokollieren sollte, ergäbe ein `mailsur` Eintrag von

```
'(.)' '(.)' '<S=256;C=*; /usr/lib/mail/surrcmd/logger \\\1 \\\2'
```

immer `Continue`.

Erweiterungsbefehle werden durch `rmail` direkt ausgeführt. Wenn eine Shell-Syntax erforderlich ist (Metazeichen, Umleitung, usw.), muß der Erweiterungsbefehl die folgende Form haben:

```
sh -c "Shell Befehlszeile..."
```

Es ist besonders darauf zu achten, alle entwerteten Gegenschrägstriche oder andere Sonderzeichen für die Shell ordnungsgemäß zu entwerten, wie im Abschnitt *"Translate"* oben angegeben.

Wenn es keine zugehörigen `<` Befehle gibt, oder alle zugehörigen `<` Befehle mit einer `Continue`-Anzeige abschließen, versucht `rmail` die Nachricht selbst zuzustellen, indem angenommen wird, daß der Empfänger lokal ist und die Nachricht an `/var/mail/Empfänger` zuzustellen ist.

> Befehl

Der Sinn eines `>` Befehls ist, daß er nach einer erfolgreichen Zustellung aufgerufen wird, um eine eventuell notwendige Nachbearbeitung durchzuführen. Die zugehörigen `>` Befehle werden nur ausgeführt, wenn ein `<` Befehl eine erfolgreiche Zustellung (siehe den vorhergehenden Abschnitt) anzeigt oder die lokale Nachrichten-Übermittlung erfolgreich ist. Die `mailsur` Datei wird neu durchsucht und alle übereinstimmenden `>` Befehle, nicht nur diejenigen, die dem erfolgreichen `<` Befehl folgen, werden der Reihenfolge nach ausgeführt. Der Ende-Status eines `>` Befehls wird ignoriert.

Defaults: Zeile

Die Standard-Einstellungen können durch Erstellen einer eigenen Zeile in der `mailsur` Datei in folgender Form neu definiert werden:

```
Defaults: [S=...;][C=...;][F=...;]
```

Defaults: Zeilen werden erkannt und die angegebenen Standard-Werte neu definiert, sobald die Zeile während der normalen Verarbeitung der `mailsur` Datei angetroffen wird. Um die Standardwerte global neu zu definieren, sollte die **Defaults:** Zeile die erste Zeile in der Datei sein. Es ist möglich, mehrere **Defaults:** Zeilen in der `mailsur` Datei zu haben, wobei jede nachfolgende Default-Zeile die vorangegangene überschreibt.

Ersetzung von Schlüsselwörtern in Erweiterungs-Befehlen.

Bestimmte Zeichenfolgen werden textlich in Erweiterungs-Befehlen ersetzt, bevor diese aufgerufen werden:

%n	Der volle Name des Empfängers.
%R	Der volle Pfad für Rückmeldungen an den Absender (nützlich für das Senden von Antworten, Zustellungs-Fehler-Meldungen usw.)
%c	Inhalt von Content-Type: Kopfzeile (wenn vorhanden).
%C	text oder binary, abhängig von einem aktuellen Durchsuchen des Inhalts. Dies ist unabhängig von dem Inhalt der zugehörigen Content-Type Kopfzeile (nützlich, falls ckbinar-sys aufgerufen wird.)
%S	Der Inhalt von Subject: Kopfzeile (falls vorhanden).
%l	Inhalt der Content-Length: Kopfzeile.
%L	Der lokale Systemname. Dieser wird entweder CLUSTER aus mailcnfg sein oder der Wert, der von uname ausgegeben wird.
%U	Der lokale Systemname, wie er von uname ausgegeben wird.
%X	Der Wert von SMARTERHOST in mailcnfg.
%D	Der lokale Bereichsname. Dies wird entweder DOMAIN aus mailcnfg sein oder der Wert, der von getdomainname ausgegeben wird.
\\n	Wie oben beschrieben, die entsprechende (...) Teilzeichenkette in den verglichenen Mustern. Dies besagt, daß die regexp Beschränkung von 9 Teilzeichenketten für die RA von Absender- und Empfänger-Feld zusammen gilt.
%schlüsselwörter	Weitere Schlüsselwörter sind in /etc/mail/mailcnfg definiert. Siehe mailcnfg(4).

Die Folgen %L, %U, %D und %schlüsselwörter sind innerhalb der Absender- und Empfänger-Felder ebenso erlaubt wie in den Befehls-Feldern.

Ein Beispiel des mailsurr Eintrags, welcher die eingebaute uux Funktionalität aus vorhergehenden Versionen von rmail ersetzt, ist:

```
'.' '+' '([@!]+)!(.)' '< /usr/bin/uux - \\!rmail (\\2)'
```

Beispiele für Befehlserweiterungen

Die folgenden Beispiele für Mail-Befehlserweiterungen beinhalten die Zustellung von Mitteilungen für eingetroffene Post an Benutzer in einem lokalen Netz sowie die Steuerung von entsprechenden Anzeigelampen, ebenso das Senden von Ausgabedaten an einem Drucker, und schließlich die Aufzeichnung aller rmail Aufträge zwischen zwei fernen Rechnern (deren Nachrichten durch das lokalen System laufen). Es folgt ein Beispiel für eine mailsurr Datei:

```
#
# Es folgen einige häufig verwendete Mail Erweiterungen. Um einige
# oder alle davon zu aktivieren, ist das '#' (Kommentarzeichen)
# am Beginn der entsprechenden Zeilen zu entfernen. Wichtig ist, daß sie
# in der Reihenfolge, in der sie in der Datei stehen, ausgewertet werden,
# deshalb müssen bevorzugte Erweiterungen an den Dateianfang gesetzt werden.
```

```
# Vermeidet alle Shell-Metazeichen
'.+' '.*[';&|^<>()].*' 'Deny'

# Bildet alle Namen in der Form local-machine!user ab auf user
'.+' '%L!(.)' 'Translate R=\1'

# Bildet alle Namen in der Form uname!user ab auf user
# Müssen eingeschaltet sein, wenn mail in einer Cluster-Umgebung benutzt wird.
'.+' '%U!(.)' 'Translate R=\1'

# Bildet alle Namen in der Form user@Host ab auf Host!user
'.+' '([^\[]+)(.+) 'Translate R=\2!\1'

# Bildet alle Namen in der Form Host.uucp!user ab auf Host!user
'.+' '([^\[]+)\.uucp!(.)' 'Translate R=\1!\2'

# Bildet alle Namen in der Form Host.local-domain!user ab auf Host!user
# domain= innerhalb /etc/mail/mailcnfg wird getdomainname(3) außer Kraft setzen.
'.+' '([^\[]+)%D!(.)' 'Translate R=\1!\2'

# Erlaubt Zugriff auf 'attmail' vom fernen Rechner 'sysa'
'sysa!.*' 'attmail!.*' 'Accept'

# Lehnt Zugriff auf 'attmail' von allen anderen fernen Rechnern ab
'.+!.*' 'attmail!.*' 'Deny'

# Sendet mail für 'laser' zum dazugehörigen Laser Drucker
# Stellt sicher, daß Fehler durch Rückmeldung berichtet werden.
'.+' 'laser' '< S=0;F=*; lp -dlaser'

# Läßt alle lokalen Namen vom mail Abkürzungs-Prozessor interpretieren
#
'.+' '[^\[]+' 'Translate R=/usr/bin/mailalias %n'

# Für Fernpost durch nusend
'.+' '([^\[]+)!(.)' '< /usr/bin/nusend -d \1 -s -e -!*rmail \2 -'

# Für Fernpost durch usend
'.+' '([^\[]+)!(.)' '< /usr/bin/usend -s -d\1 -uNoLogin -!*rmail \2 -'

# Für Fernpost durch uucp
'.+' '([^\[]+)!.+' '<S=256;C=0;
      /usr/lib/mail/surrcmd/ckbinarsys -t %C -s \1'
'.+' '([^\[]+)!(.)' '< /usr/bin/uux - \1!rmail (\2)'

# Für Fernpost durch smtp
'.+' '([^\[]+)!(.)' '< /usr/lib/mail/surrcmd/smtpqer %R %n'
```

```
#      Wenn keiner der obigen Befehle funktioniert, dann ändert ein Router die
#      Adresse.
'.+' '.*[@].*' 'Translate R=| /usr/lib/mail/surrcmd/smail -A %n'

#      Wenn keiner der obigen Befehle funktioniert, wird Fernpost an
#      einen intelligenten Host geschickt.
#      Stellt sicher, daß SMARTERHOST= in /etc/mail/mailcnfg definiert ist.
'.+' '.*[@].*' 'Translate R=%X!\n'

#      Zeichnet die erfolgreichen Nachrichten-Zustellungen auf
'(.+)' '(.+)' '>/usr/lib/mail/surrcmd/logger \1 \2'
```

Es ist zu beachten, daß der Aufruf von mail, um Post zu lesen, nichts mit der Datei mailsurr oder der Erweiterung von Befehlen zu tun hat.

Zugriffsschutz

Erweiterungs-Befehle werden mit der Berechtigung von mail (Benutzernummer UID des Aufrufers, Gruppennummer GID von mail) ausgeführt. Dies erlaubt den Erweiterungs-Befehlen, zu überprüfen, ob ihre effektive Gruppennummer GID zur Aufrufzeit mail war. Dies erfordert, daß alle Zusätze für mailsurr vor Einfügung genau geprüft werden, um einen unerlaubten Zugriff auf die Nachrichten eines Benutzers zu verhindern. Alle Erweiterungs-Befehle werden ausgeführt mit dem Pfad /usr/lib/mail/surrcmd:/usr/bin.

Fehlersuche in neuen mailsurr Einträgen

Zur Fehlersuche in mailsurr Dateien wird der Parameter -T des mail Befehls benutzt. Der Parameter -T verlangt ein Argument, das den Pfadnamen einer zu testenden mailsurr Datei angibt. Wenn dieser NULL ist (wie in -T ""), wird die systemweite mailsurr Datei verwendet. Geben Sie ein

```
mail -T TestMDatei Empfänger
```

und eine einfache Meldung (wie zum Beispiel "testing"), gefolgt durch eine Zeile mit entweder nur einem Punkt (".") oder einem Ctrl-D. Das Ergebnis der Benutzung des -T Parameters wird auf die Standard-Ausgabe geschrieben und zeigt die Eingabedaten sowie die resultierenden Transformationen an, die das mail Dienstprogramm mit der mailsurr Datei für den angegebenen Empfänger durchgeführt hat.

Mail-Nachrichten werden nicht gesendet oder zugestellt, wenn der Parameter -T benutzt wird.

DATEIEN

```
/etc/mail/mailsur
/usr/lib/mail/surrcmd/* Erweiterungs-Befehle
/etc/mail/mailcnfg Initialisierungs-Information für mail
```

SIEHE AUCH

ckbinarsys(1M), mailcnfg(4)
 mail(1), sh(1), uux(1), ed(1), egrep(1) in den *Kommandos*.
 exec(2), exit(2), wait(2), popen(3), regexp(5), getdomainname(3) im *Referenzhandbuch für Programmierer*

mailsurr (4)

mailsurr (4)

HINWEIS

Es ist nicht ratsam, neue Einträge in die systemweite Datei `mailsurr` zu schreiben, ohne mindestens ihre syntaktische Korrektheit durch `'mail -T ...'`, wie oben beschrieben, zu überprüfen.

BEZEICHNUNG

mdevice – Master-Datei für Geräte-Spezifikationen

ÜBERSICHT

mdevice

BESCHREIBUNG

Die Datei mdevice ist im Verzeichnis /etc/conf/cf.d abgelegt. Sie enthält eine einzeilige Beschreibung jedes Gerätetreibers und konfigurierbaren Software-Moduls in dem System, das erzeugt werden soll (zur Beschreibung von Dateisystemtypen siehe mfsys(4)). Jede Zeile in mdevice bezeichnet die Master-Dateikomponente von einem Treibersoftware-Paket (DSP). Dieses kann entweder mit dem Grundsystem geliefert oder später durch idinstall installiert werden.

Jede Zeile enthält mehrere Felder, die durch einen Zwischenraum getrennt sind; sie sind unten beschrieben. Jedes Feld muß mit einem Wert oder einem "-" (Bindestrich) versehen sein.

1. *Gerätename*: Dieses Feld enthält den internen Namen des Geräts oder des Moduls und kann bis zu 8 Zeichen lang sein. Das erste Zeichen des Namens muß ein Buchstabe sein; die restlichen Zeichen können Buchstaben, Ziffern oder Unterstriche sein.
2. *Funktionsliste*: Dieses Feld enthält eine Zeichenkette aus Buchstaben, die jeweils eine verfügbare Treiberfunktion repräsentieren. Ist einer der unten aufgelisteten Buchstaben in der Zeichenkette vorhanden, dann muß der Treiber eine Einsprungsadresse (Funktion) für die dem Buchstaben entsprechende Routine haben. Wenn keine Funktionen aus der folgenden Liste verfügbar sind, muß dies durch einen Bindestrich angezeigt werden.

- o – open Routine
- c – close Routine
- r – read Routine
- w – write Routine
- i – ioctl Routine
- s – startup Routine
- x – exit Routine
- f – fork Routine
- e – exec Routine
- I – init Routine
- h – halt Routine
- p – poll Routine
- E – kenter Routine
- X – kexit Routine

L – chpoll Routine (poll Systemaufruf)

M – zeichenorientierte mmap Routine

S – zeichenorientierte segmap Routine

z – size Routine

Beachten Sie bitte: Handelt es sich um ein blockorientiertes Gerät (siehe *Treibercharakteristika* unten), dann müssen standardmäßig eine *strategy* Routine und eine *print* Routine vorhanden sein.

3. *Treibercharakteristika*: Dieses Feld enthält verschiedene Zeichen, die die Merkmale des Treibers angeben. Wenn keines der unten aufgelisteten Zeichen eingetragen ist, muß das Feld einen Bindestrich enthalten. Zulässige Zeichen für dieses Feld sind:

i – Der Geräte-Treiber ist installierbar.

n – Der Geräte-Treiber ist nicht installierbar.

a – Der Geräte-Treiber ist automatisch installierbar.

o – Das Gerät darf nur einen *sdevice* Eintrag haben.

s – Zum Gerät gibt es kein *device count field*.

r – Dieses Gerät ist in allen Konfigurationen des Systemkerns erforderlich. Diese Option ist ausschließlich für Treiber bestimmt, die mit dem Grundsystem geliefert werden. Geräteknoten (Geräte-dateien in dem Verzeichnis */dev*), die einmal für diese Einheit erzeugt wurden, werden nie wieder entfernt. Siehe *idmknod*.

c – Das Gerät ist ein zeichenorientiertes Gerät.

b – Das Gerät ist ein blockorientiertes Gerät.

[b/c] –

Gibt an, welche Geräteknoten zu welcher Geräteklasse gehören. Siehe *idmknod*.

t – Das Gerät ist ein Terminal (tty).

e – Der Treiber ist ein Exec-Switch.

m – Der Treiber ist ein STREAMS-Modul.

H – Der Geräte-Treiber steuert Hardware. Diese Option unterscheidet Treiber, die Hardware unterstützen, von reinen Software-Treibern (Pseudo-Geräten).

N – Kein Treiber (keine Datei *Driver.o* und *space.c* vorhanden).

O – Der Ein/Ausgabe-Adreßbereich (IOA Bereich) dieses Geräts darf sich mit dem eines anderen überschneiden.

G – Das Gerät benutzt keinen Interrupt, obwohl in dem Eintrag *sdevice* ein Interrupt angegeben ist. Damit wird ein Gerät einer spezifischen Gerätegruppe zugeordnet.

- R – Reset Routine.
 - S – Der Geräte-Treiber ist ein installierbares STREAMS-Modul.
 - D – Der Geräte-Treiber kann seinen DMA-Kanal mit anderen Treibern gemeinsam nutzen.
 - d – Der Treiber implementiert eine Scheduling-Klasse (dispatcher), wie z.B. rt, ts.
 - u – Zusammen mit dem Zeichen "M" stellt dieser Buchstabe sicher, daß der Treiber Geräteklassennummern (major device numbers) erhält, die für block- und zeichenorientierte Geräte mit der gleichen Zahl beginnen.
 - M – Diese Option zeigt an, daß das Gerät mehrere Geräteklassennummern (major device numbers) benötigt.
 - p – (Nur MX500) Der Treiber oder STREAMS-Modul arbeitet völlig parallel. Ohne diese Option versucht das System, den Treiber oder STREAMS-Modul seriell zu betreiben. Damit ist es möglich, auf dem Multiprozessorsystem Pakete zu installieren, die von anderen ABI-kompatiblen Systemen stammen.
 - f – (Nur MX300) Der Treiber verwendet Schnittstellen von System VR4 (SysVR4 (new style) Interfaces).
 - A – Diese Option zeigt an, daß das Gerät autokonfiguriert wird von einem MBAd (MB-I Adapter). Der Treiber muß zusätzliche `probe` und `boot` Routinen unterstützen. Der `sdevice` Eintrag hat ein spezielles Format, das aus 12 Feldern besteht.
 - P – Diese Option zeigt an, daß das Gerät ein autokonfiguriertes Pseudo-Gerät ist, ohne eigene Hardware. Der Treiber muß eine zusätzliche `boot` Routine unterstützen. Der `sdevice` Eintrag hat das übliche Format, das aus 10 Feldern besteht. Die meisten dieser Felder haben aber keine Bedeutung.
 - B – (Nur MX500) Diese Option zeigt an, daß das Gerät vom SCED (SCSI/Ethernet/Diagnostics Board) autokonfiguriert wird. Der Treiber muß zusätzliche `probe` und `boot` Routinen unterstützen. Der `sdevice` Eintrag hat ein spezielles Format, das aus 15 Feldern besteht.
4. *Präfix für Behandlungsroutine:* Die Zeichenkette dieses Feldes wird allen nach außen bekannten SteuerROUTinen vorangestellt, die mit diesem Treiber zusammenhängen. Die Zeichenkette kann bis zu 4 Zeichen lang sein.
 5. *Geräteklassennummer (major device number) für blockorientierte Geräte:* In einer DSP-Master-Datei sollte dieses Feld auf Null gesetzt sein. Einem blockorientierten Gerät wird während der Installation durch `idinstall` ein Wert zugeordnet. Bei Geräten mit mehreren Geräteklassennummern wird dieses Feld benutzt, um den "Bereich" von Geräteklassennummern zu spezifizieren. Der Bereich 3-6 wird z.B. als vier Geräteklassennummern von 3 bis 6 einschließlich interpretiert.

6. *Geräteklassennummer (major device number) für zeichenorientierte Geräte:* In einer DSP-Master-Datei sollte dieses Feld auf Null gesetzt sein. Ist das Gerät ein zeichenorientiertes Gerät oder ein STREAMS-Modul, wird während der Installation durch `idinstall` ein Wert zugeordnet. Bei Geräten mit mehreren Geräteklassennummern wird dieses Feld benutzt, um den "Bereich" von Geräteklassennummern zu spezifizieren. Der Bereich 3-6 wird z.B. als vier Geräteklassennummern von 3 bis 6 einschließlich interpretiert.
7. *Minimale Geräteanzahl:* In diesem Feld steht eine Zahl, die die minimale Anzahl von Geräten spezifiziert, die in der Datei `sdevice` angegeben sein muß.
8. *Maximale Geräteanzahl:* Dieses Feld enthält einen ganzzahligen Wert, der die maximale Anzahl von Einheiten spezifiziert, die in der Datei `sdevice` angegeben sein darf.
9. *DMA Kanal:* Dieses Feld enthält einen ganzzahligen Wert, der den DMA-Kanal spezifiziert, der von dieser Einheit benutzt werden kann. Wenn das Gerät DMA nicht verwendet, muß dieses Feld mit '-1' belegt werden. Hier ist zu beachten, daß ein DMA-Kanal von mehr als einer Einheit gemeinsam genutzt werden kann (bisher nicht gestattet). Auf der MX500 sollte dieses Feld '-1' enthalten. Auf der MX300 wird es nicht benutzt (vgl. `sdevice(4): map-` Eintrag).

Spezifikation von STREAMS-Einheiten und -Modulen

STREAMS-Module und -Treiber werden in allen UNIX Systemen anders behandelt als sonstige andere Treiber. Ihre Konfiguration spiegelt diesen Unterschied wieder. Um einen STREAMS-Gerätetreiber zu spezifizieren, sollte in der Datei `mdevice` im Feld *Treibercharakteristika* sowohl ein 'S' als auch ein 'c' eingetragen sein (siehe *Treibercharakteristika*). Damit ist klar, daß es sich um einen STREAMS-Treiber handelt und ein Eintrag in der SINIX Betriebssystemkern-Tabelle `cdevsw` erforderlich ist. In dieser Tabelle werden normalerweise die STREAMS-Treiber für das System konfiguriert.

Ist ein STREAMS-Modul kein Gerätetreiber, sondern z.B. ein Übertragungsprozedur-Modul (line discipline module), sollte sein Eintrag in der Datei `mdevice` im Feld *Treibercharakteristika* ein 'S', aber kein 'c' - wie ein Gerätetreiber - enthalten.

mdevice(4)

(MX300, MX500)

mdevice(4)

BEISPIEL

Das folgende Beispiel zeigt `mdevice` Einträge auf der MX500 für einen NCR Host Adapter (ADP32 Controller) und für ein EXABYTE 8mm Video Bandgerät.

*	Func	Character.	prefix	bmajor	cmajor	min	max
adp32	oi	AHicf	adp32	0	0	0	1
exa	irwoc	Picf	exa	0	5	0	3

EBENE

Multiprozessor-Erweiterung

SIEHE AUCH

`mfsys(4)`, `sdevice(4)`.

`idinstall(1M)` im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

mdevice – Master-Datei für Geräte-Spezifikationen

ÜBERSICHT

mdevice

BESCHREIBUNG

Die Datei mdevice ist im Verzeichnis /etc/conf/cf.d abgelegt. Sie enthält eine einzeilige Beschreibung jedes Gerätetreibers und konfigurierbaren Software-Moduls in dem System, das erzeugt werden soll (zur Beschreibung von Dateisystemtypen siehe mfsys(4)). Jede Zeile in mdevice bezeichnet die Master-Dateikomponente von einem Treibersoftware-Paket (DSP). Dieses kann entweder mit dem Grundsystem geliefert oder später durch idinstall installiert werden.

Jede Zeile enthält mehrere Felder, die durch einen Zwischenraum getrennt sind; sie sind unten beschrieben. Jedes Feld muß mit einem Wert oder einem "-" (Bindestrich) versehen sein.

1. *Gerätename*: Dieses Feld enthält den internen Namen des Geräts oder des Moduls und kann bis zu 8 Zeichen lang sein. Das erste Zeichen des Namens muß ein Buchstabe sein; die restlichen Zeichen können Buchstaben, Ziffern oder Unterstriche sein.
2. *Funktionsliste*: Dieses Feld enthält eine Zeichenkette aus Buchstaben, die jeweils eine verfügbare Treiberfunktion repräsentieren. Ist einer der unten aufgelisteten Buchstaben in der Zeichenkette vorhanden, dann muß der Treiber eine Einsprungsadresse (Funktion) für die dem Buchstaben entsprechende Routine haben. Wenn keine Funktionen aus der folgenden Liste verfügbar sind, muß dies durch einen Bindestrich angezeigt werden.

- o – open Routine
- c – close Routine
- r – read Routine
- w – write Routine
- i – ioctl Routine
- s – startup Routine
- x – exit Routine
- f – fork Routine
- e – exec Routine
- I – init Routine
- h – halt Routine
- p – poll Routine
- E – kenter Routine
- X – kexit Routine

Beachten Sie bitte: Handelt es sich um ein blockorientiertes Gerät (siehe *Treibercharakteristika* unten), dann müssen standardmäßig eine *strategy* Routine und eine *print* Routine vorhanden sein.

3. *Treibercharakteristika*: Dieses Feld enthält verschiedene Zeichen, welche die Merkmale des Treibers angeben. Wenn keines der unten aufgelisteten Zeichen eingetragen ist, muß das Feld einen Bindestrich enthalten. Zulässige Zeichen für dieses Feld sind:
 - i – Der Geräte-Treiber ist installierbar.
 - c – Das Gerät ist ein zeichenorientiertes Gerät.
 - b – Das Gerät ist ein blockorientiertes Gerät.
 - [b/c] –
Gibt an, welche Geräteknoten zu welcher Geräteklasse gehören. Siehe *idmknod*.
 - t – Das Gerät ist ein Terminal (tty).
 - o – Das Gerät darf nur einen *sdevice* Eintrag haben.
 - r – Dieses Gerät ist in allen Konfigurationen des Systemkerns erforderlich. Diese Option ist ausschließlich für Treiber bestimmt, die mit dem Grundsystem geliefert werden. Geräteknoten (Geräte-dateien in dem Verzeichnis */dev*), die einmal für diese Einheit erzeugt wurden, werden nie wieder entfernt. Siehe *idmknod*.
 - u – Zusammen mit einem 'M' stellt dieser Buchstabe sicher, daß der Treiber Geräteklassennummern (major device numbers) erhält, die für block- und zeichenorientierte Geräte mit der gleichen Zahl beginnen.
 - S – Der Geräte-Treiber ist ein installierbares STREAMS-Modul.
 - H – Der Geräte-Treiber steuert die Hardware. Diese Option unterscheidet Treiber, die Hardware unterstützen, von reinen Software-Treibern (Pseudo-Geräten).
 - G – Dieses Gerät benutzt keinen Interrupt, obwohl in dem Eintrag *sdevice* ein Interrupt angegeben ist. Damit wird ein Gerät einer spezifischen Gerätegruppe zugeordnet.
 - D – Der Geräte-Treiber kann seinen DMA-Kanal mit anderen Treibern gemeinsam nutzen.
 - M – Diese Option zeigt an, daß das Gerät mehrere Geräteklassennummern benötigt.
 - O – Der Ein/Ausgabe-Adreßbereich (IOA Bereich) dieses Geräts darf sich mit dem eines anderen überschneiden.
4. *Präfix für Behandlungsroutine*: Die Zeichenkette dieses Feldes wird allen nach außen bekannten Steuerroutinen vorangestellt, die mit diesem Treiber zusammenhängen. Die Zeichenkette kann bis zu 4 Zeichen lang sein.

5. *Geräteklassennummer (major device number) für blockorientierte Geräte:* In einer DSP-Master-Datei sollte dieses Feld auf Null gesetzt sein. Einem blockorientierten Gerät wird während der Installation durch `idinstall` ein Wert zugeordnet. Bei Geräten mit mehreren Geräteklassennummern wird dieses Feld benutzt, um den "Bereich" von Geräteklassennummern zu spezifizieren. Der Bereich 3-6 wird z.B. als vier Geräteklassennummern von 3 bis 6 einschließlich interpretiert.
6. *Geräteklassennummer (major device number) für zeichenorientierte Geräte:* In einer DSP-Master-Datei sollte dieses Feld auf Null gesetzt sein. Ist das Gerät ein zeichenorientiertes Gerät oder ein STREAMS-Modul, wird während der Installation durch `idinstall` ein Wert zugeordnet. Bei Geräten mit mehreren Geräteklassennummern wird dieses Feld benutzt, um den "Bereich" von Geräteklassennummern zu spezifizieren. Der Bereich 3-6 wird z.B. als vier Geräteklassennummern von 3 bis 6 einschließlich interpretiert.
7. *Minimale Geräteanzahl:* In diesem Feld steht eine Zahl, die die minimale Anzahl von Geräten spezifiziert, die in der Datei `sdevice` angegeben sein muß.
8. *Maximale Geräteanzahl:* Dieses Feld enthält einen ganzzahligen Wert, der die maximale Anzahl von Einheiten spezifiziert, die in der Datei `sdevice` angegeben sein darf.
9. *DMA Kanal:* Dieses Feld enthält einen ganzzahligen Wert, der den DMA-Kanal spezifiziert, der von dieser Einheit benutzt werden kann. Wenn das Gerät DMA nicht verwendet, muß dieses Feld mit '-1' belegt werden. Hier ist zu beachten, daß ein DMA-Kanal von mehr als einer Einheit gemeinsam genutzt werden kann (bisher nicht gestattet).

Spezifikation von STREAMS-Einheiten und -Modulen

STREAMS-Module und -Treiber werden in allen UNIX Systemen anders behandelt als sonstige andere Treiber. Ihre Konfiguration spiegelt diesen Unterschied wieder. Um einen STREAMS-Gerätetreiber zu spezifizieren, sollte in der Datei `mdevice` im Feld *Treibercharakteristika* sowohl ein 'S' als auch ein 'c' eingetragen sein (siehe *Treibercharakteristika*). Damit ist klar, daß es sich um einen STREAMS-Treiber handelt und ein Eintrag in der SINIX Betriebssystemkern-Tabelle `cdevsw` erforderlich ist. In dieser Tabelle werden normalerweise die STREAMS-Treiber für das System konfiguriert.

Ist ein STREAMS-Modul kein Gerätetreiber, sondern z.B. ein Übertragungsprozedur-Modul (line discipline module), sollte sein Eintrag in der Datei `mdevice` im Feld *Treibercharakteristika* ein 'S', aber kein 'c' - wie ein Gerätetreiber - enthalten.

mdevice (4)

(WX200)

mdevice (4)

EBENE

Multiprozessor-Erweiterung

SIEHE AUCH

mfsys(4), sdevice(4).

idinstall(1M) im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

mfsys – Datei zur Konfiguration von Dateisystemtypen

ÜBERSICHT

mfsys

BESCHREIBUNG

Die Datei *mfsys* informiert über die Konfiguration von Dateisystemtypen, die in den Betriebssystemkern integriert werden sollen, sobald ein neuer Kern generiert wird. Die Datei steht im Verzeichnis */etc/conf/cf.d* und enthält eine einzeilige Beschreibung für jeden Dateisystemtyp. Die Datei *mfsys* entsteht aus Dateien im Verzeichnis */etc/conf/mfsys.d*. Jede Zeile enthält die folgenden durch Leerzeichen getrennten Felder:

1. *name*: Dieses Feld enthält den internen Namen für den Dateisystemtyp (z.B. S51K oder DUFST). Der Name darf höchstens 32 Zeichen lang sein und darf nur Ziffern und Großbuchstaben enthalten.
2. *prefix*: Dieses Feld gibt eine Zeichenkette an, die an *fstypsw*-Behandlungsfunktionen übergeben wird, die für diesen Dateisystemtyp definiert sind (z.B. s5, du). Das Präfix darf höchstens 8 Zeichen lang sein.
3. *flags*: Dieses Feld enthält eine Hexadezimalzahl der Form "0xNN". Sie wird als Eintrag in der Datenstruktur *vfssw* für diesen Dateisystemtyp benutzt. Bei der MX500 (aber nicht bei der MX300 und WX200) gibt die Marke 0x8000 an, daß das Dateisystem als voll parallelisierter Multiprozessor-Code implementiert ist. Falls diese Marke nicht gesetzt ist (Standardeinstellung), wird das Dateisystem als "single-thread" behandelt. Diese Marke darf auf keinen Fall geändert werden!

BEISPIEL

Das folgende Beispiel zeigt, daß einige der Dateisysteme im Hauptspeicher als "single-thread" laufen, während alle Dateisysteme auf Festplatte voll parallelisiert sind:

Name	Prefix	Flags
bfs	bf	0x00
events	ev_	0x00
fdfs	fdfs	0x00
fifofs	fifo	0x8000
namefs	name	0x00
nfs	nfs	0x00
proc	pr	0x8000
s5	s5	0x8000
specfs	spec	0x8000
ufs	ufs	0x8000

SIEHE AUCH

sfsys(4).
idinstall(1M), idbuild(1M) im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

mnttab – Tabelle der eingehängten Dateisysteme

ÜBERSICHT

```
#include <sys/mnttab.h>
```

BESCHREIBUNG

Die Datei /etc/mnttab enthält Informationen über Geräte, die durch den Befehl mount eingehängt wurden. Die Information hat die folgende in <sys/mnttab.h> definierte Struktur:

```
struct    mnttab {
    char   *mnt_special;
    char   *mnt_mountp;
    char   *mnt_fstype;
    char   *mnt_mntopts;
    char   *mnt_time;
};
```

Die Felder in der Mount-Tabelle sind durch Leerzeichen getrennt und zeigen folgende Informationen: das blockorientierte Gerät, den Einhängpunkt, den Dateisystem-Typ des eingehängten Dateisystems, die Optionen und den Zeitpunkt, zu dem das Dateisystem eingehängt wurde.

SIEHE AUCH

mount(1M), getmntent(1M), setmnt(1M)
Abschnitt 4 des *Leitfadens für Systemverwalter*

BEZEICHNUNG

mtune – Einstellbare Systemparameter

ÜBERSICHT

mtune

BESCHREIBUNG

Die Datei `mtune` enthält Informationen über alle einstellbaren Systemparameter. Jeder davon steht in einer eigenen Zeile. Jede Zeile enthält die folgenden Felder, die durch Leerzeichen voneinander getrennt sind:

1. *Parametername*: Eine Zeichenkette, die maximal 20 Zeichen lang ist. Sie wird benutzt, um die prozessorspezifische Anweisung `"#define ..."` zu konstruieren. Diese übergibt den Wert an das System, wenn es neu erzeugt wird.
2. *Voreinstellung*: Dies ist der Standardwert des einstellbaren Parameters. Wenn in der Datei `stune` kein Wert spezifiziert ist, wird dieser Wert verwendet, sobald das System erzeugt wird.
3. *Minimalwert*: Dies ist der minimale zulässige Wert für den einstellbaren Parameter. Wenn der Parameter in der Datei `stune` angegeben ist, überprüfen die Konfigurationswerkzeuge, ob der neue Wert gleich oder größer ist als dieser Wert.
4. *Maximalwert*: Dies ist der maximale zulässige Wert für den einstellbaren Parameter. Wenn der Parameter in der Datei `stune` angegeben ist, kontrollieren die Konfigurationswerkzeuge, ob der neue Wert gleich oder kleiner ist als dieser Wert.

Die Datei `mtune` steht normalerweise in `/etc/conf/cf.d`. Jedoch sollte ein Benutzer oder ein zusätzliches Paket niemals die Datei `mtune` direkt bearbeiten, um die Einstellung eines Systemparameters zu ändern oder die Datei zu ergänzen. Stattdessen ist hierfür der Befehl `idtune` zu verwenden.

Damit die neuen Werte wirksam werden, muß der Systemkern neu erzeugt und dann das System neu gestartet werden.

SIEHE AUCH

`stune(4)`.

`idbuild(1M)`, `idtune(1M)`, `iddefaults(1M)`

BEZEICHNUNG

netconfig – Datenbank für Netzwerk-Konfiguration

ÜBERSICHT

```
#include <netconfig.h>
```

BESCHREIBUNG

Die Datenbank für Netzwerk-Konfiguration `/etc/netconfig` ist eine Systemdatei. Hier werden Informationen über Netzwerke gespeichert, die mit dem System verbunden sind und benutzt werden können. Die Datenbank `netconfig` und die Routinen, die darauf zugreifen (siehe `getnetconfig(3N)`), sind Teil der Netzwerk-Auswahlkomponente. Die Netzwerk-Auswahlkomponente enthält auch die Umgebungsvariable `NETPATH` und eine Anzahl von Routinen. Diese Routinen greifen auf die Datenbank `netconfig` zu, indem sie `NETPATH`-Komponenten als Verweise zu Einträgen in `netconfig` verwenden. `NETPATH` ist in `sh(1)` beschrieben; die `NETPATH` Zugriffsroutinen werden in `getnetpath(3N)` behandelt.

`netconfig` enthält einen Eintrag für jedes Netzwerk, das auf dem System verfügbar ist. Einträge werden durch Neue-Zeile-Zeichen getrennt. Felder werden durch Leerzeichen getrennt und treten in der unten beschriebenen Reihenfolge auf. Leerzeichen innerhalb eines Feldes können als `"\blank"` oder `"\tab"` dargestellt werden. Gegenschrägstriche müssen als `"\"` dargestellt werden. Jedes Feld entspricht einem Element in der Struktur `struct netconfig`. `struct netconfig` und die auf dieser Handbuchseite beschriebenen Bezeichner sind in `/usr/include/netconfig.h` definiert.

Netzwerkennung

Diese Zeichenkette wird verwendet, um ein Netzwerk eindeutig zu identifizieren. *Netzwerkennung* besteht aus einzelnen Zeichen (außer Null). Sie hat eine Länge von mindestens 1. Es ist keine maximale Länge vorgegeben. Die Vergabe des Namens ist Aufgabe des Systemverwalters. Alle *Netzwerkennungen* auf einem System müssen eindeutig sein.

Semantik

Diese Zeichenkette stellt die Semantik (Liste von Diensten) des Netzwerks durch die Angabe der Service-Schnittstelle dar. Das Feld *Semantik* ist obligatorisch. Folgende Semantiken werden erkannt:

<code>tpi_clts</code>	Transportschnittstelle, verbindungslos
<code>tpi_cots</code>	Transportschnittstelle, verbindungsorientiert
<code>tpi_cots_ord</code>	Transportschnittstelle, verbindungsorientiert, unterstützt ordnungsgemäße Freigabe.

Anzeige

In diesem Feld werden bestimmte zweiwertige ("wahr" und "falsch") Netzwerk-Attribute eingetragen. Die Zeichenkette *Anzeige* besteht aus einer Kombination von Zeichen, von denen jedes den Wert des entsprechenden Attributs anzeigt. Ist das Zeichen vorhanden, ist das Attribut "wahr" wenn nicht, ist das Attribut "falsch". Ein "-" zeigt an, daß keines der Attribute vorhanden ist. Gegenwärtig wird nur ein Zeichen erkannt:

v Visible network (Standard). Es wird benutzt, wenn die Umgebungsvariable `NETPATH` nicht gesetzt ist.

Protokollfamilie

Die Felder *Protokollfamilie* und *Protokollname* sind für protokollspezifische Anwendungen vorgesehen.

Das Feld *Protokollfamilie* enthält eine Zeichenkette, die eine Protokoll-Familie bezeichnet. Der Bezeichner von *Protokollfamilie* folgt denselben Regeln wie der für *Netzwerkennung*. Das heißt, die Zeichenkette darf keine Null enthalten, hat eine Länge von mindestens 1 und es ist keine maximale Länge angegeben. Ein "-" in diesem Feld zeigt an, daß sich kein Bezeichner einer Protokollfamilie darauf bezieht, d.h. das Netzwerk ist experimentell. Es folgen einige Beispiele:

loopback	Rücksprung (vom lokalen Rechner zum Host).
inet	Netzverbund: UDP, TCP, etc.
implink	ARPANET imp Adressen
pup	PUP Protokolle: z.B. BSP
chaos	MIT CHAOS Protokolle
ns	XEROX NS Protokolle
nbs	NBS Protokolle
ecma	European Computer Manufacturers Association
datakit	DATAKIT Protokolle
ccitt	CCITT Protokolle, X.25, etc.
sna	IBM SNA
decnet	DECNET
dli	Direktes Datenleitung Interface
lat	LAT
hylink	NSC Hyperkanal
appletalk	Apple Talk
nit	Netzwerkschnittstelle Tap
ieee802	IEEE 802.2; auch ISO 8802
osi	Gemeinsame Bezeichnung für alle Familien, die von OSI benutzt werden (z.B. protosw lookup)
x25	Nur für CCITT X.25
osinet	AFI = 47, IDI = 4
gossip	U.S. Government OSI

Protokollname

Diese Zeichenkette gibt den Namen des benutzten Protokolls an. Der Bezeichner für *Protokollname* folgt denselben Regeln wie denen für *Netzwerkennung* (siehe oben). Die nachfolgenden Protokoll-Namen werden erkannt. Ein “-” zeigt an, daß keiner der aufgelisteten Namen verwendet wird.

tcp	Transmission Control Protocol (TCP)
udp	User Datagram Protocol (UDP)
icmp	Internet Control Message Protocol (ICMP)

Netzwerkgerät

Dies ist der volle Pfadname des verwendeten Geräts, das eine Verbindung zum Transport-Provider zur Verfügung stellt. Typischerweise steht dieses Gerät in dem Verzeichnis /dev. Die Angabe von *Netzwerkgerät* ist obligatorisch.

Suchbibliotheken für Verzeichnisse

Die *Suchbibliotheken für Verzeichnisse* unterstützen einen “Verzeichnis Service” (Name/Adreß-Abbildung) für das Netzwerk. Dieser Dienst wird durch die Name/Adreß-Abbildungs-Funktion des Systems vorgenommen. Wenn ein Netzwerk nicht mit einer solchen Bibliothek ausgestattet ist, wird die *netdir*-Funktion nicht zur Verfügung stehen. Ein “-” in diesem Feld zeigt an, daß keine Suchbibliotheken vorhanden sind. In diesem Fall kann die Name/Adreß-Abbildung für das Netzwerk nicht verwendet werden. Das Feld für die Suchbibliotheken besteht aus einer durch Komma getrennten Liste von vollen Pfadnamen der Bibliotheken, auf die dann dynamisch verwiesen wird. In den Pfadnamen enthaltene Kommas können als “\,” und Gegenschrägstriche als “\\” dargestellt werden.

Zeilen in /etc/netconfig, die mit einem Nummernzeichen (#) in Feld 1 beginnen, werden als Kommentar behandelt.

Die Struktur struct netconfig enthält folgende Elemente. Sie entsprechen den Feldern in den netconfig Datenbank-Einträgen:

char * nc_netid	Netzwerkennung	mit
	abschließendem Nullzeichen	
unsigned long nc_semantics	Semantik	
unsigned long nc_flag	Schalter	
char * nc_protobufmly	Protokollfamilie	
char * nc_proto	Protokollname	
char * nc_device	Voller Pfadname des Netzwerkgerätes	

netconfig(4)

netconfig(4)

unsigned long nc_nlookups	Anzahl der Verzeichnis-Suchbibliotheken
char ** nc_lookups	Volle Pfadnamen der Verzeichnis-Suchbibliotheken
unsigned long nc_unused[9]	Reserviert für zukünftige Erweiterungen (nicht auf Benutzerebene zu verwenden)

Das Feld `nc_semantics` enthält die folgenden Werte, entsprechend den oben beschriebenen Semantiken:

NC_TPI_CLTS
NC_TPI_COTS
NC_TPI_COTS_ORD

Das Feld `nc_flag` ist ein Bit-Feld. Gegenwärtig wird das unten angegebene Bit erkannt. Es entspricht dem oben beschriebenen Attribut. `NC_NOFLAG` zeigt an, daß keinerlei Attribute existieren.

NC_VISIBLE

DATEIEN

/etc/netconfig
/usr/include/netconfig.h

SIEHE AUCH

`netdir_getbyname(3N)`, `getnetconfig(3N)`, `getnetpath(3N)`, `netconfig(4)`
in *CES, Leitfaden und Werkzeuge für die Programmierung mit C, Netzwerk-Schnittstellen*
Leitfaden für Systemverwalter

BEZEICHNUNG

netmasks – Netzmasken-Datenbank

ÜBERSICHT

/etc/netmasks

BESCHREIBUNG

Die in der Datei `netmasks` enthaltenen Netzmasken werden zur Implementierung von IP Standard-Teilnetzen verwendet. Für jedes Netzwerk, das teilvernetzt ist, muß eine eigene Zeile in dieser Datei existieren. Jede Zeile enthält die Netzwerknummer, eine beliebige Anzahl von Leerzeichen oder Tabulatoren und die Netzmaske, die in diesem Netzwerk zu benutzen ist. Netzwerknummern und Masken können in der gewohnten Punktschreibweise angegeben werden (ähnlich wie IP-Host Adressen, wobei im Host-Anteil Nullen stehen). Zum Beispiel legt

```
128.32.0.0 255.255.255.0
```

fest, daß das Klasse B Netzwerk 128.32.0.0 je acht Bits im Teilnetz- und im Host-Feld haben sollte, zusätzlich zu den standardmäßigen sechzehn Bits im Netzwerk-Feld.

DATEIEN

/etc/netmasks

SIEHE AUCH`ifconfig(1M)`

Postel, Jon, und Mogul, Jeff, *Internet Standard Subnetting Procedure*, RFC 950, Network Information Center, SRI International, Menlo Park, Calif., August 1985.

BEZEICHNUNG

netrc – Datei für ftp Fern-Login Daten

ÜBERSICHT

\$HOME/.netrc

BESCHREIBUNG

Die Datei `.netrc` enthält Daten für die Anmeldung bei einem fernen Host über das Netzwerk zur Dateiübertragung durch `ftp(1)`. Diese Datei befindet sich in dem Home-Verzeichnis des Benutzers auf der Maschine, die die Dateiübertragung beginnt. Seine Zugriffsrechte sollten so eingestellt sein, daß Lese-Zugriff durch seine Gruppe und andere (others) nicht erlaubt ist (siehe `chmod(1)`).

Folgende Tokens, voneinander durch Leerzeichen, Tabulatoren oder Neue-Zeile-Zeichen getrennt, werden erkannt:

`machine name`

Bezeichnet den Namen eines fernen Systems. Nach dem Aufruf eines `ftp`- oder `open`-Befehls durchsucht die Auto-Login-Prozedur die Datei `.netrc` nach einem Eintrag für `machine`, der mit dem verlangten fernen System übereinstimmt. Sobald einer angetroffen wurde, werden die nachfolgenden `.netrc` Token eingelesen, bis das Dateiende erreicht ist oder ein anderes `machine`-Schlüsselwort angetroffen wird.

`login name`

Bezeichnet einen Benutzer auf dem fernen System. Ist dieses Schlüsselwort vorhanden, löst die Auto-Login-Prozedur eine Anmeldung mit dem angegebenen Namen aus.

`password zeichenkette`

Liefert ein Paßwort. Ist dieses Schlüsselwort vorhanden, stellt die Auto-Login-Prozedur die angegebene Zeichenkette zur Verfügung, wenn der ferne Server ein Paßwort als Teil des Anmeldeprozesses erwartet. Anmerkung: Steht dieses Schlüsselwort in der Datei `.netrc`, bricht `ftp` den Auto-Login-Vorgang ab, wenn `.netrc` durch irgendjemand anderen als den Benutzer gelesen werden kann.

`account zeichenkette`

Liefert ein zusätzliches Paßwort für Abrechnungsinformation. Wenn der ferne Server ein zusätzliches Paßwort verlangt, liefert die Auto-Login-Prozedur die angegebene Zeichenkette. Andernfalls löst sie einen `ACCT`-Befehl aus.

`macrodef name`

Definiert ein Makro. Dieses Schlüsselwort hat dieselbe Wirkung wie der Befehl `ftp macrodef`. Ein Makro ist durch die Angabe des Namens definiert. Sein Inhalt beginnt in der nächsten `.netrc`-Zeile und setzt sich fort, bis eine sogenannte Nullzeile (aufeinanderfolgende Neue-Zeile-Zeichen) angetroffen wird. Ist ein Makro mit dem Namen `init` definiert, wird es automatisch als letzter Schritt des Auto-Login-Vorgangs ausgeführt.

BEISPIEL

Eine `.netrc`-Datei, die folgende Zeile enthält,

```
machine ray login demo password mypassword
```

erlaubt ein Auto-Login zur Maschine `ray` und verwendet den Benutzernamen `demo` mit dem Paßwort `mypassword`.

DATEIEN

`~/.netrc`

SIEHE AUCH

`chmod(1)`, `ftp(1)`, `ftpd(1M)`.

BEZEICHNUNG

networks – Datenbank für Netzwerk-Namen

ÜBERSICHT

/etc/networks

BESCHREIBUNG

Die Datei `networks` enthält Informationen bezüglich der bekannten Netzwerke, aus denen das DARPA Internet besteht. Für jedes Netzwerk muß eine einzelne Zeile mit folgender Information vorhanden sein:

offizieller_Netzwerkname Netzwerknummer Aliase

Die einzelnen Teile werden durch eine beliebige Anzahl von Leerzeichen und/oder Tabulatoren getrennt. Ein '#' am Anfang der Zeile leitet einen Kommentar ein. Zeichen bis zum Ende dieser Zeile werden nicht von Routinen interpretiert, die die Datei abarbeiten. Diese Datei wird normalerweise von der offiziellen Netzwerk-Datenbank verwaltet. Sie wird vom Network Information Control Center (NIC) gepflegt. Trotzdem können lokale Änderungen erforderlich sein, um sie auf den neuesten Stand zu bringen. Dies betrifft nicht-offizielle Aliase und/oder unbekannte Netzwerke.

Die Netzwerknummer kann in der gewohnten Punktschreibweise angegeben werden unter Verwendung der `inet_network` Bibliotheksroutine aus der Internet Address Manipulation Library, `inet(7)`. Netzwerknamen dürfen außer Feld-Trennzeichen, Neue-Zeile-Zeichen oder Kommentarzeichen jedes beliebige abdruckbare Zeichen enthalten.

DATEIEN

/etc/networks

SIEHE AUCH

`getnetent(3N)`, `inet(7)`.

HINWEIS

Anstelle einer statischen Datei sollte ein Name-Server verwendet werden. Für schnellen Zugriff sollte ein binär indiziertes Dateiformat verfügbar sein.

BEZEICHNUNG

`.ott` – FACE Objektarchitektur-Information

ÜBERSICHT

`.ott`

BESCHREIBUNG

Die FACE Objektarchitektur speichert Informationen über Objekt-Typen in der ASCII-Datei `.ott` (Objekt-Typ-Tabelle). Sie ist in jedem Verzeichnis enthalten. Diese Datei beschreibt alle Objekte in diesem Verzeichnis. Jede Zeile der Datei `.ott` enthält Informationen über ein Objekt in Feldern, die durch ein Pipe-Zeichen voneinander getrennt sind. Die Felder sind in der nachstehend aufgeführten Reihenfolge angegeben:

<i>name</i>	Der Name der aktuellen Systemdatei.
<i>anzeige_name</i>	Das ist der Name, der dem Benutzer angezeigt werden soll. Hier steht ein Punkt, wenn es derselbe ist wie der Name der Datei.
<i>beschreibung</i>	Die Beschreibung des Objekts, oder ein Punkt, wenn die Beschreibung Standard ist (dieselbe wie Objekt-Typ).
<i>objektyp</i>	Der FACE-interne Objekttyp-Name.
<i>schalter</i>	Objektspezifische Schalter.
<i>änd_zeit</i>	Der Zeitpunkt, zu dem FACE das Objekt zuletzt verändert hat. Die Zeit wird hexadezimal als die Anzahl Sekunden seit 1/1/1970 angegeben.
<i>obj_spezif_daten</i>	Ein optionales Feld. Es enthält eine Anzahl Felder der Form <i>name=wert</i> , die durch Strichpunkte voneinander getrennt sind. In diesen Feldern kann FACE jede beliebige zusätzliche Information speichern, die zur Beschreibung dieses Objekt notwendig ist.

DATEIEN

`.ott` wird in jedem Verzeichnis erstellt, das durch FACE geöffnet wird.

BEZEICHNUNG

passwd – Paßwortdatei

ÜBERSICHT

/etc/passwd

BESCHREIBUNG

Die Datei /etc/passwd enthält zu jedem Benutzer die folgenden Informationen:

- Login-Name
- Dummy-Paßwort
- Benutzernummer
- Gruppennummer
- Kommentar
- aktuelles Arbeitsverzeichnis nach dem Login
- Programm, das als Shell benutzt wird

passwd ist eine ASCII-Datei. Die einzelnen Felder in einem Benutzereintrag sind durch Doppelpunkte getrennt. Das Kommentarfeld kann jede beliebige Information enthalten. Jeder Benutzereintrag wird vom nächsten durch ein Neue-Zeile-Zeichen getrennt. Wenn im Feld für die Shell kein Eintrag vorhanden ist, wird /usr/bin/sh verwendet.

Diese Datei enthält Benutzer-Login-Informationen und hat allgemeine Leseberechtigung. Sie kann deshalb beispielsweise dazu benutzt werden, den Login-Namen zu einer Benutzernummer (UID) herauszufinden.

Das Feld für das Paßwort besteht aus dem Zeichen x. Dieses Feld bleibt nur aus Kompatibilitätsgründen bestehen. Die Paßwort-Information ist in der Datei /etc/shadow enthalten; siehe shadow(4).

DATEIEN

/etc/passwd
/etc/shadow

SIEHE AUCH

pwconv(1M), useradd(1M), usermod(1M), userdel(1M), shadow(4), group(4),
unistd(4)
a64l(3C), getpwent(3C), putpwent(3C) im *Referenzhandbuch für Programmierer*
login(1), passwd(1) in den *Kommandos*.

BEZEICHNUNG

pathalias – Aliasdatei für FACE

ÜBERSICHT

`$VMSYS/pathalias $HOME/pref/pathalias`

BESCHREIBUNG

Die pathalias-Dateien enthalten Zeilen in der Form `alias=path`, wobei *path* aus einem oder mehreren, durch Doppelpunkt getrennten, Verzeichnissen bestehen kann. Immer wenn ein FACE-Benutzer sich auf einen Pfad bezieht, der nicht mit einem `"/` beginnt, wird diese Datei kontrolliert. Wenn die erste Komponente des Pfadnamens mit der linken Seite des Gleichheitszeichens übereinstimmt, wird die rechte Seite durchsucht, ähnlich der `$PATH`-Variablen des Betriebssystems. Dies erlaubt den Benutzern, auf den Ordner `$HOME/FILECABINET` durch Eingabe von `filecabinet` Bezug zu nehmen.

Es gibt eine systemweite pathalias Datei mit dem Namen `$VMSYS/pathalias`. Außerdem kann jeder Benutzer auch eine lokale Alias-Datei besitzen, die als `$HOME/pref/pathalias` bezeichnet wird. Einstellungen in der Benutzer-Alias-Datei setzen Einstellungen in der systemweiten Datei außer Kraft. Die systemweite Datei wird mit mehreren Standard-FACE-Aliasen geliefert, wie z.B. `filecabinet`, `wastebasket`, `preferences`, `other_users`, usw.

HINWEIS

Im Gegensatz zu Befehls-Schlüsselwörtern ist der teilweise Vergleich eines Pfad-Alias nicht erlaubt. Pfad-Aliase unterscheiden jedoch nicht zwischen Groß- und Kleinschreibung. Der Name eines Alias muß alphabetisch sein und darf auf keinen Fall Sonderzeichen wie `"/`, `"\"` oder `"=` enthalten. Es gibt keine festgelegte Grenze für die Anzahl der erlaubten Aliase. Alias-Dateien werden nur einmal bei der Anmeldung eingelesen und werden im Hauptspeicher bis zur Abmeldung gehalten. Deshalb wird eine Veränderung einer Alias-Datei während einer Sitzung erst bei der nächsten Sitzung wirksam.

DATEIEN

`$HOME/pref/pathalias`
`$VMSYS/pathalias`

pnch (4)

pnch (4)

BEZEICHNUNG

pnch – Dateiformat für die Abbildung einer Lochkarte

BESCHREIBUNG

Das Format PNCH ist eine bequeme Darstellung für Dateien, die aus Lochkarten-Abbildungen in einem beliebigen Code bestehen.

Eine PNCH-Datei ist eine einfache Aneinanderreihung von Kartensätzen. Ein Karten-Eintrag besteht aus einem einzelnen Steuerbyte, gefolgt von einer variablen Anzahl von Datenbytes. Das Steuerbyte gibt die Zahl der darauffolgenden Datenbytes an. Sie muß innerhalb des Bereichs 0–80 liegen. Die Datenbytes sind 8-Bit Codes, die das Karten-Abbild bilden. Wenn weniger als 80 Datenbytes vorhanden sind, heißt das, daß der Rest des Karten-Bildes aus nachfolgenden Leerzeichen besteht.

BEZEICHNUNG

/proc – Prozeß-Dateisystem

ÜBERSICHT

/proc

BESCHREIBUNG

/proc ist ein Dateisystem, das Zugriff auf das Abbild jedes aktiven Prozesses im System ermöglicht. Der Name jedes Eintrags in dem Verzeichnis /proc ist eine Dezimalzahl, die der jeweiligen Prozeßnummer (PID) entspricht. Der Eigentümer jeder "Datei" wird durch die Benutzernummer UID des Prozesses festgelegt.

Der Zugriff auf /proc Dateien erfolgt über die Standard-Schnittstellen der folgenden Systemaufrufe: open, close, read, write und ioctl. Zur Prozeßsteuerung muß die /proc Datei mit Lese- und Schreibberechtigung geöffnet werden; ist nur die Leseberechtigung gegeben, kann eine Überwachung des Prozesses stattfinden. Wie normale Dateien können auch /proc Dateien von mehreren Prozessen gleichzeitig geöffnet werden. Zur kollisionsfreien Prozeßsteuerung gibt es den Exklusiv-Modus (O_EXCL) für den Open-Aufruf. Ein solcher Aufruf zum Schreiben mißlingt, wenn die Datei bereits geöffnet ist. Ist eine Datei im Exklusiv-Modus zum Schreiben geöffnet, mißlingen nachfolgende Versuche, die Datei zum Schreiben zu öffnen. Dabei ist es gleich, ob diese Versuche den Exklusiv-Modus beanspruchen oder nicht. (Ausnahme: Der Systemverwalter kann sich darüber hinwegsetzen; er kann eine Datei sogar ohne Beanspruchung des Exklusiv-Modus öffnen, selbst wenn diese bereits im Exklusiv-Modus geöffnet ist.) Wird nur die Leseberechtigung gefordert, kann eine Datei beliebig oft geöffnet werden, selbst wenn sie im Exklusiv-Modus zum Schreiben bereits geöffnet ist.

Daten können von oder zu beliebigen Stellen im Adreßbereich des überwachten Prozesses übertragen werden. In der Datei wird durch Anwendung von lseek auf die gewünschte virtuelle Adresse positioniert. Danach folgt ein read oder write Aufruf. Mit der PIOCMAP Operation können die zugänglichen "Mappings" (Bereiche) des Adreßraums bestimmt werden. Einem zusammenhängenden Bereich des Adreßraums können mehrere Mappings mit verschiedenen Lese-/Schreib-/Ausführungs-Berechtigungen zugeordnet sein. E/A-Übertragungen können sich über aufeinanderfolgende Mappings hinweg erstrecken. Eine E/A-Anforderung, die in einen nicht gemappten Bereich hineinreicht, wird an der Übergangsstelle abgeschnitten.

Informations- und Steueroperationen werden durch ioctl zur Verfügung gestellt. Diese haben die Form:

```
#include <sys/types.h>
#include <sys/signal.h>
#include <sys/fault.h>
#include <sys/syscall.h>
#include <sys/procfs.h>
void *p;
retval = ioctl(fildes, code, p);
```

Das Argument *p* ist ein generischer Zeiger, dessen Typ von dem spezifischen ioctl-Code abhängt. Wenn es weiter unten nicht ausdrücklich anders angegeben ist, ist es stets der Nullzeiger. <sys/procfs.h> enthält die Definitionen von ioctl-Codes und die Datenstrukturen, die von diesen Operationen benutzt

werden. Bestimmte Operationen können nur dann ausgeführt werden, wenn die Prozeßdatei zum Schreiben geöffnet ist; dies schließt alle Operationen ein, die die Prozeßsteuerung beeinflussen.

Die Operationen zur Prozeßsteuerung und Gewinnung von Prozeßdaten schließen die Verwendung von Schaltermengen ein. Die Mengentypen `sigset_t`, `fltset_t` und `sysset_t` beziehen sich auf die Aufzählungstypen von Signal-, Fehler- und Systemaufrufen, wie sie in `<sys/signal.h>`, `<sys/fault.h>` und `<sys/syscall.h>` definiert sind. Jeder Mengentyp ist groß genug, um die Schalter für seine zugehörigen Aufzählungstypen aufzunehmen. Trotz verschiedener Größe haben sie eine gemeinsame Struktur und können mit den folgenden Makros behandelt werden:

```
prfillset(&set);      /* Setzen aller Schalter in set */
premptyset(&set);     /* Rücksetzen aller Schalter in set */
praddset(&set, flag); /* Setzen des angegebenen Schalters */
prdelset(&set, flag); /* Rücksetzen des angegebenen Schalters */
r = prismember(&set, flag); /* != 0, falls flag gesetzt ist */
```

Bevor `set` in irgendeiner Operation verwendet werden kann, muß es zuvor durch eines der beiden Makros `prfillset` oder `premptyset` initialisiert werden. `flag` muß ein Element der Aufzählung sein, auf die sich `set` bezieht.

Es folgen die zulässigen `ioctl`-Codes. Die Aufrufe, die Schreibzugriff erfordern, sind mit einem Stern (*) markiert. Wenn nicht anders angegeben, liefert ein `ioctl`-Aufruf auf einen bereits beendeten Prozeß den Fehler `ENOENT`.

PIOCSTATUS

Es wird Statusinformation für den Prozeß zurückgegeben; `p` ist ein Zeiger auf eine `prstatus` Struktur:

```
typedef struct prstatus {
    longpr_flags; /* Prozeßstatus */
    short pr_why; /* Ursache für Prozeß-Stop (wenn angehalten) */
    short pr_what; /* detailliertere Ursache */
    struct siginfo pr_info; /* Information im Zusammenhang mit Signal oder Fehler */
    short pr_cursig; /* aktuelles Signal */
    sigset_t pr_sigpend; /* Satz anderer wartender Signale */
    sigset_t pr_sighold; /* Satz gehaltener Signale */
    struct sigaltstack pr_altstack; /* Information über zusätzlichen Signal-Stapel */
    struct sigaction pr_action; /* Signalaktion für aktuelles Signal */
    pid_t pr_pid; /* Prozeßnummer */
    pid_t pr_ppid; /* Vaterprozeßnummer */
    pid_t pr_pgrp; /* Prozeßgruppennummer */
    pid_t pr_sid; /* Sitzungs-ID */
    timestruc_t pr_utime; /* verbrauchte CPU-Zeit durch den Benutzer */
    timestruc_t pr_stime; /* verbrauchte CPU-Zeit durch das System */
    timestruc_t pr_outime; /* Summe der Benutzer-Zeiten aller Sohn-Prozesse */
    timestruc_t pr_cstime; /* Summe der Systemzeiten aller Sohn-Prozesse */
    char pr_cname[8]; /* Name der Zuteilungsklasse */
    long pr_filler[20]; /* freier Bereich für zukünftige Erweiterungen */
    long pr_instr; /* gegenwärtiger Maschinenbefehl */
    gregset_t pr_reg; /* allgemeine Register */
} prstatus_t;
```

`pr_flags` ist eine Bitmaske, die folgende Schalter enthält:

<code>PR_STOPPED</code>	Prozeß wurde angehalten
<code>PR_ISTOP</code>	Prozeß wurde angehalten bei einem erwarteten Ereignis (siehe <code>PIOCSTOP</code>)
<code>PR_DSTOP</code>	Eine stop Anweisung auf diesen Prozeß ist aktiv (siehe <code>PIOCSTOP</code>)
<code>PR_ASLEEP</code>	Prozeß befindet sich in einem unterbrechbaren Schlaf innerhalb eines Systemaufrufs
<code>PR_FORK</code>	Prozeß hat seinen <code>inherit-on-fork</code> Schalter gesetzt (siehe <code>PIOCSFORK</code>)
<code>PR_RLC</code>	Prozeß hat seinen <code>run-on-last-close</code> Schalter gesetzt (siehe <code>PIOCSRLC</code>)
<code>PR_PTRACE</code>	Der Prozeßablauf wird verfolgt durch <code>ptrace</code>
<code>PR_PCINVAL</code>	Prozeßbefehlszähler verweist auf eine ungültige Adresse
<code>PR_ISSYS</code>	Prozeß ist ein Systemprozeß (siehe <code>PIOCSTOP</code>)

Die beiden Werte `pr_why` und `pr_what` zusammen beschreiben für einen angehaltenen Prozeß den Grund des Anhaltens. Mögliche Werte für `pr_why` sind:

`PR_REQUESTED` zeigt an, daß der Prozeß anhielt, weil `PIOCSTOP` angewandt wurde; `pr_what` ist in diesem Fall unbenutzt.

`PR_SIGNALLED` zeigt an, daß der Prozeß beim Empfang eines Signals anhielt (siehe `PIOCTRACE`); `pr_what` enthält die Signal-Nummer, die den Halt verursachte (für einen von neuem angehaltenen Prozeß befindet sich derselbe Wert in `pr_cursig`).

`PR_FAULTED` zeigt an, daß der Prozeß beim Antreffen eines Hardware-Fehlers anhielt (siehe `PIOCSFAULT`); `pr_what` enthält die Fehlernummer, die den Halt verursachte.

`PR_SYSENTRY` und `PR_SYSEXIT` zeigen einen Halt auf Grund eines Eintrags in oder Ausgang von einem Systemaufruf (siehe `PIOCSENTRY` und `PIOCSEXIT`); `pr_what` enthält die Systemaufruf-Nummer.

`PR_JOBCONTROL` zeigt an, daß der Prozeß-Stop auf eine Standardaktion eines Auftragssteuerungs-Haltesignals zurückzuführen ist (siehe `sigaction`); `pr_what` enthält die Stoppsignal-Nummer.

`pr_info`, enthält zusätzliche Informationen über ein Signal oder einen Fehler, wenn sich der Prozeß in einem `PR_SIGNALLED`- oder `PR_FAULTED`-Halt befindet (siehe `<sys/signinfo.h>`).

`pr_cursig` bezeichnet das aktuelle Signal, das heißt, das nächste Signal, das an den Prozeß übergeben wird. In `pr_sigpend` stehen die übrigen wartenden Signale. In `pr_sighold` stehen jene Signale, deren Zustellung sich verzögert, wenn sie an den Prozeß gesendet werden.

`pr_altstack` enthält die Information für den zusätzlichen Signal-Stapel des Prozesses (siehe `sigaltstack`). `pr_action` enthält die Signal-Aktions-Information bezüglich des aktuellen Signals (siehe `sigaction`); es ist undefiniert, wenn `pr_cursig` Null ist.

`pr_pid`, `pr_ppid`, `pr_pgrp` und `pr_sid` bezeichnen die Prozeßnummer, die Kennung des Vaterprozesses, die Prozeßgruppennummer und die Sitzungskennung des Prozesses.

`pr_utime`, `pr_stime`, `pr_cutime`, und `pr_cstime` bezeichnen die Benutzer- und System-Rechnerzeit, die der Prozeß verbraucht und die aufaddierte Benutzer- und System-Rechnerzeit aller Sohn-Prozesse. Die Angabe erfolgt in Sekunden und Nanosekunden.

`pr_clname` enthält den Namen der Prozeß-Klasse.

Der `pr_filler` Bereich ist reserviert für zukünftige Anwendungen.

`pr_instr` enthält die Maschinenanweisung, auf die sich der Programmzähler bezieht. Wenn sich der Programmzähler auf eine ungültige Adresse bezieht, wird `PR_PCINVAL` gesetzt und `pr_instr` ist undefiniert.

Das Feld `pr_reg` enthält den Inhalt der Register.

PIOCSTOP*, PIOCWSTOP

`PIOCSTOP` weist den Prozeß an, anzuhalten und wartet, bis dieser angehalten hat; `PIOCWSTOP` wartet nur auf das Anhalten des Prozesses. Diese Operationen sind in dem Moment abgeschlossen, wenn der Prozeß bei einem erwarteten Ereignis anhält. Ist `p` nicht der Nullzeiger, verweist er auf ein Objekt vom Typ `prstatus_t`, welches mit der Statusinformation für den angehaltenen Prozeß ausgefüllt werden soll.

Ein "erwartetes Ereignis" ist entweder ein `PR_REQUESTED`-Halt oder ein Halt, der in den Überwachungs-Schaltern für den Prozeß angegeben war (wird durch `PIOCTRACE`, `PIOCSFAULT`, `PIOCSENTRY` und `PIOCSEXIT` gesetzt). Speziell `PR_JOBCONTROL`-Stop ist kein erwartetes Ereignis. (Ein Prozeß kann infolge eines Haltesignals zweimal anhalten. Wenn nämlich das Signal `PR_SIGNALED` überwacht wird, hält er das erste Mal an und zeigt dieses Signal an. Wird er dann ohne Freigabe dieses Signals fortgesetzt, hält er ein zweites Mal an und zeigt `PR_JOBCONTROL` an.) Wenn der Prozeß durch `ptrace` gesteuert wird, kommt es beim Empfang irgendeines Signals zu einem `PR_SIGNALED`-Stop; dies ist nur dann ein erwartetes Ereignis, wenn sich das Signal in dem Überwachungs-Signalsatz befindet. Wenn `PIOCSTOP` auf einen angehaltenen Prozeß angewandt wird, wird die Halteanweisung wirksam, sobald der Prozeß durch den Zuteilungsmechanismus neu gestartet wurde (außer bei einem erwarteten Ereignis); zu diesem Zeitpunkt geht der Prozeß in einen `PR_REQUESTED`-Halt über, bevor irgendein Code auf Benutzerebene ausgeführt wird.

`ioctl`s sind unterbrechbar durch Signale, so daß beispielsweise ein alarm eingestellt werden kann, um endloses Warten auf einen Prozeß zu vermeiden, der vielleicht nie bei einem erwarteten Ereignis anhält. Wenn `PIOCSTOP` unterbrochen wird, bleibt die Halteanweisung wirksam, auch wenn `ioctl` einen Fehler ausgibt.

Ein Systemprozeß (angezeigt durch den `PR_ISSYS` Schalter) wird niemals auf Benutzerebene ausgeführt, hat keinen durch `/proc` sichtbaren Benutzerebenen-Adreßbereich und kann nicht angehalten werden. Wird `PIOCSTOP` oder `PIOCWSTOP` auf einen Systemprozeß angewendet, entsteht der Fehler `EBUSY`.

PIOCRUN*

Der überwachte Prozeß wurde nach einem Halt erneut gestartet. Ist *p* nicht der Nullzeiger, verweist er auf eine *prun*-Struktur, in der zusätzlich auszuführende Aktionen aufgeführt sind:

```
typedef struct prun {
    long    pr_flags;      /* Schalter */
    sigset_t pr_trace;     /* Signale, die zu verfolgen sind */
    sigset_t pr_sighold;   /* Signale, die zu halten sind */
    fltset_t pr_fault;     /* Fehler, die zu verfolgen sind */
    caddr_t pr_vaddr;      /* Virtuelle Adresse, bei der wieder fortgesetzt
                           werden soll */
    long    pr_filler[8];  /* freier Bereich für zukünftige Erweiterungen */
} prun_t;
```

pr_flags ist eine Bitmaske, die optionale Aktionen beschreibt; die verbleibenden Einträge sind nur von Bedeutung, wenn die entsprechenden Bits in *pr_flags* gesetzt sind. *pr_filler* ist für zukünftige Anwendungen reserviert; dieser Bereich muß durch das Anwenderprogramm mit Nullen gefüllt werden. Definition der Schalter:

PRCSIG löscht das aktuelle Signal, sofern vorhanden (siehe PIOCSSID).

PRCFAULT löscht den aktuellen Fehler, sofern vorhanden (siehe PIOCFAULT).

PRSTRACE stellt die Menge der überwachten Signale entsprechend *pr_trace* ein (siehe PIOCSTRACE).

PRSHOLD stellt die Menge der Haltesignale entsprechend *pr_sighold* ein (siehe PIOCSHOLD).

PRSFault stellt die Menge der überwachten Fehler entsprechend *pr_fault* ein (siehe PIOCFAULT).

PRSVADDR stellt die Adresse, bei der die Ausführung fortgesetzt wird, auf *pr_vaddr* ein.

PRSTEP veranlaßt den Prozeß zu Einzel-Schritten z.B., um eine einzelne Maschinenanweisung ablaufen und ausführen zu lassen. Nach Ausführung der Maschinenanweisung wird eine Hardware-Abfangfunktion erzeugt. Wird FLTTRACE verfolgt, stoppt der Prozeß, andernfalls wird das Signal SIGTRAP gesendet; wird SIGTRAP verfolgt und nicht gehalten, stoppt der Prozeß. Diese Operation verlangt Hardware-Unterstützung und ist eventuell nicht auf allen Prozessoren realisiert.

PRSAORT ist nur dann von Bedeutung, wenn sich der Prozeß in einem PR_SYSENTRY Halt befindet oder als PR_ASLEEP gekennzeichnet ist; er weist den Prozeß an, die Ausführung des Systemaufrufs abubrechen (siehe PIOCSENTRY, PIOCSEXIT).

PRSTOP veranlaßt den Prozeß, sobald wie möglich nach der fortgesetzten Ausführung erneut anzuhalten (siehe PIOCSTOP). Für den Fall, daß der Prozeß bei PR_SIGNALLED oder PR_FAULTED angehalten wird, wird der nächste Stop PR_REQUESTED angezeigt. An dieser Stelle ist kein anderer Stop dazwischengekommen, und der Prozeß hat keinerlei Codes auf Benutzerebene ausgeführt.

PIOCNICE*

Die nice Priorität des überwachten Prozesses wird um den Betrag erhöht, der in dem durch *p* adressierten *int* enthalten ist. Nur der Systemverwalter darf eine Prozeßpriorität auf diese Weise heraufsetzen, jedoch kann jeder Benutzer die Prozeßpriorität herabsetzen.

PIOCPSINFO

Diese Operation gibt sonstige Prozeßinformationen zurück, wie z.B. die, die durch *ps(1)* ausgegeben werden. *p* ist ein Zeiger auf die Struktur *prpsinfo* und enthält mindestens die folgenden Felder:

```
typedef struct prpsinfo {
    char    pr_state; /* numerischer Prozeß-Modus (siehe pr_sname) */
    char    pr_sname; /* druckbares Zeichen, das pr_state darstellt */
    char    pr_zomb; /* !=0: Zombie-Prozeß */
    char    pr_nice; /* Prozeßpriorität für CPU Verwendung */
    u_long  pr_flag; /* Prozeß-Schalter */
    uid_t   pr_uid; /* reale Benutzernummer */
    gid_t   pr_gid; /* reale Gruppennummer */
    pid_t   pr_pid; /* eindeutige Prozeßnummer */
    pid_t   pr_ppid; /* Vaterprozeßnummer */
    pid_t   pr_pgrp; /* PID des Prozeßgruppenleiters */
    pid_t   pr_sid; /* Sitzungs-ID */
    caddr_t pr_addr; /* physikalische Adresse des Prozesses */
    long    pr_size; /* Größe des Prozeß-Abbilds in Seiten */
    long    pr_rssize; /* Anzahl Seiten des residenten Teils */
    caddr_t pr_wchan; /* Warte-Adresse für schlafenden Prozeß */
    timestruc_t pr_start; /* Prozeß-Startzeit, sec+nsec v. Systemuhr */
    timestruc_t pr_time; /* Benutzer- und System-CPU-Zeit für diesen Prozeß */
    long    pr_pri; /* Priorität, hoher Wert ist hohe Priorität */
    char    pr_oldpri; /* vor SVR4, niedriger Wert ist hohe Priorität */
    char    pr_cpu; /* vor SVR4, CPU-Verwendung für Zuteilungs-Planung */
    dev_t   pr_ttydev; /* das kontrollierende Terminal (PRNODEV wenn
                        nicht vorhanden) */
    char    pr_clname[8]; /* Name der Zuteilungsklasse */
    char    pr_fname[16]; /* letztes Element des ausgeführten Pfadnamens */
    char    pr_psargs[PRARGSZ]; /* Anfangs-Zeichen der Argumentliste */
    long    pr_filler[20]; /* freier Bereich für zukünftige Erweiterungen */
} prpsinfo_t;
```

Einige der Einträge in *prpsinfo*, wie *pr_state* und *pr_flag*, sind systemspezifisch. Es kann nicht erwartet werden, daß sie ihre Bedeutungen auf verschiedenen Versionen des Betriebssystems beibehalten. *pr_addr* ist ein Relikt aus der Vergangenheit und hat keine wirkliche Bedeutung in aktuellen Systemen.

PIOCPSINFO kann auf einen Zombie-Prozeß angewandt werden. (ein Prozeß, der beendet wurde, dessen Vater jedoch noch kein *wait* darauf ausgeführt hat).

PIOCNMAP, PIOCMAp

Diese Operationen liefern Informationen über die Speicherzuordnungen = Map-pings (virtuelle Adressbereiche) für den überwachten Prozeß. PIOCnMAP gibt in einem durch *p* adressierten *int* die Anzahl der gegenwärtig aktiven Zuordnungen zurück. Mit Hilfe dieses Resultats kann man einen Speicher reservieren, der einer PIOCMAp Operation zur Verfügung gestellt wird, welche die Liste der aktuell aktiven Zuordnungen ausgibt. Für PIOCMAp verweist der Zeiger *p* auf ein Feld von Elementen des Typs *prmap_t*; für jede Zuordnung wird ein Feldelement (eine Struktur) zurückgegeben; ein weiteres Element, das nur Nullen enthält, markiert das Ende der Liste.

```
typedef struct prmap {
    caddr_t  pr_vaddr;      /* virtuelle Basis-Adresse */
    u_long   pr_size;       /* Größe der Zuordnung in Bytes */
    off_t    pr_off;        /* Relativ-Zeiger in entsprechendem Objekt,
                             falls vorhanden */
    long     pr_mflags;     /* Schutz- und Attribut-Bits */
    long     pr_filler[4];  /* Freier Bereich für zukünftige Erweiterungen */
} prmap_t;
```

pr_vaddr ist die virtuelle Basis-Adresse (die Untergrenze) der Zuordnung innerhalb des überwachten Prozesses und *pr_size* ist seine Größe in Bytes. *pr_off* ist die Distanz innerhalb des abgebildeten Objekts, auf welche die Basis-Adresse abgebildet ist.

pr_mflags ist eine Bitmaske von Schutz- und Attribut-Bits:

MA_READ	Zuordnung ist von dem überwachten Prozeß lesbar
MA_WRITE	Zuordnung ist von dem überwachten Prozeß schreibbar
MA_EXEC	Zuordnung ist von dem überwachten Prozeß ausführbar
MA_SHARED	Zuordnungs-Änderungen gelten allgemein
MA_BREAK	Zuordnung ist durch den <i>brk</i> Systemaufruf gewachsen
MA_STACK	Zuordnung ist durch Stapelfehler gewachsen

PIOCOPENM

Der Return-Code *retval* stellt einen Nur-Lese-Dateideskriptor für ein dem überwachten Prozeß zugeordnetes Objekt zur Verfügung. Ist *p* gleich Null, ist die durch *exec* ausgeführte Datei des überwachten Prozesses (seine *a.out* Datei) gefunden worden. Dies ermöglicht es einem Fehlersuchprogramm, die Symbolta-belle der Objektdatei zu finden, ohne den Pfadnamen der ausführbaren Datei zu kennen. Ist *p* nicht Null, verweist es auf die Struktur *caddr_t*, die die virtuelle Adresse innerhalb des überwachten Prozesses und des abgebildeten Objekts enthält, sofern dieses existiert. Es kann verwendet werden, um einen Dateideskrip-tor für eine gemeinsam benutzte Bibliothek zu finden, der zu dem Prozeß gehört. Im Fehlerfall wird -1 zurückgegeben (ungültige Adresse oder kein abgebildetes Objekt für die angegebene Adresse).

PIOCCRED

Holt den Satz von Berechtigungsnachweisen, die mit dem Prozeß verbunden sind. *p* verweist auf ein Objekt der Struktur `prcred_t`, die durch den Aufruf aufgefüllt wird:

```
typedef struct prcred {
    uid_t    pr_euid; /* effektive Benutzernummer */
    uid_t    pr_ruid; /* tatsächliche Benutzernummer */
    uid_t    pr_suid; /* gesicherte Benutzernummer (von exec) */
    uid_t    pr_egid; /* effektive Gruppennummer */
    uid_t    pr_rgid; /* tatsächliche Gruppennummer */
    uid_t    pr_sgid; /* gesicherte Gruppennummer (von exec) */
    u_int    pr_ngroups; /* Anzahl der zusätzlichen Gruppen */
} prcred_t;
```

PIOCGROUPS

Holt die zusätzlichen Gruppennummern (GIDs) dieses Prozesses. *p* verweist auf ein Feld mit Elementen des Typs `uid_t`; dieses Feld wird durch die Funktion versorgt. Um die Anzahl der zurückgelieferten Gruppen (`pr_ngroups`) und damit den benötigten Speicherbedarf für deren Aufnahme zu bestimmen, kann zuvor der `PIOCCRED` Aufruf verwendet werden.

PIOCGETPR, PIOCGETU

Diese Operationen kopieren jeweils die `proc` Struktur des überwachten Prozesses und dessen Benutzer-Bereich in den durch *p* adressierten Puffer. Sie existieren der Vollständigkeit halber. Eigentlich sollte es nicht notwendig sein, auf eine dieser Strukturen direkt zuzugreifen, da die relevante Statusinformation durch andere Kontrolloperationen verfügbar ist. Da ein Programm, das diese Operationen verwendet, an eine bestimmte Version des Betriebssystems gebunden ist, wird von deren Anwendung abgeraten.

`PIOCGETPR` kann auf einen Zombie-Prozeß angewandt werden (siehe `PIOCPSINFO`).

HINWEIS

Jede Operation (`ioctl` oder I/O) ist garantiert atomar unter Berücksichtigung des überwachten Prozesses, außer wenn sie auf einen Systemprozeß angewandt wird.

Aus Gründen der Sicherheit scheitert das Öffnen einer `proc` Datei, wenn nicht sowohl die Benutzernummer (UID) als auch die Gruppennummer (GID) des aufrufenden Programms mit denen des überwachten Prozesses übereinstimmen und die Objektdatei des Prozesses durch das aufrufende Programm gelesen werden kann. Dies gilt nicht für den Systemverwalter. Dateien, die sich auf `setuid` und `setgid` Prozesse beziehen, können ausschließlich durch den Systemverwalter geöffnet werden. Wenn der überwachte Prozeß ein `exec` auf eine `setuid`/`setgid` Objektdatei ausführt oder auf eine Objektdatei, die er nicht lesen kann, wird ein offener Prozeß-Dateideskriptor sogar dann ungültig, wenn er durch den Systemverwalter verwendet wird. Jede Operation, die auf einem ungültigen Dateideskriptor durchgeführt wird, ausgenommen `close`, scheitert mit `EAGAIN`. Falls vorhandene Überwachungs-Schalter gesetzt sind und die Prozeßdatei zum Schreiben geöffnet ist, wird der Prozeß in dieser Situation angewiesen, anzuhalten und sein `run-on-last-close` Schalter wird gesetzt (siehe `PIOCSRLC`). Dadurch kann ein kontrollierender Prozeß (falls er die Berechtigung dazu hat), die Prozeßdatei wieder öffnen, um eine neue gültige Dateikennzahl zu erhalten, den ungültigen Dateideskriptor zu schließen, und fortzufahren. Wird der ungültige

Dateideskriptor lediglich geschlossen, setzt der überwachte Prozeß seine Ausführung ohne gesetzte Überwachungs-Schalter fort. Ein Prozeß, der gegenwärtig nicht zum Schreiben über `/proc` geöffnet ist, jedoch übriggebliebene Überwachungs-Schalter von einem vorangegangenen Open-Aufruf hat und ein `exec` auf eine `setuid/setgid` oder unlesbare Objektdatei ausführt, wird nicht angehalten. Seine Überwachungs-Schalter werden jedoch alle zurückgesetzt.

Aus Gründen der Symmetrie und Effizienz gibt es mehr Steueroperationen, als unbedingt notwendig.

DATEIEN

`/proc` Verzeichnis (Liste der aktiven Prozesse)
`/proc/nnnnn` Prozeß-Abbild

SIEHE AUCH

`open(2)`, `ptrace(2)`, `sigaction(2)`, `signal(2)`, `sigset(2)`

FEHLERMELDUNGEN

Es folgen zusätzliche Fehler, die neben den normalerweise im Zusammenhang mit Zugriffen auf Dateisysteme vorkommenden Fehlern auftreten können:

ENOENT	Der überwachte Prozeß wurde beendet, nachdem er geöffnet wurde.
EIO	E/A wurde auf eine unzulässige Adresse in dem überwachten Prozeß versucht.
EBADF	Eine E/A- oder <code>ioctl</code> -Operation, die Schreibzugriff verlangt, wurde auf einen Dateideskriptor versucht, der nicht zum Schreiben verwendet werden kann.
EBUSY	<code>PIOCSTOP</code> oder <code>PIOCWSTOP</code> wurde auf einen Systemprozeß angewandt; ein exklusiver <code>open</code> wurde auf eine Prozeßdatei versucht, die bereits zum Schreiben geöffnet war; ein <code>open</code> zum Schreiben wurde versucht, obwohl die Prozeßdatei exklusiv geöffnet war; <code>PIOCRUN</code> , <code>PIOCSREG</code> oder <code>PIOCSFPREG</code> wurde auf einen Prozeß angewandt, der nicht bei einem erwarteten Ereignis angehalten hat; es wurde versucht, <code>/proc</code> einzuhängen, obwohl es schon eingehängt war.
EPERM	Jemand anders als der Systemverwalter versuchte eine Prozeßpriorität durch den Aufruf von <code>PIOCNICE</code> heraufzusetzen.
ENOSYS	Es wurde versucht, eine nicht unterstützte Operation (wie <code>create</code> , <code>remove</code> , <code>link</code> oder <code>unlink</code>) auf einen Eintrag in <code>/proc</code> auszuführen.
EFAULT	Eine E/A- oder <code>ioctl</code> -Anforderung verwies auf eine ungültige Adresse in dem kontrollierenden Prozeß.
EINVAL	Im allgemeinen bedeutet dies, daß irgendein ungültiges Argument an einen Systemaufruf übergeben wurde. Die Liste der Bedingungen, die diesen Fehler hervorrufen, enthält: der <code>ioctl</code> Code ist undefiniert; eine <code>ioctl</code> Operation auf einen Dateideskriptor, der sich auf das <code>/proc</code> Verzeichnis bezieht, wurde aufgerufen; eine ungültige Signalnummer wurde in <code>PIOCSIG</code> , <code>PIOCKILL</code> , oder <code>PIOCUNKILL</code> angegeben; <code>SIGKILL</code> wurde in <code>PIOCUNKILL</code> verwendet; eine unzulässige virtuelle Adresse wurde in einer <code>PIOCOPENM</code> Anforderung verwendet;

proc(4)

proc(4)

PIOCGFPREG oder PIOCSFPREG wurde auf einer Maschine ohne Gleitkomma-Hardware verwendet.

EINTR Ein Signal wurde durch den kontrollierenden Prozeß empfangen, als er darauf wartete, daß der überwachte Prozeß durch PIOCSTOP oder PIOCWSTOP angehalten wird.

EAGAIN Der überwachte Prozeß hat ein exec auf eine setuid/setgid Objektdatei oder eine Objektdatei, die er nicht lesen kann, ausgeführt; alle weiteren Operationen auf den Prozeß-Dateideskriptor (außer close) rufen diesen Fehler hervor.

BEZEICHNUNG

profile – Einstellung einer Shell-Umgebung unmittelbar nach der Anmeldung

ÜBERSICHT

/etc/profile
\$HOME/.profile

BESCHREIBUNG

Für alle Benutzer, die als Kommandointerpreter die Shell `sh(1)` besitzen, werden die Befehle in diesen Dateien als Teil ihrer Anmelde-Prozedur ausgeführt.

/etc/profile ermöglicht dem Systemverwalter, Dienste zu definieren, die bei der Anmeldung eines jeden Benutzers ausgeführt werden. Typische Dienste beinhalten: die Ankündigung von System-Neuigkeiten, Benutzerpost und die Einstellung der Standard-Umgebungsvariablen. Gelegentlich werden von /etc/profile besondere Aktionen für die `root`-Anmeldung oder den `su`-Befehl durchgeführt. Auf Computern, die außerhalb der östlichen US-Zeitzone verwendet werden, muß die Zeile

```
. /etc/TIMEZONE
```

am Anfang von /etc/profile eingetragen sein (siehe `timezone(4)`). /etc/profile wird vor \$HOME/.profile ausgeführt.

Die Datei \$HOME/.profile wird für die Einstellung von Umgebungsvariablen (und deren Export) und Terminalmodi für Benutzer verwendet. Ein typischer Inhalt (abgesehen von den Kommentaren) wäre:

```
# Export einiger Umgebungsvariablen
export MAIL PATH TERM
# Dateierstellungsmaske setzen
umask 022
# Nachricht beim Eingang neuer Post
MAIL=/var/mail/$LOGNAME
# Hinzufügen des eigenen Verzeichnisses bin zum
# Shell-Suchpfad
PATH=$PATH:$HOME/bin
# Terminal initialisieren und Tabulatoren setzen
# Korrekturtaste als Löschzeichen einstellen
stty erase '^H' echoe
```

DATEIEN

/etc/TIMEZONE	Zeitzone-Umgebung
\$HOME/.profile	Benutzerspezifische Umgebung
/etc/profile	Systemweite Umgebung

SIEHE AUCH

`env(1)`, `login(1)`, `mail(1)`, `sh(1)`, `stty(1)`, `su(1M)`, `tput(1)`
`terminfo(4)`, `timezone(4)`, `environ(5)`, `term(5)` im
Referenzhandbuch für Systemverwalter
Leitfaden für Benutzer

HINWEIS

Bei der Bereitstellung systemweiter Dienste in `/etc/profile` muß sorgfältig vorgegangen werden. Persönliche `.profile`-Dateien sind für die Bedienung fast aller globalen Anforderungen besser geeignet.

BEZEICHNUNG

protocols – Datenbank für Protokollnamen

ÜBERSICHT

/etc/protocols

BESCHREIBUNG

Die Datei protocols enthält Informationen über die bekannten Protokolle, die im DARPA Internet verwendet werden. Für jedes Protokoll muß eine eigene Zeile mit der folgenden Information vorhanden sein:

offizieller_Protokollname Protokollnummer Aliase

Die einzelnen Teile sind durch eine beliebige Anzahl von Leer-/Tabulatorzeichen voneinander getrennt. Ein '#' zeigt den Beginn eines Kommentars an; Zeichen bis zum Zeilenende werden nicht durch Routinen interpretiert, die die Datei abarbeiten.

Protokoll-Namen dürfen außer Feldbegrenzungszeichen, Neue-Zeile-Zeichen oder Kommentarzeichen jedes darstellbare Zeichen enthalten.

BEISPIEL

für eine solche Datenbank:

```
#
# Internet (IP) Protokolle
#
ip      0      IP      # internet protocol, pseudo-protocol number
icmp    1      ICMP    # internet control message protocol
ggp      3      GGP     # gateway-gateway protocol
tcp      6      TCP     # transmission control protocol
pup     12      PUP     # PARC universal packet protocol
udp     17      UDP     # user datagram protocol
```

DATEIEN

/etc/protocols

SIEHE AUCH

getprotoent(3N)

HINWEIS

Anstelle einer statischen Datei muß ein Name-Server verwendet werden. Ein binär indiziertes Dateiformat sollte für schnellen Zugriff verfügbar sein.

BEZEICHNUNG

resolv.conf – Konfigurationsdatei für Name-Server-Routinen

ÜBERSICHT

/etc/resolv.conf

BESCHREIBUNG

Die Resolver-Routinen lesen die Informationen in dieser Datei, wenn sie das erste Mal in einem Prozeß aufgerufen werden. Die Datei enthält eine Liste von Paaren Schlüsselwort-Wert in lesbarer Form. Diese stellen verschiedene Arten von Resolver-Informationen bereit.

schlüsselwort wert

Die verschiedenen Konfigurationsmöglichkeiten sind:

nameserver adresse Die Internet-Adresse (in Punktschreibweise) eines Name-Servers, die der Resolver abfragen soll. Es sollte mindestens ein Name-Server aufgelistet sein. Bis zu MAXNS (gegenwärtig 3) Name-Server können angegeben werden. In diesem Fall werden sie von den Resolver-Bibliothek-Abfragen in der aufgelisteten Reihenfolge geprüft. Der verwendete Algorithmus prüft zuerst einen Name-Server. Wenn die Datenbank-Abfrage das Zeitlimit überschreitet, prüft er den nächsten, bis kein Name-Server mehr vorhanden ist. Dann werden die Versuche bei allen Name-Servern wiederholt, bis eine maximale Anzahl von Versuchen erreicht ist.

domain name Dieser Standard-Bereich kann an Namen angehängt werden, die keinen Punkt enthalten.

address adresse Eine Internet-Adresse (in Punktschreibweise) eines beliebigen bevorzugten Netzwerks. Die Liste der Adressen, die durch den Resolver zurückgegeben werden, wird sortiert, um die Adresse dieses Netzwerks vor jede andere zu setzen.

Das Paar Schlüsselwort-Wert muß in einer eigenen Zeile stehen und der Schlüsselbegriff (beispielsweise *nameserver*) muß am Zeilenanfang stehen. Der Wert ist vom Schlüsselwort durch ein Leer-/Tabulatorzeichen getrennt.

DATEIEN

/etc/resolv.conf

SIEHE AUCH

named(1M), gethostent(3N), resolver(3N).

BEZEICHNUNG

routing – System-Unterstützung für Paket-Routing

ÜBERSICHT

```
#include <net/route.h>
```

BESCHREIBUNG

Zu den Möglichkeiten des Netzwerks gehört ein allgemeines Paket-Routing. Die Verwaltung der Routing-Tabellen kann durch Anwenderprozesse vorgenommen werden.

Eine Routing-Tabelle besteht aus einem einfachen Satz von Datenstrukturen. Sie wird zur Auswahl der passenden Netzwerkschnittstelle bei der Übertragung von Paketen verwendet. Diese Tabelle enthält einen einzelnen Eintrag für jeden Weg zu einem bestimmten Netzwerk oder Host. Die Routing-Tabelle wurde zur Unterstützung des Routing für das Internet Protokoll (IP) entworfen. Seine Implementierung ist jedoch protokollunabhängig und kann deshalb andere Protokolle ebenfalls bedienen. Anwenderprogramme dürfen diese Datenbank mit Hilfe der zwei `ioctl(2)` Befehle `SIOCADDRT` und `SIOCDELRT` verändern. Diese Befehle erlauben das Einfügen bzw. Löschen eines einzelnen Eintrags einer Routing-Tabelle. Eine Veränderung der Routing-Tabelle darf nur durch privilegierte Benutzer ausgeführt werden.

Ein Eintrag in der Routing-Tabelle hat die folgende Form, wie in `/usr/include/net/route.h` definiert:

```
struct rtenry {
    u_long  rt_hash;           /* beschleunigte Suche */
    struct  sockaddr rt_dst;    /* Schlüssel */
    struct  sockaddr rt_gateway; /* Wert */
    short   rt_flags;          /* Host/Netz */
    short   rt_refcnt;          /* Anzahl Verweise */
    u_long  rt_use;            /* Anzahl zeichenorientierter Pakete */
#ifdef STRNET
    struct  ip_provider *rt_prov; /* Vermittlung benutzen */
#else
    struct  ifnet *rt_ifp;        /* Schnittstelle benutzen */
#endif /* STRNET */
};
```

Wobei `rt_flags` definiert ist durch:

```
#define RTF_UP      0x1 /* Weg benutzbar */
#define RTF_GATEWAY 0x2 /* Ziel ist ein Gateway */
#define RTF_HOST    0x4 /* Host Eintrag (sonst Netz) */
```

Es gibt drei Arten von Einträgen in der Routing-Tabelle: für einen bestimmten Host, für alle Hosts auf einem bestimmten Netzwerk und für ein beliebiges Ziel, das nicht mit Einträgen der ersten zwei Arten übereinstimmt (ein Wildcard-Pfad). Jede Netzwerkschnittstelle nimmt einen Eintrag in eine Routing-Tabelle vor, wenn sie initialisiert wird. Normalerweise spezifiziert die Schnittstelle den Weg, der einen direkten Anschluß zum Ziel-Host oder Netzwerk angibt. Die Transportebene einer

Protokoll-Familie verlangt bei einem direkten Weg gewöhnlich, daß das Paket an denselben Host, der darin angegeben ist, geschickt werden soll. Ist das nicht der Fall, wird die Schnittstelle aufgefordert, das Paket an eine andere Einheit als den ursprünglichen Empfänger zu adressieren (das heißt, das Paket wird weitergeleitet).

Einträge in Routing-Tabellen, die durch einen Benutzerprozeß vorgenommen werden, dürfen keine Hash-, Verweiszähler-, Anwendungs- oder Schnittstellenfelder sein; diese werden durch die Routing-Routinen ausgefüllt. Wird ein Weg gerade benutzt, wenn er gelöscht wird (`rt_refcnt` ist ungleich Null), werden alle damit zusammenhängenden Ressourcen so lange nicht freigegeben, bis alle Verweise darauf entfernt sind.

Benutzer-Prozesse lesen die Routing-Tabellen über das `/dev/kmem` Gerät.

Das Feld `rt_use` enthält die Anzahl der Pakete, die auf diesem Weg gesendet wurden. Dieser Wert wird dazu benutzt, unter mehreren Wegen solche auszuwählen, die zum gleichen Ziel führen. Existieren mehrere, wird der am wenigsten verwendete Weg ausgewählt.

Ein Wildcard Routing-Eintrag kann vorgenommen werden, indem man keinen Wert für die Zieladresse angibt. Wildcard-Strecken werden nur verwendet, um im Falle eines Systemausfalls einen Weg zum Ziel-Host und Netzwerk zu finden. Die Kombination von Wildcard-Pfaden und Routing-Umlenkungen stellt einen einfachen Mechanismus für die Regelung des Routing-Verkehrs zur Verfügung.

DATEIEN

`/dev/kmem`

SIEHE AUCH

`ioctl(2)`, `route(1M)`, `routed(1M)`.

FEHLERMELDUNGEN

EEXIST	Das Verdoppeln eines bestehenden Eintrags wurde angefordert.
ESRCH	Das Löschen eines nicht bestehenden Eintrags wurde angefordert.
ENOBUFS	Die Ressourcen reichen nicht aus, um einen neuen Weg zu installieren.

BEZEICHNUNG

rt_dptbl – Parametertabelle für den Echtzeit-Dispatcher

ÜBERSICHT

```
#include <sys/rt.h>
```

BESCHREIBUNG

Der Prozeß-Scheduler (oder Dispatcher/Zuteileroutine) ist der Anteil des Systemkerns, der die Belegung der CPU für Prozesse kontrolliert. Der Scheduler unterstützt den Begriff von Zuteilungsklassen. Jeder Klasse entspricht eine Zuteilungsmethode, die verwendet wird, um den Prozessen innerhalb dieser Klasse CPU-Zeit zuzuteilen. Verbunden mit jeder Zuteilungsklasse ist ein Satz von Warteschlangen, in denen startbereite Prozesse Schlange stehen. Diesen Warteschlangen wird durch die System-Konfiguration ein Satz von globalen Zuteilungsprioritäten zugeordnet, die für Prozesse innerhalb der Klasse verfügbar sind. (Die Zuteileroutine wählt zur Ausführung immer den Prozeß mit der höchsten globalen Zuteilungspriorität im System aus.) Die mit einer vorgegebenen Klasse verbundenen Warteschlangen werden durch diese Klasse angesehen wie ein lückenloser Satz von Prioritätsebenen, numeriert von 0 (niedrigste Priorität) bis *n* (höchste Priorität - dieser Wert ist konfigurationsabhängig). Der Satz der globalen Zuteilungsprioritäten, dem die Warteschlangen für eine vorgegebene Klasse zugeordnet werden, müssen nicht bei Null anfangen und dürfen Lücken enthalten (abhängig von der Konfiguration).

Die Echtzeit-Klasse verwaltet eine Tabelle im Hauptspeicher mit einem Eintrag für jede Prioritätsebene und vergibt die Merkmale für diese Ebene. Diese Tabelle hat den Namen Real-Time Dispatcher Parameter Table (rt_dptbl). Die rt_dptbl besteht aus einem Feld von Parameter-Strukturen (struct rt_dpent), je eine für jede der *n* Prioritätsebenen. Die Eigenschaften einer vorgegebenen Prioritätsebene *i* sind angegeben durch die *i*-te Parameter-Struktur in diesem Feld (rt_dptbl).

Eine Parameter-Struktur besteht aus folgenden Elementen. Diese sind auch in der /usr/include/sys/rt.h Include-Datei beschrieben.

rt_globpri	Die mit dieser Prioritätsebene verbundene globale Zuteilungspriorität. Die Zuordnung zwischen Echtzeit-Prioritätsebenen und globalen Zuteilungsprioritäten wird beim Hochfahren durch die Systemkonfiguration festgelegt. Die rt_globpri Werte können nicht mit dispadmin(1M) geändert werden.
rt_quantum	Die Länge der Zeiteinheit in Takten (HZ), die Prozessen auf dieser Ebene zugeteilt wird. Der Zeiteinheits-Wert ist nur ein Standard- oder Startwert für Prozesse auf einer bestimmten Ebene. Diese Zeiteinheit eines Echtzeit-Prozesses kann durch den Benutzer mit dem Befehl priocntl oder dem Systemaufruf priocntl geändert werden.

Ein Systemverwalter kann das Verhalten des Echtzeit-Anteils des Schedulers durch Neukonfigurieren der rt_dptbl beeinflussen. Hierfür sind zwei Methoden verfügbar.

DISPADMIN Konfigurationsdatei

Die `rt_quantum`-Werte in der `rt_dptbl` können auf einem laufenden System durch den Befehl `dispadmin(1M)` geprüft und verändert werden. Der Aufruf von `dispadmin` für die Echtzeit-Klasse erlaubt dem Systemverwalter die aktuelle `rt_dptbl`-Konfiguration der Tabelle im Hauptspeicher zu laden oder diese mit Werten aus einer Konfigurationsdatei zu überschreiben. Die für Eingaben in `dispadmin` verwendete Konfigurationsdatei muß mit dem unten beschriebenen Format übereinstimmen.

Leerzeilen werden ignoriert und ein beliebiger Teil einer Zeile rechts von einem `#`-Symbol wird als Kommentar behandelt. Die erste Zeile, die weder Kommentar noch leer ist, muß die Auflösung anzeigen, die zur Interpretation der Zeiteinheiten-Werte benutzt wird. Die Auflösung ist angegeben als

`RES=res`

wobei *res* eine positive Ganzzahl zwischen 1 und 1.000.000.000 und die verwendete Auflösung der Kehrwert von *res* in Sekunden ist. (Zum Beispiel `RES=1000` bezeichnet Auflösung in Millisekunden.) Obgleich eine sehr genaue (Nanosekunden) Auflösung angegeben sein kann, werden die Längen der Zeiteinheiten aufgerundet auf das nächste ganzzahlige Vielfache der Systemuhr-Auflösung. Beispielsweise ist die höchste gegenwärtig auf der 3B2 verfügbare Auflösung 10 Millisekunden (1 "Takt"). Wäre *res* 1000, würde ein Zeiteinheiten-Wert von 34 eine Einheit von 34 Millisekunden benennen, welcher auf der 3B2 auf 4 Takte (40 Millisekunden) aufgerundet würde.

Die restlichen Zeilen in der Datei werden verwendet, um die `rt_quantum`-Werte für jede Echtzeit-Prioritätsebene anzugeben. Die erste Zeile bezeichnet die Einheit der Echtzeit-Ebene 0, die zweite Zeile bezeichnet die Einheit der Echtzeit-Ebene 1, usw.. Für jede konfigurierte Echtzeit-Prioritätsebene muß genau eine Zeile vorhanden sein. Jeder `rt_quantum`-Eintrag muß entweder eine positive Ganzzahl sein, die die gewünschte Zeiteinheit spezifiziert (in der durch *res* vorgegebenen Auflösung), oder das Symbol-`RT_TQINF`, das eine unbegrenzte Zeiteinheit für diese Ebene anzeigt.

BEISPIEL

Der folgende Auszug einer `dispadmin`-Konfigurationsdatei veranschaulicht das Format. Es ist zu beachten, daß es für jede Zeile, die eine Zeiteinheit spezifiziert, einen Kommentar gibt, der auf die entsprechende Prioritätsebene hinweist. Diese Ebenen-Nummern zeigen die Priorität innerhalb der Echtzeit-Klasse an, und die Zuordnung zwischen diesen Echtzeit-Prioritäten und den entsprechenden globalen Zuteilungsprioritäten wird durch die in der `rt` Masterdatei angegebene Konfiguration festgelegt. Genaugenommen dienen die Ebenen-Nummern dem Komfort des Systemverwalters beim Lesen der Datei und werden wie jeder beliebige Kommentar von `dispadmin` ignoriert. `dispadmin` geht davon aus, daß die Zeilen in der Datei in aufeinanderfolgender aufsteigender Reihenfolge von Prioritätsebenen (von 0 bis zur maximalen konfigurierten Echtzeit-Priorität) geordnet sind. Die Ebenen-Nummern in den Kommentaren sollten normalerweise mit dieser Anordnung übereinstimmen; wenn dies aus irgendeinem Grund nicht der Fall ist, hat das jedoch keine Auswirkung auf `dispadmin`.

rt_dptbl(4)

rt_dptbl(4)

```
# Konfigurationsdatei für die Echtzeit-Zuteileroutine
RES=1000
#      Zeiteinheit                Priorität
#      (rt_quantum)              Ebene
      100                        #      0
      100                        #      1
      100                        #      2
      100                        #      3
      100                        #      4
      100                        #      5
      90                         #      6
      90                         #      7
      .                          .      .
      .                          .      .
      .                          .      .
      10                         #      58
      10                         #      59
```

DATEIEN

/usr/include/sys/rt.h

SIEHE AUCH

dispadm(1M), priocntl(1), priocntl(2),

Kapitel "Prozeß-Scheduler" im *Leitfaden für Systemverwalter*

BEZEICHNUNG

savetab – Liste zu sichernder Slices

BESCHREIBUNG

sysssave(1M) erwartet die Liste der zu sichernden Slices in einer Datei, die bei der Option `-t` angegeben werden kann. Standardmäßig ist dies `/var/sysssave/savetab`.

Die Datei ist eine editierbare ASCII-Datei. Pro Zeile steht ein Slice- oder Platten-Name (z.B. `shd1t5s6` oder `c0d1`). Die Slices können auf Platten am Interphase Storage oder am NCR Host Adapter liegen. Leerzeilen in der Datei werden überlesen. Kommentare beginnen mit dem Zeichen `"#"` und reichen bis zum Zeilenende.

Wird Slice 0 angegeben (z.B. `c0d0s0`), dann werden alle Slices dieses Laufwerks gesichert, bis auf Slices vom Typ `V_BACKUP`, `V_BOOT`, `V_OTHER`, `V_ALATTRK` oder `V_ALTS`.

Anstatt eines Slice-Namens kann auch ein Laufwerk-Name ohne Sliceangabe (z.B. `c0d0`) angegeben werden. Dieser wird wie `c0d0s0` behandelt, d.h. alle Slices des Laufwerks werden gesichert.

Slicetypen, die nicht gesichert werden

`V_BOOT` Enthält das VTOC und den `pdinfo`-Block.

`V_BACKUP`

Erstreckt sich über die gesamte UNIX-Partition einer Platte. Slice 0 ist von diesem Typ.

`V_ALTS` Wird z.B. bei AT386-Systemen für die Zuweisung von Ersatzsektoren verwendet. Bei der MX300 wird diese Slice nicht verwendet, da dort die Zuweisung über den Controller erfolgt.

`V_ALATTRK`

Wird z.B. bei AT386-Systemen für die Zuweisung von Ersatzspuren verwendet. Bei der MX300 wird diese Slice nicht verwendet, da dort die Zuweisung über den Controller erfolgt.

`V_OTHER`

Nicht-UNIX-Slice. Eine eventuell vorhandene DOS-Partition ist ebenfalls im VTOC eingetragen. Diese Slice hat dann den Typ `V_OTHER`.

BEISPIEL

```
#
# Beispiel für /var/sysssave/savetab
#
c0d1s0      # whole drive 1
c1d0s3      # data base slice
shd0t5s6    # some other slice
shd1t5      # whole external drive
```

DATEIEN

`/var/sysssave/savetab`, `/var/sysssave/LOGFILE`.

savetab (4)

(MX300)

savetab (4)

SIEHE AUCH

`sysrestore(1M) syssave(1M)`

BEZEICHNUNG

sccsfile – Format einer SCCS-Datei

BESCHREIBUNG

Eine SCCS (Source Code Control System) Datei ist eine ASCII-Datei. Sie besteht aus sechs logischen Teilen: der Prüfsumme, der Delta-Tabelle (enthält Informationen über jede Veränderung), Benutzernamen (Login-Namen und/oder Gruppennummern (GIDs) von Benutzern, die Deltas einfügen dürfen), Flags (Definitionen von internen Schlüsselwörtern), Kommentaren (beliebige beschreibende Informationen über die Datei), und dem Hauptteil (die eigentlichen Textzeilen, vermischt mit Steuerzeilen).

Innerhalb einer SCCS-Datei gibt es Zeilen, die mit dem ASCII-Zeichen SOH (start of heading; oktal 001) beginnen. Dieses Zeichen wird in dieser Dokumentation als Steuerzeichen bezeichnet und als @ dargestellt. Jede unten beschriebene Zeile, die nicht ausdrücklich mit dem Steuerzeichen beginnt, darf dieses auch nicht enthalten.

Einträge in der Form DDDDD stellen eine 5-stellige Ziffernfolge (eine Zahl zwischen 00000 und 99999) dar.

Jeder logische Teil einer SCCS-Datei ist unten im Detail beschrieben.

Prüfsumme

Die Prüfsumme ist die erste Zeile einer SCCS-Datei. Die Zeile hat die Form:

```
@h      DDDDD
```

Der Wert der Prüfsumme ist die Summe über alle Zeichen, ausgenommen die der ersten Zeile. Die Zeile @h liefert eine Dateiformatkennung (magic number) von (oktal) 064001, abhängig von der Reihenfolge der Bytes.

Delta

Die Delta-Tabelle besteht aus einer variablen Anzahl von Einträgen. Diese können eine der folgenden Formen haben:

```
@s      DDDDD/      DDDDD/      DDDDD
@i <type> <SCCS ID> yr/mo/da hr:mi:se <pgmr>      DDDDD      DDDDD
@i      DDDDD ...
@x      DDDDD ...
@g      DDDDD ...
@m < MR number>
...
@c <comments> ...
...
@e
```

Die erste Zeile (@s) enthält die jeweilige Anzahl der Zeilen, die eingefügt, gelöscht bzw. unverändert sind. Die zweite Zeile (@i) enthält den Typ des Deltas (normal: D oder entfernt: R), die SCCS ID des Deltas, das Datum und den Zeitpunkt der Erstellung des Deltas, den Benutzernamen entsprechend der realen Benutzernummer/UID zur Zeit, als das Delta angelegt wurde, und die Seriennummern des Deltas beziehungsweise seiner Vorgänger.

Die Zeilen @i, @x und @g enthalten die Serien-Nummern von Deltas, die enthalten, ausgeschlossen bzw. ignoriert sind. Diese Zeilen sind optional.

Die @m-Zeilen (optional) enthalten jeweils eine MR-Nummer, die dem Delta zugeordnet ist. Die @c-Zeilen enthalten dem Delta zugeordnete Kommentare. Die Zeile @e beendet den Delta-Eintrag.

Benutzernamen

Die Liste der Login-Namen und/oder Gruppennummern/GIDs von Benutzern, die Deltas in der Datei einfügen dürfen. Sie werden durch Neue-Zeile-Zeichen voneinander getrennt. Die Zeilen, die diese Login-Namen und/oder numerischen Gruppennummern/GIDs enthalten, sind von den Zeilen @u und @U eingeschlossen. Eine leere Liste erlaubt jedem Benutzer, eine Veränderung vorzunehmen. Eine Zeile, die mit einem ! beginnt, untersagt der dort angegebenen Gruppe bzw. dem Benutzer, Veränderungen vorzunehmen.

Flags

Intern verwendete Schlüsselwörter. Für weitere Informationen über ihre Anwendung siehe `admin(1)`. Jede Flag-Zeile hat folgende Form:

```
@f <flag>    <optional text>
```

Die folgenden Flags sind definiert:

```
@f t <Programm Typ>
@f v <Programm Name>
@f i <Schlüsselwort Zeichenkette>
@f b
@f m <Modul Name>
@f f <ganzzahlig abgerundeter Wert>
@f c <ganzzahlig aufgerundeter Wert>
@f d <Standard-SID>
@f n
@f j
@f l <gesperrte Versionen>
@f q <definiert durch Benutzer>
@f z <reserviert für Anwendung in Schnittstellen>
```

Das t-Flag definiert den Ersatz für das %Y% Schlüsselwort. Das v-Flag steuert die Eingabeaufforderung für MR-Nummern zusätzlich zu Kommentaren. Wenn optionaler Text vorhanden ist, definiert es ein Kontroll-Programm für die Gültigkeit der MR-Nummer. Das i-Flag steuert den Warnung/Fehler-Aspekt der Meldung "No id keywords". Wenn das i-Flag nicht vorhanden ist, ist diese Meldung nur eine Warnung. Ist es jedoch vorhanden, verursacht diese Meldung einen Abbruchfehler (die Datei wird nicht "geholt" bzw. das Delta nicht erstellt). Wenn das b-Flag vorhanden ist, kann die Option -b in dem Befehl `get` verwendet werden, um einen Zweig im Deltabaum zu erzeugen. Das m-Flag definiert die erste Wahl für den Ersatz-Text des %M%-Schlüsselworts. Das f-Flag definiert die "unterste" Freigabe. Unter dieser Freigabe können keine Deltas mehr hinzugefügt werden. Das c-Flag definiert die "oberste" Freigabe. Oberhalb dieser Freigabe können keine Deltas mehr hinzugefügt werden. Das d-Flag definiert die Standard-SID, die benutzt wird, wenn in einem `get`-Befehl kein SID angegeben wird. Das n-Flag veranlaßt `delta` ein Null-Delta (ein Delta, in dem keine Änderungen vorgenommen wurden) in die

Versionen einzufügen, die übersprungen werden, wenn ein Delta in einer neuen Freigabe erzeugt wird (z.B. wenn Delta 5.1 nach Delta 2.7 erstellt wurde, werden die Versionen 3 und 4 übersprungen). Das Fehlen des `n`-Flags verursacht, daß übersprungene Versionen vollständig leer sind. Das `j`-Flag gibt `get` die Möglichkeit, dieselbe Basis-SID gleichzeitig mehrfach zu bearbeiten. Das `l`-Flag definiert eine Liste von Freigaben, die gegen Bearbeitung gesperrt sind. Das `q`-Flag definiert den Ersatz für das `%Q%` Schlüsselwort. Das `z`-Flag wird in speziellen Schnittstellen-Programmen verwendet.

Kommentar

Beliebiger Text, der mit den Zeilen `@t` und `@T` geklammert ist, enthält üblicherweise eine Beschreibung des Verwendungszwecks der Datei.

Hauptteil

Der Hauptteil besteht aus Textzeilen und Steuerzeilen. Textzeilen beginnen im Gegensatz zu Steuerzeilen nicht mit dem Steuerzeichen. Es gibt drei Arten von Steuerzeilen: einfügen, löschen und Ende, dargestellt durch:

<code>@I</code>	<code>DDDDD</code>	(einfügen)
<code>@D</code>	<code>DDDDD</code>	(löschen)
<code>@E</code>	<code>DDDDD</code>	(Ende)

Die Ziffernfolge ist die Seriennummer, die dem Delta für die Steuerzeile entspricht.

SIEHE AUCH

`admin(1)`, `delta(1)`, `get(1)`, `prs(1)`.

BEZEICHNUNG

sdevice – System-Datei für Geräte-Spezifikationen

ÜBERSICHT

sdevice

BESCHREIBUNG

Die Datei `sdevice` enthält Informationen über die lokale System-Konfiguration für jedes Gerät, das in der Datei `mdevice` eingetragen ist. `sdevice` kann einen oder mehrere Einträge für jedes angegebene Gerät enthalten. Die Datei `sdevice` steht im Verzeichnis `/etc/conf/cf.d` und setzt sich aus Komponenten-Dateien im Verzeichnis `/etc/conf/sdevice.d` zusammen. Die Dateien in `/etc/conf/sdevice.d` sind die System-Datei-Komponenten, die entweder mit dem Grundsystem geliefert oder später durch `idinstall` installiert wurden.

Der Eintrag in der Datei `sdevice` kann vier verschiedene Formate annehmen. Das Format ist abhängig davon, ob in der Datei `mdevice` das Feld für die Treibercharakteristik den Buchstaben A, B oder P oder keinen dieser drei Buchstaben enthält.

Enthält das Feld für die Treibercharakteristik in der Datei `mdevice` keinen der drei Buchstaben A, B oder P, dann muß jeder Eintrag in der Datei `sdevice` die folgenden, durch Leerzeichen getrennten, Felder enthalten:

1. *gerätename*: Dieses Feld enthält den internen Namen des Treibers. Dieser muß mit einem der Namen im ersten Feld eines `mdevice`-Dateieintrags übereinstimmen.
2. *konfiguration*: Dieses Feld muß das Zeichen 'Y' enthalten, wenn das Gerät im Systemkern installiert werden soll. Für Test-Zwecke kann man 'N' eintragen, d.h. das Gerät soll nicht installiert werden.
3. *gerät*: Dieses Feld kann einen geräteabhängigen numerischen Wert enthalten, der normalerweise die Anzahl der untergeordneten Geräte auf einem Controller oder Pseudo-Gerät angibt. Der Wert muß innerhalb der Minimal- und Maximal-Werte liegen, die in den Feldern 7 und 8 des `mdevice`-Eintrags stehen.
4. *ipl*: Das Feld *ipl* gibt die ipl-Ebene des Geräts an, mit der die Interrupt-Behandlungsroutine des Treibers im neuen Systemkern laufen wird. Gültige Werte sind 0 bis 8. Wenn der Treiber keine Interrupt-Behandlungsroutine hat, ist eine 0 in dieses Feld zu setzen.

5. *typ*: Dieses Feld bezeichnet den Typ des Interrupt-Schemas, das vom Gerät verlangt wird. Zulässige Werte sind:
 - 0 – Das Gerät erfordert keine Interrupt-Leitung.
 - 1 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, verlangt jeder Controller einen eigenen Interrupt.
 - 2 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, benutzen alle Controller denselben Interrupt.
 - 3 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, teilen sich alle Controller denselben Interrupt. Mehrere Geräte-Treiber, die dieselbe ipl-Ebene haben, können diesen Interrupt gemeinsam benutzen.
6. *vektor*: Dieses Feld enthält die Nummer des Interrupt-Vektors, die das Gerät benutzt. Wenn das Feld *typ* den Wert 0 enthält (d.h. es ist kein Interrupt erforderlich), sollte dieses Feld ebenfalls mit einer 0 versehen sein. Es ist darauf zu achten, daß sich mehrere Geräte eine Interrupt-Nummer teilen können.
7. *SIOA*: Das Feld *SIOA* (Start I/O Address) enthält die Start-Adresse auf dem Ein-/Ausgabe-Bus, über den das Gerät kommuniziert. Die Start-Adresse muß zwischen 0x1 und 0xFFFFE liegen. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
8. *EIOA*: Das Feld *EIOA* (End I/O Address) enthält die End-Adresse auf dem Ein-/Ausgabe-Bus, über den das Gerät kommuniziert. Die End-Adresse muß größer sein als die Start-Adresse, die im siebten Feld angegeben ist. Die End-Adresse darf aber nicht größer als 0xFFFFE sein. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
9. *SCMA*: Das Feld *SCMA* (Start Controller Memory Address) wird von Controllern verwendet, die einen internen Speicher haben. Es bezeichnet die Start-Adresse dieses Speichers. Die Anfangsadresse ist 0X10000. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
10. *ECMA*: Das Feld *ECMA* (End Controller Memory Address) bezeichnet das Ende des internen Speichers für das Gerät. Sein Wert muß größer sein als der Wert des achten Felds. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.

Enthält das Feld für die Treibercharakteristik in der Datei `mdevice` den Buchstaben A, dann handelt es sich um ein autokonfiguriertes Gerät am MBAd (MB-I Adapter). Der Eintrag in der Datei `sdevice` muß dann die folgenden 12, durch Leerzeichen getrennte, Felder enthalten:

1. *gerätename*: Dieses Feld enthält den internen Namen des Treibers. Dieser muß mit einem der Namen im ersten Feld eines `mdevice`-Dateieintrags übereinstimmen.
2. *konfiguration*: Dieses Feld muß das Zeichen 'Y' enthalten, wenn das Gerät im Systemkern installiert werden soll. Für Test-Zwecke kann man 'N' eintragen, d.h. das Gerät soll nicht installiert werden.
3. *gerät*: Dieses Feld kann einen geräteabhängigen numerischen Wert enthalten, der normalerweise die Anzahl der untergeordneten Geräte auf einem Controller oder Pseudo-Gerät angibt. Der Wert muß innerhalb der Minimal- und Maximal-Werte liegen, die in den Feldern 7 und 8 des `mdevice`-Eintrags stehen.
4. *ipl*: Das Feld *ipl* gibt die ipl-Ebene des Geräts an, mit der die Interrupt-Behandlungsroutine des Treibers im neuen Systemkern laufen wird. Gültige Werte sind 0 bis 8. Wenn der Treiber keine Interrupt-Behandlungsroutine hat, ist eine 0 in dieses Feld zu setzen.
5. *typ*: Dieses Feld bezeichnet den Typ des Interrupt-Schemas, das vom Gerät verlangt wird. Zulässige Werte sind:
 - 0 – Das Gerät erfordert keine Interrupt-Leitung.
 - 1 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, verlangt jeder Controller einen eigenen Interrupt.
 - 2 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, benutzen alle Controller denselben Interrupt.
 - 3 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, teilen sich alle Controller denselben Interrupt. Mehrere Geräte-Treiber, die dieselbe ipl-Ebene haben, können diesen Interrupt gemeinsam benutzen. Treiber am MBAd sollten so implementiert sein, daß sie Interrupt-Leitungen gemeinsam benutzen. Die Anzahl der Interrupt-Vektoren ist nämlich sehr eingeschränkt.
 - 4 oder >6 – Das Gerät erfordert eine Interrupt-Leitung.
Das Gerät teilt sich diesen Interrupt mit allen typengleichen Controllern. Identische Controller müssen den Interrupt gemeinsam benutzen.
 - 5 – Das Gerät erfordert eine Interrupt-Leitung.
Dieser Typ entspricht in etwa Typ 6 und wird nur aus Gründen der Rückwärtskompatibilität unterstützt.
 - 6 – Das Gerät erfordert eine Interrupt-Leitung.
An den Boards kann ein Interrupt eingestellt werden, solange er nicht mit andersartigen Boards kollidiert. Der Treiber erkennt die Interruptnummer dynamisch. Dieser Interrupttyp ist nur für Geräte am MBAD implementiert.

6. *vektor*: Dieses Feld enthält die Nummer des Interrupt-Vektors, die das Gerät benutzt. Wenn das Feld *typ* den Wert 0 enthält (d.h. es ist kein Interrupt erforderlich), sollte dieses Feld ebenfalls mit einer 0 versehen sein. Mehrere Geräte können sich eine Interrupt-Nummer teilen. Ein Wert -1 in diesem Feld zusammen mit einem Wert 6 im Feld *typ* zeigen an, daß der Treiber die Nummer des Interrupt-Vektors dynamisch bestimmen muß.
7. *SIOA*: Das Feld *SIOA* (Start I/O Address) enthält die Start-Adresse im MB-I Speicher oder im Ein-/Ausgabe-Speicher, über den das Gerät kommuniziert. Sie wird abgebildet auf das Feld *mbad_dev->md_flags*. Die Startadresse muß innerhalb 0x1 und 0xFFFF liegen. Wird das *SIOA*-Feld nicht verwendet, ist es auf Null zu setzen.
8. *EIOA*: Das Feld *EIOA* (End I/O Address) enthält die End-Adresse im MB-I Speicher oder im Ein-/Ausgabe-Speicher, über den das Gerät kommuniziert. Die End-Adresse muß größer sein als der im siebten Feld angegebene Wert und darf nicht größer als 0xFFFF sein. Der Wert des Feldes wird nur daraufhin überprüft, aber nicht weiter ausgewertet. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
9. *SCMA*: Das Feld *SCMA* (Start Controller Memory Address) enthält die Start-Adresse im MB-I Speicher oder im Ein-/Ausgabe-Speicher, über den das Gerät kommuniziert. Sie wird abgebildet auf das Feld *mbad_dev->md_csr*. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
10. *ECMA*: Das Feld *ECMA* (End Controller Memory Address) enthält die End-Adresse im MB-I Speicher oder im Ein-/Ausgabe-Speicher, über den das Gerät kommuniziert. Die End-Adresse muß größer sein als der Wert des achten Felds. Der Wert des *ECMA*-Feldes wird nur daraufhin überprüft, aber nicht weiter ausgewertet. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
11. *maps*: Das Feld *maps* enthält die Anzahl von Maps, die der Treiber verwendet. Diese Maps dienen dazu, der DMA am MB-I Zugriff auf den Hauptspeicher zu ermöglichen. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
12. *MBAd*: (Nur MX500) Das Feld *MBAd* (MB-I Adapter) enthält die Nummer des MBAd, an dem das Gerät konfiguriert ist. Die Nummer liegt zwischen 0 und 2. Der Wert -1 bezeichnet einen beliebigen MBAd.

Enthält das Feld für die Treibercharakteristik in der Datei *mdevice* den Buchstaben P, dann handelt es sich um ein autokonfiguriertes Pseudo-Gerät ohne eigene Hardware. Der Eintrag in der Datei *sdevice* muß dann die folgenden 10, durch Leerzeichen getrennte, Felder enthalten:

1. *gerätename*: Dieses Feld enthält den internen Namen des Treibers. Dieser muß mit einem der Namen im ersten Feld eines *mdevice*-Dateieintrags übereinstimmen.
2. *konfiguration*: Dieses Feld muß das Zeichen 'Y' enthalten, wenn das Gerät im Systemkern installiert werden soll. Für Test-Zwecke kann man 'N' eintragen, d.h. das Gerät soll nicht installiert werden.

3. *gerät*: Dieses Feld kann einen geräteabhängigen numerischen Wert enthalten, der normalerweise die Anzahl der untergeordneten Geräte auf einem Controller oder Pseudo-Gerät angibt. Der Wert muß innerhalb der Minimal- und Maximal-Werte liegen, die in den Feldern 7 und 8 des `mdevice`-Eintrags stehen.
4. *ipl*: Dieses Feld hat keine Bedeutung für Pseudo-Geräte und sollte auf Null gesetzt sein.
5. *typ*: Dieses Feld hat keine Bedeutung für Pseudo-Geräte und sollte auf Null gesetzt sein.
6. *vektor*: Dieses Feld hat keine Bedeutung für Pseudo-Geräte und sollte auf Null gesetzt sein.
7. *SIOA*: Dieses Feld hat keine Bedeutung für Pseudo-Geräte und sollte auf -1 gesetzt sein.
8. *EIOA*: Dieses Feld hat keine Bedeutung für Pseudo-Geräte und sollte auf -1 gesetzt sein.
9. *SCMA*: Dieses Feld hat keine Bedeutung für Pseudo-Geräte und sollte auf -1 gesetzt sein.
10. *ECMA*: Dieses Feld hat keine Bedeutung für Pseudo-Geräte und sollte auf -1 gesetzt sein.

(Nur MX500) Enthält das Feld für die Treibercharakteristik in der Datei `mdevice` den Buchstaben B, dann handelt es sich um ein Gerät am SCED-Board. Der Eintrag in der Datei `sdevice` muß dann die folgenden 15, durch Leerzeichen getrennte, Felder enthalten:

1. *gerätename*: Dieses Feld enthält den internen Namen des Treibers. Dieser muß mit einem der Namen im ersten Feld eines `mdevice`-Dateieintrags übereinstimmen.
2. *konfiguration*: Dieses Feld muß das Zeichen 'Y' enthalten, wenn das Gerät im Systemkern installiert werden soll. Für Test-Zwecke kann man 'N' eintragen, d.h. das Gerät soll nicht installiert werden.
3. *gerät*: Dieses Feld kann einen geräteabhängigen numerischen Wert enthalten, der normalerweise die Anzahl der untergeordneten Geräte auf einem Controller oder Pseudo-Gerät angibt. Der Wert muß innerhalb der Minimal- und Maximal-Werte liegen, die in den Feldern 7 und 8 des `mdevice`-Eintrags stehen.
4. *bin*: Das Feld *bin* gibt die ipl-Ebene des Geräts an, mit der die Interrupt-Behandlungsroutine des Treibers im neuen Systemkern laufen wird. Gültige Werte sind 0 bis 8. Wenn der Treiber keine Interrupt-Behandlungsroutine hat, ist eine 0 in dieses Feld zu setzen.
5. *typ*: Dieses Feld bezeichnet den Typ des Interrupt-Schemas, das vom Gerät verlangt wird. Zulässige Werte sind:
 - 0 – Das Gerät erfordert keine Interrupt-Leitung.

- 1 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, verlangt jeder Controller einen eigenen Interrupt.
 - 2 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, benutzen alle Controller denselben Interrupt.
 - 3 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, teilen sich alle Controller denselben Interrupt. Mehrere Geräte-Treiber, die dieselbe ipl-Ebene haben, können diesen Interrupt gemeinsam benutzen. Verschiedene Controller können sich denselben Interrupt-Vektor teilen, müssen es aber nicht. Dieser Typ von Implementierung wird für Treiber am SCED-Board empfohlen.
 - 4 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, teilen sich alle Controller denselben Interrupt. Mehrere Geräte-Treiber, die dieselbe ipl-Ebene haben, können diesen Interrupt gemeinsam benutzen. Typengleiche Controller teilen sich denselben Interrupt-Vektor.
6. *vektor*: Dieses Feld hat keine Bedeutung und sollte auf Null gesetzt sein.
 7. *SIOA*: Dieses Feld hat keine Bedeutung und sollte auf Null gesetzt sein.
 8. *EIOA*: Dieses Feld hat keine Bedeutung und sollte auf Null gesetzt sein.
 9. *SCMA*: Dieses Feld hat keine Bedeutung und sollte auf Null gesetzt sein.
 10. *ECMA*: Dieses Feld hat keine Bedeutung und sollte auf Null gesetzt sein.
 11. *rq*: Das *rq*-Feld (request queue length) gibt die Länge der Auftragswarteschlange für das Geräteprogramm an.
 12. *dq*: Das *dq*-Feld (done queue length) gibt an, wieviele Aufträge aus der Auftragschlange des Geräteprogramms bereits ausgeführt sind.
 13. *SCED*: Das *SCED*-Feld enthält die Nummer des SCED-Boards, an dem der Treiber konfiguriert ist. Die Nummer muß zwischen 0 und 3 liegen. Der Wert -1 bezeichnet ein beliebiges SCED-Board.
 14. *target*: Das *target*-Feld wird für SCSI-Geräte verwendet. Es bezeichnet die Targetnummer, die zwischen 0 und 7 liegen kann. Der Wert -1 in diesem Feld bedeutet, daß der Treiber die Targetnummer festlegt.
 15. *lun*: Das *lun*-Feld (logical unit) wird für SCSI-Geräte verwendet. Bei anderen Geräten ist dies die Nummer eines untergeordneten Geräts. Es bezeichnet die Einheitsnummer, die zwischen 0 und 7 liegen kann. Der Wert -1 in diesem Feld bedeutet, daß der Treiber die Einheit festlegt.

sdevice(4)

(MX300, MX500)

sdevice(4)

BEISPIEL

Das folgende Beispiel zeigt *sdevice* Einträge auf der MX500 für einen NCR Host Adapter (ADP32 Controller) und für ein EXABYTE 8mm Video Bandgerät.

*	conf	unit	ipl	type	vector	sioa	eioa	scma	ecma	maps	
exa	Y	0	0	0	0	-1	-1	-1	-1	0	
*	conf	unit	ipl	type	vector	sioa	eioa	scma	ecma	maps	MBAd
adp32	Y	0	5	6	-1	0	0	6000	6000	36	-1

EBENE

Multiprozessor-Erweiterung

SIEHE AUCH

mdevice(4).

idinstall(1m) im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

sdevice – System-Datei für Geräte-Spezifikationen

ÜBERSICHT

sdevice

BESCHREIBUNG

Die Datei `sdevice` enthält Informationen über die lokale System-Konfiguration für jedes Gerät, das in der Datei `mdevice` eingetragen ist. `sdevice` kann einen oder mehrere Einträge für jedes angegebene Gerät enthalten. Die Datei `sdevice` steht im Verzeichnis `/etc/conf/cf.d` und setzt sich aus Komponenten-Dateien im Verzeichnis `/etc/conf/sdevice.d` zusammen. Die Dateien in `/etc/conf/sdevice.d` sind die System-Datei-Komponenten, die entweder mit dem Grundsystem geliefert oder später durch `idinstall` installiert wurden.

Jeder Eintrag in der Datei `sdevice` muß die folgenden, durch Leerzeichen getrennten, Felder enthalten:

1. *gerätename*: Dieses Feld enthält den internen Namen des Treibers. Dieser muß mit einem der Namen im ersten Feld eines `mdevice`-Dateieintrags übereinstimmen.
2. *konfiguration*: Dieses Feld muß das Zeichen 'Y' enthalten, wenn das Gerät im Systemkern installiert werden soll. Für Test-Zwecke kann man 'N' eintragen, d.h. das Gerät soll nicht installiert werden.
3. *gerät*: Dieses Feld kann einen geräteabhängigen numerischen Wert enthalten, der normalerweise die Anzahl der untergeordneten Geräte auf einem Controller oder Pseudo-Gerät angibt. Der Wert muß innerhalb der Minimal- und Maximal-Werte liegen, die in den Feldern 7 und 8 des `mdevice`-Eintrags stehen.
4. *ipl*: Das Feld *ipl* gibt die ipl-Ebene des Geräts an, mit der die Interrupt-Behandlungsroutine des Treibers im neuen Systemkern laufen wird. Gültige Werte sind 0 bis 8. Wenn der Treiber keine Interrupt-Behandlungsroutine hat, ist eine 0 in dieses Feld zu setzen.

5. *typ*: Dieses Feld bezeichnet den Typ des Interrupt-Schemas, das vom Gerät verlangt wird. Zulässige Werte sind:
 - 0 – Das Gerät erfordert keine Interrupt-Leitung.
 - 1 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, verlangt jeder Controller einen eigenen Interrupt.
 - 2 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, benutzen alle Controller denselben Interrupt.
 - 3 – Das Gerät erfordert eine Interrupt-Leitung.
Wenn der Treiber mehr als einen Hardware-Controller unterstützt, teilen sich alle Controller denselben Interrupt. Mehrere Geräte-Treiber, die dieselbe ipl-Ebene haben, können diesen Interrupt gemeinsam benutzen.
6. *vektor*: Dieses Feld enthält die Nummer des Interrupt-Vektors, die das Gerät benutzt. Wenn das Feld *typ* den Wert 0 enthält (d.h. es ist kein Interrupt erforderlich), sollte dieses Feld ebenfalls mit einer 0 versehen sein. Es ist darauf zu achten, daß sich mehrere Geräte eine Interrupt-Nummer teilen können.
7. *SIOA*: Das Feld *SIOA* (Start I/O Address) enthält die Start-Adresse auf dem Ein-/Ausgabe-Bus, über den das Gerät kommuniziert. Die Start-Adresse muß zwischen 0x1 und 0xFFFFE liegen. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
8. *EIOA*: Das Feld *EIOA* (End I/O Address) enthält die End-Adresse auf dem Ein-/Ausgabe-Bus, über den das Gerät kommuniziert. Die End-Adresse muß größer sein als die Start-Adresse, die im siebten Feld angegeben ist. Die End-Adresse darf aber nicht größer als 0xFFFFE sein. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
9. *SCMA*: Das Feld *SCMA* (Start Controller Memory Address) wird von Controllern verwendet, die einen internen Speicher haben. Es bezeichnet die Start-Adresse dieses Speichers. Die Anfangsadresse ist 0X10000. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.
10. *ECMA*: Das Feld *ECMA* (End Controller Memory Address) bezeichnet das Ende des internen Speichers für das Gerät. Sein Wert muß größer sein als der Wert des achten Felds. Wird dieses Feld nicht verwendet, ist es auf Null zu setzen.

EBENE

Multiprozessor-Erweiterung

SIEHE AUCH*mdevice(4).**idinstall(1m)* im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

services – Internet-Dienste und Aliase

ÜBERSICHT

/etc/inet/services

BESCHREIBUNG

Die Datei `services` enthält einen Eintrag für jeden Dienst, der durch DARPA Internet verfügbar ist. Jeder Eintrag besteht aus einer Zeile in der Form:

service-name *anschluß / protokoll* *alias*

service-name Dies ist der offizielle Internet Service-Name.

anschluß / protokoll Dieses Feld ist aus der Anschlußnummer und dem Protokoll zusammengesetzt, durch das der Service zur Verfügung gestellt wird (zum Beispiel `512/tcp`).

alias Dies ist eine Liste möglicher Namen, durch die der Service angefordert werden könnte.

Die Felder können durch eine beliebige Zahl von Leer-/Tabulatorzeichen getrennt sein. Ein '#' (Nummernzeichen) bezeichnet den Anfang eines Kommentars, der bis zum Zeilenende reicht. Der Inhalt von Kommentarzeilen wird nicht interpretiert.

Mit Ausnahme von Feldtrennzeichen, Neue-Zeile-Zeichen oder Kommentarzeichen können Service-Namen jedes druckbare Zeichen enthalten

DATEIEN

/etc/inet/services

SIEHE AUCH

`getservent(3N)`, `inetd.conf(4)`.

HINWEIS

Anstelle einer statischen Datei sollte ein Name-Server verwendet werden.

BEZEICHNUNG

sfsys – Typen von Dateisystemen

ÜBERSICHT

sfsys

BESCHREIBUNG

Die Datei `sfsys` enthält lokale System-Informationen über jeden Dateisystem-Typ, der in der Datei `mfsys` angegeben ist. Sie steht in dem Verzeichnis `/etc/conf/cf.d` und enthält einen einzeiligen Eintrag für jeden Dateisystem-Typ in der Datei `mfsys`. Die Datei `sfsys` setzt sich aus Komponenten-Dateien in dem Verzeichnis `/etc/conf/sfsys.d` zusammen. Jede Zeile in dieser Datei ist ein durch Leerzeichen getrennter Satz von Feldern folgenden Inhalts:

1. *name*: Dieses Feld enthält den internen Namen des Dateisystem-Typs (z.B. DUFST, S51K). Dieser Name ist bis zu 32 Zeichen lang und kann aus allen numerischen Zeichen und Großbuchstaben bestehen.
2. *Y/N*: Dieses Feld enthält entweder den Großbuchstaben 'Y' (für "yes") oder den Großbuchstaben 'N' (für "no"), um anzuzeigen, ob der genannte Dateisystem-Typ in den nächsten Betriebssystemkern, der erzeugt wird, konfiguriert werden soll.

SIEHE AUCH

`mfsys(4)`.

`idinstall(1M)`, `idbuild(1M)` im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

shadow – Shadow-Paßwortdatei

ÜBERSICHT

/etc/shadow

BESCHREIBUNG

/etc/shadow ist eine ASCII Systemdatei mit eingeschränktem Zugriff. Die Felder für jeden Benutzereintrag sind durch Doppelpunkte voneinander getrennt. Jeder Benutzereintrag steht in einer eigenen Zeile. Im Gegensatz zu der Datei /etc/passwd hat /etc/shadow keine allgemeine Leseberechtigung.

/etc/shadow enthält folgende Felder:

<i>benutzername</i>	Benutzernummer (UID).
<i>paßwort</i>	Dieses Feld enthält ein 13 Zeichen langes, verschlüsseltes Paßwort für den Benutzer und, wenn diese Anmeldung nicht zugänglich ist, eine <i>lock</i> -Zeichenkette. Ist kein Eintrag vorhanden, gibt es kein Paßwort für die Anmeldung.
<i>letzte änderung</i>	Die Anzahl von Tagen zwischen dem 1. Januar 1970 und dem Datum, an dem das Paßwort zuletzt verändert wurde.
<i>minimum</i>	Die Mindest-Anzahl von Tagen, die zwischen Paßwortänderungen erforderlich ist.
<i>maximum</i>	Die Höchstzahl von Tagen, die das Paßwort gültig ist.
<i>warnung</i>	Die Anzahl von Tagen, die zwischen der Warnung an den Benutzer und dem Ablauf der Gültigkeit des Paßworts liegt.
<i>inaktiv</i>	Die höchste zulässige Zahl an Tagen, während derer diese Benutzerkennung nicht benutzt wurde.
<i>verfall</i>	Ein absolutes Datum, das das Verfallsdatum aller Anmeldungen festlegt.
<i>schalter</i>	Reserviert für zukünftige Anwendungen. Wird gegenwärtig nicht benutzt und ist auf Null gesetzt.

Das verschlüsselte Paßwort besteht aus 13 Zeichen, die aus einem 64-Zeichen Alphabet (., /, 0–9, A–Z, a–z) gewählt werden.

Zur Aktualisierung dieser Datei dienen die Befehle `passwd`, `useradd`, `usermod` oder `userdel`.

DATEIEN

/etc/shadow

SIEHE AUCH

`useradd(1M)`, `usermod(1M)`, `userdel(1M)`, `passwd(4)`
`putspent(3X)`, `getspent(3X)` im Referenzhandbuch für Programmierer
`login(1)`, `passwd(1)` in den Kommandos.

BEZEICHNUNG

sharetab – Tabelle gemeinsam über Netz benutzter Dateisysteme

ÜBERSICHT

/etc/dfs/sharetab

BESCHREIBUNG

sharetab befindet sich im Verzeichnis /etc/dfs und enthält eine Tabelle der lokalen Ressourcen, welche durch den Befehl share gemeinsam benutzt werden.

Jede Zeile der Datei besteht aus den folgenden Feldern:

pfadname ressource ds-typ spez_optionen beschreibung

pfadname Gibt den Pfadnamen der gemeinsamen Ressource an.

ressource Gibt den symbolischen Namen an, über den ferne Rechner auf die Ressource zugreifen können.

ds-typ Gibt den Dateisystem-Typ der gemeinsamen Ressource an.

spez_optionen Gibt die für den Dateisystem-Typ spezifischen Optionen an, die dem Befehl share übergeben wurden, wenn die Ressource gemeinsam benutzt wurde.

beschreibung Ist eine Beschreibung der gemeinsamen Ressource, die bei der gemeinsamen Benutzung der Ressource durch den Systemverwalter geliefert wird.

SIEHE AUCH

share(1M)

BEZEICHNUNG

strcf – STREAMS-Konfigurationsdatei für STREAMS TCP/IP

ÜBERSICHT

/etc/inet/strcf

BESCHREIBUNG

/etc/inet/strcf enthält die Prozedur, die durch `slink(1M)` ausgeführt wird, um die Konfigurations-Operationen für STREAMS, die für STREAMS TCP/IP erforderlich sind, durchzuführen.

Die Standard-Datei /etc/inet/strcf enthält mehrere Funktionen, die diverse Konfigurations-Operationen ausführen, darunter auch eine Beispiel-Funktion für `boot`. Normalerweise muß ausschließlich die `boot`-Funktion verändert werden, um die Konfiguration für eine vorgegebene Installation anzupassen. In einigen Fällen kann es jedoch notwendig sein, existierende Funktionen zu ändern oder neue Funktionen hinzuzufügen.

Die folgenden Funktionen führen grundlegende Verknüpfungsoperationen durch:

Die Funktion `tp` wird verwendet, um die Verbindung zwischen einem Transport-Provider, wie TCP und IP, einzurichten.

```
#
# tp - konfiguriere Transport-Provider (z.B. tcp, udp, icmp)
# Verwendung: tp devname
#
tp {
    p = open $1
    ip = open /dev/ip
    link p ip
}
```

Die Funktion `linkint` verbindet die angegebenen STREAMS und führt eine `sifname`-Operation mit dem vorgegebenen Namen aus.

```
#
# linkint - link Schnittstelle zu ip oder arp
# Verwendung: linkint top bottom ifname
#
linkint {
    x = link $1 $2
    sifname $1 x $3
}
```

Die Funktion `aplinkint` führt für eine Schnittstelle, die das Modul `app` benutzt, dieselbe Funktion aus wie `linkint`.

```
#
# aplinkint - ähnlich linkint, aber app wurde in den Stapel
# gestellt auf dev
# Verwendung: aplinkint top bottom ifname
#
aplinkint {
    push $2 app
    linkint $1 $2 $3
}
```

Die folgenden Funktionen werden verwendet, um verschiedene Arten von Ethernet-Schnittstellen zu konfigurieren:

Die Funktion `uenet` wird verwendet, um eine Ethernet-Schnittstelle für einen geklonten Gerätetreiber zu konfigurieren, der den `unit select ioctl`-Aufruf benutzt, um die gewünschte Schnittstelle auszuwählen. Der Schnittstellenname wird durch Verketten des angegebenen Präfix und der Gerätenummer gebildet.

```
#
# uenet - konfiguriere Ethernet-Schnittstelle für geklonten Treiber,
#         der unit select verwendet
# Verwendung: uenet ip-fd devname ifprefix unit
#
uenet {
    ifname = strcat $3 $4
    dev = open $2
    unitselect dev $4
    aplinkint $1 dev ifname
    dev = open $2
    unitselect dev $4
    arp = open /dev/arp
    linkint arp dev ifname
}
```

Die Funktion `denet` führt dieselbe Funktion wie `uenet` aus, mit der Ausnahme, daß `DL_ATTACH` verwendet wird anstelle von `unit select`.

```
#
# denet - konfiguriere Ethernet-Schnittstelle für geklonten Treiber
#         der DL_ATTACH verwendet
# Verwendung: denet ip-fd devname ifprefix unit
#
denet {
    ifname = strcat $3 $4
    dev = open $2
    dlattach dev $4
    aplinkint $1 dev ifname
    dev = open $2
    dlattach dev $4
    arp = open /dev/arp
    linkint arp dev ifname
}
```

Die Funktion `cenet` wird verwendet, um eine Ethernet-Schnittstelle für einen geklonten Gerätetreiber, der verschiedene Geräteklassennummern für jede Schnittstelle verwendet, zu konfigurieren. Der Gerätenamen-Präfix und der Gerätenummer bilden den Schnittstellen-Namen. Der Schnittstellen-Name wird in einer ähnlichen Weise unter Verwendung des Schnittstellen-Namen-Präfix gebildet.

```
#
# cenet - konfiguriere Ethernet-Schnittstelle für geklonten
#         Treiber mit einer Geräteklasse je Schnittstelle
# Verwendung: cenet ip-fd devprefix ifprefix unit
#
cenet {
    devname = strcat $2 $4
    ifname = strcat $3 $4
    dev = open devname
    aplinkint $1 dev ifname
    dev = open devname
    arp = open /dev/arp
    linkint arp dev ifname
}
```

Die Funktion `senet` wird verwendet, um eine Ethernet-Schnittstelle für einen nicht geklonten Gerätetreiber zu konfigurieren. Für IP und ARP müssen zwei verschiedene Geräteknotten angegeben sein.

```
# senet - konfiguriere Ethernet-Schnittstelle für nicht geklonten
#         Treiber
# Verwendung: senet ip-fd ipdevname arpdevname ifname
#
senet {
    dev = open $2
    aplinkint $1 dev $4
    dev = open $3
    arp = open /dev/arp
    linkint arp dev $4
}
```

Die Funktion `senetc` ist ähnlich wie `senet`, mit der Ausnahme, daß sie die Spezifikation eines Konvergenz-Moduls erlaubt, das mit dem Ethernet-Treiber benutzt wird (wie z.B. für den 3B2 emd Treiber).

```
#
# senetc - konfiguriere Ethernet-Schnittstelle für nicht geklonten
#          Treiber, der Konvergenz-Modul verwendet
# Verwendung: senetc ip-fd convergence ipdevname arpdevname ifname
#
senetc {
    dev = open $3
    push dev $2
    aplinkint $1 dev $5
    dev = open $4
    push dev $2
    arp = open /dev/arp
    linkint arp dev $5
}
```

Die Funktion `loopback` wird verwendet, um die Loopback-Schnittstelle zu konfigurieren.

```
#
# loopback - konfiguriere Loopback-Einheit
# Verwendung: loopback ip-fd
#
loopback {
    dev = open /dev/Schleife
    linkint $1 dev lo0
}
```

Die Funktion `slip` wird verwendet, um eine SLIP-Schnittstelle zu konfigurieren. Diese Funktion wird beim Hochfahren normalerweise nicht ausgeführt. Stattdessen führt der Befehl `slattach(1M)` `slink` aus und spezifiziert `slip` in der Befehlszeile.

```
#
# slip - konfiguriere slip-Schnittstelle
# Verwendung: slip unit
#
slip {
    ip = open /dev/ip
    s = open /dev/slip
    ifname = strcat sl $1
    unitssel s $1
    linkint ip s ifname
}
```

Die Funktion `boot` wird standardmäßig aufgerufen, wenn `slink` ausgeführt wird. Normalerweise muß ausschließlich der Abschnitt `interfaces` und eventuell der Abschnitt `queue params` für eine vorgegebene Installation angepaßt werden. Für die verschiedenen Ethernet-Treiberarten gibt es Beispiele.

strcf(4)

strcf(4)

```
#
# boot - Ladezeit-Konfiguration
#
boot {
    #
    # Warteschlangen Parameter
    #
    initqp /dev/udp rq 8192 40960
    initqp /dev/ip muxrq 8192 40960 rq 8192 40960
    #
    # Transport
    #
    tp /dev/tcp
    tp /dev/udp
    tp /dev/icmp
    tp /dev/rawip
}
```

DATEIEN

/etc/inet/strcf

SIEHE AUCH

slattach(1M), slink(1M).

BEZEICHNUNG

strptime – Sprachspezifische Zeichenketten

BESCHREIBUNG

Für jede Region kann es eine druckbare Datei geben, die die regionalen Formate für Datum und Zeit festlegt. Diese Dateien müssen in dem Verzeichnis `/usr/lib/locale/<locale>/LC_TIME` abgelegt sein. Diese Dateien haben folgenden Inhalt:

1. Abgekürzte Monatsnamen (sortiert)
2. Monatsnamen (sortiert)
3. Abgekürzte Wochentagsnamen (sortiert)
4. Wochentagsnamen (sortiert)
5. Voreingestellte Zeichenketten, die die Formate für die lokale Zeit (%X) und das lokale Datum (%x) spezifizieren.
6. Standardformat für cftime, falls das Argument für cftime 0 oder NULL ist.
7. AM (ante meridiem) Zeichenfolge
8. PM (post meridiem) Zeichenfolge

Jede Zeichenfolge steht in einer eigenen Zeile. Alle Zwischenräume sind wichtig! Die angegebene Reihenfolge der Zeichenketten in der obigen Liste muß in der Datei beibehalten werden.

BEISPIEL

`/usr/lib/locale/C/LC_TIME`

```
Jan
Feb
...
Januar
Februar
...
So
Mo
...
Sonntag
Montag
...
%H:%M:%S
%m/%d/%y
%eine %b %d %T %Z %Y
AM
PM
```

DATEIEN

`/usr/lib/locale/<locale>/LC_TIME`

strftime(4)

strftime(4)

SIEHE AUCH

`ctime(3C), setlocale(3C), strftime(3C).`

BEZEICHNUNG

stune – Dateiformat

ÜBERSICHT

stune

BESCHREIBUNG

Die Datei `stune` enthält lokale Systemeinstellungen für einstellbare Parameter. Die Parametereinstellungen in dieser Datei ersetzen die Standardeinstellungen, die in der Datei `mtune` angegeben sind, falls die neuen Werte innerhalb des Gültigkeitsbereichs für die Parameter liegen, der in `mtune` angegeben ist. Die Datei enthält eine Zeile für jeden Parameter, der neu gesetzt werden soll. Jede Zeile enthält zwei durch Leerzeichen getrennte Felder:

1. *external name*: Dies ist der externe Name der einstellbaren Parameter, der in der Datei `mtune` verwendet wird.
2. *wert*: Dieses Feld enthält den neuen Wert für die einstellbaren Parameter.

Die Datei `stune` befindet sich normalerweise in `/etc/conf/cf.d`. Jedoch sollte ein Benutzer oder ein Zusatzpaket die Datei `mtune` niemals direkt bearbeiten. Stattdessen sollte der Befehl `idtune` verwendet werden.

Damit die neuen Werte in Kraft treten, muß der Systemkern neu erzeugt und das System anschließend neu gestartet werden.

SIEHE AUCH

`mtune(4)`.
`idbuild(1M)`, `idtune(1M)`, `iddefaults(1M)`

BEZEICHNUNG

su – Optionen für su

ÜBERSICHT

/etc/default/su

BESCHREIBUNG

Optionen für das Programm su können in der Datei /etc/default/su über Schlüsselwörter eingestellt oder geändert werden. Die folgenden Schlüsselwörter werden von su erkannt:

SULOG=*dateiname*

Gibt die Protokolldatei für (erfolgreiche und erfolglose) su-Aufrufe an.

CONSOLE=*gerät*

Versucht ein Benutzer sich an einem Gerät, das nicht als Konsole festgelegt ist, unter root anzumelden, erfolgt auf der Konsole eine entsprechende Meldung, um den dortigen Benutzer darüber zu informieren.

PATH=*dateiverzeichnisliste*

Der Suchpfad des Benutzers wird auf diese Zeichenkette eingestellt, wenn er mit su in eine andere Kennung wechselt, die nicht root ist. Standardmäßig ist dies ":/bin:/usr/bin".

SUPATH=*dateiverzeichnisliste*

Der Suchpfad des Benutzers wird auf diese Zeichenkette eingestellt, wenn er mit su zur root wechselt. Der Standard ist ":/bin:/etc:/usr/bin".

DATEIEN

/etc/default/su

BEZEICHNUNG

term – Format der übersetzten Datei term

ÜBERSICHT

/usr/lib/share/terminfo

BESCHREIBUNG

Die Beschreibungen der übersetzten Datei terminfo(4) stehen in dem Verzeichnis /usr/share/lib/terminfo. Um eine lineare Suche eines sehr großen Verzeichnisses zu vermeiden, wird ein zweistufiges Schema verwendet: /usr/share/lib/terminfo/c/name wobei name der Name des Terminals ist und c das erste Zeichen von name. Auf diese Weise kann 97801 in der Datei /usr/share/lib/terminfo/9/97801 gefunden werden. Gleiche Namen für dasselbe Terminal sind durch mehrere Verweise auf dieselbe übersetzte Datei realisiert.

So wird erreicht, daß das Format auf jeder Hardware dasselbe ist. Es wird von einem 8-Bit-Byte ausgegangen. Es gibt jedoch keine Vorschriften über Byte-Reihenfolge oder Vorzeichen-Erweiterung. Auf diese Weise können die binären terminfo-Dateien auf andere Hardware mit 8-Bit-Bytes übertragen werden.

Ganze Zahlen vom Typ short integer werden in zwei 8-Bit-Bytes gespeichert. Das erste Byte (LSB) enthält die niederwertigen 8 Bits des Werts und das zweite Byte (MSB) enthält die höherwertigen 8 Bits (somit ist der dargestellte Wert $256 \cdot \text{MSB} + \text{LSB}$). Der Wert -1 wird dargestellt durch 0377, 0377, der Wert -2 wird dargestellt durch 0376, 0377; andere negative Werte sind unzulässig. -1 bedeutet generell, daß eine Eigenschaft dieses Terminals fehlt. -2 bedeutet, daß die Eigenschaft in der terminfo- Kommandodatei ungültig gemacht wurde und auch als fehlend zu betrachten ist.

Die übersetzte Datei wird von den Beschreibungen der Terminals in der Quelldatei erzeugt (siehe die -I Option von infocmp). Hierfür wird der terminfo-Compiler tic benutzt und die Quelldatei von der Routine setupterm gelesen (siehe curses(3X)). Die Datei wird in sechs Teile, in der folgenden Reihenfolge, aufgeteilt: die Kopfzeile, Terminalnamen, boolesche Schalter, Zahlen, Zeichenketten und Zeichenkettentabelle.

Der Kopfbereich steht am Anfang der Datei. Dieser Abschnitt enthält sechs ganze Zahlen vom Typ short integer in dem unten beschriebenen Format. Diese ganzen Zahlen sind:

1. die Dateikennung (oktal 0432);
2. die Größe des Namensbereichs in Bytes;
3. die Anzahl der Bytes in dem booleschen Bereich;
4. die Anzahl der ganzen Zahlen vom Typ short integer in dem Zahlenbereich;
5. die Anzahl der Offsets (short integer) in dem Zeichenkettenbereich;
6. die Größe der Zeichenkettentabelle in Bytes.

Als nächstes folgt der Terminal-Namensbereich. Er enthält die erste Zeile der terminfo-Beschreibung, der die verschiedenen Namen für das Terminal auflistet, getrennt durch das Pipe-Symbol (|) (siehe term(5)). Der Abschnitt wird durch das ASCII-Zeichen NUL beendet.

Die booleschen Schalter belegen je ein Byte. Dieses Byte ist entweder 0 oder 1, je nachdem, ob der Schalter gesetzt ist oder nicht. Der Wert 2 bedeutet, daß die Option außer Kraft gesetzt ist. Die Eigenschaften sind in derselben Reihenfolge wie die Datei `<term.h>`.

Um sicherzustellen, daß der Zahlenbereich mit einem geraden Byte-Offset beginnt, wird zwischen dem booleschen Bereich und dem Zahlenbereich gegebenenfalls ein Nullbyte eingefügt. Alle ganzen Zahlen vom Typ `short integer` sind auf eine Datenwort-Grenze dieses Typs ausgerichtet.

Der Zahlenbereich ist dem Abschnitt der booleschen Schalter ähnlich. Jede Eigenschaft verwendet zwei Bytes und wird als eine ganze Zahl vom Typ `short integer` gespeichert. Wenn der dargestellte Wert -1 oder -2 ist, wird angenommen, daß diese Eigenschaft fehlt.

Der Zeichenkettenbereich ist auch ähnlich. Jede Eigenschaft wird als eine ganze Zahl vom Typ `short integer` im selben Format wie oben gespeichert. Ein Wert von -1 oder -2 bedeutet, daß die Eigenschaft fehlt. Andernfalls wird der Wert als Distanzadresse (offset) vom Anfang der Zeichenketten-Tabelle verwendet. Sonderzeichen in ``X` oder `\c` Schreibweise werden in ihrer übersetzten Form gespeichert, nicht in der Druckdarstellung. Füllzeichen (`$<nn>`) und Parameter-Informationen (`%x`) werden unverändert in nicht übersetzter Form gespeichert.

Der letzte Abschnitt ist die Tabelle der Zeichenketten. Sie enthält alle Werte von Zeichenketten-Eigenschaften, auf die im Zeichenketten-Bereich Bezug genommen wird. Jede Zeichenkette wird durch ein Nullzeichen abgeschlossen.

Es ist zu beachten, daß `setupterm` andere Eigenschaften als die gegenwärtig in der Datei vorhandenen erwarten kann. Entweder wurde die Datenbank aktualisiert seit `setupterm` neu übersetzt wurde (resultierend in zusätzlichen unerkannten Einträgen in der Datei), oder das Programm wurde erst neu übersetzt, nachdem die Datenbank aktualisiert wurde (resultierend in fehlenden Einträgen). Die Routine `setupterm` muß auf beide Eigenschaften vorbereitet sein - deshalb sind die Zahlen und Größen enthalten. Auch müssen neue Eigenschaften immer am Ende der Listen von booleschen, Zahl- und Zeichenketten-Eigenschaften hinzugefügt werden.

Hier ist ein Beispiel einer Terminal-Information, wie sie sich auf dem 97801 Terminal befindet. Sie kann durch den Befehl `infoamp -I 97801` ausgegeben werden.

term(4)

term(4)

```
standard|97801|97808,
am,
cols#80, lines#24,
acsc=+K,L.N-Mf?jEkClBmDnJqAtFuGvIwHx@~, , bel=^G,
blink=E[5m, cbt=E[Z, civis=E[6p, clear=E[HE[2J,
cnorm=E[7p, cr=\r, csr=E[%i%pl%d;%p2%dr,
cub=E[%p1%dD, cubl=\b, cud=E[%p1%dB, cudl=E[B,
cuf=E[%p1%dC, cufl=E[C, cup=E[%i%pl%d;%p2%dH,
cuu=E[%p1%dA, cuul=E[A, dch=E[%p1%dP, dchl=E[P,
dim=E[2m, dl=E[%p1%dM, dll=E[M, ed=E[0J, el=E[0K,
home=E[H, ht=\t, ich=E[%p1%d@, ichl=E[@,
il=E[%p1%dL, ill=E[L, ind=E[S, indn=E[%p1%dS,
invis=E[8m, kbs=\b, kcbt=E[Z, kcubl=E[D,
kcudl=E[B, kcufl=E[C, kcuul=E[A, kdchl=E[P,
kdll=E[M, kdw=Ep, kfl=E@, kfl0=EJ, kfl1=EK,
kfl2=EL, kfl3=EM, kfl4=EN, kfl5=EO, kfl6=EP,
kfl7=E0, kfl8=E_, kfl9=Ed, kf2=EA, kf20=ET,
kf21=EV, kf22=EX, kf23=Es, kf24=E;, kf25=E",
kf26=E#, kf27=E$, kf28=E%, kf29=E&, kf3=EB,
kf30=E', kf31=E<, kf32=E=, kf33=E*, kf34=E+,
kf35=E,, kf36=E-, kf37=E., kf38=E/, kf39=E1,
kf4=EC, kf40=E2, kf41=E3, kf42=EU, kf43=EW,
kf44=EY, kf5=ED, kf6=EF, kf7=EG, kf8=EH, kf9=EI,
khlp=E>, khome=E[H, kichl=E[@, kill=E[L,
kind=E[T, kiw=Eo, kmenu=\n, kmode=E4, kprrt=Eg,
kri=E[S, krst=Em, krst=Em, lf0=E>, lf2=^D,
nel=EE, pctrm=USE_TERM:s97801pc:, rc=E[u, rev=E[7m,
ri=E[T, rin=E[%p1%dT, rmacs=^O, rmso=E[0m,
rmul=E[0m, sbt=E9, sc=E[s,
sgr=E[0%?%p1%t;7%;%?%p2%t;4%;%?%p3%t;7%;%?%p4%t;5%;%?%p5%t;
2%;%?%p6%t;7%;%?%p7%t;8m%em%;%?%p9%t^N%e^O%;,
sgr0=E[0m^O, sht=E:, smacs=^N,
smcup=E[1;24rE[m^OE)w, smso=E[7m, smul=E[4m,
```

term(4)

term(4)

Und dies ist ein oktaler Speicherabzug der term-Datei, erzeugt durch den Befehl `od -c /usr/share/lib/terminfo/9/97801:`

```
0000000 032 001 025      %      !      211 001 232 002      s      t      a      n
0000020      d      a      r      d      |      9      7      8      0      1      |      9      7      8      0      8
0000040      001
0000060
0000100                                P      377 377 030      377 377 377 377
0000120 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
*
0000200 377 377 377 377 377 377 377 377      *      (      = 001
0000220 377 377      Y      O      T      377 377 377 377      2      G
0000240 272      027 001      ,      377 377 034 001      C      377 377      K
0000260 377 377      ) 001      a      377 377 377 377 372      S 001 377 377
0000300      \b 001 377 377      N 001 377 377 352      377 377 357      376
0000320      1 001 377 377 374      364      377 377 377 377 377 377 003 001
0000340      6 001 377 377 377 377 377 377 377 377 377 377 377 377 377
0000360      ! 001      i      377 377 006 002 377 377 377 377 377 377 - 001
0000400      e      \ 001 377 377 377 377 377 377 377 377      1 001 207 001
0000420      o 001      r 001      u 001      x 001      { 001      ~ 001 201 001 204 001
0000440      h 001      % 001      m      d 001 377 377 377 377 377 377      ' 001
0000460      }      u      377 377      X 001 377 377 377 377 360 001 377 377
0000500 377 377 363 001 377 377 377 377 377 377 377 377 377 377 377 377
0000520 377 377 377 377 377 377 223      377 377 341      307      226
0000540 330      201      276      250      261      212      237      377 377
0000560 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
0000600 377 377 377 377 324      377 377 320      y      q      025 002
0000620 377 377 377 377      ; 001 377 377 377 377 377 377 377 377 377 377
0000640 377 377 377 377 377 377 377 377 377 377 377      w 002 377 377
0000660      \b 002 377 377 377 377 377 377 377 377 377 377 377 377 377
0000700 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
0000720 377 377 377 377 377 377 377 377 370 001 377 377 377 377 377 377
0000740 377 377 377 377 377 377 377 365 001 377 377 377 377 377 377
0000760 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
*
0001060 377 377 377 377 377 377 377 377 212 001 215 001 220 001 223 001
0001100 226 001 231 001 234 001 237 001 242 001 245 001 250 001 253 001
0001120 256 001 261 001 264 001 267 001 272 001 275 001 300 001 303 001
0001140 306 001 311 001 314 001 317 001 322 001 325 001 330 001 333 001
0001160 336 001 341 001 344 001 347 001 352 001 355 001 377 377 377 377
0001200 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
*
0001600 377 377 377 377 377 377 025      377 377 377 377 001 002 376 001
0001620 373 001 004 002      \f 002 017 002 022 002      s      t      a      n      d      a
0001640      r      d      |      9      7      8      0      1      |      9      7      8      0      8      U
0001660      S      E      _      T      E      R      M      :      s      9      7      8      0      1      p      c
0001700      :      \r      007      \b      033      [      Z      033      [      %      i
0001720      %      p      1      %      d      ;      %      p      2      %      d      H      033      [      C
0001740      033      [      B      033      [      A      033      [      0      K      033      [
0001760      0      J      033      [      H 033      [      2      J      033      [      M      033
```

term(4)**term(4)**

```

0002000 [ M 033 [ L 033 [ L 033 [ T 033
0002020 [ S 033 [ S 033 [ T 033 [ % p 1
0002040 % d S 033 [ % p 1 % d T 033 E
0002060 033 [ % p 1 % d B 033 [ % p 1 % d
0002100 A 033 [ % p 1 % d D 033 [ % p 1
0002120 % d C 033 [ H 033 [ % p 1 % d L
0002140 033 [ % p 1 % d M 033 [ s 033 [
0002160 u 033 [ % p 1 % d @ 033 [ % p 1
0002200 % d P 033 [ 8 m 033 [ 7 m 033 [
0002220 0 m 017 016 017 033 [ 7 m 033 [ 0
0002240 m 033 [ 1 ; 2 4 r 033 [ m 017 033 ) w
0002260 033 [ 6 p 033 [ 7 p 033 [ @ 033
0002300 [ @ 033 [ P 033 [ P 033 [ 4 m
0002320 033 [ 0 m \t 033 [ % i % p 1 % d
0002340 ; % p 2 % d r 033 [ 2 m 033 [ 5
0002360 m 033 [ A 033 [ B 033 [ C 033 [
0002400 D 033 [ H 033 @ 033 A 033 B 033
0002420 C 033 D 033 F 033 G 033 H 033 I
0002440 033 J 033 K 033 L 033 M 033 N
0002460 033 O 033 P 033 0 033 _ 033 d 033
0002500 T 033 V 033 X 033 033 ; 033 "
0002520 033 # 033 $ 033 % 033 & 033 '
0002540 033 < 033 = 033 * 033 + 033 , 033
0002560 - 033 . 033 / 033 1 033 2 033 3
0002600 033 U 033 W 033 Y 033 > 004 033
0002620 g 033 > 033 m 033 9 033 : \n .br
0002640 \b 033 [ Z 033 4 033 p 033 o 033
0002660 [ 0 % ? % p 1 % t ; 7 % ; % ? %
0002700 p 2 % t ; 4 % ; % ? % p 3 % t ;
0002720 7 % ; % ? % p 4 % t ; 5 % ; % ?
0002740 % p 5 % t ; 2 % ; % ? % p 6 % t
0002760 ; 7 % ; % ? % p 7 % t ; 8 m % e
0003000 m % ; % ? % p 9 % t 016 % e 017 % ;
0003020 + K , L . N - M f ? j E k C 1
0003040 B m D n J q A t F u G v I w H x
0003060 @ ~ ,
0003064

```

Als Einschränkung gilt: vollständig übersetzte Einträge dürfen 4096 Bytes nicht überschreiten und die Gesamtheit der Einträge im Namens-Feld darf nicht länger als 128 Bytes sein.

DATEIEN

/usr/lib/share/terminfo

Übersetzte Datenbank für die Terminal-Beschreibung

/usr/include/term.h terminfo Include-Datei

SIEHE AUCH

curses(3X).

infocmp(1M), terminfo(4), term(5)

BEZEICHNUNG

terminfo – Datenbank der Terminaleigenschaften

ÜBERSICHT

/usr/share/lib/terminfo/?/*

BESCHREIBUNG

terminfo ist eine durch tic erzeugte Datenbank, in der die Eigenschaften solcher Geräte wie Terminals oder Drucker beschrieben werden. terminfo-Quelldateien beschreiben diese Geräte, indem sie eine Reihe ihrer Eigenschaften sowie weitere Aspekte angeben und bestimmte Zeichensequenzen aufführen, die ihr Verhalten steuern. Oft benutzen bildschirmorientierte Programme wie vi und curses diese Datenbank, wie auch einige SINIX-Systemkommandos wie beispielsweise ls und more. Damit ist es ohne weitere Programmänderungen möglich, eine Vielzahl von Geräten anzusprechen und damit zu arbeiten.

Quelldateien für terminfo enthalten eine oder mehrere Gerätebeschreibungen, zu denen es jeweils einen Beschreibungskopf gibt, der in Spalte 1 beginnt und dem dann eine oder mehrere Zeilen mit der Beschreibung der jeweiligen Geräte-Eigenschaften folgen. In einer Quelldatei für terminfo muß jede Zeile mit einem Komma (,) aufhören. Mit Ausnahme des Beschreibungskopfs muß jede Zeile mit einem Leer-/Tabulatorzeichen beginnen.

Die Einträge in Quelldateien für terminfo bestehen aus einer Reihe durch Kommata getrennter Felder. Leer-/Tabulatorzeichen nach einem Komma werden ignoriert. Kommata als Teil eines Eintrags müssen durch einen vorangestellten Gegenschrägstrich entwertet werden. Das folgende Beispiel eines terminfo-Eintrags zeigt dessen Format.

```
alias1 | alias2 | ... | aliasn | langer_name,  
<zwischenraum> am, lines #24,  
<zwischenraum> home=\Eeh,
```

Die erste Zeile einer Beschreibung wird gewöhnlich als Kopfzeile bezeichnet. Sie beginnt in Spalte 1 und muß mindestens zwei synonyme Bezeichnungen enthalten, die durch einen senkrechten Strich voneinander getrennt sind. Der letzte Eintrag der Kopfzeile sollte eine ausführliche Gerätebezeichnung in beliebiger Form sein. Synonyme dürfen mit keinem anderen Namen in der terminfo-Datenbank kollidieren und müssen den Namenskonventionen für Dateibezeichnungen in SINIX-Systemen genügen [siehe tic(1M)]. Sie dürfen beispielsweise keine Leer-/Tabulatorzeichen oder Schrägstriche enthalten.

Jedem Gerät sollte ein Name, wie zum Beispiel "97801", zugeordnet werden. Mit Ausnahme des letzten (ausführlichen) sollte man bei der Wahl der Gerätenamen die folgenden Konventionen beachten. Zunächst sollte der Name keinen Bindestrich enthalten, da dieser zur Anbindung eines Suffixes dient, um bestimmte Betriebsarten zu kennzeichnen.

Diese besonderen Betriebsarten können hardwaremäßig eingestellt sein oder entsprechen Benutzerpräferenzen. Um einem bestimmten Gerät eine besondere Betriebsart zuzuordnen, wird seinem Namen ein Bindestrich und ein Suffix angehängt, welches die Betriebsart kennzeichnet. Beispielsweise bedeutet das Suffix -w (wide mode) einen "breiten Modus", in dem statt der üblichen 80 Spalten 132 benutzt werden können. Um also ein vt100-Gerät mit 132 Spalten zu betreiben, gibt

man dafür den Namen "vt100-w" an. Nach Möglichkeit sollten die folgenden Suffixe verwendet werden.

Suffix	Bedeutung	Beispiel
-w	Breiter Modus (mehr als 80 Spalten)	5410-w
-am	Autom. Ränder (normalerw. Voreinst.)	vt100-am
-nam	Ohne autom. Ränder	vt100-nam
-n	Anzahl der Bildschirmzeilen	2300-40
-na	Keine Pfeiltasten (bleiben lokal)	c100-na
-np	Anzahl der Speicherseiten	c100-4p
-rv	Inverse Darstellung	4415-rv

Die Seiten zu `terminfo` im Referenzhandbuch sind in zwei Abschnitte aufgeteilt: "Geräte-Eigenschaften" und "Drucker-Eigenschaften".

Teil 1: Geräte-Eigenschaften

Die in `terminfo` beschriebenen Eigenschaften gliedern sich in drei Typen: Boolesche Werte (die das Vorhandensein oder das Fehlen bestimmter Merkmale angeben), numerische Werte (die bestimmte Gerätemerkmale quantifizieren) und Zeichenketten (die bei der Ausgabe bestimmte Operationen anstoßen).

In der folgenden Tabelle stellt **Variable** den Namen dar, über den ein C-Programmierer Geräte-Eigenschaften (auf der Ebene von `terminfo`) abruft. **Cap-Name** ist die Kurzbezeichnung einer Eigenschaft, wie sie in der Quelldatei für `terminfo` steht. Diese Kurzbezeichnung wird für die Aktualisierung der Quelldatei und von dem Kommando `tput` verwendet. Der **Termcap Code** ist ein aus zwei Buchstaben bestehender interner Name einer Eigenschaft, der dem früher in `termcap` verwendeten entspricht. (Es ist zu beachten, das `termcap` nicht länger unterstützt wird.)

Es gibt keine strengen Einschränkungen bezüglich der Länge von Cap-Namen, es hat sich jedoch eine informelle Begrenzung auf 5 Zeichen eingebürgert, um eine gewisse Kürze beizubehalten. Nach Möglichkeit werden dieselben oder ähnliche Namen wie im ANSI X3.64-1979 Standard verwendet mit der gleichen Bedeutung wie dort.

Mit Ausnahme der für die Eingabe vorgesehenen können für alle über Zeichenketten definierten Eigenschaften Füllzeichen definiert sein. Namen von Eigenschaften für die Eingabe, die im Abschnitt **Zeichenketten** aufgeführt sind, beginnen mit `key_`. Das in der folgenden Tabelle unter der Rubrik Beschreibung auftauchende Symbol `#i` bedeutet den *i*-ten Parameter.

Boolesche Werte

Variable	Cap- Name	Termcap- Code	Beschreibung
auto_left_margin	bw	bw	cub1 in Spalte 0 bewirkt Sprung zur letzten Spalte
auto_right_margin	am	am	Automat. Randeinstellung vorh.
back_color_erase	bce	be	Bildschirm löschen mit Hintergrundfarbe
can_change	ccc	cc	Terminal kann vorhandene Farbe redefinieren
ceol_standout_glitch	xhp	xs	Hervorhebung wird durch Überschreiben nicht gelöscht (hp)
col_addr_glitch	xhpa	YA	Nur positive Bewegungsrichtung für hpa/mhpa
cpi_changes_res	cpix	YF	Änderung des Zeichenabstands ändert Auflösung
cr_cancels_micro_mode	crxm	YB	cr schaltet Mikro-Modus aus
eat_newline_glitch	xenl	xn	Neue Zeile nach Spalte 80 wird ignoriert (Concept)
erase_overstrike	eo	eo	Übereinandergeschriebene Zeichen können mit einem Leerzeichen gelöscht werden
generic_type	gn	gn	Generischer Leitungstyp (z.B., Wählleitung)
hard_copy	hc	hc	Hardcopy Gerät
hard_cursor	chts	HC	Cursor ist schwer erkennbar
has_meta_key	km	km	Meta-Taste vorh. (Shift, Paritätsbit)
has_print_wheel	daisy	YC	Zeichensatzwechsel erfordert manuellen Austausch des Typenrads
has_status_line	hs	hs	Gesonderte Statuszeile vorhanden
hue_lightness_saturation	hls	hl	Für Farbe wird ausschließlich die HLS-Notation (Tektronix) verwendet
insert_null_glitch	in	in	Einfügemodus kennt Nullzeichen
lpi_changes_res	lpix	YG	Änderung des Zeilenabstands ändert Auflösung
memory_above	da	da	Bildschirminhalt kann über den Bildschirmrand hinaus nach oben verschoben und festgehalten werden
memory_below	db	db	Bildschirminhalt kann über den Bildschirmrand hinaus nach unten verschoben und festgehalten werden
move_insert_mode	mir	mi	Positionieren im Einfügemodus mögl.
move_standout_mode	msgr	ms	Positionieren im Hervorhebungsmodus mögl.
needs_xon_xoff	nxon	rx	Keine Füllzeichen möglich, xon/xoff-Quittungsbetrieb erforderlich
no_esc_ctlc	xsbc	xb	Beehive (f1=escape, f2=ctrl C)
non_rev_rmcup	nrrmc	NR	Keine Umkehrung von rmcup durch smcup
no_pad_char	npc	NP	Kein Füllzeichen vorhanden
over_strike	os	os	Übereinanderschreiben möglich
prtr_silent	mc5i	5i	Drucker gibt am Bildschirm kein Echo aus
row_addr_glitch	xvpa	YD	Nur positive Bewegungsrichtung für vpa/mvpa
semi_auto_right_margin	sam	YE	Zeichenausgabe in letzter Spalte erzeugt cr
status_line_esc_ok	eslok	es	Escape kann in Statuszeile verwendet werden

terminfo (4)**terminfo (4)**

Variable	Cap- Name	Termcap- Code	Beschreibung
dest_tabs_magic_smo	xt	xt	Überschreibende Tabulatoren, <i>smso</i> Magic-Zeichen(t1061)
tilde_glitch	hz	hz	Tilde (~) nicht darstellbar (Hazeline)
transparent_underline	ul	ul	Unterstreichen möglich
xon_xoff	xon	xo	xon/xoff Quittungsbetrieb

Numerische Angaben

Variable	Cap- Name	Termcap Code	Beschreibung
buffer_capacity	bufsz	Ya	Kapazität des Druckerpuffers
columns	cols	co	Anzahl Spalte pro Zeile
dot_vert_spacing	spinv	Yb	Vertikaler Nadelabstand in Nadeln pro Zoll
dot_horz_spacing	spinh	Yc	Horizontaler Punktabstand in Punkten pro Zoll
init_tabs	it	it	Tabulatoren auf bei Start auf jeder #-ten Position
label_height	lh	lh	Anzahl der Zeilen eines Bildschirm-Etiketts
label_width	lw	lw	Anzahl der Spalten eines Bildschirm-Etiketts
lines	lines	li	Anzahl Zeilen pro Bildschirm oder Speicherseite
lines_of_memory	lm	lm	Zeilen im Speicher, wenn > lines; 0 bedeutet: variabel
magic_cookie_glitch	xmc	sg	Anzahl Leerzeichen nach <i>smso</i> or <i>rmso</i> (Hervorhebungsmodus ein/aus)
max_colors	colors	Co	Maximale Anzahl an Bildschirmfarben
max_micro_address	maddr	Yd	Maximalwert in <i>micro_..._address</i>
max_micro_jump	mjump	Ye	Maximalwert in <i>parm_..._micro</i>
max_pairs	pairs	pa	Maximale Anzahl von Farbpaaren am Bildschirm
micro_col_size	mcs	Yf	Schrittweite für Zeichen im Mikro-Modus
micro_line_size	mls	Yg	Schrittweite für Zeilen im Mikro-Modus
no_color_video	ncv	NC	Bildschirmattribut darf nicht mit Farbe kombiniert werden
number_of_pins	npins	Yh	Anzahl der Nadeln im Druckkopf
num_labels	nlab	Nl	Anzahl der Bildschirm-Etiketten (Numerierung beginnt mit 1)
output_res_char	orc	Yi	Horizontale Auflösung in Einheiten pro Zeichen
output_res_line	orl	Yj	Vertikale Auflösung in Einheiten pro Zeile
output_res_horz_inch	orhi	Yk	Horizontale Auflösung in Einheiten pro Zoll
output_res_vert_inch	orvi	Yl	Vertikale Auflösung in Einheiten pro Zoll
padding_baud_rate	pb	pb	Niedrigste Übertragungsgeschwindigkeit, ab der Verzögerungen für CR/NL erforderlich sind
virtual_terminal	vt	vt	Log. Terminalnummer (SINIX-System)
wide_char_size	wides	Yn	Schrittweite für Zeichen bei Breitschrift
width_status_line	ws1	ws	Anzahl Spalten in Statuszeile

Strings

Variable	Cap- Name	Termcap Code	Beschreibung
acs_chars	acsc	ac	Zeichensatz-Paare für Grafik: aAbBcC
alt_scancode_esc	scsca	S8	Alternatives Escape für Scan-Code-Emulation (Voreinstellung gilt für vt100)
back_tab	cbt	bt	Zeichen für Tabulator-Rücksprung
bell	bel	b1	Akustisches Signal (Klingel)
bit_image_repeat	birep	Zy	Rastergrafik-Element #1 #2-Mal wiederholen (tparm verwenden)
bit_image_newline	binel	Zz	Auf nächste Zeile der Rastergrafik positionieren (tparm verwenden)
bit_image_carriage_return	bicr	Yv	An den Anfang der gleichen Zeile positionieren (tparm verwenden)
carriage_return	cr	cr	Wagenrücklauf
change_char_pitch	cpi	ZA	Anzahl der Zeichen pro Zoll ändern
change_line_pitch	lpi	ZB	Anzahl der Zeilen pro Zoll ändern
change_res_horz	chr	ZC	Horizontale Auflösung ändern
change_res_vert	cvr	ZD	Vertikale Auflösung ändern
change_scroll_region	csr	cs	Scroll-Bereich wechseln (Zeilen #1 bis #2) (vt100)
char_padding	rmp	rP	Wie ip, jedoch im Ersetzungsmodus
char_set_names	csnm	Zy	Liste aller Zeichensätze
clear_all_tabs	tbc	ct	Tabulatoren löschen
clear_margins	mgc	MC	Alle Ränder löschen (oben, unten, seitlich)
clear_screen	clear	cl	Bildschirm löschen und Cursor auf Bildschirmanfang positionieren
clr_bol	ell	cb	Ab Zeilenanfang bis Cursorposition löschen
clr_eol	el	ce	Löschen bis Zeilenende
clr_eos	ed	cd	Löschen bis Bildschirmende
code_set_init	csin	ci	Initialisierungssequenz für mehrfache Zeichensätze
color_names	colorm	Yw	Name für die Farbe #1
column_address	hpa	ch	Cursor auf Spalte positionieren
command_character	cmdch	CC	Prototyp-Kommandozeichen für Terminal
cursor_address	cup	cm	Cursor bildschirmbezogen positionieren, Zeile #1, Spalte #2
cursor_down	cuDl	do	Cursor eine Zeile nach unten
cursor_home	home	ho	Cursor an Bildschirmanfang (falls cup nicht vorhanden)
cursor_invisible	civis	vi	Cursor unsichtbar machen
cursor_left	cub1	le	Cursor eine Spalte nach links
cursor_mem_address	mrcup	CM	speicherbezogene Cursorpositionierung
cursor_normal	cnorm	ve	Normaldarstellung des Cursors (vs/vi rücksetzen)
cursor_right	cuf1	nd	Cursor nach rechts ohne Überschreiben/Löschen (Cursor oder Druckkopf)
cursor_to_ll	ll	ll	Cursor auf letzte Zeile, erste Spalte (falls cup nicht gilt)
cursor_up	cuu1	up	Cursor eine Zeile nach oben
cursor_visible	cvvis	vs	Cursor hervorgehoben darstellen
define_bit_image_region	defbi	Yx	Rechteck-Bereich für Rastergrafik definieren

Variable	Cap- Name	Termcap Code	Beschreibung
			(tparm verwenden)
define_char	defc	ZE	Zeichen in einem Zeichensatz definieren †
delete_character	dchl	dc	Zeichen löschen
delete_line	dll	dl	Zeile löschen
device_type	devt	dv	Anzeige der Unterstützung von Sprache/ Zeichensatz
dis_status_line	dsl	ds	Statuszeile ausschalten
display_pc_char	dispc	S1	PC-Zeichen ausgeben
down_half_line	hd	hd	Halbe Zeile nach unten (halber Zeilenvorschub)
ena_acs	enacs	eA	Alternativen Zeichensatz einschalten
end_bit_image_region	endbi	Yy	Ende des Bereichs für Rastergrafik (tparm verwenden)
enter_alt_charset_mode	smacs	as	Alternativen Zeichensatz einschalten
enter_am_mode	smam	SA	Automatische Randeinstellung einschalten
enter_blink_mode	blink	mb	Blinken einschalten
enter_bold_mode	bold	md	Hell- oder Fettmodus einschalten
enter_ca_mode	smcup	ti	Steuersequenz, die vor cup ausgegeben werden muß
enter_delete_mode	smdc	dm	Löschmodus einschalten
enter_dim_mode	dim	mh	Halbhellmodus einschalten
enter_doublewide_mode	swidm	ZF	Breitschrift einschalten
enter_draft_quality	sdrfq	ZG	Entwurfsschrift einschalten
enter_insert_mode	smir	im	Einfügemodus einschalten
enter_italics_mode	sitm	ZH	Schrägschrift einschalten
enter_leftward_mode	slm	ZI	Richtungsumkehr für Druckkopf
enter_micro_mode	smicm	ZJ	Eigenschaften für Mikro-Modus einstellen
enter_near_letter_quality	snlq	ZK	NLQ-Modus einstellen
enter_normal_quality	snrmq	ZL	Normal-Modus für Drucken für Drucken einstellen
enter_pc_charset_mode	smpch	S2	Zeichendarstellung für PC einschalten
enter_protected_mode	prot	mp	Modus "geschützt" einschalten
enter_reverse_mode	rev	mr	Inversdarstellung einschalten
enter_scancode_mode	smsc	S4	Scan-Code-Modus für PC einschalten
enter_secure_mode	invis	mk	Dunkelmodus einschalten (Zeichen unsichtbar)
enter_shadow_mode	sshm	ZM	Schattendruck einschalten
enter_standout_mode	smso	so	Hervorhebungsmodus einschalten
enter_subscript_mode	ssubm	ZN	Tiefstellung einschalten
enter_superscript_mode	ssupm	ZO	Hochstellung einschalten
enter_underline_mode	smul	us	Unterstreichungsmodus einschalten
enter_upward_mode	sum	ZP	Wagenrücklauf nach oben einschalten
enter_xon_mode	smxon	SX	Xon/xoff Quittungsbetrieb einschalten
erase_chars	ech	ec	#1 Zeichen löschen (mit Leerzeichen überschreiben)
exit_alt_charset_mode	rmacs	ae	Alternativen Zeichensatz ausschalten
exit_am_mode	rmam	RA	Automatische Randeinstellung ausschalten
exit_attribute_mode	sgr0	me	alle Darstellungsarten ausschalten
exit_ca_mode	rmcup	te	Beendet Programme, die cup verwenden
exit_delete_mode	rmdc	ed	Löschmodus ausschalten
exit_doublewide_mode	rwidm	ZQ	Breitschrift ausschalten

Variable	Cap- Name	Termcap Code	Beschreibung
exit_insert_mode	rmir	ei	Einfügemodus ausschalten
exit_italics_mode	ritm	ZR	Schrägschrift ausschalten
exit_leftward_mode	rlm	ZS	Aus Umkehrmodus nach links in Normal-Modus zurückschalten
exit_micro_mode	rmicm	ZT	Mikro-Modus ausschalten
exit_pc_charset_mode	rmpch	S3	Zeichendarstellung für PC ausschalten
exit_scancode_mode	rmsc	S5	Scan-Code-Modus für PC ausschalten
exit_shadow_mode	rshm	ZU	Schattendruck ausschalten
exit_standout_mode	rmso	se	Hervorhebungsmodus ausschalten
exit_subscript_mode	rsubm	ZV	Tiefstellung ausschalten
exit_superscript_mode	rsupm	ZW	Hochstellung ausschalten
exit_underline_mode	rmul	ue	Unterstreichungsmodus ausschalten
exit_upward_mode	rum	ZX	Aus Umkehrmodus nach unten in Normal-Modus zurückschalten
exit_xon_mode	rmxon	RX	Xon/xoff Quittingsbetrieb ausschalten
flash_screen	flash	vb	Optisches Signal (ohne Cursorbewegung)
form_feed	ff	ff	Seitenvorschub bei Hardcopy-Gerät
from_status_line	fs1	fs	Rücksprung von Statuszeile
init_1string	is1	i1	Terminalinitialisierung
init_2string	is2	is	Terminalinitialisierung
init_3string	is3	i3	Terminalinitialisierung
init_file	if	if	Name der Initialisierungsdatei
init_prog	iprog	iP	Pfadname des Initialisierungsprogramms
initialize_color	initc	Ic	Definition einer Farbe initialisieren
initialize_pair	initp	Ip	Farbpaar initialisieren
insert_character	ich1	ic	Zeichen einfügen
insert_line	ill1	al	Leerzeile einfügen
insert_padding	ip	ip	Einfügen von Füllzeichen

Zeichenketten der Art "key_" werden von speziellen Tasten abgegeben. Die Beschreibung dazu enthält das in `curses.h` definierte Makro, welches den durch die `curses` Routine `getch` zurückgelieferten Code abgibt, wenn diese Taste betätigt wird [siehe `curs_getch(3X)`].

terminfo (4)

terminfo (4)

Variable	Cap- Name	Termcap Code	Beschreibung
key_a1	ka1	K1	KEY_A1, Linke obere Ecke der Tastatur
key_a3	ka3	K3	KEY_A3, Rechte obere Ecke der Tastatur
key_b2	kb2	K2	KEY_B2, Mitte der Tastatur
key_backspace	kbs	kb	KEY_BACKSPACE, Code der Backspace-Taste
key_beg	kbeg	@1	KEY_BEG, Code der Taste Beg(inn)
key_btab	kcbt	kB	KEY_BTAB, Code der Taste Tab-Rückschritt
key_c1	kc1	K4	KEY_C1, Linke untere Ecke der Tastatur
key_c3	kc3	K5	KEY_C3, Rechte untere Ecke der Tastatur
key_cancel	kcan	@2	KEY_CANCEL, Code der Taste 'Abbruch'
key_catab	ktbc	ka	KEY_CATAB, Code der Taste alle-Tabs-löschen
key_clear	kclr	kC	KEY_CLEAR, Code der Bildschirm-löschen- oder Lösch-Taste
key_close	kclro	@3	KEY_CLOSE, Code der Taste 'Schließen'
key_command	kcmd	@4	KEY_COMMAND, Code der cmd-Taste (Kommando)
key_copy	kcpy	@5	KEY_COPY, Code der Taste 'Kopieren'
key_create	kert	@6	KEY_CREATE, Code der Erzeuge-Taste
key_ctab	ktab	kt	KEY_CTAB, Code der Taste Tabulator-löschen
key_dc	kdch1	kD	KEY_DC, Code der Taste Zeichen-löschen
key_dl	kdl1	kL	KEY_DL, Code der Taste Zeile-löschen
key_down	kcud1	kd	KEY_DOWN, Code der Taste Pfeile-nach-unten
key_eic	krmir	kM	KEY_EIC, Code von rmir oder smir (Einfügemodus aus/ein)
key_end	kend	@7	KEY_END, Code der Ende-Taste
key_enter	kent	@8	KEY_ENTER, Code der Enter-/Send-Taste
key_eol	kel	kE	KEY_EOL, Code der Taste Lösche-bis-Zeilenende
key_eos	ked	kS	KEY_EOS, Code der Taste Lösche-bis-Bildschirmende
key_exit	kext	@9	KEY_EXIT, Code der Exit-Taste
key_f0	kf0	k0	KEY_F(0), Code der Funktionstaste f0
key_f1	kf1	k1	KEY_F(1), Code der Funktionstaste f1
key_f2	kf2	k2	KEY_F(2), Code der Funktionstaste f2
key_f3	kf3	k3	KEY_F(3), Code der Funktionstaste f3
key_f4	kf4	k4	KEY_F(4), Code der Funktionstaste f4
key_f5	kf5	k5	KEY_F(5), Code der Funktionstaste f5
key_f6	kf6	k6	KEY_F(6), Code der Funktionstaste f6
key_f7	kf7	k7	KEY_F(7), Code der Funktionstaste f7
key_f8	kf8	k8	KEY_F(8), Code der Funktionstaste f8
key_f9	kf9	k9	KEY_F(9), Code der Funktionstaste f9
key_f10	kf10	k;	KEY_F(10), Code der Funktionstaste f10
key_f11	kf11	F1	KEY_F(11), Code der Funktionstaste f11
key_f12	kf12	F2	KEY_F(12), Code der Funktionstaste f12
key_f13	kf13	F3	KEY_F(13), Code der Funktionstaste f13
key_f14	kf14	F4	KEY_F(14), Code der Funktionstaste f14
key_f15	kf15	F5	KEY_F(15), Code der Funktionstaste f15
key_f16	kf16	F6	KEY_F(16), Code der Funktionstaste f16

terminfo (4)

terminfo (4)

Variable	Cap- Name	Termcap Code	Beschreibung
key_f17	kf17	F7	KEY_F(17), Code der Funktionstaste f17
key_f18	kf18	F8	KEY_F(18), Code der Funktionstaste f18
key_f19	kf19	F9	KEY_F(19), Code der Funktionstaste f19
key_f20	kf20	FA	KEY_F(20), Code der Funktionstaste f20
key_f21	kf21	FB	KEY_F(21), Code der Funktionstaste f21
key_f22	kf22	FC	KEY_F(22), Code der Funktionstaste f22
key_f23	kf23	FD	KEY_F(23), Code der Funktionstaste f23
key_f24	kf24	FE	KEY_F(24), Code der Funktionstaste f24
key_f25	kf25	FF	KEY_F(25), Code der Funktionstaste f25
key_f26	kf26	FG	KEY_F(26), Code der Funktionstaste f26
key_f27	kf27	FH	KEY_F(27), Code der Funktionstaste f27
key_f28	kf28	FI	KEY_F(28), Code der Funktionstaste f28
key_f29	kf29	FJ	KEY_F(29), Code der Funktionstaste f29
key_f30	kf30	FK	KEY_F(30), Code der Funktionstaste f30
key_f31	kf31	FL	KEY_F(31), Code der Funktionstaste f31
key_f32	kf32	FM	KEY_F(32), Code der Funktionstaste f32
key_f33	kf33	FN	KEY_F(33), Code der Funktionstaste f33
key_f34	kf34	FO	KEY_F(34), Code der Funktionstaste f34
key_f35	kf35	FP	KEY_F(35), Code der Funktionstaste f35
key_f36	kf36	FQ	KEY_F(36), Code der Funktionstaste f36
key_f37	kf37	FR	KEY_F(37), Code der Funktionstaste f37
key_f38	kf38	FS	KEY_F(38), Code der Funktionstaste f38
key_f39	kf39	FT	KEY_F(39), Code der Funktionstaste f39
key_f40	kf40	FU	KEY_F(40), Code der Funktionstaste f40
key_f41	kf41	FV	KEY_F(41), Code der Funktionstaste f41
key_f42	kf42	FW	KEY_F(42), Code der Funktionstaste f42
key_f43	kf43	FX	KEY_F(43), Code der Funktionstaste f43
key_f44	kf44	FY	KEY_F(44), Code der Funktionstaste f44
key_f45	kf45	FZ	KEY_F(45), Code der Funktionstaste f45
key_f46	kf46	Fa	KEY_F(46), Code der Funktionstaste f46
key_f47	kf47	Fb	KEY_F(47), Code der Funktionstaste f47
key_f48	kf48	Fc	KEY_F(48), Code der Funktionstaste f48
key_f49	kf49	Fd	KEY_F(49), Code der Funktionstaste f49
key_f50	kf50	Fe	KEY_F(50), Code der Funktionstaste f50
key_f51	kf51	Ff	KEY_F(51), Code der Funktionstaste f51
key_f52	kf52	Fg	KEY_F(52), Code der Funktionstaste f52
key_f53	kf53	Fh	KEY_F(53), Code der Funktionstaste f53
key_f54	kf54	Fi	KEY_F(54), Code der Funktionstaste f54
key_f55	kf55	Fj	KEY_F(55), Code der Funktionstaste f55
key_f56	kf56	Fk	KEY_F(56), Code der Funktionstaste f56
key_f57	kf57	Fl	KEY_F(57), Code der Funktionstaste f57
key_f58	kf58	Fm	KEY_F(58), Code der Funktionstaste f58
key_f59	kf59	Fn	KEY_F(59), Code der Funktionstaste f59
key_f60	kf60	Fo	KEY_F(60), Code der Funktionstaste f60
key_f61	kf61	Fp	KEY_F(61), Code der Funktionstaste f61

terminfo (4)

terminfo (4)

Variable	Cap- Name	Termcap Code	Beschreibung
key_f62	kf62	Fq	KEY_F(62), Code der Funktionstaste f62
key_f63	kf63	Fr	KEY_F(63), Code der Funktionstaste f63
key_find	kfind	@0	KEY_FIND, Code der Find-Taste
key_help	khlp	*1	KEY_HELP, Code der Help-Taste
key_home	khome	kh	KEY_HOME, Code der Home-Taste (Bildschirmanfang)
key_ic	kich1	kI	KEY_IC, Code der Taste Zeichen-einfügen/ Einfügemodus-einschalten
key_il	kill	kA	KEY_IL, Code der Taste Zeile-einfügen
key_left	kcub1	k1	KEY_LEFT, Code der Taste Pfeil-nach-links
key_ll	kll	kH	KEY_LL, Code der Taste linke-untere-Bildschirmecke
key_mark	kmrk	*2	KEY_MARK, Code der Markiere-Taste
key_message	kmsg	*3	KEY_MESSAGE, Code der Meldungs-Taste
key_move	kmov	*4	KEY_MOVE, Code der Positionierungs-Taste
key_next	knxt	*5	KEY_NEXT, Code der Taste nächstes-Objekt
key_npage	kn	kN	KEY_NPAGE, Code der Taste nächste-Seite
key_open	kopn	*6	KEY_OPEN, Code der Open-Taste
key_options	kopt	*7	KEY_OPTIONS, Code der Taste 'Optionen'
key_ppage	kpp	kP	KEY_PPAGE, Code der Taste 'vorige-Seite'
key_previous	kprv	*8	KEY_PREVIOUS, Code der Taste 'voriges-Objekt'
key_print	kprt	*9	KEY_PRINT, Code der Taste 'Druck/Kopie'
key_redo	krdo	*0	KEY_REDO, Code der Taste 'Wiederholung'
key_reference	kref	&1	KEY_REFERENCE, Code der Ref(erenz)-Taste
key_refresh	krfr	&2	KEY_REFRESH, Code der Taste 'Bildschirm-neu'
key_replace	krpl	&3	KEY_REPLACE, Code der Ersetze-Taste
key_restart	krst	&4	KEY_RESTART, Code der Neustart-Taste
key_resume	kres	&5	KEY_RESUME, Code der Taste 'neu-aufsetzen'
key_right	kcuf1	kr	KEY_RIGHT, Code der Taste Pfeil-nach-rechts
key_save	ksav	&6	KEY_SAVE, Code der Sichern-Taste
key_sbeg	KBEG	&9	KEY_SBEG, Code der Taste Beg(inn) (plus Shift)
key_scancel	kCAN	&0	KEY_SCANCEL, Code der Taste 'Abbruch' (plus Shift)
key_scommand	kCMD	*1	KEY_SCOMMAND, Code der Kommando-Taste (plus Shift)
key_scopy	kCPY	*2	KEY_SCOPY, Code der Taste 'Kopieren' (plus Shift)
key_screate	kCRT	*3	KEY_SCREATE, Code der Taste 'Erzeugen' (plus Shift)
key_sdc	kDC	*4	KEY_SDC, Code der Taste 'Zeichen-löschen' (plus Shift)
key_sdl	kDL	*5	KEY_SDL, Code der Taste 'Zeile-löschen' (plus Shift)
key_select	kslt	*6	KEY_SELECT, Code der Auswahl-Taste
key_send	kEND	*7	KEY_SEND, Code der End-Taste (plus Shift)
key_seol	KEOL	*8	KEY_SEOL, Code der Taste 'Zeile-löschen' (plus Shift)

terminfo (4)

terminfo (4)

Variable	Cap- Name	Termcap Code	Beschreibung
key_sexit	kEXT	*9	KEY_SEXIT, Code der Exit-Taste (plus Shift)
key_sf	kind	kF	KEY_SF, Code der Taste vorwärts/nach-unten-Scrollen
key_sfind	kFND	*0	KEY_SFIND, Code der Find-Taste (plus Shift)
key_shelp	kHLP	#1	KEY_SHELP, Code der Help-Taste (plus Shift)
key_shome	kHOM	#2	KEY_SHOME, Code der Home-Taste (plus Shift)
key_sic	kIC	#3	KEY_SIC, Code der Taste 'Eingabe' (plus Shift)
key_sleft	kLFT	#4	KEY_SLEFT, Code der Taste 'Pfeil-nach-rechts' (plus Shift)
key_smessage	kMSG	%a	KEY_SMESSAGE, Code der Taste 'Meldung' (plus Shift)
key_smove	kMOV	%b	KEY_SMOVE, Code der Taste 'Positionierung' (plus Shift)
key_snext	kNXT	%c	KEY_SNEXT, Code der 'nächste'-Taste (plus Shift)
key_soptions	kOPT	%d	KEY_SOPTIONS, Code der Taste 'Optionen' (plus Shift) key
key_sprevious	kPRV	%e	KEY_SPREVIOUS, Code der 'vorige'-Taste (plus Shift) key
key_sprint	kPRT	%f	KEY_SPRINT, Code der Drucke-Taste (plus Shift)
key_sr	kri	kR	KEY_SR, Code der Taste rückwärts/nach-oben-Scrollen
key_sredo	kRDO	%g	KEY_SREDO, Code der Taste 'Wiederholung' (plus Shift)
key_sreplace	kRPL	%h	KEY_SREPLACE, Code der Taste 'Ersetzen' (plus Shift)
key_sright	kRIT	%i	KEY_SRIGHT, Code der Taste 'Pfeil-nach-rechts' (plus Shift)
key_sresume	kRES	%j	KEY_SRESUME, Code der Taste 'neu-aufsetzen' (plus Shift)
key_ssave	kSAV	!1	KEY_SSAVE, Code der Sichern-Taste (plus Shift)
key_ssuspend	kSPD	!2	KEY_SSUSPEND, Code der suspend-Taste (plus Shift) key
key_stab	khts	kT	KEY_STAB, Code der Taste 'Tabulatoren-setzen'
key_sundo	kUND	!3	KEY_SUNDO, Code der undo-Taste (plus Shift)
key_suspend	kspd	&7	KEY_SUSPEND, Code der suspend-Taste
key_undo	kund	&8	KEY_UNDO, Code der undo-Taste
key_up	kcuu1	ku	KEY_UP, Code der Taste 'Pfeil-nach-oben'
keypad_local	mnkx	ke	Übertragungsmodus der Tastatur ausschalten
keypad_xmit	smkx	ks	Übertragungsmodus der Tastatur einschalten
lab_f0	lf0	10	Beschriftung von Funktionstaste f0 falls ungleich f0
lab_f1	lf1	11	Beschriftung von Funktionstaste f1 falls ungleich f1
lab_f2	lf2	12	Beschriftung von Funktionstaste f2 falls ungleich f2
lab_f3	lf3	13	Beschriftung von Funktionstaste f3 falls ungleich f3
lab_f4	lf4	14	Beschriftung von Funktionstaste f4 falls ungleich f4
lab_f5	lf5	15	Beschriftung von Funktionstaste f5 falls ungleich f5
lab_f6	lf6	16	Beschriftung von Funktionstaste f6 falls ungleich f6
lab_f7	lf7	17	Beschriftung von Funktionstaste f7 falls ungleich f7

terminfo (4)

terminfo (4)

Variable	Cap- Name	Termcap Code	Beschreibung
lab_f8	lf8	l8	Beschriftung von Funktionstaste f8 falls ungleich f8
lab_f9	lf9	l9	Beschriftung von Funktionstaste f9 falls ungleich f9
lab_f10	lf10	l10	Beschriftung von Funktionstaste f10 falls ungleich f10
label_off	rmln	LF	Soft-Etiketten ausschalten
label_on	smln	LO	Soft-Etiketten einschalten
meta_off	rmm	mo	Meta-Modus ausschalten (8. Bit)
meta_on	smm	mm	Meta-Modus einschalten (8. Bit)
micro_column_address	mhpa	ZY	Wie column_address für Mikro-Modus
micro_down	mcudl	ZZ	Wie cursor_down für Mikro-Modus
micro_left	mcubl	Za	Wie cursor_left für Mikro-Modus
micro_right	mcufr	Zb	Wie cursor_right für Mikro-Modus
micro_row_address	mvpa	Zc	Wie row_address für Mikro-Modus
micro_up	mcuul	Zd	Wie cursor_up für Mikro-Modus
newline	nel	nw	Neue-Zeile (wie cr plus lf)
order_of_pins	porder	Ze	Abbildung von Software-Bits auf Druckkopf-Nadeln
orig_colors	oc	oc	Farben(-Paare) auf ursprünglichen Wert zurücksetzen
orig_pair	op	op	Ursprüngl. Voreinstellung des Farbpaares herstellen
pad_char	pad	pc	Füllzeichen (ungleich Null)
parm_dch	dch	DC	#1 Zeichen löschen
parm_delete_line	dl	DL	#1 Zeilen löschen
parm_down_cursor	cud	DO	Cursor #1 Zeilen nach unten
parm_down_micro	mcud	Zf	Wie parm_down_cursor für Mikro-Modus
parm_ich	ich	IC	Einfügen von #1 Leerzeichen
parm_index	indn	SF	Um #1 Zeilen vorwärts Scrollen
parm_insert_line	il	AL	Einfügen von #1 Leerzeilen
parm_left_cursor	cub	LE	Cursor um #1 Stellen nach links
parm_left_micro	mcub	Zg	Wie parm_left_cursor für Mikro-Modus
parm_right_cursor	cuf	RI	Cursor um #1 Stellen nach rechts
parm_right_micro	mcuf	Zh	Wie parm_right_cursor für Mikro-Modus
parm_rindex	rin	SR	Um #1 Zeilen rückwärts Scrollen
parm_up_cursor	cuu	UP	Cursor um #1 Stellen nach oben
parm_up_micro	mcuu	Zi	Wie parm_up_cursor für Mikro-Modus
pc_term_options	pctrm	S6	Optionen für PC-Terminal
pkey_key	pfkey	pk	Programmierbare Funktionstaste #1 mit Zeichenkette #2 für Ausgabe belegen
pkey_local	pfloc	pl	Programmierbare Funktionstaste #1 mit Zeichenkette #2 für Ausführung belegen
pkey_plab	pfxl	xl	Programmierbare Funktionstaste #1 mit Zeichenkette #2 für Übertragung und mit Zeichenkette #3 für Anzeige belegen
pkey_xmit	px	px	Programmierbare Funktionstaste #1 mit Zeichenkette #2 für Übertragung belegen
plab_norm	pln	pn	Programmierbare Funktionstaste #1 mit Zeichenkette #2 für Anzeige belegen
print_screen	mc0	ps	Bildschirminhalt auf Drucker ausgeben

terminfo(4)

terminfo(4)

Variable	Cap- Name	Termcap Code	Beschreibung
prtr_non	mc5p	pO	Drucker für Ausgabe von #1 Bytes anschalten
prtr_off	mc4	pf	Drucker anschalten
prtr_on	mc5	po	Drucker ausschalten
repeat_char	rep	rp	Zeichen #1 #2-Mal wiederholen
req_for_input	rfi	RF	Nächstes Eingabezeichen schicken (für ptys)
reset_1string	rs1	r1	Terminal in Normalzustand rücksetzen
reset_2string	rs2	r2	Terminal in Normalzustand rücksetzen
reset_3string	rs3	r3	Terminal in Normalzustand rücksetzen
reset_file	rf	rf	Name der Datei mit den Zeichenketten zum Rücksetzen
restore_cursor	rc	rc	Cursorposition mit letztem sc wiederherstellen
row_address	vpa	cv	Absolute vertikale Position
save_cursor	sc	sc	Cursorposition sichern
scancode_escape	scesc	S7	Escape für Scan-Code Emulation
scroll_forward	ind	sf	Scrollen nach oben
scroll_reverse	ri	sr	Scrollen nach unten
select_char_set	sos	Zj	Zeichensatz wählen
set0_des_seq	s0ds	s0	Wechsel in Zeichensatz 0 (EUC set 0, ASCII)
set1_des_seq	s1ds	s1	Wechsel in Zeichensatz 1
set2_des_seq	s2ds	s2	Wechsel in Zeichensatz 2
set3_des_seq	s3ds	s3	Wechsel in Zeichensatz 3
set_a_background	setab	AB	Hintergrundfarbe mit ANSI-Escape setzen
set_a_foreground	setaf	AF	Vordergrundfarbe mit ANSI-Escape setzen
set_attributes	sgr	sa	Bildschirmattribut #1-#9 setzen
set_background	setb	Sb	Aktuelle Hintergrundfarbe setzen
set_bottom_margin	smgb	Zk	Unteren Rand auf aktuelle Zeile setzen
set_bottom_margin_parm	smgbp	Zl	Unteren Rand auf Zeile #1 oder #2 Zeilen vor der untersten setzen
set_color_band	setcolor	Yz	Wechsel beim Farbband auf Farbe #1
set_color_pair	scp	sp	Aktuelles Farbpaar setzen
set_foreground	setf	Sf	Aktuelle Vordergrundfarbe1 setzen
set_left_margin	smgl	ML	Linken Rand an aktueller Spalte setzen
set_left_margin_parm	smglp	Zm	Linken (rechten) Rand auf Spalte #1 (#2) setzen
set_lr_margin	smglr	ML	Linken und rechten Rand setzen
set_page_length	slines	YZ	Seitenlänge auf #1 Zeilen setzen (tparm verwenden)
set_right_margin	smgr	MR	Rechten Rand an aktueller Spalte setzen
set_right_margin_parm	smgrp	Zn	Rechten Rand an Spalte #1 setzen
set_tab	hts	st	An aktueller Spalte in allen Zeilen einen Tabulator setzen
set_tb_margin	smgtb	MT	Oberen und unteren Rand setzen
set_top_margin	smgt	Zo	Oberen Rand auf aktuelle Zeile setzen
set_top_margin_parm	smgtp	Zp	Oberen (unteren) Rand auf Zeile #1 (#2) setzen
set_window	wind	wi	Aktuelles Fenster belegt die Zeilen #1-#2 und die Spalten #3-#4
start_bit_image	sbim	Zq	Druck von Pixelgrafik einschalten
start_char_set_def	sosd	Zr	Beginn einer Zeichensatz-Definition

terminfo (4)

terminfo (4)

Variable	Cap- Name	Termcap Code	Beschreibung
stop_bit_image	rbim	Zs	Druck von Pixelgrafik ausschalten
stop_char_set_def	rcsd	Zt	Ende einer Zeichensatz-Definition
subscript_characters	subcs	Zu	Liste tiefstellbarer Zeichen
superscript_characters	supcs	Zv	Liste hochstellbarer Zeichen
tab	ht	ta	Sprung zur nächsten Hardware-Tabulatorposition (Tab-Distanz = 8)
these_cause_cr	docr	Zw	Ausdruck dieser Zeichen verursacht cr
to_status_line	tsl	ts	Sprung in Statuszeile, Spalte #1
underline_char	uc	uc	Ein Zeichen unterstreichen und dahinter positionieren
up_half_line	hu	hu	Halbe Zeile hoch (umgekehrter Halbzeilenvorschub)
xoff_character	xoffc	XF	X-off Zeichen
xon_character	xonc	XN	X-on Zeichen
zero_motion	zerom	Zx	Keine Druckkopfbewegung nach dem Drucken des nächsten Einzelzeichens

Beispieleintrag

Der folgende Eintrag zeigt einen der komplexeren Einträge in der terminfo-Datei; er beschreibt das 97801 Terminal.

```
standard|97801|97808,
    am,
    cols#80, lines#24,
    acsc=+K,L,N-Mf??jEkClBmDnJqAtFuGvIwHx@~, , bel=^G,
    blink=E[5m, cbt=E[Z, civis=E[6p, clear=E[HE[2J,
    cnorm=E[7p, cr=\r, csr=E[%i%p1%d;%p2%dr,
    cub=E[%p1%dD, cubl=\b, cud=E[%p1%dB, cudl=E[B,
    cuf=E[%p1%DC, cuf1=E[C, cup=E[%i%p1%d;%p2%dH,
    cuu=E[%p1%DA, cuu1=E[A, dch=E[%p1%DP, dchl=E[P,
    dim=E[2m, dl=E[%p1%DM, dl1=E[M, ed=E[0J, el=E[0K,
    home=E[H, ht=\t, ich=E[%p1@d@, ich1=E[@,
    il=E[%p1%DL, ill=E[L, ind=E[S, indn=E[%p1%DS,
    invis=E[8m, kbs=\b, kcbt=E[Z, kcubl=E[D,
    kcucl=E[B, kcufl=E[C, kcuul=E[A, kdchl=E[P,
    kdl1=E[M, kdw=Ep, kf1=E@, kf10=EJ, kf11=EK,
    kf12=EL, kf13=EM, kf14=EN, kf15=EO, kf16=EP,
    kf17=E0, kf18=E_, kf19=Ed, kf2=EA, kf20=ET,
    kf21=EV, kf22=EX, kf23=Es, kf24=E;, kf25=E",
    kf26=E#, kf27=E$, kf28=E%, kf29=E&, kf3=EB,
    kf30=E', kf31=E<, kf32=E=, kf33=E*, kf34=E+,
    kf35=E,, kf36=E-, kf37=E., kf38=E/, kf39=E1,
    kf4=EC, kf40=E2, kf41=E3, kf42=EU, kf43=EW,
    kf44=EY, kf5=ED, kf6=EF, kf7=EG, kf8=EH, kf9=EI,
    khlp=E>, khome=E[H, kich1=E[@, kill=E[L,
    kind=E[T, kiw=Eo, kmenu=0 kmode=E4, kprt=Eg,
    kri=E[S, krst=Em, krst=Em, lf0=E>, lf2=^D,
    nel=EE, pctrm=USE_TERM:s97801pc:, rc=E[u, rev=E[7m,
    ri=E[T, rin=E[%p1%dT, rmacs=^O, rmso=E[0m,
    rmul=E[0m, sbt=E9, sc=E[s,
    sgr=E[0%?%p1%t;7%;??%p2%t;4%;??%p3%t;7%;??%p4%t;5%;??%p5%t;
    2%;??%p6%t;7%;??%p7%t;8m%em%;??%p9%t^N%e^O%;,
    sgr0=E[0m^O, sht=E:, smacs=^N,
    smcup=E[1;24rE[m^OE)w, smso=E[7m, smul=E[4m,
```

Typen der Eigenschaften im Beispieleintrag

Das Beispiel zeigt die drei Arten, in terminfo Eigenschaften zu beschreiben, nämlich boolesche, numerische und Zeichenketten-Werte. Alle aufgezählten Eigenschaften müssen mit einem Komma abgeschlossen sein; das gilt auch für die letzte. In terminfo Quelldateien wird über den **Cap-Namen** auf die Eigenschaften Bezug genommen (siehe obige Tabellen).

Boolesche Werte werden lediglich aufgezählt und durch Kommata getrennt.

Numerischen Werten folgt das Zeichen '#' und ein positiver ganzzahliger Wert. So ist beispielsweise dem Eintrag `cols` im obigen Beispiel für 97801 der Wert 80 zugeordnet (`cols` gibt die am Gerät verfügbare Spaltenanzahl an). Die Angaben für numerische Werte können sowohl in dezimaler als auch in oktaler oder hexadezimaler Schreibweise angegeben werden, wobei den Konventionen der

Programmiersprache C zu folgen ist.

Eigenschaften, die über Zeichenketten beschrieben werden, wie zum Beispiel `e1` (Sequenz für "Lösche bis zum Zeilenende") sind folgendermaßen aufgebaut: zuerst kommt ein Cap-Name aus zwei bis fünf Zeichen, dem ein Gleichheitszeichen (=) und die zugehörige Zeichenkette folgt, die bis zum nächsten Komma reicht. In einer solchen Beschreibung kann z.B. an jeder Stelle eine Verzögerungszeit vorkommen, die in Millisekunden angegeben wird. Sie ist durch ein vorangestelltes `$` Zeichen gekennzeichnet und in spitze Klammern eingeschlossen (z.B. `e1=\E[0K`). Die Füllzeichen werden durch `tput` geliefert. Eine Verzögerung wird in der Form einer Zahl, einer Zahl gefolgt von einem Stern (z.B. `5*`), einem Schrägstrich (z.B. `5/`) oder von beiden (z.B. `5*/`) angegeben. Die Angabe von `*` bedeutet, daß die erforderliche Anzahl von Füllzeichen proportional zur Anzahl der von der Operation betroffenen Zeilen ist und die angegebene Menge als die Anzahl der Füllzeichen pro betroffene Einheit zu verstehen ist. (Für den Fall von Zeicheneinfügung ist der Faktor weiterhin die Anzahl der betroffenen Zeilen und ist stets 1, sofern das Gerät nicht über die Eigenschaft `in` verfügt und die Software davon Gebrauch macht.) Im Zusammenhang mit `*` ist es manchmal nützlich, die Verzögerung in der Form von `3.5` anzugeben, d.h. in Zehntel-Millisekunden. Dabei darf nicht mehr als eine Dezimalstelle angegeben werden.

Ein Schrägstrich bedeutet, daß die Angabe von Füllzeichen obligatorisch ist. Verfügt ein Gerät über die `xon` Eigenschaft, sind Füllzeichen empfohlen, werden jedoch nur für Aufwandsschätzungen verwendet oder beim Betrieb im Raw-Modus. Ganz gleich wie `xon` gesetzt ist, werden obligatorische Füllzeichen immer übertragen. Wenn für `bel` oder `flash` (empfohlene oder obligatorische) Füllzeichen angegeben sind, werden sie immer verwendet, unabhängig davon, ob `xon` angegeben ist oder nicht.

Für die Codierung von Sonderzeichen gilt in `terminfo` die folgende Notation. Das Zeichen ESCAPE wird sowohl durch `\E` als auch durch `\e` dargestellt. Für jedes geeignete `x` steht `^x` für Control-`x` und die Sequenzen `\n`, `\l`, `\r`, `\t`, `\b`, `\f` und `\s` ergeben Neue-Zeile, Zeilenvorschub, Wagenrücklauf, Tabulator, Backspace, Seitenvorschub und Leerzeichen. Ferner ist `\^` der Code für Zirkumflex (`^`); `\` ist der Code für Gegenschrägstrich (`\`); `\:` ist der Code für Doppelpunkt (`:`) und `\0` steht für Null. Aus `\0` wird effektiv `\200` erzeugt, was somit kein Abschlußzeichen einer Zeichenkette ist, aber von den meisten Geräten als Nullzeichen interpretiert wird, sofern CS7 vereinbart ist (siehe `stty(1)`). Außerdem können Zeichen auch noch durch einen Gegenschrägstrich, gefolgt von drei Oktalziffern, angegeben werden.

Manchmal ist es notwendig, eine bestimmte Eigenschaft zu deaktivieren. Dies wird durch einen Punkt vor dem Namen der betreffenden Eigenschaft realisiert (siehe das zweite `ind` im obigen Beispiel). Es ist zu beachten, daß die Eigenschaften in der Reihenfolge von links nach rechts definiert werden und daher eine weitere Definition die frühere überschreibt.

Erstellen von Beschreibungen

Die effektivste Methode, eine Beschreibung zu erstellen, besteht darin, die Beschreibung eines ähnlichen Terminals in `terminfo` zu kopieren, die neue schrittweise aufzubauen und Teilbeschreibungen mit `vi` auf Fehler hin zu überprüfen. Bei einem sehr ungewöhnlichen Terminal kann sich jedoch

herausstellen, daß es mit der `terminfo` Datei nur unzulänglich beschrieben werden kann oder daß `vi` an diesem Gerät auf einmal nicht mehr funktioniert. Eine neue Terminalbeschreibung läßt sich auf einfache Weise testen, indem man die Umgebungsvariable `TERMINFO` mit dem Pfadnamen des Verzeichnisses belegt, das die übersetzte Beschreibung enthält, an der man gerade arbeitet; dann greifen Programme auf dieses Verzeichnis zu statt auf `/usr/share/lib/terminfo`. Ein "Härtetest", um die korrekte Verzögerung für das Einfügen von Zeilen zu ermitteln (sofern sie nicht vom Hersteller in der Dokumentation angegeben ist), sieht folgendermaßen aus: Mit `vi` eine große Datei mit 9600 Baud editieren, in der Mitte des Bildschirms etwa 16 Zeilen löschen und dann rasch hintereinander mehrmals die `u`-Taste (`undo`) betätigen. Wenn dies den Bildschirm durcheinander bringt, heißt das gewöhnlich, daß eine größere Verzögerung benötigt wird. Einen ähnlichen Test kann man für die Funktion Zeichen-Einfügen durchführen.

Abschnitt 1-1: Grundlegende Eigenschaften

Mit der numerischen Eigenschaft `cols` gibt man die Anzahl der Spalten pro Zeile für das Gerät an. Handelt es sich um ein Terminal, wird die Anzahl der Bildschirmzeilen mit `lines` angegeben. Springt der Cursor bei Erreichen des rechten Randes an den Anfang der nächsten Zeile, sollte `am` angegeben werden. Kann das Terminal den Bildschirm löschen und steht der Cursor danach in der linken oberen Ecke, so wird das durch die Zeichenketten-Eigenschaft `clear` angegeben. Können mehrere Zeichen übereinandergeschrieben werden (statt daß das erste Zeichen durch das zweite ersetzt wird), ist `os` anzugeben. Für eine Druckerstation ohne Bildschirmaufzeichnung ist sowohl `hc` als auch `os` anzugeben. Falls es möglich ist, den Cursor auf die erste Spalte der aktuellen Zeile zu positionieren, ist hierfür `cr` anzugeben. (Normalerweise ist das der Wagenrücklauf, d.h. Control-M.) Wenn die Möglichkeit zur Erzeugung eines akustischen Signals existiert (wie Klingel oder Piepston), wird hierfür `bel` angegeben. Wenn dieses Gerät wie die meisten das `xon-xoff` Protokoll für die Flußsteuerung verwendet, ist `xon` anzugeben.

Ein eventuell vorhandener Code für die Positionierung des Cursors um eine Stelle nach links (z.B. Backspace) ist mit `cubl` anzugeben. Entsprechend sind die Codes für die Positionierung des Cursors nach rechts, nach oben bzw. nach unten mit `cuf1`, `cuu1`, bzw. `cud1` anzugeben. Dabei darf der Text nicht verändert werden, über den der Cursor bei der Positionierung hinwegläuft. So wird man normalerweise nicht "`cuf1=\s`" angeben, weil sonst die Zeichen, die der Cursor passiert, mit Leerzeichen überschrieben würden.

In diesem Zusammenhang ist auf folgenden wichtigen Punkt hinzuweisen: die in `terminfo` kodierten lokalen Cursorbewegungen sind hinsichtlich des linken und des oberen Bildschirmrandes undefiniert. Deshalb sollte ein Programm nie einen Rückwärtsschritt über den linken Bildschirmrand hinaus versuchen (es sei denn, `bw` ist angegeben) oder nach oben über den oberen Bildschirmrand hinaus. Ein Bildlauf nach oben wird dadurch erreicht, daß im Programm erst auf die linke untere Bildschirmecke positioniert wird und dann die Zeichenkette `ind` (Index) abgesetzt wird.

Ein Bildlauf nach unten wird so programmiert, daß erst auf die linke obere Bildschirmecke positioniert wird und dann die Zeichenkette `ri` (reverser Index) abgesetzt wird. Die Zeichenketten `ind` und `ri` sind undefiniert, wenn nicht vorher auf die entsprechende Bildschirmecke positioniert worden ist.

Zu den Bildlauf-Sequenzen existieren auch parametrisierte Varianten, nämlich `lndr` und `rin`. Diese haben dieselbe Bedeutung wie `ind` und `ri`, außer daß sie mit einem Parameter versorgt werden und gemäß dieser Angabe um die entsprechende Anzahl von Zeilen nach oben bzw. unten blättern. Auch sie sind undefiniert, wenn nicht vorher auf die entsprechende Bildschirmecke positioniert worden ist.

Mit `am` wird angegeben, ob der Cursor bei der Ausgabe am rechten Bildschirmrand stehen bleibt; diese Angabe gilt jedoch nicht notwendigerweise auch für ein `cuf1` in der letzten Spalte. Die einzige in Bezug auf den linken Rand definierte Cursorbewegung nach links ist die mit `bw`. Diese Angabe bewirkt, daß ein `cub1` am linken Rand an den rechten Rand der vorhergehenden Zeile springt. Ohne `bw` ist diese Cursorbewegung undefiniert. Diese Funktion kann z.B. gut dazu verwendet werden, um einen Rahmen rund um den Bildschirm zu zeichnen. Verfügt das Terminal über umschaltbare automatische Randeinstellungen, sollte in der Quelldatei für `terminfo` `am` angegeben sein. In diesem Fall sollten Zeichenketten zur Initialisierung diese nach Möglichkeit einschalten. Gibt es ein Kommando, das auf die erste Spalte der nächsten Zeile positioniert, so kann dies mit `nel` (Neue-Zeile) angegeben werden. Dabei ist unwichtig, ob das Kommando den Rest der aktuellen Zeile löscht. Auf diese Weise kann auch bei einem Terminal, das kein `cr` und `lf` hat, eine brauchbare Position auf den Anfang der nächsten Zeile erreicht werden.

Mit diesen Angaben können Hardcopygeräte und Bildschirmterminals ausreichend beschrieben werden. So sieht die Beschreibung des AT&T 5320 Hardcopygeräts folgendermaßen aus:

```
5320|att5320|AT&T 5320 hardcopy terminal,
    am, hc, os,
    cols#132,
    bel=^G, cr=\r, cub1=\b, cnd1=\n,
    dch1=\E[P, dll=\E[M,
    ind=\n,
```

und das Lear Siegler ADM-3 wird so beschrieben:

```
adm3|lsi adm3,
    am, bel=^G, clear=^Z, cols#80, cr=^M, cub1=^H,
    cud1=^J, ind=^J, lines#24,
```

Abschnitt 1-2: Parametrisierte Steuersequenzen

Cursor-Adressierung und andere Funktionen, die mit Parametern versorgt werden müssen, werden durch parametrisierte Steuersequenzen beschrieben, wobei wie in `printf` Escape-Sequenzen des Formats `%x` verwendet werden. So braucht beispielsweise die Cursorpositionierung mit `cup` zwei Parameter, die Spalten- und Zeilennummer, wohin positioniert werden soll. (Zeilen und Spalten werden beginnend mit Null durchnummeriert; die Angaben beziehen sich auf den für den Benutzer sichtbaren Bildschirm und nicht etwa auf einen unsichtbaren Speicherbereich.) Erlaubt das Terminal eine speicherbezogene Cursor-Adressierung, kann dies mit `mrcup` angegeben werden.

Die Parameterversorgung erfolgt mittels eines Stacks und spezieller `%`-Codes, wobei die Postfix-Notation (umgekehrte polnische Notation) verwendet wird. Typischerweise trägt eine Steuersequenz einen ihrer Parameter im Stack ein, um ihn dann in einem bestimmten Format auszugeben. Oft werden aber auch komplexere

Operationen benötigt, wobei dann die Operanden in normaler Reihenfolge und die Operationen in Postfix-Notation angegeben sind. So wird beispielsweise vom ersten Parameter 5 abgezogen, indem man `%p1%(5)%-` angibt.

Die %-Codes haben folgende Bedeutung (dabei steht die Funktion `push` für "in den Stack eintragen" und die Funktion `pop` für "oberstes Element dem Stack entnehmen"):

`%%` '%' ausgeben

`%[[:]schalter][stellen].dez_stellen][doxXs]`
wie in `printf`, Schalter sind `[-+#]` und Leerzeichen

`%c` oberstes Stack-Element im Format `%c` ausgeben

`%p[1-9]`
i-ten Parameter im Stack eintragen

`%P[a-z]`
dynamische Variable `[a-z]` mit oberstem Stack-Element versorgen

`%g[a-z]`
dynamische Variable `[a-z]` im Stack eintragen

`%P[A-Z]`
statische Variable `[a-z]` mit oberstem Stack-Element versorgen

`%g[A-Z]`
statische Variable `[a-z]` im Stack eintragen

`%'c'` Zeichenkonstante `c` im Stack eintragen

`%{nn}` Dezimalkonstante `nn` im Stack eintragen

`%l` Zeichenkette vom Stack nehmen, Länge ermitteln, diese im Stack eintragen
`push strlen(pop)`

`%+ %- %* %/ %m`
arithmetische Operatoren (`%m` = modulo): `push(pop integer2 op pop integer1)`

`%& %| %^`
Bit Operationen: `push(pop integer2 op pop integer1)`

`%= %> %<`
logische Operationen: `push(pop integer2 op pop integer1)`

`%^ %&` logische Operationen: und, oder

`%! %~` unäre Operationen: `push(op pop)`

`%i` (gilt für ANSI Terminals) addiere 1 zum ersten Parameter, falls einer angegeben ist, bzw. zu den beiden ersten, falls mehr als einer angegeben ist.

`%? expr %t thenpart %e elsepart %;`
if-then-else, `%e elsepart` ist wahlweise; else-if's können auch wie in Algol 68 geschrieben werden: `%? c1 %t b1 %e c2 %t b2 %e c3 %t b3 %e c4 %t b4 %e re %;`
`ci` sind Bedingungen, `bi` sind Rümpfe.

Wenn der Schalter “-” zusammen mit “[doxS]” angegeben wird, muß zwischen “%” und “-” ein Doppelpunkt (:) angegeben werden, damit sich der Schalter von der binären Operation “%-” unterscheidet, z.B. “%:-16.16s”.

Als Beispiel sei die Hewlett-Packard 2645 betrachtet. Um hier den Cursor auf Zeile 3 und Spalte 12 zu positionieren, muß `\Ea12c03Y` plus eine Verzögerung von 6 Millisekunden abgesetzt werden. Man beachte, daß hier Zeile und Spalte in umgekehrter Reihenfolge angegeben werden und beide mit zwei Ziffern spezifiziert sein müssen, wobei gegebenenfalls eine Null eingefügt wird. Bei `cup` ist daher anzugeben:

```
cup=\Ea%p2%2.2dc%p1%2.2dY$<6>
```

Beim Micro-Term ACT-IV muß der Zeile und Spalte ein `^T` vorangestellt sein und die Zeilen- und Spaltenangabe selbst erfolgt binär: “`cup=^T%p1%c%p2%c`”. Geräte, die “%c” verwenden, müssen über die Eigenschaft `cub1` (Cursor um ein Zeichen nach links, dabei das Zeichen löschen; Backspace) verfügen und den Cursor eine Bildschirmzeile nach oben verschieben können (`cuu1`). Das ist deswegen nötig, weil die Übertragung von `\n`, `^D` und `\r` nicht immer zuverlässig ist (sie könnten vom System geändert oder weggeworfen werden). (Die von `terminfo` verwendeten Bibliotheksfunktionen setzen den TTY-Modus so, daß Tabulatoren keinesfalls expandiert werden; die Übertragung von `\t` ist daher sicher. Dies ist besonders für das Ann Arbor 4080 wichtig.)

Zum Schluß sei die LSI ADM-3a betrachtet, wo als Distanzwert (offset) ein Leerzeichen verwendet wird. Damit ist “`cup=\E=%p1%\s' %+%c%p2%\s' %+%c`”. Nachdem “`\E=`” abgesetzt ist, wird der erste Parameter im Stack eingetragen, dann der ASCII-Code für Leerzeichen (32), dann werden beide addiert (wobei die Summe anstelle der beiden Werte im Stack eingetragen wird) und das Resultat als Zeichen ausgegeben wird. Anschließend werden für den zweiten Parameter dieselben Operationen durchgeführt. Die Verwendung des Stacks erlaubt auch wesentlich komplexere Arithmetik.

Abschnitt 1-3: Cursorbewegungen

Verfügt das Terminal über die Fähigkeit, den Cursor rasch an den Bildschirmanfang (die linke obere Ecke) zu positionieren, so kann dies mit `home` angegeben werden; eine entsprechende Fähigkeit zur Positionierung in die linke untere Ecke wird mit `ll` angegeben. Dabei kann diese Positionierung auch durch ein `cuu1` ab Bildschirmanfang erreicht werden; dies sollte jedoch nie direkt in einem Programm erfolgen (es sei denn, `ll` sorgt dafür), da in einem Programm keinerlei Annahmen bezüglich der Auswirkungen einer Positionierung nach oben vom Bildschirmanfang getroffen werden können. Es ist zu beachten, daß `home` auf die gleiche Stelle positioniert wie bei der Angabe von (0,0), nämlich in die linke obere Ecke des Bildschirms und nicht etwa des Speichers. (Daher kann an den Hewlett-Packard Terminals die Sequenz `\EH` nicht für `home` verwendet werden, ohne dabei nicht auch einige andere Eigenschaften am Terminal zu verlieren.)

Erlaubt das Gerät eine absolute Zeilen- bzw. Spaltenpositionierung, so kann dies mit den nur einen Parameter erfordernden Steuersequenzen `hpa` (horizontale Positionierung absolut) und `vpa` (vertikale Positionierung absolut) angegeben werden. Diese Angaben sind manchmal kürzer als die allgemeineren Sequenzen mit zwei Parametern (siehe Beispiel zu Hewlett-Packard 2645) und ihre Verwendung ist der von `cup` vorzuziehen. Existieren lokale Cursorbewegungen (z.B., `n` Stellen nach

rechts), so sind diese mit `cud`, `cub`, `cuf` und `cuu` anzugeben, wobei der einzige Parameter die Anzahl Stellen angibt, um die der Cursor bewegt werden soll. Diese Funktionen sind insbesondere bei Geräten wie z.B. Tektronix 4025 nützlich, die nicht über die Funktion `cup` verfügen.

Wenn sich ein Gerät in einem speziellen Modus befinden muß, damit Programme, die diese Eigenschaften verwenden, ablaufen können, dann wird mit `smcup` in diesen Zustand gewechselt und er wird mit `rmcup` wieder verlassen. Dies tritt bei Terminals auf, wie beispielsweise Concept, die über mehr als eine Speicherseite verfügen. Hat ein solches Gerät nur speicherbezogene Cursor-Adressierung und keine relative Adressierung, dann muß für deren korrektes Funktionieren ein Fenster von der Größe des Bildschirms fest im Gerät verankert werden. Auch beim tektronix 4025 wird dies verwendet. Dort wird das von `terminfo` verwendete Kommandozeichen durch `smcup` gesetzt. Wenn nach der Ausgabe einer `rmcup` Sequenz durch eine `smcup` Sequenz der Bildschirm nicht wieder in den Zustand vor der `rmcup` Ausgabe rückversetzt wird, muß `nrrmc` angegeben werden.

Abschnitt 1-4: Bereich löschen

Erlaubt das Terminal Löschen ab der aktuellen Cursorposition bis Zeilenende, wobei der Cursor an der aktuellen Position bleibt, sollte dies mit `el` angegeben werden. Kann das Terminal vom Zeilenanfang bis zur aktuellen Cursorposition inklusive löschen, wobei der Cursor auf der aktuellen Position verbleibt, so ist dies mit `el` anzugeben. Ist Löschen ab der aktuellen Cursorposition bis zum Bildschirmende möglich, so ist dies mit `ed` anzugeben. `ed` ist so definiert, daß es nur ab der ersten Spalte einer Zeile wirkt. Daher kann man diese Funktion auch simulieren, indem man ein Kommando zum Löschen einer großen Anzahl Zeilen eingibt, wenn `ed` selbst nicht vorhanden ist.

Abschnitt 1-5: Zeile einfügen/löschen

Die Fähigkeit eines Terminals, oberhalb der aktuellen Zeile eine neue leere Zeile zu erzeugen, wird mit `il1` angegeben. Diese Steuersequenz ist so definiert, daß sie nur wirkt, wenn der Cursor in der ersten Spalte der Zeile steht. Nach dem Einfügen der Zeile steht der Cursor in der neuen leeren Zeile. Umgekehrt wird die Fähigkeit eines Terminals, die aktuelle Zeile (in der der Cursor steht) zu löschen, mit `dl1` angegeben. Auch diese Steuersequenz ist so definiert, daß sie nur wirkt, wenn der Cursor in der ersten Spalte der zu löschenden Zeile steht. Entsprechende Funktionen, bei denen über einen Parameter die Anzahl der einzufügenden bzw. zu löschenden Zeilen angegeben wird, können mit `il` und `dl` angegeben werden.

Kann bei Terminals (wie z.B. VT100) ein bestimmter Bildlauf-Bereich gesetzt werden, so wird das entsprechende Kommando mit `csr` beschrieben; dabei werden zwei Parameter verwendet, die die erste und die letzte Zeile des Bildlauf-Bereichs angeben. Leider ist die Position des Cursors nach diesem Kommando undefiniert. Allerdings ist es möglich, die Auswirkung dieser Kommandos zum Einfügen oder Löschen von Zeilen zu bestimmen. In diesem Zusammenhang sind auch die Kommandos `sc` und `rc` recht nützlich, die die aktuelle Cursorposition vormerken und wiederherstellen. Bei vielen Terminals, die keine expliziten Funktionen zum Einfügen und Löschen von Zeilen haben, können am oberen oder unteren Bildschirmrand Zeilen auch mit `ri` bzw. `ind` eingefügt werden; häufig ist dies sogar schneller, auch wenn die Terminals über die genannten Funktionen verfügen.

Um zu bestimmen, ob ein Terminal erhaltende oder nicht-erhaltende Bildlauf-Bereiche hat, geht man folgendermaßen vor: In der Mitte des Bildschirms wird ein Bildlauf-Bereich erzeugt und in der letzten Zeile des Bereichs werden Daten abgelegt. Dann wird der Cursor auf die erste Zeile des Bildlauf-Bereichs positioniert und die Funktion `ri` (reverser index) gefolgt von `d11` (Zeile löschen) oder `ri` angestoßen. Wenn die ursprünglich in der letzten Zeile des Bildlauf-Bereichs stehenden Daten durch `d11` oder `ind` im Bildlauf-Bereich wiederhergestellt werden, handelt es sich um einen erhaltenden Bildlauf-Bereich. Andernfalls hat das Terminal nicht-erhaltende Bildlauf-Bereiche. Im letzteren Fall sollte `csr` nicht angegeben werden, es sei denn, daß `ind`, `ri`, `indn`, `rin`, `d1` und `d11` alle nicht-erhaltendes Scrollen simulieren.

Kann an einem Terminal ein Fenster als Speicherbereich definiert werden, auf das sich alle Kommandos auswirken, so sollte dies mit der parametrisierten Steuersequenz `wind` angegeben werden. Die vier hier benötigten Parameter bezeichnen die erste und die letzte Zeile und die erste und die letzte Spalte im Speicherbereich (in dieser Reihenfolge).

Kann der Bildschirminhalt über den Bildschirm hinaus nach oben verschoben und festgehalten werden, so ist `da` anzugeben. Kann der Bildschirminhalt über den Bildschirm hinaus nach unten verschoben und festgehalten werden, so ist `db` anzugeben. Dies bedeutet, daß beim Löschen einer Zeile oder beim Scrollen eines ganzen Bildschirms beschriebene Zeilen von unten — bzw. beim Rückwärts-Scrollen mit `ri` von oben — auf dem Bildschirm ausgegeben werden können.

Abschnitt 1-6: Zeichen einfügen/löschen

Es gibt hinsichtlich der Funktionen zum Einfügen bzw. Löschen von Zeichen zwei Grundtypen programmierbarer Terminals, die mittels `terminfo` beschrieben werden können. Die gebräuchlichsten Operationen zum Einfügen und Löschen von Zeichen wirken nur auf die Zeichen der aktuellen Zeile und verschieben die Zeichen am Ende der Zeile so, daß sie verloren gehen. Bei anderen Terminals (z.B. Concept 100 und Perkin Elmer Owl) wird zwischen über die Tatstatur eingegebenen und nicht belegten Leerstellen am Bildschirm in der Weise unterschieden, daß beim Einfügen bzw. Löschen nur eine nicht belegte Leerstelle verdoppelt bzw. überschrieben wird. Zu welchem Typ Ihr Terminal gehört, können Sie dadurch feststellen, das Sie zunächst den Bildschirm löschen und dann Text mit eingestreuten Cursorpositionierungen eingeben, beispielsweise "`abc def`", wobei zwischen `abc` und `def` keine Leerzeichen eingegeben werden, sondern die Pfeiltasten für Cursor-nach-rechts benutzt werden. Anschließend wird der Cursor vor `abc` positioniert und für das Terminal der Einfügemodus eingestellt. Wenn Sie jetzt weiteren Text eingeben und dadurch der Rest der Zeile entsprechend verschoben wird, sodaß die Zeichen am Zeilenende verloren gehen, unterscheidet Ihr Terminal nicht zwischen eingegebenen Leerzeichen und nicht belegten Leerstellen. Wird andererseits durch die Texteingabe zunächst `abc` bis an `def` herangeschoben und dann beides bis zum Zeilenende und darüberhinaus in die nächste Zeile verschoben, gehört Ihr Terminal zum zweiten Typ; d.h., es sollte die Eigenschaft `in` ("insert null") eingegeben werden. Zwar handelt es sich hier um zwei von der Logik her unterschiedliche Eigenschaften (einzeiliger bzw. mehrzeiliger Einfügemodus und besondere Behandlung nicht belegter Leerzeichen), doch sind bislang keine Terminals bekannt, die nicht mit einem einzigen Merkmal beschrieben werden könnten.

Mit `terminfo` lassen sich sowohl Terminals beschreiben, die einen eigenen Einfügemodus haben, als auch solche, die eine einfache Steuersequenz zum Einfügen einer Leerstelle in der aktuellen Zeile absetzen. Mit `smir` gibt man die Steuersequenz zum Einschalten des Einfügemodus an, mit `rmir` die Sequenz zum Ausschalten des Einfügemodus und mit `ich1` eine beliebige Sequenz, die unmittelbar vor dem einzufügenden Zeichen ausgegeben werden muß. Für die meisten Terminals mit eigenem Einfügemodus braucht `ich1` nicht angegeben zu werden, wohl aber für solche, die zum Einfügen von Leerzeichen auf dem Bildschirm eine Steuersequenz brauchen. Verfügt Ihr Terminal über beides, ist gewöhnlich der Einfügemodus der Funktion `ich1` vorzuziehen. Von beidem zusammen sollte nur dann Gebrauch gemacht werden, wenn das Terminal auch tatsächlich eine Kombination davon benötigt. Nach dem Einfügen eventuell erforderliche Verzögerungen sind in Millisekunden mit `ip` (als Zeichenketten-Option) anzugeben. Hiermit kann man auch andere Zeichenketten angeben, die gegebenenfalls nach dem Einfügen eines einzelnen Zeichens abgesetzt werden müssen. Muß an Ihrem Terminal sowohl der Einfügemodus eingeschaltet sein als auch ein spezieller Code vor jedem einzufügenden Zeichen abgesetzt werden, kann sowohl `smir/rmir` als auch `ich1` angegeben werden und es wird beides verwendet. Die Funktion `ich` benötigt einen Parameter `n` und fügt `n` Leerzeichen ein.

Falls bei der Eingabe von Zeichen, während man nicht im Einfügemodus ist, eine Verzögerung nötig ist, wird diese in Millisekunden mit `rmp` angegeben.

Manchmal ist es erforderlich, im Einfügemodus mit Cursorbewegungen an andere Stellen der gleichen Zeile zu positionieren, um dort Zeichen zu löschen (weil beispielsweise hinter der Einfügeposition ein Tabulator steht). Erlaubt Ihr Terminal Positionierungen im Einfügemodus, kann das Einfügen durch die Angabe von `mir` beschleunigt werden. Wird `mir` nicht angegeben, wirkt sich das lediglich auf die Geschwindigkeit aus, mit der Einfügungen vorgenommen werden können. Bei manchen Terminals (speziell denen von Datamedia) darf `mir` aufgrund der Funktionsweise des Einfügemodus nicht angegeben werden.

Schließlich kann man mit `dch1` die Funktion zum Löschen eines Zeichens angeben und die Funktion `dch` mit einem Parameter `n` löscht `n` Zeichen; der Modus zum Löschen wird mit `smdc` und `rmde` ein- bzw. ausgeschaltet (Modus, in dem sich das Terminal befinden muß, damit `dch1` funktioniert).

Mit `ech` plus einem Parameter `n` kann ein Kommando zum Löschen von `n` Zeichen angegeben werden, was dem Überschreiben von `n` Zeichen mit Leerstellen ohne Cursorbewegung entspricht.

Abschnitt 1-7: Hervorhebung, Unterstreichung und optische Signale

Ihr Gerät mag über eine oder mehrere Darstellungsarten zur Hervorhebung bestimmter Zeichen am Bildschirm verfügen. Dazu gehören die folgenden Darstellungsarten (wobei in Klammern der Name zur Aktivierung genannt ist): Blinken (`blink`), fett bzw. besonders hell (`bold`), blaß bzw. halbhell (`dim`), Bildschirm dunkel bzw. Text unsichtbar (`invis`), geschützter Text (`prot`), Inversdarstellung (`rev`) und ein alternativer Zeichensatz (der mit `smacs` ein- und mit `rmacs` ausgeschaltet wird). Falls es zum Aktivieren des alternativen Zeichensatzes ein eigenes Kommando braucht, wird das mit `enacs` angegeben. Das Einschalten einer dieser Darstellungsarten kann (muß aber nicht) dazu führen, daß andere ausgeschaltet werden.

Zum Ausschalten der erweiterten Darstellungsarten wird `sgr0` verwendet. Es sollte stets spezifiziert sein, da es die einzige Möglichkeit ist, manche Eigenschaften wie z.B. `dim` oder `blink` auszuschalten.

Eine Darstellungsart sollte als *standout* (Hervorhebungsmodus) definiert werden [siehe `curses(3X)`], um Fehlermeldungstext und sonstigen Text hervorzuheben, der Aufmerksamkeit auf sich ziehen soll. Dafür sollte eine klare, kontrastreiche Anzeige gewählt werden, die angenehm für die Augen ist (halbhelle Inversdarstellung oder Invers allein hat sich gut bewährt). Die Steuersequenzen zum Ein- und Ausschalten des Hervorhebungsmodus werden mit `sms0` bzw. `rms0`, angegeben. Hinterläßt der Code zum Ein- bzw. Ausschalten des Hervorhebungsmodus, wie beispielsweise beim TVI 912 und Teleray, ein oder zwei Leerzeichen am Bildschirm, so kann man sich mittels `xmc` über die genaue Anzahl der hinterlassenen Leerzeichen informieren.

Steuersequenzen zum Ein- und Ausschalten von Unterstreichungen können mit `smul` und `rmul`, angegeben werden. Verfügt das Gerät (wie beispielsweise das Micro-Term MIME), über die Möglichkeit, das aktuelle Zeichen zu unterstreichen und anschließend den Cursor um eine Stelle nach rechts zu positionieren, so ist dies mit `uc` anzugeben.

Terminals mit der Eigenschaft (`xmc`) ("magic cookie") legen bei Empfang von Steuersequenzen für bestimmte Darstellungsarten spezielle Zeichen zur Auswahl des Anzeige-Algorithmus (sog. "Magic Cookies") ab, statt etwa für jedes einzelne Zeichen ein Extrabit zu setzen. Manche Terminals (wie das Hewlett-Packard 2621) verlassen beim Wechsel in eine neue Zeile oder bei Cursoradressierung den Hervorhebungsmodus automatisch. In Programmen, die den Hervorhebungsmodus verwenden, sollte dieser vor Cursoradressierungen oder Zeilenwechseln ausgeschaltet werden, es sei denn, daß die Eigenschaft `magr` vorhanden ist; diese gibt an, daß Positionierungen im Hervorhebungsmodus möglich sind.

Verfügt das Terminal über die Möglichkeit, den gesamten Bildschirm blinken zu lassen, um (als Ersatz für ein akustisches Signal) einen Fehler anzuzeigen, so kann dies mit `flash` angezeigt werden; der Cursor darf durch das Blinken nicht bewegt werden. Eine recht gute Möglichkeit, diese Eigenschaft zu simulieren, besteht darin, den Bildschirm auf Inversdarstellung umzuschalten, eine Verzögerung von 200 Millisekunden zu bewirken und anschließend wieder in Normaldarstellung zurückzuschalten.

Soll der Cursor stärker als normal auffallen, wenn er zum Beispiel nicht in der untersten Zeile ist oder um ihn statt eines nicht blinkenden Unterstreichungszeichens als besser zu findendes Blocksymbol oder als blinkenden Unterstrich darzustellen, kann hierfür die Steuersequenz `cvvis` angegeben werden. Außerdem sollte die boolesche Angabe `chts` erfolgen. Kann die Darstellung des Cursors am Bildschirm völlig unterdrückt werden, wird dies mit `civis` angegeben. Mit `cnorm` wird die Auswirkung dieser Darstellungsarten wieder aufgehoben.

Die Fähigkeit eines Terminals, unterstrichene Zeichen durch Hinzufügen des Unterstrichs zu produzieren, ohne dafür besondere Steuersequenzen zu benötigen, auch wenn die Überschreibfähigkeit für andere Zeichen fehlt, wird mit `ul` angegeben. Bei Terminals, die beim Überschreiben eines Zeichens mit einem anderen beide auf dem Bildschirm darstellen, ist `os` anzugeben. Können übereinandergeschriebene Zeichen mit einem Leerzeichen gelöscht werden, ist hierfür `eo` anzugeben.

Ist eine Sequenz vorhanden, mit der beliebige Kombinationen von Darstellungsarten gesetzt werden können, wird diese mit `sgr` (Attribute setzen) angegeben. Diese Funktion arbeitet mit 9 Parametern, wobei jeder entweder 0 oder ungleich Null ist, je nachdem, ob die entsprechende Eigenschaft ein- oder ausgeschaltet ist. Die neun Parameter beziehen sich (in dieser Reihenfolge) auf: Hervorhebung, Unterstreichen, Inversdarstellung, Blinken, halbhell, fett/hell, dunkel, geschützt, alternativer Zeichensatz. Es ist nicht nötig, daß `sgr` alle diese Darstellungsarten unterstützt; dies gilt nur für diejenigen, die mit einem gesonderten Kommando angesprochen werden. Es sei beispielsweise angenommen, daß das betrachtete Terminal die folgenden Steuersequenzen zum Einschalten bestimmter Betriebsarten benötigt:

tparm Parameter	Attribut	Steuersequenz
	keines	<code>\E{0m</code>
p1	hervorheben	<code>\E[0;4;7m</code>
p2	unterstreichen	<code>\E[0;3m</code>
p3	invers	<code>\E[0;4m</code>
p4	blinken	<code>\E[0;5m</code>
p5	halbhell	<code>\E[0;7m</code>
p6	fett/hell	<code>\E[0;3;4m</code>
p7	dunkel	<code>\E[0;8m</code>
p8	geschützt	not available
p9	altern. Zeichensatz	<code>^O (off) ^N (on)</code>

Hierbei ist zu beachten, daß in jeder Steuersequenz anfangs eine 0 vorkommt, die dazu dient, andere Attribute auszuschalten, bevor das dazu gehörige eingeschaltet wird. Außerdem ist hier, wie weiter oben vorgeschlagen, der Hervorhebungsmodus hervorheben (standout) als Kombination von halbhell (dim) und Inversdarstellung invers (reverse) eingestellt. Weiterhin ist der Modus fett/hell (bold) ersatzweise durch die Kombination von Inversdarstellung plus Unterstreichen dargestellt, da es diesen Modus an diesem Terminal nicht gibt. Zum Einschalten einer Kombination von Unterstreichen und Blinken (underline+blink) müßte man `\E[0;3;5m` angeben. Darüberhinaus hat dieses Terminal keinen Geschützt-Modus, sodaß der achte Parameter p8 ignoriert wird. Das Umschalten zum alternativen Zeichensatz ist je nachdem, ob man sich gerade darin befindet oder nicht, durch die Sequenz `^N` bzw. `^O` gegeben. Die Sequenz zum Einschalten aller anderen Modi wäre `\E[0;3;4;5;7;8m^N`.

Im folgenden sei auf das Verhalten bei der Ausgabe unterschiedlicher Sequenzen hingewiesen. Beispielsweise wird für den Fall, daß entweder p2 oder p6 gilt, die Zeichenkette `;3` ausgegeben, d.h. wenn entweder Unterstreichen (underline) oder fett/hell (bold) eingeschaltet ist. Die Ausgabe der obigen Sequenzen ergibt mit den zugehörigen Abhängigkeiten folgendes:

Sequenz	auszugeben, wenn	übersetzte Form in terminfo
<code>\E[0</code>	immer	<code>\E[0</code>
<code>;3</code>	falls <code>p2</code> oder <code>p6</code>	<code>%%p2%p6% &t;3%;</code>
<code>;4</code>	falls <code>p1</code> oder <code>p3</code> oder <code>p6</code>	<code>%%p1%p3% &p6% &t;4%;</code>
<code>;5</code>	falls <code>p4</code>	<code>%%p4&t;5%;</code>
<code>;7</code>	falls <code>p1</code> oder <code>p5</code>	<code>%%p1%p5% &t;7%;</code>
<code>;8</code>	falls <code>p7</code>	<code>%%p7&t;8%;</code>
<code>m</code>	immer	<code>m</code>
<code>^N</code> oder <code>^O</code>	falls <code>p9</code> <code>^N</code> , sonst <code>^O</code>	<code>%%p9&t^N%e^O%;</code>

All dies zusammen ergibt in der `sgr` Sequenz:

```
sgr=\E[0%%p2%p6%|&t;3%;%%p1%p3%|&p6%
|&t;4%;%%p5&t;5%;%%p1%p5%
|&t;7%;%%p7&t;8%;m%%p9&t^N%e^O%;,
```

Es sei daran erinnert, daß `sgr` und `sgr0` immer angegeben werden müssen.

Abschnitt 1-8: Tastatur

Hat das Gerät eine Tastatur, die bestimmte Sequenzen überträgt, wenn Tasten gedrückt werden, kann diese Information ebenfalls angegeben werden. Allerdings ist hier zu beachten, daß das nicht für Geräte gilt, deren Tastatur nur lokal wirkt (was beispielsweise für die Tasten ohne Shift beim Hewlett-Packard 2621 zutrifft). Kann die Tastatur zwischen Übertragen und Nicht-Übertragen umgeschaltet werden, ist hierfür `smkx` bzw. `rmkx` anzugeben. Ist ein Umschalten nicht möglich, wird angenommen, daß sich die Tastatur stets im Übertragungsmodus befindet.

Die von den Pfeiltasten für 'nach links', 'nach rechts', 'nach oben', 'nach unten' erzeugten Sequenzen werden mittels `kcub1`, `kcufl`, `kcuu1`, `kcucl`, und `khome` angegeben. Die Sequenzen eventuell vorhandener Funktionstasten wie `f0`, `f1`, ..., `f63` werden mit `kf0`, `kf1`, ..., `kf63` angegeben. Unterscheiden sich die Aufschriften der ersten 11 Funktionstasten von der Voreinstellung `F0` bis `F10`, können diese durch `lf0`, `lf1`, ..., `lf10` angegeben werden. Die von Sondertasten übermittelten Codes werden folgendermaßen angegeben: `kll` (linke obere Ecke, eins nach unten), `kbs` (backspace), `ktbc` (alle Tabulatoren löschen), `ktab` (Tabulator dieser Spalte rücksetzen) `kclr` (Bildschirm oder Taste löschen), `kdch1` (Zeichen löschen), `kdll` (Zeile löschen), `knmir` (Einfügemodus ausschalten), `kel` (Zeilenrest löschen), `ked` (Bildschirmrest löschen), `kich1` (Zeichen oder Wort einfügen), `kill` (Zeile einfügen), `knp` (nächste Seite), `kpp` (vorhergehende Seite), `kind` (vorwärts/abwärts Scrollen), `kri` (rückwärts/aufwärts Scrollen), `khts` (Tabulator an dieser Spalte setzen). Ferner können bei einem 3-mal-3 Funktionstastenblock, zu dem u. a. die vier Pfeiltasten gehören, die übrigen fünf Tasten mit `ka1`, `ka3`, `kb2`, `kc1` und `kc3` angegeben werden. Diese Tasten sind in solchen Fällen nützlich, bei denen die Funktionen eines 3-mal-3 Pfeiltastenfeldes benötigt werden. In der obigen Liste sind auch noch die Eigenschaften weiterer Tasten aufgeführt.

Zeichenketten zur Programmierung von Funktionstasten werden mit `pfkey`, `pfloc`, und `px` angegeben. Eine Zeichenkette zur Programmierung von Bildschirm-Etiketten wird mit `pln` angegeben. Jede dieser Zeichenketten benutzt zwei Parameter, einen Bezeichner für die Funktionstaste und die Zeichenkette, mit der sie zu programmieren ist. `pfkey` bewirkt, daß die Betätigung dieser Taste

gleichbedeutend dazu ist, als ob der Benutzer die entsprechende Zeichenkette eingegeben hätte, `pfloc` übergibt die Zeichenkette zur lokalen Ausführung an das Terminal und `px` überträgt die Zeichenkette zum Rechner. Die Anzahl programmierbarer Bildschirm-Etiketten sowie ihre Breite und Höhe werden mit `nlab`, `lw` und `lh` angegeben. Eventuell existierende Kommandos zum Ein- und Ausschalten dieser Etiketten werden mit `smln` und `rmln` angegeben. Normalerweise wird nach einer oder mehreren Steuersequenzen der Form `pln` die Sequenz `smln` ausgegeben, um sicherzustellen, daß die Änderung am Bildschirm auch sichtbar wird.

Abschnitt 1-9: Tabulatoren und Initialisierung

Bei Terminals, die mit hardwaremäßigen Tabulatoren ausgestattet sind, kann man das Kommando, das ein Vorrücken zum nächsten Tabulator bewirkt, mit `ht` (üblicherweise Control-I) angeben. Das Kommando zum Rücksprung auf den letzten Tabulator ('backtab') kann mit `cbt` angegeben werden. Konventionsgemäß sollte in Programmen, die mit Terminals arbeiten, an denen die Tabulatoren offensichtlich rechnerintern aufgelöst werden und nicht durch das Terminal selbst, weder `ht` noch `cbt` verwendet werden, selbst wenn diese Eigenschaften vorhanden sind, weil die Tabulatoren u. U. vom Benutzer nicht korrekt gesetzt worden sind. Verfügt das Gerät über Hardware-Tabulatoren, die beim Einschalten des Geräts automatisch an jeder *n*-ten Spalte gesetzt werden, so ist mit der numerischen Angabe `it` die Schrittweite der Tabulatoren zu spezifizieren. Von dieser Information ist es normalerweise abhängig, ob das Kommando `tput init` [siehe `tput(1)`] den Modus für hardwaremäßige Tabulatorexpansion setzt oder den Modus 'Tabulatoren setzen' einschaltet. Können die Tabulatoren im nicht-flüchtigen Speicher gesichert werden, so kann in der `terminfo` Beschreibung davon ausgegangen werden, daß sie korrekt gesetzt sind. Eventuell vorhandene Kommandos zum Setzen und Rücksetzen von Tabulatoren werden mit `tbc` (Löschen aller Tabulatorpositionen) und `hts` (in allen Zeilen an der aktuellen Cursorposition einen Tabulator setzen) angegeben.

Zur Initialisierung des Geräts existieren weitere Eigenschaften, die mit `is1`, `is2` und `is3` angegeben werden. Mit `iprog` wird der Pfadname eines Programms zur Initialisierung des Geräts angegeben und in `if` steht der Name einer Datei, die lange Zeichenketten für die Initialisierung enthält. Dabei wird davon ausgegangen, daß diese Zeichenketten das Gerät in Betriebsarten versetzen, die mit der restlichen `terminfo` Beschreibung verträglich sind. Sie müssen bei jedem Login des Benutzers in der folgenden Reihenfolge an das Terminal ausgegeben werden: Initialisierungsprogramm starten mit `iprog`; Ausgabe von `is1`; Ausgabe von `is2`; Randeinstellung setzen mit `mgc`, `smgl` und `smgr`; Tabulatoren setzen mit `tbc` und `hts`; Datei ausgeben mit `if`; und schließlich Ausgabe von `is3`. Üblicherweise wird dies mit Hilfe der `init` Option von `tput` erledigt.

Der größte Teil der Initialisierung wird mit `is2` durchgeführt. Eine Wiederholung von Angaben zum Setzen spezieller Betriebsarten läßt sich dadurch vermeiden, daß man die Sequenzen für den normalen Betrieb in `is2` und diejenigen für Sonderfälle in `is1` und `is3` ablegt. Sequenzen, die ein Rücksetzen eines völlig unbekannten Zustands analog zu `is1`, `is2`, `is3` und `if` veranlassen, können entsprechend mit `rs1`, `rs2`, `rs3` und `rf` angegeben werden. (Die Methode, Dateien mittels `if` und `rf` zu verwenden, trifft nur auf wenige Terminals aus `/usr/share/lib/tabset/*` zu; generell sollten jedoch die Initialisierung und Zeichenketten zum Rücksetzen verwendet werden.) Diese Zeichenketten werden mittels `tput reset` ausgegeben, wenn das Terminal in einen Zustand der "Verklemmung" gerät. Normalerweise

werden Kommandos nur dann in `rs1`, `rs2`, `rs3` und `rf` ausgegeben, wenn sie unerwünschte Auswirkungen auf den Bildschirm haben und beim Login nicht benötigt werden. Beispielsweise ist das Kommando, um ein Terminal in den 80-Spalten-Modus zu versetzen, normalerweise Teil von `is2`. Bei manchen Terminals erzeugt dies jedoch einen unerwünschten Effekt am Bildschirm und ist völlig unnötig, weil sich das Terminal gewöhnlich schon in diesem Zustand befindet.

Falls für das Setzen von Tabulatoren komplexere Sequenzen nötig sind, als sie mittels `tbc` und `hts` beschreibbar sind, können diese in `is2` oder `if` abgelegt werden.

Das Rücksetzen beliebiger Randeinstellungen erfolgt mittels `mgc`. (Zu den Anweisungen, wie Kommandos zum Setzen und Rücksetzen von Tabulatoren angegeben werden, siehe weiter unten "Randeinstellungen" in "Druckereigenschaften".)

Abschnitt 1-10: Verzögerungen

Bestimmte Eigenschaften steuern Verzögerungen im `tty` Treiber. Sie werden hauptsächlich von Hardcopy-Geräten benötigt und von `tput init` zur Einstellung der richtigen Betriebsart verwendet. In den Eigenschaften `cr`, `ind`, `cubl`, `ff` und `tab` angegebene Verzögerungen können zum Setzen der entsprechenden Verzögerungsbits im TTY-Treiber verwendet werden. Diese Angaben sind allerdings vernachlässigbar, falls die Baudrate für Verzögerungen in `pb` angegeben ist und die Werte unterhalb von `pb` liegen.

Abschnitt 1-11: Statuszeilen

Verfügt das Terminal über eine gesonderte "Statuszeile", die normalerweise von Programmen nicht verwendet wird, kann dies mit `hs` angegeben werden; die Statuszeile wird dann als Extrazeile unter der letzten Zeile verstanden, die mit normalen Cursorbewegungen ansprechbar ist (beim Heathkit h19 ist dies die 25. Zeile und beim VT100 bei einem Scroll-Bereich von 23 Zeilen die 24. Zeile). Mit `ts1` und `fs1` gibt man spezielle Steuersequenzen an, die auf bestimmte Positionen in der Statuszeile positionieren oder aus ihr zurückspringen. Nach `fs1` muß der Cursor an derselben Stelle stehen wie vor `ts1`. Falls erforderlich, können zu diesem Zweck `sc` und `rc` in `ts1` und `fs1` integriert werden. `ts1` hat einen Parameter, der angibt, auf welche Spalte der Statuszeile der Cursor positioniert werden soll.

Funktionieren in der Statuszeile auch Escape-Sequenzen und spezielle Kommandos, wie z.B. Tabulatorsprünge, ist hierfür `eslok` anzugeben. Eine Zeichenkette, die die Statuszeile ausschaltet (oder auf andere Weise löscht), sollte mit `dsl` angegeben werden. Eventuell vorhandene Kommandos zum Sichern und Wiederherstellen der Cursorposition werden mit `sc` und `rc` angegeben. Normalerweise wird davon ausgegangen, daß die Statuszeile dieselbe Länge hat wie die restlichen Bildschirmzeilen, z.B. `cols`. Ist dies nicht der Fall (weil das Terminal möglicherweise keine ganze Zeile laden kann), wird die Länge der Statuszeile mit der numerischen Funktion `ws1` angegeben.

Abschnitt 1-12: Zeilengrafik

Verfügt das Gerät über einen alternativen Zeichensatz zum Zeichnen von Linien, wird die Abbildungsvorschrift für die Sonderzeichen durch `acsc` angegeben. Die Definition dieser Zeichenkette basiert auf dem alternativen Zeichensatz des DEC VT100 Terminals und ist um einige Zeichen des AT&T 4410v1 Terminals erweitert.

Bezeichnung	vt100+ Zeichen	97801 Zeichen
Pfeil nach rechts	+	K
Pfeil nach links	,	L
Pfeil nach unten	.	N
rechteckiges Blocksymbol	0	undefiniert
Symbol f. Laterne	I	undefiniert
Pfeil nach unten	-	M
Raute	'	undefiniert
Schachbrettfeld	a	undefiniert
Symbol f. Grad	f	?
plus/minus	g	undefiniert
Feld von Quadraten	h	undefiniert
rechte untere Ecke	j	E
rechte obere Ecke	k	C
linke obere Ecke	l	B
linke untere Ecke	m	D
plus	n	J
Scan-Zeile 1	o	undefiniert
horizontale Linie	q	A
Scan-Zeile 9	s	undefiniert
linkes T-Stück (┌)	t	F
rechtes T-Stück (┐)	u	G
unteres T-Stück (└)	v	I
oberes T-Stück (┘)	w	H
vertikale Linie	x	@
Punkt	~	,

Zur Beschreibung des Grafikzeichensatzes eines weiteren Geräts nimmt man am besten die obige Tabelle und erweitert sie um eine dritte Spalte mit den Zeichen, die nach dem Umschalten in den alternativen Zeichensatz des neuen Geräts die entsprechenden Sonderzeichen erzeugen. Beispiel:

Bezeichnung	vt100+ Zeichen	Zeichen am neuen Terminal
rechte untere Ecke	l	R
rechte obere Ecke	m	F
linke obere Ecke	k	T
linke untere Ecke	j	G
horizontale Linie	q	,
vertikale Linie	x	.

Anschließend werden die einzelnen Zeichen von links nach rechts folgendermaßen hintereinander geschrieben "acsc=1RmFkTjGq\,x.".

Außerdem lassen sich mit `terminfo` auch mehrfache Zeichensätze beschreiben. Zu Einzelheiten hierzu siehe Abschnitt 2-5.

Abschnitt 1-13: Umgang mit Farben

Im folgenden seien zwei Arten des Umgangs mit Farben beschrieben. Die Methode von Tektronix und die von Hewlett-Packard (HP). Bei Tektronix gibt es eine Menge von N vordefinierten Farben (gewöhnlich 8), aus denen der Benutzer die jeweils aktuelle Vordergrund- und Hintergrundfarbe auswählen kann. Damit unterstützt ein solches Terminal bis zu N Farben, aus denen $N*N$ Farbpaare gebildet werden können, die zur gleichen Zeit am Bildschirm dargestellt werden können. Bei HP ist es einem Benutzer nicht möglich, die Vordergrundfarbe unabhängig vom Hintergrund zu wählen, und umgekehrt. Der Benutzer muß stattdessen das komplette Farbpaar auf einmal angeben. Aus $2*M$ verschiedenen Farben können auf diese Weise bis zu M Farbpaare gebildet werden. Die meisten Terminals mit Farbe gehören einer dieser beiden Terminalklassen an.

Mit den numerischen Angaben `colors` bzw. `pairs` werden die Anzahl der gleichzeitig am Bildschirm darstellbaren Farben bzw. Farbpaare spezifiziert. Kann ein Terminal selbsttätig eine Farbdefinition ändern (z.B. Terminals der Serie 4100 und 4200 von Tektronix), sollte dies mit `ccc` (can change color, Farbwechsel möglich) angegeben werden. Zur Änderung einer Farbdefinition (Methode bei Tektronix 4200) wird `initc` (initialize color, Farbe initialisieren) verwendet. Diese Funktion benötigt vier Parameter, eine Farbnummer (aus dem Bereich von 0 bis `colors-1`) und drei RGB-Werte (rot, grün, blau) oder drei HLS-Farben (Hue, Lightness, Saturation; Farbschattierung, Helligkeit, Sättigung). Die Bereiche der RGB- und HLS-Werte hängen vom verwendeten Terminal ab.

Die Terminals der Serie 4100 von Tektronix verwenden ausschließlich HLS-Werte. Bei diesen Terminals oder solchen, die beide Arten kennen, muß man eine boolesche Variable `hls` setzen; dies veranlaßt die `init_color` Routine in `curses` dazu, ihre RGB-Argumente in HLS-Notation umzuwandeln, bevor sie sie an das Terminal schickt. Die letzten drei Argumente der `initc` Steuersequenz wären dann also HLS-Werte.

Falls ein Terminal selbsttätig Farben umdefinieren kann, dafür jedoch eine von RGB oder HLS verschiedene Notation verwendet, muß eine entsprechende Abbildung auf eines der beiden entwickelt werden.

Zum Setzen des Vordergrunds bzw. Hintergrunds auf eine bestimmte Farbe wird `setaf` (set ANSI foreground; ANSI Vordergrund setzen) bzw. `setab` (set ANSI background; ANSI Hintergrund setzen) verwendet. Diese Funktionen benötigen als Parameter die Farbnummer. Bei der HP-Methode wird zur Initialisierung eines Farbpaars die Steuersequenz `initp` (initialize pair) verwendet. Diese Funktion benötigt sieben Parameter, die Nummer des Farbpaares (aus dem Bereich von 0 bis `pairs-1`) und sechs RGB-Werte: die ersten drei für den Vordergrund und die zweiten drei für den Hintergrund. (In beiden Dreiergruppen sollte die Reihenfolge RGB eingehalten werden.) Bei der Verwendung der Steuersequenzen `initc` oder `initp` sollten jeweils die Reihenfolgen "rot, grün, blau" für RGB-Argumente und "hue, lightness, saturation (Farbschattierung, Helligkeit, Sättigung)" für HLS-Argumente beibehalten werden. Um ein Farbpaar zum aktuellen zu machen, wird `scp` (set color-pair; Farbpaar setzen) verwendet. Diese Funktion benötigt als Parameter die Nummer des Farbpaares.

Manche Terminals (und dazu zählen beispielweise die meisten Terminal-Emulatoren für PC's) verwenden die Hintergrundfarbe, um Bildschirmbereiche zu löschen. Hierfür ist `bce` (background color erase; Löschen mit Hintergrundfarbe) zu verwenden. Die Variable `op` (original pair; ursprüngliches Farbpaar) enthält eine Steuersequenz, mit der die Vorder- und Hintergrundfarbe wieder auf den Wert zurückgesetzt werden kann, der zum Zeitpunkt des Einschaltens gültig war. Analog dazu enthält `oc` (original colors; ursprüngliche Farben) eine Steuersequenz, mit der alle Farben (Tektronix-Methode) bzw. alle Farbpaare (HP-Methode) wieder auf die Werte zurückgesetzt werden können, die zum Zeitpunkt des Einschaltens gültig waren.

Manche Terminals verwenden Farben zur Darstellung von Bildschirmattributen; diese Attribute sollten nicht mit anderen Farben kombiniert werden. Die Information zu diesen Bildschirmattributen wird in der Variablen `ncv` untergebracht. Zwischen den neun niederwertigen Bits dieser Variablen und den Bildschirmattributen gibt es eine 1-zu-1 Entsprechung, die in der folgenden Tabelle dargestellt ist.

Attribut	Bit Position	Dezimal- Wert
A_STANDOUT	0	1
A_UNDERLINE	1	2
A_REVERSE	2	4
A_BLINK	3	8
A_DIM	4	16
A_BOLD	5	32
A_INVIS	6	64
A_PROTECT	7	128
A_ALTCHARSET	8	256

Wenn ein bestimmtes Bildschirmattribut nicht mit Farbe kombiniert werden soll, so ist das entsprechende Bit in der Variablen `ncv` auf 1 zu setzen; andernfalls wird es auf 0 gesetzt. Zur Bestimmung des dezimalen Wertes, der in den unteren 9 Bits von `ncv` verschlüsselt wird, geht man folgendermaßen vor. Für alle Bildschirmattribute,

die nicht mit Farbe kombinierbar sind, nimmt man ihre Dezimaläquivalente und addiert sie auf. Im folgenden Beispiel sei angenommen, daß das Terminal zur Darstellung von fett/hell (bold, Bitnummer 5 und Dezimaläquivalent 32) und für die Inversdarstellung (reverse, Bitnummer 2 und Dezimaläquivalent 4) Farben verwendet. Dann ist das entsprechende Resultat für `ncv 36 (32 + 4)`.

Section 1-14: Verschiedenes

Falls das Terminal ein anderes Füllzeichen als das Nullzeichen erfordert, so kann dies mit `pad` angegeben werden. Dabei wird von der Zeichenkette `pad` nur das erste Zeichen verwendet. Verfügt das Terminal über kein Füllzeichen, ist `npc` anzugeben.

Sind am Terminal Positionierungen um eine halbe Zeile nach oben oder unten möglich, ist dies mit `hu` (halbe Zeile nach oben) und `hd` (halbe Zeile nach unten) anzugeben. In erster Linie ist dies an Hardcopy-Geräten für Hoch- und Tiefstellung nützlich. Kann ein Hardcopy-Gerät einen Seitenvorschub ausführen, so ist dies mit `ff` (gewöhnlich Control-L) anzugeben.

Ein eventuell vorhandenes Kommando, um ein Zeichen eine bestimmte Anzahl mal zu wiederholen (wodurch man bei der Übertragung einer großen Anzahl identischer Zeichen Zeit spart), wird mit der parametrisierten Steuersequenz `rep` angegeben. Dabei ist der erste Parameter das zu wiederholende Zeichen und der zweite gibt die Anzahl der Wiederholungen an. So ergibt der Aufruf von `tparam(repeat_char, 'x', 10)` die Zeichenkette `xxxxxxxxxx`.

Arbeitet das Terminal, wie z.B. das Tektronix 4025, mit einem modifizierbaren Befehlszeichen, so wird dieses mit `cmdch` angegeben. Damit definiert man den Prototyp eines Befehlszeichens, der dann in allen folgenden Steuersequenzen verwendet wird. Dabei gilt für manche UNIX-Systeme die folgende Vereinbarung: falls die Umgebungsvariable `CC` definiert ist, werden alle Vorkommen des Prototyp-Befehlszeichens durch deren Inhalt ersetzt.

Beschreibungen von Terminals, die sich nicht auf einen bestimmten Typ bekannter Terminals beziehen, wie z.B. *switch*, *dialup*, *patch* und *network*, sollten die Angabe `gn` (generischer Typ) enthalten. Dies erlaubt es Programmen, zu signalisieren, daß sie nicht wissen, wie sie das Terminal ansprechen sollen. (Dies trifft nicht für virtuelle Terminals zu, deren Escape-Sequenzen ja bekannt sind.) Handelt es sich bei dem Terminal um eines, das vom Protokoll für virtuelle Terminals des SINIX-Systems unterstützt wird, kann dessen Terminalnummer als `vt` angegeben werden. Wenn vor Lesevorgängen eine Steuersequenz für Zeilensprung abgesetzt werden muß, wird dies in `rfl` angegeben.

Bei Terminals, die den `xon/xoff`-Quittungsbetrieb für Flußsteuerung verwenden, ist `xon` anzugeben. Auch hier sollten Verzögerungen als Information für Routinen zur Kostenermittlung angegeben werden, es werden jedoch keine Füllzeichen übertragen. Die Steuersequenzen zum Ein- und Ausschalten des `xon/xoff`-Quittungsbetriebs werden mit `stixon` und `stixon` angegeben. Falls sich die Start-/Stop-Zeichen für den Quittungsbetrieb von `^S` und `^Q` unterscheiden, werden sie durch `xonc` und `xoffc` definiert.

Mit `km` gibt man eine eventuell vorhandene "Meta-Taste" an, die sich wie die Shift-Taste verhält und in allen übermittelten Zeichen das achte Bit setzt. Andernfalls geht die Software davon aus, daß das achte Bit ein Paritätsbit ist und löscht es gewöhnlich. Mit `smm` und `rm` werden die Steuersequenzen zum Ein- und Ausschalten dieses "Meta-Modus" angegeben.

Ist die Anzahl der Zeilen im Speicher des Terminals größer als die Zahl der Zeilen, die auf ein Mal am Bildschirm ausgegeben werden können, so ist die Anzahl der Zeilen im Speicher mit `lm` anzugeben. Hat `lm` den Wert 0, so bedeutet dies, daß die Zeilenzahl im Speicher die des Bildschirms übersteigt, jedoch variabel ist.

Sequenzen für die Steuerung eines an das Terminal angeschlossenen Druckers werden mit `mc0` (Ausdrucken des Bildschirminhalts), `mc4` (Drucker ausschalten) und `mc5` (Drucker einschalten) angegeben. Bei eingeschaltetem Drucker geht jeder an das Terminal übertragene Text auch an den Drucker. Bei der Variante `mc5p` (mit einem Parameter) bleibt der Drucker eingeschaltet, bis die durch den Parameter angegebene Anzahl von Zeichen gedruckt ist und wird dann ausgeschaltet. Dabei darf der Parameter den Wert 255 nicht überschreiten. Wenn bei eingeschaltetem Drucker der Text nicht auf dem Bildschirm ausgegeben wird, muß `mc5i` angegeben werden (stiller Drucker). Solange `mc5p` gilt, wird jedweder Text einschließlich `mc4` in transparenter Form an den Drucker übertragen.

Abschnitt 1-15: Spezialfälle

Das Anwendungsmodell von `terminfo` trifft die meisten Terminals recht passabel. Dennoch kann es vorkommen, daß manche Terminals dadurch nicht vollständig beschrieben werden und zusätzliche Unterstützung durch `terminfo` benötigen. Das ist kein Hinweis auf Fehlfunktionen der Terminals, sondern heißt lediglich, daß es zwischen der Beschreibungsart und der tatsächlichen Hardware bestimmte Unterschiede gibt. Das kann daran liegen, daß es sich um ungewöhnliche Geräte handelt oder daß diese aus bestimmten Gründen nicht über alle Eigenschaften verfügen, die in `terminfo` implementiert sind.

Terminals wie das Hazeltine, die keine Tilde (~) darstellen können, sollten dies durch `hz` anzeigen.

Für Terminals (wie das Concept 100), bei denen ein Zeilenvorschub ignoriert wird, der unmittelbar auf einen `am`-Sprung folgt, sollte `xenl` angegeben werden. Für solche Terminals (wie dem VT100), bei denen der Cursor bei Erreichen der äußersten rechten Spalte dort verharrt, bis das nächste Zeichen ankommt, statt sofort auf die nächste Zeile zu springen, sollte `xenl` angegeben werden.

Kann eine Hervorhebung ('standout') nur durch die Eingabe von `e1` gelöscht werden, (und nicht durch einfaches Überschreiben mit normalem Text), so ist `xhp` anzugeben.

Für Teleray-Terminals, bei denen bei der Positionierung auf einen Tabulator alle passierten Zeichen gelöscht (mit Leerzeichen überschrieben) werden, ist `xt` (löschende Tabulatoren) anzugeben. Eine weitere Bedeutung von `xt` ist, daß der Cursor nicht auf ein "Magic Cookie" positioniert werden kann und der Hervorhebungsmodus daher nur durch Löschen und Einfügen von Zeilen rückgesetzt werden kann.

Bei den BEEHIVE Superbee Terminals, die kein Escape-Zeichen und kein Control-C übertragen, muß zum Zeichen dafür, daß für Escape die Taste f1 und für Control-C die Taste f2 betätigt werden muß, `xsB` angegeben werden.

Abschnitt 1-16: Ähnliche Terminals

Zwei einander sehr ähnliche Terminals können — mit gewissen Einschränkungen — auf dieselbe Art und Weise beschrieben werden. Mit `use` gibt man in der einen Beschreibung den Namen des anderen, ähnlichen Terminals an. Dabei werden die Eigenschaften des mit `use` aufgerufenen Terminals durch diejenigen ersetzt, die vor `use` angegeben sind. Setzt man `xx@` der Definition von `use` voran — wobei `xx` der Code für eine Eigenschaft ist —, so wird diese Eigenschaft dadurch außer Kraft gesetzt. Zum Beispiel definiert die Angabe von

```
att4424-2|Teletype 4424 in Darstellungsgruppe ii,
    rev@, sgr@, smul@, use=att4424,
```

ein AT&T 4424 Terminal, das nicht über die Eigenschaft von `rev`, `sgr`, und `smul` verfügt und daher keine Hervorhebung kann. Dies ist für unterschiedliche Betriebssystemen des Terminals oder für unterschiedliche Benutzerpräferenzen nützlich. Die Angabe von `use` kann mehrfach erfolgen.

TEIL 2: DRUCKER-EIGENSCHAFTEN

Mit der `terminfo`-Datenbank lassen sich Drucker genauso beschreiben wie Terminals. Um herauszufinden, welche Eigenschaften sich sowohl für Drucker als auch für Terminals angeben lassen, siehe die Tabellen unter "Geräte-Eigenschaften", in denen die Eigenschaftsnamen und die Variablennamen dafür aufgeführt sind.

Abschnitt 2-1: Gerundete Werte

Funktionen, die über parametrisierte Zeichenketten beschrieben werden, arbeiten stets mit ganzzahligen Werten. Daher sollten Ersteller von `terminfo`-Beschreibungen dafür sorgen, daß die entsprechenden Zeichenketten so ausgelegt sind, daß sie gerundete Werte erwarten. Entwickler sollten bei der Planung von Anwendungsprogrammen berücksichtigen, daß Werte stets auf den nächstmöglichen ganzzahligen Wert zu runden sind, bevor sie in parametrisierten Zeichenketten verwendet werden.

Abschnitt 2-2: Drucker-Auflösung

Als Auflösung eines Druckers gilt die kleinstmögliche Distanz zwischen zwei Zeichen, die er erzielen kann. Im allgemeinen ist die Auflösung in horizontaler und vertikaler Richtung voneinander unabhängig. Die vertikale Auflösung eines Druckers erhält man, indem man den kleinstmöglichen Abstand zweier aufeinanderfolgender Basislinien beim Drucken mißt, wohingegen die horizontale Auflösung durch den minimal möglichen Abstand zwischen der linken Kante aufeinanderfolgender identischer Zeichen gegeben ist.

Man geht für alle Drucker davon aus, daß sie in der Lage dazu sind, mit einer einheitlichen horizontalen und vertikalen Auflösung zu drucken. Aus der gegenwärtigen Sicht von `terminfo` betrachtet, bedeutet das Drucken die Ausgabe aller Zeichen in einer einheitlichen Matrix, in der alle Zeichen relativ zu festen Positionen der einzelnen "Zellen" dieser Matrix gedruckt werden. Außerdem gilt, daß jede dieser Zellen dieselbe Größe hat, deren Ausmaß durch die horizontale und vertikale Auflösung gegeben ist. (Wie weiter unten erläutert wird, ist diese Größe allerdings kein fester Wert, sondern kann verändert werden.)

Es gibt viele Drucker, die "proportional" ausdrucken können, d.h., daß die Breite des Zwischenraums zwischen zwei Zeichen von der Breite des ersten abhängt. In `terminfo` sind derlei Fähigkeiten zwar nicht berücksichtigt, doch enthält sie ausreichend viele Definitionen von Eigenschaften, um Applikationen die Simulation von Proportionaldruck zu ermöglichen.

Ein Drucker muß nicht nur dazu in der Lage sein, beim Ausdruck die Minimalabstände laut horizontaler und vertikaler Auflösung einzuhalten, sondern auch dazu, den Druckkopf auf eine Position zu setzen, der ein ganzzahliges Vielfaches der kleinsten Distanz von der vorigen Position entfernt ist. Damit ist es möglich, zwischen den gedruckten Zeichen Zwischenräume einzufügen, die im Extremfall die ganze Seitenbreite oder Seitenlänge einnehmen.

Manche Drucker haben in Abhängigkeit verschiedener Betriebsarten unterschiedliche Auflösungen. Im "normalen Modus" gelten die bestehenden Eigenschaften in `terminfo` genau wie bei Terminals als auf Spalten und Zeilen anwendbar. Damit gibt die bereits bekannte Eigenschaft `lines` die Seitenlänge in Zeilen und `cols` die Seitenbreite in Spalten an. Im sogenannten "Mikro-Modus" beziehen sich viele in `terminfo` aufgeführten Eigenschaften auf Teilgrößen von Zeilen und Spalten. Manche Drucker verarbeiten Mikro-Modus und normalen Modus gleichzeitig, so daß in diesem Fall alle Eigenschaften zusammen gelten.

Abschnitt 2-3: Angabe der Drucker-Auflösung

Die Auflösung eines Druckers kann unterschiedlich angegeben werden. Bei jeder davon wird die Auflösung als die Anzahl kleinstmöglicher Schritte pro Distanzeinheit angegeben.

Angabe der Drucker-Auflösung	
Characteristische Anzahl kleinster Schritte	
<code>orhi</code>	horizontale Schritte pro Zoll
<code>orvi</code>	vertikale Schritte pro Zoll
<code>orc</code>	Schritte pro Spalte
<code>orl</code>	Schritte pro Zeile

Wenn im normalen Modus ausgedruckt wird, dann wird der Druckkopf nach jedem Zeichen um eine Spalte weiter versetzt, wobei die Schrittweite der spaltenmäßigen Auflösung entspricht. Spezialfälle als Ausnahme werden weiter unten behandelt. Manche Drucker positionieren den Druckkopf automatisch in die nächste Zeile, wenn in der äußerst rechten Spalte ein Zeichen gedruckt worden ist, wobei die vertikale Schrittweite der zeilenmäßigen Auflösung entspricht. Im Mikro-Modus können diese Schrittweiten anders und bei manchen Druckern gar Null sein.

Angabe der Drucker-Auflösung
Drucken mit automat. Druckkopfbewegung

Normaler Modus:

orc	horizontale Schritte
orl	vertikale Schritte

Mikro-Modus:

mcs	horizontale Schritte
mls	vertikale Schritte

Manche Drucker können breite Zeichen drucken. Dabei kann im normalen Modus der Abstand, um den der Druckkopf weiterbewegt wird, größer sein als beim Drucken von Zeichen normaler Breite. Das Gleiche gilt für den Mikro-Modus. Allerdings gibt es hier einen Zusammenhang: Falls der Abstand für die Weiterbewegung des Druckkopfs beim Drucken normaler Zeichen in beiden Modi gleich ist ($mcs=orc$), so bewegt sich der Druckkopf auch beim Drucken breiter Zeichen in beiden Modi um die gleiche Distanz weiter. Das heißt nicht, daß der Druckkopfversatz bei normalen und breiten Zeichen gleich ist, sondern lediglich, daß die Änderung des Versatzes in beiden Modi gleich groß ist. Unterscheiden sich jedoch die Werte des Druckkopfversatzes je nach eingestelltem Modus, d.h. es gilt ($mcs < orc$), so wird davon ausgegangen, daß der Druckkopfversatz im Mikro-Modus der Breite eines breiten Zeichens in diesem Modus entspricht (vgl. folgende Tabelle).

Angabe der Drucker-Auflösung
Drucken breiter Zeichen mit automat. Druckkopfbewegung

Normaler Modus oder Mikro Modus ($mcs = orc$):

widcs	horizontale Schritte
-------	----------------------

Micro Mode ($mcs < orc$):

mcs	horizontale Schritte
-----	----------------------

Eventuell vorhandene Steuersequenzen zur Änderung des Zeichenabstandes (Anzahl der Spalten pro Zoll) und des Zeilenabstandes (Zeilen pro Zoll) können dazu verwendet werden, die Drucker-Auflösung zu verändern, jedoch ist die Art der Änderung druckerspezifisch.

Angabe der Drucker-Auflösung
Ändern des Zeichen-/Zeilenabstands

cpi	Zeichenabstand ändern, und zwar in orhi oder orc, je nachdem, ob cpix gesetzt ist oder nicht
cpix	
lpi	Zeilenabstand ändern, und zwar in orvi oder orl, je nachdem, ob lpix gesetzt ist oder nicht
lpix	
chr	Schrittweite pro Spalte ändern
cvr	Schrittweite pro Zeile ändern

Die beiden Funktionen `cpi` und `lpi` benötigen jeweils ein Argument, nämlich die Anzahl von Zeichen bzw. von Zeilen pro Zoll. Die beiden Funktionen `chr` und `cvr` benötigen ebenfalls je ein Argument, nämlich die Schrittweite pro Zeichen bzw. pro Zeile.

Die Angabe einer dieser Steuersequenzen bewirkt eine Änderung in manchen der Werte `orc`, `orhi`, `orl` und `orvi`. Außerdem ändert sich der Druckkopfversatz `widcs` bei breiten Zeichen in Abhängigkeit von `orc`. Analog dazu ändert sich der entsprechende Wert `mcs` im Mikro-Modus, wobei folgende Ausnahme gilt: Ist als Distanz 0 oder 1 angegeben, wird keine Änderung vorgenommen (siehe die mit einem † markierten Angaben in der folgenden Tabelle).

Programme, die von `cpi`, `lpi`, `chr` oder `cvr` Gebrauch machen, sollten die Druckerauflösung neu berechnen (ebenso wie andere Werte, die unter "Pixelgrafik" in "Auswirkung von Änderungen der Druckerauflösung" näher beschrieben sind).

Angabe der Drucker-Auflösung Auswirkung bei Änderung des Zeichen-/Zeilenabstands	
<i>vorher</i>	<i>nachher</i>
Verwendung von <code>cpi</code> bei rückgesetztem <code>cpix</code> :	
<code>orhi'</code>	<code>orhi</code>
<code>orc'</code>	$\text{orc} = \frac{\text{orhi}}{V_{cpi}}$
Verwendung von <code>cpi</code> bei gesetztem <code>cpix</code> :	
<code>orhi'</code>	$\text{orhi} = \text{orc} \cdot V_{cpi}$
<code>orc'</code>	<code>orc</code>
Verwendung von <code>lpi</code> bei rückgesetztem <code>lpix</code> :	
<code>orvi'</code>	<code>orvi</code>
<code>orl'</code>	$\text{orl} = \frac{\text{orvi}}{V_{lpi}}$
Verwendung von <code>lpi</code> bei gesetztem <code>lpix</code> :	
<code>orvi'</code>	$\text{orvi} = \text{orl} \cdot V_{lpi}$
<code>orl'</code>	<code>orl</code>
Verwendung von <code>chr</code> :	
<code>orhi'</code>	<code>orhi</code>
<code>orc'</code>	V_{chr}
Verwendung von <code>cvr</code> :	
<code>orvi'</code>	<code>orvi</code>
<code>orl'</code>	V_{cvr}
Verwendung von <code>cpi</code> oder <code>chr</code> :	
<code>widcs'</code>	$\text{widcs} = \text{widcs}' \cdot \frac{\text{orc}}{\text{orc}'}$
<code>mcs'</code>	$\text{mcs} = \text{mcs}' \cdot \frac{\text{orc}}{\text{orc}'}$

Dabei sind V_{cpi} , V_{lpi} , V_{chr} und V_{cvr} die Argumente für `cpi`, `lpi`, `chr` und `cvr`. Mit dem Symbol (') sind jeweils die alten Werte gekennzeichnet.

Abschnitt 2-4: Eigenschaften zur Druckkopfpositionierung

In der folgenden Beschreibung ist mit "Bewegung" die Veränderung der "aktuellen Position" gemeint. Bei Terminals ist dies der Cursor, bei einigen Druckern die Druckposition. Andere Drucker haben andere Entsprechungen. Im allgemeinen

wird unter der aktuellen Position die Position verstanden, an der ein Zeichen erscheint, wenn es gedruckt wird.

In `terminfo` gibt es Zeichenketten-Eigenschaften für Steuersequenzen, mit denen eine Anzahl voller Spalten oder Zeilen übersprungen werden kann. Ebenso gibt es Zeichenketten-Eigenschaften für Steuersequenzen, mit denen Bewegungen um eine Anzahl von kleinsten Einheiten erreicht werden kann.

Druckkopfbewegungen	
<code>mcubl</code>	1 Stelle nach links bewegen
<code>mcuf1</code>	1 Stelle nach rechts bewegen
<code>mcuul</code>	1 Stelle nach oben bewegen
<code>mcudl</code>	1 Stelle nach unten bewegen
<code>mcub</code>	<i>N</i> Stellen nach links bewegen
<code>mcuf</code>	<i>N</i> Stellen nach rechts bewegen
<code>mcuu</code>	<i>N</i> Stellen nach oben bewegen
<code>mcud</code>	<i>N</i> Stellen nach unten bewegen
<code>mhpa</code>	<i>N</i> Stellen vom linken Rand aus bewegen
<code>mvpa</code>	<i>N</i> Stellen vom oberen Rand aus bewegen

Die letzten sechs Zeichenketten werden jeweils mit einem einzigen Argument, nämlich mit *N* verwendet.

In einigen Fällen ist die Bewegung auf einen kleineren Wert als die Seitenbreite oder -länge beschränkt. Außerdem akzeptieren einige Drucker keine absolute Bewegung nach links von der aktuellen Position aus. In `terminfo` gibt es Eigenschaften, um diese Einschränkungen anzugeben.

Einschränkungen für Bewegungen	
<code>mjump</code>	Einschränkung für die Verwendung von <code>mcubl</code> , <code>mcuf1</code> , <code>mcuul</code> , <code>mcudl</code>
<code>maddr</code>	Einschränkung für die Verwendung von <code>mhpa</code> , <code>mvpa</code>
<code>xhpa</code>	Falls gesetzt, kann mit <code>hpa</code> und <code>mhpa</code> nicht nach links bewegt werden
<code>xvpa</code>	Falls gesetzt, kann mit <code>vpa</code> und <code>mvpa</code> nicht nach oben bewegt werden

Für einen Drucker, der im "Mikro-Modus" arbeiten muß, um die oben beschriebenen Bewegungen auszuführen, gibt es Zeichenketten-Eigenschaften, die Steuersequenzen enthalten, mit denen dieser Modus ein- und ausgeschaltet werden kann. Für Drucker, bei denen ein Wagenrücklauf automatisch die Rückkehr in den normalen Modus bewirkt, steht ein boolescher Wert zur Verfügung.

Ein-/Ausschalten des Mikro-Modus	
smicm	Mikro-Modus einschalten
rmicm	Mikro-Modus ausschalten
crxm	Benutzung von cr schaltet den Mikro-Modus aus

Die Bewegung, die ausgeführt wird, wenn ein Zeichen an die äußerste rechte Stelle gedruckt wird, ist für die einzelnen Drucker unterschiedlich. Einige Drucker führen keine Bewegung durch, einige positionieren auf den Beginn der nächsten Zeile, andere positionieren auf den Beginn der gleichen Zeile. In terminfo gibt es boolesche Werte, um alle diese Fälle zu beschreiben.

Was geschieht, nachdem ein Zeichen an die äußerste rechte Position gedruckt wurde

sam	Automatisches Positionieren auf den Beginn der gleichen Zeile
-----	---

Für einige Drucker kann ein Modus angegeben werden, in dem die normale Richtung der Bewegungen vertauscht wird. Dieser Modus ist dann besonders nützlich, wenn keine Eigenschaften für Bewegung nach links oder nach oben zur Verfügung stehen. In diesem Fall können diese Eigenschaften mit Hilfe der umgekehrten Bewegungseigenschaft und den Eigenschaften für die Bewegung nach rechts oder nach unten erzeugt werden. Allerdings ist es besser, die Bewegungen nach links oder nach oben der jeweiligen Anwendung zu überlassen statt sie in der terminfo Datenbank einzutragen. Dadurch können mehrere umgekehrte Bewegungen zusammengefaßt werden, ohne daß dazwischen der Umkehr-Modus ein- und ausgeschaltet werden muß.

Ein-/Ausschalten des Umkehr-Modus	
slm	Richtung für horizontale Bewegungen vertauschen
rlm	Richtung für horizontale Bewegungen wieder herstellen
sum	Richtung für vertikale Bewegungen vertauschen
rum	Richtung für vertikale Bewegungen wieder herstellen

Während die Richtung für horizontale Bewegungen vertauscht ist:

mcubl	1 Stelle nach rechts bewegen
mcuf1	1 Stelle nach links bewegen
mcub	N Stellen nach rechts bewegen
mcuf	N Stellen nach links bewegen
cubl	1 Spalte nach recht bewegen
cuf1	1 Spalte nach links bewegen
cub	N Spalten nach rechts bewegen
cuf	N Spalten nach links bewegen

Während die Richtung für vertikale Bewegungen vertauscht ist:

mcuul	1 Stelle nach unten bewegen
mcudl	1 Stelle nach oben bewegen
mcuu	N Stellen nach unten bewegen
mcud	N Stellen nach oben bewegen
cuul	1 Zeile nach unten bewegen
cudl	1 Zeile nach oben bewegen
cuu	N Zeilen nach unten bewegen
cud	N Zeilen nach oben bewegen

Der Umkehr-Modus sollte sich nicht auf die absoluten Bewegungseigenschaften `mvpa` und `mhp` auswirken. Allerdings vertauscht der Umkehr-Modus auch die Aktionen des "Zeilenumbruchs", der beim Drucken eines Zeichens an der äußerst rechten Stelle wirksam wird. Wenn also für einen Drucker die standardmäßige `terminfo` Eigenschaft `am` angegeben ist, wird auf den Beginn der vorhergehenden Zeile positioniert, nachdem im Umkehr-Modus ein Zeichen an der äußerst rechten Stelle gedruckt wurde.

Wenn irgendeine andere Bewegungs-Eigenschaft im Umkehr-Modus verwendet wird, führt dies zu undefinierten Aktionen. Daher müssen Programme den Umkehr-Modus ausschalten, bevor sie andere Bewegungseigenschaften benutzen.

Zwei weitere Eigenschaften vervollständigen die Liste von neuen Bewegungseigenschaften. Eine dieser Eigenschaften wird für Drucker benötigt, die die aktuelle Position auf den Anfang einer Zeile setzen, wenn bestimmte Kontrollzeichen wie z.B. "line-feed" oder "form-feed," benutzt werden. Die andere wird dazu verwendet, um die Bewegungen zu unterdrücken, die im Normalfall nach dem Drucken eines Zeichens ausgeführt werden.

Zeichenketten für weitere Bewegungen

<code>Locr</code>	Liste der Kontrollzeichen, die <code>cr</code> bewirken
<code>zerom</code>	Keine Druckkopfbewegung nach dem Drucken des nächsten Einzelzeichens

Randeinstellungen

`terminfo` sieht zwei Zeichenketten vor, mit denen Randeinstellungen für Terminals gesetzt werden können: eine für den linken und eine für den rechten Rand. Drucker dagegen haben zwei weitere Begrenzungen für den oberen und den unteren Rand jeder Seite. Darüberhinaus braucht für einige Drucker nicht die aktuelle Position auf eine Begrenzung gesetzt zu werden, damit die Begrenzung dort festgelegt werden kann. Stattdessen kann eine Begrenzung ohne Rücksicht auf die aktuelle Druckposition spezifiziert werden. `terminfo` bietet daher sechs weitere Zeichenketten, um Ränder für Drucker festzulegen.

Ränder festlegen	
<code>smgl</code>	Linke Begrenzung an der aktuellen Spalte festlegen
<code>smgr</code>	Rechte Begrenzung an der aktuellen Spalte festlegen
<code>smgb</code>	Untere Begrenzung auf der aktuellen Zeile festlegen
<code>smgt</code>	Obere Begrenzung auf der aktuellen Zeile festlegen
<code>smgbp</code>	Untere Begrenzung auf Zeile <i>N</i> festlegen
<code>smglp</code>	Linke Begrenzung auf Spalte <i>N</i> festlegen
<code>smgrp</code>	Rechte Begrenzung auf Spalte <i>N</i> festlegen
<code>smgtp</code>	Obere Begrenzung auf Zeile <i>N</i> festlegen

Die letzten vier Zeichenketten werden mit ein oder mehreren Argumenten benutzt, die die Position der existierenden oder festzulegenden Begrenzung angeben. Wenn sowohl `smglp` als auch `smgrp` gesetzt sind, werden beide mit einem einzigen Argument nämlich mit *f2N* benutzt, das die Spaltennummer der jeweils rechten oder linken Begrenzung angibt. Wenn `smgtp` und `smgbp` beide gesetzt sind, wird jede der beiden Zeichenketten benutzt, um die obere und die untere Begrenzung festzulegen: `smgtp` wird nur mit einem einzigen Argument, mit *N*, benutzt, das die Zeilennummer der oberen Begrenzung angibt. `smgbp` wird dagegen mit zwei Argumenten benutzt, nämlich mit *N* und *M*, die beide die Zeilennummer der unteren Begrenzung angeben; der erste Wert gibt den Abstand vom oberen Rand der Seite an, der zweite den Abstand vom unteren Rand der Seite. Damit kann die untere Seitenbegrenzung für die Drucker verschiedener Hersteller auf zwei unterschiedliche Arten spezifiziert werden. Wenn ein `terminfo` Eintrag für einen Drucker gemacht wird, für den eine untere Begrenzung festgelegt werden kann, darf nur entweder der erste oder der zweite Parameter abhängig vom Druckertyp verwendet werden. In einer Anwendung, die `smgbp` benutzt, um die untere Begrenzung festzulegen, müssen beide Argumente angegeben werden.

Wenn von den beiden Werten `smglp` und `smgrp` nur einer gesetzt ist, wird er mit zwei Argumenten benutzt: den Spaltennummern zuerst für die linke und dann die rechte Begrenzung. Entsprechendes gilt, wenn von den Werten `smgtp` und `smgbp` nur einer gesetzt ist. In diesem Fall werden zwei Argumente benutzt, von denen das erste die obere, das zweite die untere Begrenzung angibt, wobei vom oberen Rand der Seite gezählt wird. Wenn also ein `terminfo` Eintrag für einen Drucker gemacht wird, für den sowohl die linke und rechte als auch die obere und untere Begrenzung gleichzeitig anzugeben sind, dürfen nur entweder `smglp` und `smgrp` oder `smgtp` und `smgbp` angegeben werden, die jeweils anderen Werte müssen freigelassen werden. Bei einer Anwendung, die diese Zeichenketten-Eigenschaften benutzt, sollten die jeweiligen Paare zunächst daraufhin überprüft werden, ob jeder der beiden Werte eines Paares oder nur einer der Werte gesetzt ist. Dementsprechend sollten sie dann verwendet werden.

Beim Zählen der Zeilen oder Spalten ist Zeile 0 die oberste Zeile und Spalte 0 die äußerste linke Spalte. Wenn als zweites Argument für `smgbbp` der Wert 0 angegeben ist, bedeutet dies die unterste Zeile der Seite.

Alle Ränder können mit `mgc` zurückgesetzt werden.

Schattendruck, Schrägschrift, Breitschrift, Hoch-, Tiefstellung

Fünf neue Sets von Zeichenketten können zur Beschreibung der Drucker-Eigenschaften für den erweiterten Textdruck benutzt werden.

Erweiterte Druckmöglichkeiten	
<code>sshm</code>	Modus für Schattendruck einschalten
<code>rshn</code>	Modus für Schattendruck ausschalten
<code>sitm</code>	Modus für Schrägschrift einschalten
<code>ritm</code>	Modus für Schrägschrift ausschalten
<code>swidm</code>	Modus für Breitdruck einschalten
<code>rwidm</code>	Modus für Breitdruck ausschalten
<code>ssupm</code>	Modus für Hochstellung einschalten
<code>rsupm</code>	Modus für Hochstellung ausschalten
<code>supcs</code>	Liste hochstellbarer Zeichen
<code>ssubm</code>	Modus für Tiefstellung einschalten
<code>rsubm</code>	Modus für Tiefstellung ausschalten
<code>subcs</code>	Liste tiefstellbarer Zeichen

Falls ein Drucker die Steuersequenz `sshm` vor jedem Zeichen benötigt, das mit Schatten gedruckt werden soll, wird die `rshn` Zeichenfolge freigelassen. Daher müssen Programme, für die eine Steuersequenz in `sshm` aber nicht in `rshn` steht, die Steuersequenz `sshm` vor jedem Zeichen benutzen, das mit Schatten gedruckt werden soll. Oder aber es muß die Steuersequenz `sshm` einmalig vor der Reihe von Zeichen angegeben werden, die mit Schatten gedruckt werden sollen, gefolgt von `rshn`. Das Gleiche gilt für jedes der Paare `sitm/ritm`, `swidm/rwidm`, `ssupm/rsupm` und `ssubm/rsubm`.

Es ist zu beachten, daß `terminfo` auch eine Eigenschaft vorsieht, um fettgedruckten Text (**bold**) zu erzeugen. Da Schattendruck und Fettdruck beide den gedruckten Text "verdunkeln", werden diese beiden Arten des Drucks auf verschiedenen Druckern in jeweils leicht unterschiedlicher Weise erzeugt. Im allgemeinen wird Fettdruck dadurch erreicht, daß das gleiche Zeichen mehrfach gedruckt wird. Schattendruck wird ebenfalls durch Mehrfachdrucken erzeugt, wobei eine leichte Bewegung nach oben und/oder zur Seite ausgeführt wird, um das Zeichen "dicker" erscheinen zu lassen.

Die erweiterten Druckmodi sind voneinander unabhängig, sodaß es möglich wäre, beispielsweise Tiefstellungen in Schrägschrift und Schattendruck darzustellen.

Wie schon weiter oben erwähnt, muß das automatische Weiterschieben nach dem Drucken von Breitzeichen in `widcs` angegeben werden.

Falls nur eine Untermenge von abdruckbaren ASCII-Zeichen in Hoch- oder Tiefstellung gedruckt werden kann, müssen diese Zeichen jeweils in `supcs` oder `subcs` Zeichenketten angegeben werden. Falls die `ssupm` oder `ssubm` Zeichenketten Steuersequenzen enthalten, die entsprechenden `supcs` oder `subcs` Zeichenketten aber leer sind, wird angenommen, daß sämtliche abdruckbaren ASCII-Zeichen für Hoch- oder Tiefstellung zur Verfügung stehen.

Das automatische Weiterschieben nach den Drucken eines hochgestellten oder eines tiefgestellten Zeichens entspricht dem für normale Zeichen. So wird z.B. in jedem der drei folgenden Beispiele nach dem Drucken die gleiche Bewegung ausgeführt:

`B`_i `B`_i `B`ⁱ

Es ist zu beachten, daß die existierende boolesche Eigenschaft `msgr` angibt, ob Steuersequenzen für die Bewegung im "standout mode" benutzt werden dürfen. Diese Eigenschaft ist erweitert worden, um auch die zusätzlichen erweiterten Druckmodi abzudecken. `msgr` sollte für diejenigen Drucker gesetzt werden, die jede Steuersequenz für die Bewegung akzeptieren, ohne daß Schattendruck, Schrägschrift, Breitdruck, Hochstellung oder Tiefstellung davon beeinträchtigt werden. Umgekehrt sollte ein Programm diese Modi ausschalten, bevor eine Bewegung ausgeführt wird, wenn `msgr` nicht gesetzt ist.

Abschnitt 2-5: Alternative Zeichensätze

Zusätzlich zu der Möglichkeit, in `terminfo`-Zeilengrafik zu definieren (siehe Abschnitt 1-12), können auch alternative Zeichensätze angegeben werden. Die folgenden Eigenschaften ermöglichen es, für Drucker und Terminals mehrere Zeichensätze auszuwählen oder zu definieren.

Alternative Zeichensätze	
<code>scs</code>	Zeichensatz <i>N</i> auswählen
<code>scsd</code>	Definition von Zeichensatz <i>N</i> beginnen, <i>M</i> Zeichen
<code>defc</code>	Zeichen <i>A</i> definieren, <i>B</i> Punkte breit, Unterlänge <i>D</i>
<code>rcsd</code>	Definition von Zeichensatz <i>N</i> beenden
<code>csnm</code>	Liste der Namen der Zeichensätze
<code>daisy</code>	Die Typenräder des Druckers werden manuell gewechselt

Die Zeichenketten `scs`, `rcsd` und `csnm` werden mit einem einzigen Argument nämlich mit *N* benutzt, einer Zahl von 0 bis 63, die den Zeichensatz bezeichnet. Die Zeichenkette `scsd` verwendet ebenfalls das Argument *N* sowie das Argument *M*, das die Anzahl der Zeichen im Zeichensatz angibt. Die Zeichenkette `defc` wird mit drei Argumenten benutzt: *A* bezeichnet die ASCII-Abbildung für das Zeichen, *B* gibt die Breite des Zeichens in Punkten an und *D* enthält 0 oder 1 abhängig davon, ob das Zeichen eine Unterlänge hat oder nicht. Der `defc` Zeichenkette folgt eine Zeichenkette von "Rastergrafik" Bytes, die beschreiben, wie das Zeichen aussieht (siehe unten).

Der Zeichensatz 0 ist der vorgegebene Zeichensatz, der nach dem Aktivieren des Druckers vorhanden ist. Natürlich hat nicht jeder Drucker 64 Zeichensätze; wenn `scs` mit einem Argument benutzt wird, das keinen verfügbaren Zeichensatz auswählt, so führt dies zum Ergebnis Null in `tparam`.

Wenn ein Zeichensatz vor seiner Verwendung definiert werden soll, muß die Steuersequenz `scsd` benutzt werden, bevor der Zeichensatz definiert wird und die Steuersequenz `rcsd` nach der Definition. `tparm` wird ebenfalls das Ergebnis Null liefern, wenn für das Argument `N` kein entsprechender Wert angegeben wird. Muß ein Zeichensatz nach seiner Definition noch ausgewählt werden, so muß nach der Steuersequenz `scsd` noch die Steuersequenz `rcsd` angegeben werden. Ein Programm kann feststellen, welche der drei Zeichenketten es benötigt, indem es die Ergebnisse überprüft, die sich nach Verwendung der Zeichenketten `scs`, `scsd` und `rcsd` mit einer Zeichensatznummer in einem `tparm` Aufruf ergeben.

Zwischen den Zeichenketten `scsd` und `rcsd` muß die `defc` Zeichenkette benutzt werden, um jedes Zeichen zu definieren. Um ein Zeichen auf einem von `terminfo` versorgten Drucker auszugeben, wird der ASCII Code an den Drucker geschickt. Dies betrifft sowohl die Zeichen eines alternativen Zeichensatzes als auch "normale" Zeichen. Daher gehört zur Definition eines Zeichens auch der ASCII Code, auf den es abgebildet ist. Zusätzlich wird die Breite des Zeichens in Punkten angegeben zusammen mit einer Kennzeichnung, ob das Zeichen die Druckzeile nach unten überschreitet (wie das z.B. bei dem Kleinbuchstaben "g" in den meisten Zeichensätzen der Fall ist). Die Breite des Zeichens in Punkten zeigt auch die Anzahl von Rastergrafik Bytes an, die auf die Zeichenkette `defc` folgen. Diese Rastergrafik Bytes geben an, welche Rasterpunkte innerhalb des Matrix-Musters mit Tinte angespritzt werden müssen, um das Zeichen abzubilden. Die Anzahl dieser Bytes und ihr Format werden weiter unten unter "Punkt-gesteuerte Grafik" beschrieben.

Am einfachsten ist es für den Erzeuger von `terminfo`-Einträgen, wenn er jeden Zeichensatz über eine Nummer anspricht. Diese Nummern sind allerdings für Anwendungsentwickler nicht besonders aussagekräftig. Mit Hilfe der Zeichenkette `csnm` wird dieses Problem gelöst, mit der für jede Nummer ein Name zur Verfügung gestellt wird.

Wenn die `csnm` Zeichenkette beim Aufruf von `tparm` mit einer Zeichensatz-Nummer verwendet wird, erzeugt die `csnm` Zeichenkette den entsprechenden Namen. Um Zeichensätze anzusprechen, sollten ausschließlich diese Namen benutzt werden. Es gibt keine Namenskonventionen, allerdings sollte jeder, der einen `terminfo`-Eintrag für einen Drucker erzeugt, solche Namen verwenden, die mit den Namen in der Benutzerdokumentation für Drucker konsistent sind. Anwendungsentwickler sollten dem Benutzer die Möglichkeit geben, Zeichensätze sowohl über eine Nummer anzugeben (wobei es dem Benutzer überlassen bleibt, die korrekte Nummer durch Prüfen der `csnm` Zeichenkette zu ermitteln) als auch über den Namen, wobei die Anwendung die entsprechende Zeichensatz-Nummer durch Prüfen der `csnm` Zeichenkette ermittelt.

Diese Eigenschaften werden wohl ausschließlich für Punkt-Matrix-Drucker benutzt. Falls sie nicht zur Verfügung stehen, dürfen die entsprechenden Zeichenketten nicht angegeben werden. Für Drucker, bei denen die Typenräder manuell gewechselt werden oder die Kassetten für Zeichensätze haben, ist der boolesche Wert `daisy` gesetzt.

Abschnitt 2-6: Punkt-Matrix-Grafik

Punkt-Matrix-Drucker haben üblicherweise die Eigenschaft, "rastergrafische" Bilder zu reproduzieren. Es gibt drei neue numerische Eigenschaften und drei neue Zeichenketten-Eigenschaften, die beim Zeichnen eines rastergrafischen Bildes hilfreich sein können, unabhängig vom Typ des Punkt-Matrix-Druckers oder der Anzahl von Nadeln oder Punkten, die ein Drucker gleichzeitig bedienen kann.

Punkt-Matrix-Grafik	
<code>npins</code>	Anzahl der Nadeln, N , im Druckkopf
<code>spinv</code>	Vertikaler Nadelabstand in Nadeln pro Inch
<code>spinh</code>	Horizontaler Punktabstand in Punkten pro Inch
<code>porder</code>	Bildet Software-Bits auf Druckkopf-Nadeln ab
<code>sbim</code>	Druck von Rastergrafik starten (Breite B Bits)
<code>rbim</code>	Druck von Rastergrafik beenden

Die Zeichenkette `sbim` wird mit einem einzigen Argument benutzt, nämlich mit B , der Breite des Bildes in Punkten.

Das Modell der Punkt-Matrix oder Raster-Grafik, das `terminfo` bietet, entspricht der Technik der meisten Punkt-Matrix-Drucker: jeder Durchlauf des Druckkopfs erzeugt eine Punkt-Matrix, die N Punkte hoch und B Punkte breit ist. Üblicherweise ist dies ein breites, geschlossenes Rechteck aus Punkten. Die Höhe dieses Rechtecks aus Punkten variiert von einem Drucker zum nächsten; sie wird mit der numerischen Eigenschaft `npins` angegeben. Die Größe des Rechtecks wird in Zoll-Bruchteilen angegeben und variiert ebenfalls; sie kann aus den numerischen Eigenschaften `spinv` und `spinh` abgeleitet werden. Mit Hilfe dieser drei Werte kann eine Anwendung ein komplettes rastergrafisches Bild in mehrere horizontale Streifen unterteilen, wobei u.U. zur Berechnung der unterschiedlichen vertikalen und horizontalen Punktdistanzen eine Interpolation nötig ist.

Die Zeichenketten `sbim` und `rbim` werden verwendet, um ein Punkt-Matrix-Bild zu starten oder zu beenden. Die Zeichenkette `sbim` wird mit einem einzigen Argument benutzt, das die Breite der Punkt-Matrix in Punkten angibt. Nach der `sbim` Zeichenfolge und vor der `rbim` Zeichenfolge wird eine Reihe von "Rastergrafik Bytes" an den Drucker geschickt. Die Anzahl der Bytes ist ein ganzzahliges Mehrfaches der Breite der Punkt-Matrix. Der Mehrfachwert und das Format jedes Bytes wird durch die `porder` Zeichenkette wie unten beschrieben festgelegt.

Die `porder` Zeichenkette ist eine Liste, in der durch Kommata getrennt die Nummern der Nadeln und danach wahlweise eine numerische Distanzangabe erfolgt. Ist eine Distanzangabe vorgegeben, so wird sie durch ein Semikolon von der Liste getrennt. Die Position einer jeden Nadel-Nummer innerhalb der Liste entspricht einem Bit in einem 8-Bit-Datenbyte. Die Nadeln sind fortlaufend von 1 bis `npins` durchnummeriert, wobei 1 die oberste Nadel ist. Es ist zu beachten, daß die Bezeichnung Nadel hier sehr frei verwendet wird; Tintenstrahl-Punkt-Matrix-Drucker haben keine Nadeln, verwenden aber eine entsprechende Methode, um

einen einzelnen Tintenpunkt aufs Papier zu bringen. Die Bit-Positionen in `porder` sind in Gruppen zu acht zusammengefaßt, wobei die erste Stelle in einer Gruppe das höchwertigste Bit und die letzte Stelle das niederwertigste Bit enthält. Eine Anwendung erzeugt 8-Bit-Bytes in der Reihenfolge der Gruppen in `porder`.

Eine Anwendung errechnet die "Rastergrafik bytes" anhand der internen Abbildung, indem sie die vertikalen Punkt-Positionen bei jedem Durchlauf des Druckkopfs auf 8-Bit-Bytes abbildet. Dabei bedeutet ein auf 1 gesetztes Bit, daß Tinte abgegeben werden soll, ein auf 0 gesetztes Bit bedeutet, daß keine Tinte abgegeben wird. Dies kann vertauscht werden (0 Bit für Tinte, 1 Bit für keine Tinte), indem eine negative Nadel-Nummer angegeben wird. Falls eine Stelle in `porder` ausgelassen ist, wird ein Wert 0 für das Bit verwendet. Falls an einer Stelle ein kleines `x` anstelle einer Nadel-Nummer steht, wird der Wert 1 für das Bit verwendet. Aus Gründen der Konsistenz kann auch ein kleines `o` benutzt werden, um ein ausgelassenes 0-wertiges Bit darzustellen. Es muß ein Mehrfaches der 8 Bit-Positionen in `porder` entweder benutzt oder übergangen werden. Andernfalls wird das niederwertigste Bit im letzten Byte auf 0 gesetzt. Falls der Relativzeiger angegeben ist, wird er jedem Datenbyte angefügt; der Relativzeiger kann negativ sein.

Einige Beispiele sollen die Verwendung der `porder` Zeichenkette verdeutlichen. Die AT&T 470, AT&T 475 und C.Itoh 8510 Drucker sehen 8 Nadeln für Grafik vor. Diese 8 Nadeln werden von oben nach unten durch die 8 Bit eines Bytes bezeichnet und zwar vom niederwertigsten zum höchstwertigsten. Die `porder` Zeichenkette für diese Drucker wäre 8,7,6,5,4,3,2,1. Die AT&T 478 und AT&T 479 Drucker sehen ebenfalls acht Nadeln für Grafik vor. Allerdings werden diese Nadeln in umgekehrter Reihenfolge bezeichnet. Die `porder` Zeichenkette für diese Drucker wäre 1,2,3,4,5,6,7,8. Die AT&T 5310, AT&T 5320, DEC LA100 und DEC LN03 Drucker sehen sechs Nadeln für Grafik vor. Die Nadeln werden von oben nach unten durch die Dezimalwerte 1, 2, 4, 8, 16 und 32 bezeichnet. Diese entsprechen den sechs niederwertigen Bits in einem 8-Bit-Byte, obwohl die Dezimalwerte um die Distanz 63 versetzt sind. Die `porder` Zeichenkette für diese Drucker wäre entweder , ,6,5,4,3,2,1;63 oder o,o,6,5,4,3,2,1;63.

Abschnitt 2-7: Auswirkung von Änderungen der Druckauflösung

Wenn die Steuersequenzen zur Änderung des Zeichen- oder Zeilenabstands verwendet werden, kann sich der Nadel- oder Punktabstand ändern:

Punkt-Matrix-Grafik	
Ändern der Zeichen/Zeilenposition	
<code>cp</code>	Zeichenabstand ändern
<code>cpix</code>	Falls gesetzt, wird <code>spinh</code> von <code>cp</code> geändert
<code>lp</code>	Zeilenabstand ändern
<code>lpix</code>	Falls gesetzt, wird <code>spinv</code> von <code>lp</code> geändert

Programme, die `cp` oder `lp` verwenden, sollten den Punktabstand neu berechnen.

Punkt-Matrix-Grafik	
Auswirkungen von Änderungen des Zeichen-/Zeilenabstands	
Vorher	Nachher
Verwendung von <code>cpi</code> bei rückgesetzten <code>cpix</code> :	
<code>spinh'</code>	<code>spinh</code>
Verwendung von <code>cpi</code> bei gesetztem <code>cpix</code> :	
<code>spinh'</code>	$\text{spinh} = \text{spinh}' \cdot \frac{\text{orhi}}{\text{orhi}'}$
Verwendung von <code>lpi</code> bei rückgesetzten <code>lpix</code> :	
<code>spinv'</code>	<code>spinv</code>
Verwendung von <code>lpi</code> bei gesetztem <code>lpix</code> :	
<code>spinv'</code>	$\text{spinv} = \text{spinv}' \cdot \frac{\text{orhi}}{\text{orhi}'}$
Verwendung von <code>chr</code> :	
<code>spinh'</code>	<code>spinh</code>
Verwendung von <code>cvr</code> :	
<code>spinv'</code>	<code>spinv</code>

`orhi'` und `orhi` sind die Werte für die horizontale Auflösung in Stellen pro Inch, die vor der Verwendung von `cpi` und nach der Verwendung von `cpi`, durchgeführt wird. Gleichmaßen sind `orvi'` und `orvi` die Werte für die vertikale Auflösung in Stellen pro Inch, die vor der Verwendung von `lpi` und nach der Verwendung von `lpi`, durchgeführt wird. Demnach entspricht eine Veränderung der Punkte pro Inch bei Punkt-Matrix-Grafik den Änderungen der Stellen pro Inch bei der Druckauflösung.

Abschnitt 2-8: Druckqualität

Bei vielen Punkt-Matrix-Druckern kann der Punktabstand für gedruckten Text dahingehend geändert werden, daß NLQ-Schrift (nahezu "Briefqualität") oder aber Entwurfsschrift produziert wird. Normalerweise ist es wichtig, daß zwischen diesen beiden Möglichkeiten ausgewählt werden kann, denn mit der Verbesserung der Qualität steigt die Druckrate. Es stehen drei neue Zeichenketten zur Verfügung, um diese Eigenschaften zu beschreiben.

	Druckqualität
<code>snlq</code>	NLQ-Modus einstellen
<code>snrmq</code>	normalen Modus einstellen
<code>sdrfq</code>	Entwurfs-Modus einstellen

Die Eigenschaften sind in abnehmender Druckqualität aufgeführt. Falls ein Drucker nicht über alle drei Qualitätsstufen verfügt, sollten ein oder zwei Qualitätsstufen entsprechend freigelassen werden.

Abschnitt 2-9: Druckgeschwindigkeit und Puffergröße

Aufgrund der Tatsache, daß es keine Standardprotokolle gibt, mit denen ein Programm sich permanent mit einem Drucker synchron halten kann und daß moderne Drucker Daten puffern, bevor sie sie ausdrucken, ist es im allgemeinen nicht möglich, daß ein Programm zu einem beliebigen Zeitpunkt bestimmen kann, was bereits ausgedruckt worden ist. Um das einigermaßen abschätzen zu können, sind zwei neue numerische Werte eingeführt worden.

Druckgeschwindigkeit/Puffergröße	
cps	Nominelle Druckgeschwindigkeit in Zeichen pro Sekunde
bufsz	Pufferkapazität (Anzahl der Zeichen)

Mit cps wird die nominelle bzw. die durchschnittliche Geschwindigkeit angegeben, mit der der Drucker Zeichen ausdruckt. Ist dieser Wert nicht angegeben, sollte die Geschwindigkeit mit einem Zehntel der aktuellen Übertragungsrate angesetzt werden. Unter der Voraussetzung, daß eine geeignete Flußsteuerung verwendet wurde, gibt cps die Anzahl der Zeichen an, die nach dem sichergestellten Ausdruck eines Zeichens noch gepuffert werden können. Ist dieser Wert nicht angegeben, wird davon ausgegangen, daß der Drucker keine Zeichen puffert, sondern sie sofort beim Empfang ausdruckt.

Wenn beispielsweise ein Drucker einen Puffer hat, der 1000 Zeichen faßt, so ist dadurch garantiert, daß bei der Übertragung des Zeichens "a", dem 1000 weitere folgen, der Buchstabe "a" auch ausgedruckt wird. Falls derselbe Drucker mit einer Geschwindigkeit von 100 Zeichen pro Sekunde druckt, braucht er 10 Sekunden zum Ausdruck aller Zeichen im Puffer (und weniger, wenn er nicht voll ist). Wenn ein Programm die Puffergröße und die Druckgeschwindigkeit des Druckers kennt und außerdem Protokoll über die übertragenen Daten führt, kann es sich leicht mit dem Drucker synchronisieren.

Es ist zu beachten, daß die meisten Hersteller von Druckern die maximale Übertragungsrate empfehlen, und nicht die nominelle. Eine recht passable Methode, einen Wert für cps zu ermitteln, besteht darin, einige Seiten Text zu erzeugen, die Anzahl der darin enthaltenen Zeichen zu ermitteln und zu prüfen, wie lang der Ausdruck dauert.

Anwendungsprogramme, die diese Werte benutzen, sollten die Tatsache der variablen Druckgeschwindigkeit im Auge behalten. Einfacher Text in kurzen Zeilen ohne eingestreute Escape-Sequenzen wird wohl ziemlich genau mit der angepriesenen Druckgeschwindigkeit ausgegeben und ist wahrscheinlich schneller als der durch cps angegebene Wert. Grafische Daten mit großen Mengen von Escape-Sequenzen oder sehr langen Zeilen brauchen sicher mehr Zeit und bleiben deutlich unter dem Wert in cps. Falls ein Anwendungsprogramm den Wert cps dazu benutzt, um abzuschätzen, wie lange ein Textblock zum Ausdruck braucht, sollte es einen Sicherheitsfaktor addieren. Falls es den gleichen Wert dazu benutzt, um zu bestimmen, wieviel Text bereits ausgedruckt worden ist, sollte es einen Sicherheitsfaktor subtrahieren. Auf diese Weise kann sich das Anwendungsprogramm höchstens zu Gunsten des Benutzers irren, dem es nur darum geht, daß der Ausdruck korrekt verläuft.

DATEIEN

<code>/usr/share/lib/terminfo/?/*</code>	Datenbank mit der übersetzten Terminalbeschreibung
<code>/usr/share/lib/.COREterm/?/*</code>	Untermenge der Datenbank mit der übersetzten Terminalbeschreibung
<code>/usr/share/lib/tabset/*</code>	Tabulatorpositionen für bestimmte Terminals in einem Format, wie es an Terminals ausgegeben werden kann (Escape-Sequenzen zum Setzen von Tabulatoren und Randeinstellungen)

SIEHE AUCH

`curses(3X)`, `ls(1)`, `pg(1)`, `printf(3S)`, `stty(1)`, `tic(1M)`, `tput(1)`, `tty(1)`, `vi(1)`.

HINWEIS

Die effektivste Methode, eine Beschreibung zu erstellen, besteht darin, die Beschreibung eines ähnlichen Terminals in `terminfo` zu kopieren, die neue schrittweise aufzubauen und Teilbeschreibungen mit `vi` auf Fehler hin zu überprüfen. Eine neue Terminalbeschreibung läßt sich auf einfache Weise testen, indem man die Umgebungsvariable `TERMINFO` mit dem Pfadnamen des Verzeichnisses belegt, das die übersetzte Beschreibung enthält; dann greifen Programme auf dieses Verzeichnis zu statt auf `/usr/share/lib/terminfo`.

timezone(4)

timezone(4)

BEZEICHNUNG

timezone – Einstellung der Standard-Systemzeit-Zone

ÜBERSICHT

/etc/TIMEZONE

BESCHREIBUNG

Diese Datei setzt die Umgebungsvariable TZ für die Zeitzone und exportiert sie.

Diese Prozedur wird von anderen Prozeduren verwendet, die die Zeitzone kennen müssen.

BEISPIEL

/etc/TIMEZONE für die Ostküste:

```
#      Zeitzone
TZ=EST5EDT
export TZ
```

SIEHE AUCH

ctime(3C), environ(5).

rc2(1M), profile(4) im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

ts_dptbl – Parametertabelle für die Time-Sharing Dispatcher

ÜBERSICHT

```
#include <sys/ts.h>
```

BESCHREIBUNG

Der Prozeß-Scheduler (oder Dispatcher) ist der Teil des Systemkerns, der die Belegung der CPU durch Prozesse kontrolliert. Der Scheduler verwendet Zuteilungsklassen. Jeder Klasse entspricht eine Zuteilungsmethode, die verwendet wird, um den Prozessen innerhalb dieser Klasse CPU-Zeit zuzuteilen. Verbunden mit jeder Zuteilungsklasse sind verschiedene Warteschlangen, in denen startbereite Prozesse stehen. Diesen Warteschlangen werden durch die System-Konfiguration globale Zuteilungsprioritäten zugeordnet, die für Prozesse innerhalb einer Klasse verfügbar sind (der Dispatcher wählt zur Ausführung immer den Prozeß mit der höchsten globalen Priorität im System aus). Die mit einer vorgegebenen Klasse verbundenen Warteschlangen werden an Hand dieser Klasse als ein lückenloser Satz von Prioritätsebenen betrachtet, die von 0 (niedrigste Priorität) bis n (höchste Priorität) numeriert sind. Der Wert für die höchste Priorität ist konfigurationsabhängig. Die globalen Zuteilungsprioritäten, die einer Warteschlange mit einer vorgegebenen Klasse zugeordnet werden, müssen nicht bei Null anfangen und dürfen Lücken enthalten (abhängig von der Konfiguration).

Prozesse in der Time-Sharing-Klasse, die im Benutzer-Modus laufen (oder im Systemkern-Modus, bevor sie einschlafen), werden entsprechend den Parametern in der Parametertabelle für den Time-Sharing Dispatcher (ts_dptbl) zugeteilt. (Time-Sharing-Prozesse, die nach dem Schlafen im Systemkern-Modus laufen, laufen innerhalb eines besonderen Bereichs von Prioritäten, die für solche Prozesse reserviert sind. Sie sind nicht durch die Parameter in ts_dptbl betroffen, bis sie in den Benutzer-Modus zurückkehren). ts_dptbl besteht aus einem Feld von Parameter-Strukturen (struct ts_dpent), je eine für jede der n Prioritätsebenen, die von Time-Sharing-Prozessen im Benutzer-Modus verwendet werden. Die Eigenschaften einer vorgegebenen Prioritätsebene i sind angegeben durch die i te Parameter-Struktur in diesem Feld (ts_dptbl[i]).

Eine Parameter-Struktur besteht aus den folgenden Elementen. Diese sind auch in der Include-Datei /usr/include/sys/ts.h beschrieben.

ts_globpri	Die mit dieser Prioritätsebene verbundene globale Zuteilungspriorität. Die Zuordnung zwischen Time-Sharing-Prioritätsebenen und globalen Zuteilungsprioritäten wird beim Hochfahren durch die System-Konfiguration festgelegt. ts_globpri ist das einzige Element der ts_dptbl, das nicht mit dispadm(1M) geändert werden kann.
ts_quantum	Das Zeitlimit in Takten (HZ), das Prozessen auf dieser Ebene zugeteilt wird.
ts_tqexp	Prioritätsebene der neuen Warteschlange, auf die ein auf der aktuellen Ebene laufender Prozeß zu setzen ist, wenn er sein Zeitlimit überschreitet. Normalerweise wird dieses Feld mit einer niedrigeren Time-Sharing-Prioritätsebene verbunden, die eine längeres Zeitlimit hat.

ts_dptbl(4)

ts_dptbl(4)

<code>ts_slpret</code>	Prioritätsebene der neuen Warteschlange, auf die ein Prozeß, der vorher im Benutzer-Modus auf dieser Ebene war, zu setzen ist, wenn er nach dem Schlafen zum Benutzer-Modus zurückkehrt. Normalerweise wird dieses Feld mit einer höheren Prioritätsebene verbunden, die eine kürzeres Zeitlimit hat.
<code>ts_maxwait</code>	Ein Prozeß-Zähler; <code>ts_dispwait</code> wird jedesmal auf Null gesetzt, wenn ein Time-Sharing-Prozeß in die Dispatcher-Warteschlange zurückgestellt wird, nachdem sein Zeitlimit abgelaufen ist, oder wenn er erwacht ist (<code>ts_dispwait</code> wird nicht auf Null zurückgesetzt, wenn ein Prozeß durch einen Prozeß mit höherer Priorität verdrängt wird). Dieser Zähler wird einmal pro Sekunde für jeden Prozeß in der Dispatcher-Warteschlange erhöht. Wenn der <code>ts_dispwait</code> -Wert eines Prozesses den <code>ts_maxwait</code> -Wert für seine Ebene überschreitet, wird die Priorität des Prozesses auf den Wert geändert, der durch <code>ts_lwait</code> angegeben ist. Dieses Feld hat den Zweck, eine permanente Verdrängung von Prozessen zu verhindern
<code>ts_lwait</code>	Verschiebt einen Prozeß auf diese neue Prioritätsebene, wenn <code>ts_dispwait</code> größer ist als <code>ts_maxwait</code> .

Der Systemverwalter kann das Verhalten des Time-Sharing-Anteils des Schedulers durch Neukonfigurieren der `ts_dptbl` beeinflussen. Hierfür sind zwei Methoden verfügbar.

DISPADMIN Konfigurationsdatei

Mit Ausnahme von `ts_globpri` können alle Elemente der `ts_dptbl` auf einem laufenden System durch den Befehl `dispadmin(1M)` geprüft und verändert werden. Der Aufruf von `dispadmin` für die Time-Sharing-Klasse erlaubt dem Systemverwalter die aktuelle `ts_dptbl`-Konfiguration der Tabelle im Hauptspeicher zu laden oder diese mit Werten aus einer Konfigurationsdatei zu überschreiben. Die für Eingaben in `dispadmin` verwendete Konfigurationsdatei muß mit dem unten beschriebenen Format übereinstimmen.

Leerzeilen werden ignoriert und ein beliebiger Teil einer Zeile rechts von einem `#`-Symbol wird als Kommentar behandelt. Die erste Zeile, die weder Kommentar noch leer ist, muß die Auflösung anzeigen, die zur Interpretation des Zeitlimit-Wertes benutzt wird. Die Auflösung ist angegeben als

`RES=res`

wobei `res` eine positive Ganzzahl zwischen 1 und 1.000.000.000 und die verwendete Auflösung der Umkehrwert von `res` in Sekunden ist (beispielsweise bezeichnet `RES=1000` eine Auflösung in Millisekunden). Obgleich eine sehr genaue (Nanosekunden) Auflösung angegeben sein kann, werden die Längen des Zeitlimits auf das nächste ganzzahlige Vielfache der Systemuhr-Auflösung aufgerundet. Beispielsweise ist die höchste gegenwärtig auf der 3B2 verfügbare Auflösung 10 Millisekunden (1 "Takt"). Wäre `res` 1000, würde ein Zeitlimit-Wert von 34 eine Einheit von 34 Millisekunden benennen, der auf der 3B2 auf 4 Takte (40 Millisekunden) aufgerundet würde.

ts_dptbl(4)

ts_dptbl(4)

Die restlichen Zeilen in der Datei werden verwendet, um die Parameterwerte für jede Time-Sharing-Prioritätsebene anzugeben. Die erste Zeile bezeichnet die Parameter für Time-Sharing-Ebene 0, die zweite Zeile bezeichnet die Parameter für Time-Sharing Ebene 1, usw. Für jede konfigurierte Time-Sharing-Prioritätsebene muß genau eine Zeile vorhanden sein.

BEISPIEL

Der folgende Auszug einer `dispadmin`-Konfigurationsdatei veranschaulicht das Format. Es ist zu beachten, daß es für jede Zeile, die ein Zeitlimit spezifiziert, einen Kommentar gibt, der auf die entsprechende Prioritätsebene hinweist. Diese Ebenen-Nummern zeigen die Priorität innerhalb der Time-Sharing-Klasse an. Die Zuordnung zwischen diesen Time-Sharing-Prioritäten und den entsprechenden globalen Zuteilungsprioritäten wird durch die in der `ts`-Masterdatei angegebene Konfiguration festgelegt. Genaugenommen dienen die Ebenen-Nummern dem Komfort des Systemverwalters beim Lesen der Datei und werden von `dispadmin` ignoriert, wie auch jeglicher Kommentar. `dispadmin` geht davon aus, daß die Zeilen in der Datei in aufeinanderfolgender, aufsteigender Reihenfolge von Prioritätsebenen (von 0 bis zur maximalen konfigurierten Echtzeit-Priorität) geordnet sind. Die Ebenen-Nummern in den Kommentaren sollten normalerweise mit dieser Anordnung übereinstimmen; wenn dies aus irgendeinem Grund nicht der Fall ist, hat das jedoch keine Auswirkung auf `dispadmin`.

Konfigurationsdatei für die Time-Sharing-Zuteilerroutine RES=1000

#	ts_quantum	ts_tqexp	ts_slpret	ts_maxwait	ts_lwait	Prioritätsebene
	500	0	10	5	10	# 0
	500	0	11	5	11	# 1
	500	1	12	5	12	# 2
	500	1	13	5	13	# 3
	500	2	14	5	14	# 4
	500	2	15	5	15	# 5
	450	3	16	5	16	# 6
	450	3	17	5	17	# 7
	.	.	.	.	.	. .
	.	.	.	.	.	. .
	.	.	.	.	.	. .
	50	48	59	5	59	# 58
	50	49	59	5	59	# 59

DATEIEN

/usr/include/sys/ts.h

SIEHE AUCH

`dispadmin(1M)`, `priocntl(1)`, `priocntl(2)`,

Kapitel "Prozeß-Scheduler" im *Leitfaden für Systemverwalter*

ts_dptbl(4)

ts_dptbl(4)

HINWEIS

dispadmin führt einige begrenzte Plausibilitätskontrollen der Werte aus der Konfigurationsdatei durch. Die Plausibilitätskontrolle ist vorgesehen, um sicherzustellen, daß die neuen ts_dptbl-Werte das System nicht zum Absturz bringen. Die Plausibilitätskontrolle versucht nicht zu analysieren, welche Wirkung die neuen Werte auf den Durchsatz des Systems haben werden. Ungewöhnliche ts_dptbl-Konfigurationen können einen sehr negativen Einfluß auf den Durchsatz haben.

ts_dptbl-Werte, die in der ts-Masterdatei angegeben sind, werden nicht auf Plausibilität kontrolliert. Die Spezifizierung einer widersprüchlichen oder unsinnigen ts_dptbl-Konfiguration durch die ts-Masterdatei könnte ernsthafte Durchsatz-Probleme verursachen und/oder das System zum Absturz bringen.

BEZEICHNUNG

ttsrch — Suchliste von Verzeichnissen für ttyname

ÜBERSICHT

/etc/ttsrch

BESCHREIBUNG

ttsrch ist eine optionale Datei, die von der ttyname-Bibliothek-Routine benutzt wird. Diese Datei enthält die Namen von Verzeichnissen in /dev, die Terminaldateien enthalten und solche von terminalabhängigen Geräten. Der Zweck dieser Datei ist, den Durchsatz von ttyname zu verbessern. Dies geschieht durch den Hinweis, welche Unterverzeichnisse in /dev terminalabhängige Gerätedateien enthalten und zuerst durchsucht werden sollen. Diese Unterverzeichnis-Namen müssen auf separaten Zeilen erscheinen und mit /dev beginnen. Pfadnamen, die nicht mit /dev beginnen, werden ignoriert und auf der Konsole wird eine Warnung ausgegeben. Leerzeilen (Zeilen, die nur Leerzeichen enthalten) und Zeilen, die mit dem Kommentarzeichen "#" beginnen, werden ignoriert. Für jede aufgelistete Datei (ausgenommen mit dem besonderen Eintrag /dev), durchsucht ttyname rekursiv die Unterverzeichnisse nach einer Übereinstimmung. Wenn /dev in der ttsrch-Datei erscheint, wird das /dev-Verzeichnis selbst durchsucht, jedoch keine rekursive Durchsuchung seiner Unterverzeichnisse durchgeführt.

ttyname durchsucht die Geräte-Dateien nach einer Datei, deren Geräteklasse/Gerätenummer, Dateisystem-Bezeichner und I-Node-Nummer mit dem Dateideskriptor übereinstimmen, der als Argument vorgegeben wurde. Wenn keine übereinstimmende Datei gefunden wurde, wird sie nur noch nach Geräteklasse/Gerät und Dateisystem-Bezeichner verglichen, wenn ein Dateisystem-Bezeichner gefunden werden kann. Ist jedoch der Dateideskriptor mit einem geklonten Gerät verbunden (siehe clone(7)), funktioniert dieser Algorithmus nicht. Das kommt daher, daß die I-Node-Nummer der Geräte-Datei, die mit einem klonbaren Gerät verbunden ist, niemals mit der I-Node-Nummer des Dateideskriptors übereinstimmt, der von dem Open-Aufruf des klonbaren Geräts zurückgegeben wird. Um in diesen Situationen Abhilfe zu schaffen, können Einträge in der Datei /etc/ttsrch vorgenommen werden, die den Durchsatz verbessern, wenn geklonte Geräte als Terminals in einem System verwendet werden (z.B. für Fern-Login). Dies ist jedoch nur nützlich, wenn die Geräte, die mit einem geklonten Gerät verbunden sind, in einem Unterverzeichnis stehen. (Es ist wichtig zu beachten, daß Geräte-Dateien für geklonte Geräte nicht zwingend vorhanden sein müssen. Wenn dies der Fall ist, könnte ttyname möglicherweise fehlschlagen.) Wenn zum Beispiel /dev/starlan ein geklontes Gerät ist, könnte ein Unterverzeichnis /dev/slan die Dateien 0, 1, 2 usw. enthalten, welche den Geräten des Starlan-Treibers entsprechen. Ein optionales zweites Feld wird in der Datei /etc/ttsrch verwendet, um die Vergleichs-Kriterien anzugeben. Dieses Feld ist durch einen Zwischenraum (einer beliebigen Kombination von Leerzeichen oder Tabulatoren) vom vorhergehenden getrennt. Der Buchstabe M steht für die major/minor Gerätenummer, F steht für den Dateisystem-Bezeichner, und I steht für die I-Node-Nummer. Fehlt dieses Feld für einen Eintrag, ist der Standard MFI. Das bedeutet, daß der Vergleich auf alle drei Kriterien durchgeführt werden muß. Für geklonte Geräte sollte das Feld MF enthalten. Das zeigt an, daß es nicht notwendig ist, daß der Vergleich auf die I-Node-Nummer paßt.

Ohne die Datei `/etc/ttysrch` durchsucht `ttyname` das Verzeichnis `/dev`, indem zuerst in den Verzeichnissen `/dev/term`, `/dev/pts` und `/dev/xt` gesucht wird. Sind in einem System Terminal-Geräte in anderen Verzeichnissen als diesen installiert, verbessert es den Durchsatz, wenn die Datei `ttysrch` angelegt ist und eine Liste dieser Verzeichnisse enthält.

BEISPIEL

Eine Musterdatei `/etc/ttysrch`:

```
/dev/term MFI
/dev/pts MFI
/dev/xt MFI
/dev/slan MF
```

Diese Datei weist `ttyname` an, zuerst die aufgelisteten Verzeichnisse zu durchsuchen. Wird dann bei der Durchsuchung des Verzeichnisses `/dev/slan` eine Datei angetroffen, deren Geräteklasse/Gerät und Dateisystem-Bezeichner mit denen des Dateideskriptor-Arguments in `ttyname` übereinstimmen, ist ein passender Gerätenamen gefunden worden.

DATEIEN

`/etc/ttysrch`

SIEHE AUCH

`ttyname(3C)`, `clone(7)`

BEZEICHNUNG

unistd – Include-Datei für symbolische Konstanten

ÜBERSICHT

```
#include <unistd.h>
```

BESCHREIBUNG

Die Include-Datei <unistd.h> definiert die symbolischen Konstanten und Strukturen, die nicht in anderen Include-Dateien definiert oder erklärt sind. Der Inhalt dieser Include-Datei ist unten beschrieben.

Für die Funktion `access` sind folgende symbolische Konstanten definiert (siehe `access(2)`):

R_OK	Test auf Leseberechtigung
W_OK	Test auf Schreibberechtigung
X_OK	Test auf Ausführen (Sucherlaubnis)
F_OK	Test auf Existenz der Datei

Die Konstanten F_OK, R_OK, W_OK und X_OK und die Ausdrücke R_OK | W_OK, R_OK | X_OK und R_OK | W_OK | X_OK haben unterscheidbare Werte.

Deklaration der Konstante

NULL Null-Zeiger

Die folgenden symbolischen Konstanten sind für die Funktion `lockf` definiert (siehe `lockf(3C)`):

F_ULOCK	Entsperren eines vorher gesperrten Bereichs
F_LOCK	Sperren eines Bereichs für ausschließliche Anwendung
F_TLOCK	Test und Sperren eines Bereichs für ausschließliche Anwendung
F_TEST	Test eines Bereichs sperrt diesen für andere Prozesse

Die folgenden symbolischen Konstanten sind für die Funktionen `lseek` (siehe `lseek(2)`) und `fcntl` definiert (siehe `fcntl(2)`) (alle Werte sind verschieden):

SEEK_SET	Distanzzeiger der Datei auf <i>distanz</i> (offset) setzen
SEEK_CUR	Distanzzeiger der Datei auf aktuellen plus <i>distanz</i> setzen
SEEK_END	Distanzzeiger der Datei auf EOF plus <i>distanz</i> setzen

Die folgenden symbolischen Konstanten sind definiert (mit fest vorgegebenen Werten):

_POSIX_VERSION	Ganzzahliger Wert; zeigt Version des POSIX Standard an
_XOPEN_VERSION	Ganzzahliger Wert; zeigt Version des XPG an, mit der das System übereinstimmt

Die folgenden symbolischen Konstanten dienen dazu, die vorhandenen Optionen anzuzeigen:

<code>_POSIX_JOB_CONTROL</code>	Implementierung unterstützt Auftragssteuerung
<code>_POSIX_SAVED_IDS</code>	Die Exec-Funktionen (siehe <code>exec(2)</code>) sichern die effektive Benutzer- und Gruppennummer
<code>_POSIX_VDISABLE</code>	Mit diesem Zeichen können Sonderzeichen des Terminals, wie sie in <code><termios.h></code> definiert sind, deaktiviert werden (siehe <code>termio(7)</code>).

Die folgenden symbolischen Konstanten sind für `sysconf` definiert (siehe `sysconf(3C)`):

```

_SC_ARG_MAX
_SC_CHILD_MAX
_SC_CLK_TCK
_SC_JOB_CONTROL
_SC_NGROUPS_MAX
_SC_OPEN_MAX
_SC_PAGESIZE
_SC_PASS_MAX
_SC_SAVED_IDS
_SC_VERSION
_SC_XOPEN_VERSION

```

Die folgenden symbolischen Konstanten sind für `pathconf` definiert (siehe `pathconf(3C)`):

```

_PC_CHOWN_RESTRICTED
_PC_LINK_MAX
_PC_MAX_CANON
_PC_MAX_INPUT
_PC_NAME_MAX
_PC_NO_TRUNC
_PC_PATH_MAX
_PC_PIPE_BUF
_PC_VDISABLE

```

Die folgenden symbolischen Konstanten sind für die Datei STREAMS definiert:

STDIN_FILENO	Dateinummer von <code>stdin</code> ; sie ist 0.
STDOUT_FILENO	Dateinummer von <code>stdout</code> ; sie ist 1.
STDERR_FILENO	Dateinummer von <code>stderr</code> ; sie ist 2.

Die folgenden Pfadnamen sind definiert:

GF_PATH	Pfadname der Gruppendatei.
PF_PATH	Pfadname der Paßwort-Datei.

HINWEIS

Die folgenden Werte für Konstanten sind definiert:

_POSIX_VERSION	198808L
_XOPEN_VERSION	3

SIEHE AUCH

`access(2)`, `exec(2)`, `fcntl(2)`, `lseek(2)`, `termios(2)`, `fpathconf(3C)`, `sysconf(3C)`, `group(4)`, `passwd(4)`, `termio(7)`.

BEZEICHNUNG

utmp, wtmp – Format der Einträge in utmp und wtmp

ÜBERSICHT

```
#include <utmp.h>
```

BESCHREIBUNG

Diese Dateien beinhalten Benutzer- und Verwaltungsinformationen für Befehle wie `who`, `write` und `login`, und haben die folgende Struktur, die in `<utmp.h>` definiert ist:

```
#define UTMP_FILE "/var/adm/utmp"
#define WTMP_FILE "/var/adm/wtmp"
#define ut_name    ut_user

struct    utmp {
    char    ut_user[8];        /* Benutzer Login-Name */
    char    ut_id[4];          /* /etc/inittab id (angelegt durch einen
                                Prozeß, der einen Eintrag in utmp setzt) */
    char    ut_line[12];        /* Geräte-Name (Konsole, lxxx) */
    short    ut_pid;            /* Prozeßnummer */
    short    ut_type;           /* Art des Eintrags */
    struct    exit_status {
        short    e_termination; /* Ende-Status des Prozesses */
        short    e_exit;         /* Exit-Status des Prozesses */
    } ut_exit;                 /* Ende-Status eines Prozesses,
                                markiert als DEAD_PROCESS */
    time_t    ut_time;          /* Zeit-Eintrag wurde vorgenommen */
};

/* Definitionen für ut_type */

#define EMPTY        0
#define RUN_LVL      1
#define BOOT_TIME    2
#define OLD_TIME     3
#define NEW_TIME     4
#define INIT_PROCESS 5          /* Prozeß erzeugt durch "init" */
#define LOGIN_PROCESS 6         /* ein "getty" Prozeß im Wartestatus
                                für Anmeldung */
#define USER_PROCESS 7          /* ein Benutzerprozeß */
#define DEAD_PROCESS 8
#define ACCOUNTING   9
#define UTMAXTYPE    ACCOUNTING /* max. gültiger Wert von ut_type */

/* Unten sind besondere Zeichenketten oder Formate verwendet, die im */
/* Feld "ut_line" verwendet werden, wenn die Verwaltung sich */
/* nicht auf einen Prozeß bezieht. */
/* Die Zeichenkette für das ut_line Feld darf nicht mehr als */
/* 11 Zeichen + ein Null-Zeichen lang sein. */
```

utmp(4)

```
#define RUNLVL_MSG "run-level %c"
#define BOOT_MSG   "system boot"
#define OTIME_MSG  "old time"
#define NTIME_MSG  "new time"
```

DATEIEN

/var/adm/utmp
/var/adm/wtmp

SIEHE AUCH

getut(3C).
login(1), who(1), write(1) in den *Kommandos*.

utmp(4)

BEZEICHNUNG

utmpx, wtmpx – Formate der Einträge in utmpx und wtmpx

ÜBERSICHT

```
#include <utmpx.h>
```

BESCHREIBUNG

utmpx(4) ist eine erweiterte Version von utmp(4).

Diese Dateien beinhalten Benutzer- und Verwaltungsinformationen für Befehle wie who, write und login, und haben die folgende Struktur, wie in <utmpx.h> definiert:

```
#define UTMPX_FILE    "/var/adm/utmpx"
#define WTMPX_FILE    "/var/adm/wtmpx"
#define ut_name       ut_user
#define ut_xtime       ut_tv.tv_sec

struct utmpx {
    char    ut_user[32];    /* Benutzer-Login-Name */
    char    ut_id[4];      /* inittab ID */
    char    ut_line[32];    /* Geräte-Name (Konsole, lnxx) */
    pid_t    ut_pid;        /* Prozeßnummer */
    short    ut_type;       /* Art des Eintrags */
    struct exit_status ut_exit; /* Prozeßbeendigungs-/Exit-Status */
    struct timeval ut_tv;   /* Zeit-Eintrag wurde vorgenommen */
    long    ut_session;    /* Sitzungskennung, verwendet für Fenstertechnik */
    long    pad[5];        /* reserviert für zukünftige Anwendungen */
    short    ut_syslen;     /* signifikante Länge von ut_host */
    /* einschließlich abschließender NULL */
    char    ut_host[257];   /* Name des Fern-Systems */
};
/* Definitionen für ut_type */

#define EMPTY        0
#define RUN_LVL      1
#define BOOT_TIME    2
#define OLD_TIME     3
#define NEW_TIME     4
#define INIT_PROCESS 5 /* Prozeß erzeugt durch "init" */
#define LOGIN_PROCESS 6 /* ein "getty" Prozeß im Wartestatus für Anmeldung */
#define USER_PROCESS 7 /* ein Benutzerprozeß */
#define DEAD_PROCESS 8
#define ACCOUNTING   9

#define UTMXATYPE    ACCOUNTING /* größter zugelassener Wert von ut_type */

/* Unten sind besondere Zeichenketten oder Formate, die im Feld "ut_line"
/* verwendet werden, wenn die Verwaltung sich nicht auf einen Prozeß bezieht.
/* Die Zeichenkette für das ut_line Feld darf nicht mehr als 11 Zeichen +
/* ein Null-Zeichen lang sein. */

#define RUNLVL_MSG    "run-level %c"
#define BOOT_MSG      "system boot"
#define OTIME_MSG     "old time"
#define NTIME_MSG     "new time"
#define MOD_WIN       10
```

utmpx(4)

utmpx(4)

DATEIEN

/var/adm/utmpx
/var/adm/wtmpx

SIEHE AUCH

getutx(3C).
login(1), who(1), write(1) in den *Kommandos*.

BEZEICHNUNG

vfstab – Tabelle von Dateisystem-Standardwerten

ÜBERSICHT

```
#include <sys/fstyp.h>
#include <sys/param.h>
#include <sys/vfstab.h>
```

BESCHREIBUNG

Die Datei `/etc/vfstab` beschreibt Standardwerte für jedes Dateisystem. Diese Information hat folgende Struktur, wie sie in `<sys/vfstab.h>` definiert ist:

```
struct vfstab {
    char    *vfs_special;
    char    *vfs_fsckdev;
    char    *vfs_mountpt;
    char    *vfs_fstype;
    char    *vfs_fsckpass;
    char    *vfs_automnt;
    char    *vfs_mntopts;
};
```

Die Felder in der Tabelle sind durch Leerzeichen getrennt und enthalten den Namen des blockorientierten Geräts bzw. der Ressource des zeichenorientierten Geräts, das Standardverzeichnis zum Einhängen und den Dateisystem-Typ. Weiterhin enthalten sie einen numerischen Wert, der `fsck` zur Entscheidung dient, ob das Dateisystem automatisch überprüft werden soll und ob es durch `mountall` automatisch eingehängt werden soll. Außerdem stehen hier auch die Optionen für das Einhängen des Dateisystems. Ein `'-'` kennzeichnet einen leeren Eintrag im Feld.

Mit den Routinen der `getvfsent(3C)`-Familie kann auf `/etc/vfstab` gelesen und geschrieben werden.

SIEHE AUCH

`fsck(1M)`, `mount(1M)`, `setmnt(1M)`
`getvfsent(3C)` im *Referenzhandbuch für Programmierer*
 Kapitel 4 im *Leitfaden für Systemverwalter*

5

Sonstiges

Sonstiges

5-1

intro (5)

intro (5)

BEZEICHNUNG

intro – Einführung zum Kapitel Sonstiges

BESCHREIBUNG

In diesem Kapitel sind verschiedene Dinge beschrieben, wie z.B. Makropakete, Zeichensatztabellen u. a.

ascii(5)

ascii(5)

BEZEICHNUNG

ascii – Abbildung des ASCII-Zeichensatzes

ÜBERSICHT

/usr/pub/ascii

BESCHREIBUNG

Die Datei `ascii` enthält die Abbildung des ASCII-Zeichensatzes. Für jedes Zeichen ist der äquivalente Hexadezimal- und Oktalwert angegeben, der bei Bedarf für den Ausdruck verwendet werden kann. Die Datei enthält die beiden folgenden Tabellen:

ASCII OKTAL

000 nul	001 soh	002 stx	003 etx	004 eot	005 enq	006 ack	007 bel
010 bs	011 ht	012 nl	013 vt	014 np	015 cr	016 so	017 si
020 dle	021 dc1	022 dc2	023 dc3	024 dc4	025 nak	026 syn	027 etb
030 can	031 em	032 sub	033 esc	034 fs	035 gs	036 rs	037 us
040 sp	041 !	042 "	043 #	044 \$	045 %	046 &	047 '
050 (	051)	052 *	053 +	054 ,	055 -	056 .	057 /
060 0	061 1	062 2	063 3	064 4	065 5	066 6	067 7
070 8	071 9	072 :	073 ;	074 <	075 =	076 >	077 ?
100 @	101 A	102 B	103 C	104 D	105 E	106 F	107 G
110 H	111 I	112 J	113 K	114 L	115 M	116 N	117 O
120 P	121 Q	122 R	123 S	124 T	125 U	126 V	127 W
130 X	131 Y	132 Z	133 [	134 \	135]	136 ^	137 _
140 `	141 a	142 b	143 c	144 d	145 e	146 f	147 g
150 h	151 i	152 j	153 k	154 l	155 m	156 n	157 o
160 p	161 q	162 r	163 s	164 t	165 u	166 v	167 w
170 x	171 y	172 z	173 {	174	175 }	176 ~	177 del

ASCII HEXADEZIMAL

00 nul	01 soh	02 stx	03 etx	04 eot	05 enq	06 ack	07 bel
08 bs	09 ht	0a nl	0b vt	0c np	0d cr	0e so	0f si
10 dle	11 dc1	12 dc2	13 dc3	14 dc4	15 nak	16 syn	17 etb
18 can	19 em	1a sub	1b esc	1c fs	1d gs	1e rs	1f us
20 sp	21 !	22 "	23 #	24 \$	25 %	26 &	27 '
28 (	29)	2a *	2b +	2c ,	2d -	2e .	2f /
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7
38 8	39 9	3a :	3b ;	3c <	3d =	3e >	3f ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G
48 H	49 I	4a J	4b K	4c L	4d M	4e N	4f O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W
58 X	59 Y	5a Z	5b [	5c \	5d]	5e ^	5f _
60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g
68 h	69 i	6a j	6b k	6c l	6d m	6e n	6f o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w
78 x	79 y	7a z	7b {	7c	7d }	7e ~	7f del

ascii(5)

ascii(5)

DATEIEN

/usr/pub/ascii

BEZEICHNUNG

environ – Benutzerumgebung

BESCHREIBUNG

Beim Start eines Prozesses stellen die `exec`-Routinen in einem Feld (array) von Zeichenketten, die sogenannte Umgebung, zur Verfügung (siehe `exec(2)`). Üblicherweise sind diese Zeichenketten von der Form *variable=wert*, z.B.: `PATH=/sbin:/usr/sbin`. Mit diesen Umgebungsvariablen haben Programme die Möglichkeit, ihre jeweilige Umgebung abzufragen. Von den folgenden Umgebungsvariablen wird erwartet, daß sie in der Ablaufumgebung eines Anwendungsprogramms gesetzt sind und demzufolge im Programm verwendet werden können.

HOME Diese Variable enthält das Home-Verzeichnis des Benutzers; sie wird durch `login(1)` mit der Information aus der Paßwortdatei versorgt (siehe `passwd(4)`).

LANG Diese Variable enthält Information über regionale Gegebenheiten und ermöglicht es Benutzern, unterschiedliche nationale Gebräuche zu beachten. Der Inhalt dieser Variablen wird von der Funktion `setlocale` untersucht, wenn sie mit dem Argument "" für Region aufgerufen wird. Ist die entsprechende Umgebungsvariable für eine bestimmte Kategorie nicht gesetzt, wird `LANG` als Voreinstellung für die jeweilige Region benutzt.

Angenommen, `setlocale(3C)` wird in der folgenden Form aufgerufen:

```
setlocale(LC_CTYPE, "")
```

Dann prüft `setlocale()` zunächst, ob die Umgebungsvariable `LC_CTYPE` existiert und nicht leer ist. Ist das nicht der Fall, prüft `setlocale(3C)` als nächstes, ob die Umgebungsvariable `LANG` existiert und nicht leer ist. Fällt das Ergebnis in beiden Fällen negativ aus, wird die C-Voreinstellung für die Region verwendet, um damit die Kategorie `LC_CTYPE` zu setzen.

Die meisten Kommandos werden den Aufruf

```
setlocale(LC_ALL, "")
```

absetzen, bevor sie etwas anderes tun. Damit ist es möglich, Programme ohne Änderung des Quellcodes an verschiedene nationale Umgebungen anzupassen

Die folgenden, den einzelnen Kategorien von `setlocale(3C)` entsprechenden Umgebungsvariablen, werden unterstützt:

LC_COLLATE Mit dieser Kategorie wird die aktuelle Sortierfolge spezifiziert. Die dazu gehörige Information ist in einer von `colltbl(1M)` erzeugten Datenbank abgelegt. Der Inhalt dieser Umgebungsvariablen beeinflusst die Funktionen `strcoll(3C)` und `strxfrm(3C)`.

LC_CTYPE	Mit dieser Kategorie wird die Zeichenklasse, die Zeichenkonvertierung und die Breite von Zeichen, die mehr als ein Byte belegen, festgelegt. Die dazu gehörige Information ist in einer von <code>chrtbl(1M)</code> erzeugten Datenbank abgelegt. Die C-Voreinstellung für die Region entspricht hier dem 7-Bit ASCII-Zeichensatz. Diese Umgebungsvariable wird von <code>ctype(3C)</code> und <code>mbchar(3C)</code> benutzt; außerdem verwenden sie viele andere Kommandos wie z.B. <code>cat(1)</code> , <code>ed(1)</code> , <code>ls(1)</code> und <code>vi(1)</code> .
LC_MESSAGES	Mit dieser Kategorie wird festgelegt, welche Sprache für die Datenbank der Meldungstexte verwendet wird, wenn z.B. ein Anwenderprogramm über eine Datenbank mit französischen und eine mit deutschen Meldungen verfügt. Zur Erzeugung von Meldungsdatenbanken dient das Kommando <code>mkmsgs(1M)</code> . Diese Umgebungsvariable wird von <code>exstr(1)</code> , <code>gettext(1)</code> , <code>gettext(3C)</code> und <code>srchtxt(1)</code> verwendet.
LC_MONETARY	Mit dieser Kategorie werden die Symbole und Trennzeichen für die Darstellung von Geldbeträgen festgelegt, wie sie in verschiedenen Ländern üblich sind. Die dazu gehörige Information ist in einer von <code>montbl(1M)</code> erzeugten Datenbank abgelegt. Diese Umgebungsvariable wird von <code>localeconv(3C)</code> verwendet.
LC_NUMERIC	Mit dieser Kategorie werden die Symbole und Trennzeichen für Dezimalpunkt und Tausenderstellen festgelegt. Die dazu gehörige Information ist in einer von <code>chrtbl(1M)</code> erzeugten Datenbank abgelegt. Der C-Voreinstellung für die Region entspricht hier der Punkt "." als Dezimalpunkt und keine Verwendung von Trennzeichen bei Tausenderstellen. Diese Umgebungsvariable wird von <code>localeconv(3C)</code> , <code>printf(3C)</code> und <code>strtod(3C)</code> verwendet.
LC_TIME	Mit dieser Kategorie wird das Format für Datum und Zeit festgelegt. Die dazu gehörige Information ist in einer durch <code>strptime(4)</code> spezifizierten Datenbank abgelegt. Der C-Voreinstellung für die Region entspricht hier das USA-Format für Datum und Zeit. Diese Umgebungsvariable wird von zahlreichen Kommandos und Funktionen verwendet, wie z.B. <code>at(1)</code> , <code>calendar(1)</code> , <code>date(1)</code> , <code>strptime(3C)</code> und <code>getdate(3C)</code> .

MSGVERB	Damit wird gesteuert, welches Standardformat für Meldungskomponenten von <code>fmtmsg</code> verwendet wird, wenn auf die Standardfehlerausgabe Meldungen ausgegeben werden (siehe <code>fmtmsg(1)</code> und <code>fmtmsg(3C)</code>).
SEV_LEVEL	Damit werden Fehlergewichte definiert, mit entsprechenden Zeichenketten gekoppelt und in Fehlermeldungen im Standardformat ausgegeben (siehe <code>addseverity(3C)</code> , <code>fmtmsg(1)</code> , und <code>fmtmsg(3C)</code>).
NETPATH	Diese Variable enthält eine Liste von durch Doppelpunkte getrennten Netzwerk-Identifikatoren, die in Form von Zeichenketten angegeben sind. Sie werden von der Netzwerkauswahl-Komponente des Systems dafür benutzt, um anwendungsspezifische Suchpfade für Netzwerke zur Verfügung zu stellen. Ein Netzwerk-Identifikator muß aus mindestens einem Zeichen bestehen und darf keine Nullzeichen enthalten. Zu seiner Länge gibt es keine Einschränkung. Netzwerk-Identifikatoren werden normalerweise vom Systemverwalter erstellt. Sie sind stets das erste Feld eines Eintrags in der Datei <code>/etc/netconfig</code> . Folglich stellt <code>NETPATH</code> eine Verbindung in die Datei <code>/etc/netconfig</code> zu der Netzwerkinformation in diesem Eintrag her. Die Datei <code>/etc/netconfig</code> wird vom Systemverwalter gepflegt. Die Umgebungsvariable <code>NETPATH</code> wird von den Bibliotheksfunktionen verwendet, die in <code>getnetpath(3N)</code> beschrieben sind.
NLSPATH	Diese Umgebungsvariable enthält eine Reihe von schablonenartigen Pfadnamen, die <code>catopen(3C)</code> zur Auffindung von Meldungskatalogen dienen. Jede dieser Schablonen besteht aus einem wahlfreien Präfix, einem oder mehreren Feldern mit Platzhaltern, einem Dateinamen und einem wahlfreien Suffix.

Beispiel:

```
NLSPATH="/system/nlslib/%N.cat"
```

veranlaßt `catopen()`, alle Meldungskataloge in dem Verzeichnis `/system/nlslib` zu suchen und den Namen des Katalogs folgendermaßen zu konstruieren: das Kürzel `%N` steht für den an `catopen()` übergebenen Parameter *name*, dem noch das Suffix `.cat` anzufügen ist.

Ein Feld mit Platzhalter besteht aus dem Zeichen %, dem ein Einzelzeichen als Schlüsselwort folgt. Als Schlüsselwörter gelten zur Zeit:

%N	Der Inhalt des an <code>catopen()</code> übergebenen Parameters <i>name</i> .
%L	Inhalt von LANG
%l	Sprache-Anteil in LANG
%t	Staat-Anteil in LANG
%c	Zeichensatz-Anteil in LANG
%%	Einzelzeichen % selbst

Ist der angegebene Wert undefiniert, wird er durch die leere Zeichenreihe ersetzt. Die Trennzeichen „_“ und „.“ können nicht Teil der Ersetzungen sein, die durch %t und %c vorgenommen werden.

Die einzelnen Schablonen in NLSPATH sind durch einen Doppelpunkt (:) voneinander getrennt. Beginnt der Inhalt von NLSPATH mit einem oder zwei Doppelpunkt(en), so ist das gleichbedeutend mit der Spezifikation %N.

Die Belegung

```
NLSPATH=":%N.cat:/nlslib/%L/%N.cat"
```

bedeutet beispielsweise, daß `catopen()` der Reihe nach in *name.name.cat* und */nlslib/\$LANG/name.cat* nach dem gewünschten Meldungskatalog suchen soll.

PATH Diese Umgebungsvariable enthält eine Aufzählung von Verzeichnissen, die nach dem entsprechenden ausführbaren Programm durchsucht werden, wenn Kommandos nur mit relativem Pfadnamen aufgerufen werden. Kommandos wie z.B. `sh(1)`, `time(1)`, `nice(1)`, `nohup(1)` usw. machen Gebrauch von dieser Umgebungsvariablen. Die einzelnen Verzeichnisse sind durch einen Doppelpunkt (:) voneinander getrennt. Durch `login(1)` wird die Variable folgendermaßen belegt `PATH=/usr/bin` (siehe `sh(1)` für detailliertere Angaben).

TERM Diese Umgebungsvariable enthält den Namen des angeschlossenen Terminals. Kommandos wie z.B. `rm(1)` oder `vi(1)` können dann in der zugehörigen Beschreibung spezielle Fähigkeiten dieses Terminals nachschlagen und nutzen.

TZ Diese Umgebungsvariable enthält Information über die Zeitzone. TZ wird von den Funktionen `ctime(3C)`, `localtime()` (siehe `ctime(3C)`), `strftime(3C)` und `mktime(3C)` dazu benutzt, den Voreinstellungswert für die Zeitzone außer Kraft zu setzen. Ist das erste Zeichen von TZ ein Doppelpunkt (:), erfolgt die Interpretation implementierungsabhängig. Ansonsten hat TZ das folgende Format:

std diff [*dst* [*diff*] , [*start* [/zeit] , *ende* [/zeit]]]

std und *dst*

Die beiden Einträge *std* und *dst* bestehen aus je drei oder mehr Bytes und geben die Standardzeit (*std*) und die Sommerzeit an (daylight savings time: (*dst*)). Die Angabe von *std* ist erforderlich. Fehlt *dst* , gilt in diesem Gebiet keine Sommerzeit. Es können sowohl Groß- als auch Kleinbuchstaben verwendet werden sowie alle Zeichen außer einem führenden Doppelpunkt, Ziffern, einem Komma, einem Minus- oder Pluszeichen.

diff Damit wird die Differenz spezifiziert, die zur lokalen Zeit addiert werden muß, um "Coordinated Universal Time" zu erhalten. Diese Differenz ist im folgenden Format angegeben:

hh [: *mm* [: *ss*]]

Die Angabe der Minuten (*mm*) und Sekunden (*ss*) ist optional. Die Angabe der Stunde (*hh*) ist Pflicht; sie darf auch eine einzelne Ziffer sein. Die Angabe von *diff* hinter *std* ist ebenfalls Pflicht. Falls hinter *dst* kein *diff* angegeben ist, wird die Sommerzeit als eine Stunde vor der Normalzeit angenommen. Der Wert (von einer oder mehreren Ziffern) wird stets als Dezimalwert interpretiert. Die Stunde muß aus dem Bereich von 0 bis 24 und die Minuten (und ggf. die Sekunden) aus dem Bereich von 0 bis 59 stammen, andernfalls sind die Resultate rein zufällig. Ein führendes Minuszeichen bestimmt die Zeitzone als östlich des Hauptmeridians, im anderen Fall gilt sie als westlich davon (was optional durch ein führendes Pluszeichen angegeben werden kann).

start/zeit,ende/zeit

Damit wird angegeben, wann auf Sommerzeit umgeschaltet und wann wieder auf Normalzeit zurückgeschaltet wird. Das Umschalten wird durch *start/zeit* und das Zurückschalten durch *ende/zeit* festgelegt, wobei in dem Feld *zeit* in lokaler Zeitangabe der Zeitpunkt des Umschaltens steht.

Die folgenden Formate sind für *start* und *ende* möglich:

- $\text{J}n$ Der Julianische Tag n ($1 \leq n \leq 365$). Dabei zählen keine Schalttage. Es gibt keinen 29. Februar. Anders gesagt, der 28. Februar ist der 59. Tag und der 1. März der 60. Tag.
- n Der Julianische Tag einschließlich der Null ($0 \leq n \leq 365$). Es gelten Schalttage und damit gibt es auch den 29. Februar.

Mm.n.d

Der *d*-te Tag der Woche *n* des Monats *m* des Jahres. Dabei gilt für *d*, *n* und *m*: ($0 \leq d \leq 6$, $1 \leq n \leq 5$, $1 \leq m \leq 12$). Dabei bedeutet Woche 5 "den letzten Tag *d* des Monats *m*", der entweder in der vierten oder der fünften Woche vorkommt. Woche 1 ist die Woche, in der der Tag *d* vorkommt. Tag 0 ist der Sonntag.

Sind die optionalen Felder *start* und *ende* nicht angegeben, werden implementierungsbedingte Voreinstellungen benutzt.

Die Angabe von *zeit* erfolgt im gleichen Format wie *diff*, allerdings ist hier kein führendes Plus- oder Minuszeichen erlaubt. Ist *zeit* nicht angegeben, wird 02:00:00 als Voreinstellung verwendet.

Durch Argumente der Shell *sh*(1) in der Form *name=wert* sowie mittels des Kommandos *export* oder der Funktion *exec*(2) können weitere Umgebungsvariable definiert werden. Es ist nicht ratsam, solche Shellvariablen zu redefinieren, die häufig mit Hilfe der Datei *.profile* exportiert werden, wie: MAIL, PS1, PS2, IFS (siehe *profile*(4)).

SIEHE AUCH

chrtbl(1M), *colltbl*(1M), *mkmsgs*(1M), *monthbl*(1M), *netconfig*(4), *strftime*(4), *passwd*(4), *profile*(4) im *Referenzhandbuch für Systemverwalter*.

exec(2), *addseverity*(3C), *catopen*(3C), *ctime*(3C), *ctype*(3C), *fmtmsg*(3C), *getdate*(3C), *gettext*(3C), *localeconv*(3C), *mbchar*(3C), *mktime*(3C), *printf*(3C), *strcoll*(3C), *strftime*(3C), *strtod*(3C), *strxfrm*(3C), *strftime*(4), *timezone*(4).
cat(1), *date*(1), *ed*(1), *fmtmsg*(1), *ls*(1), *login*(1), *nice*(1), *nohup*(1), *sh*(1), *sort*(1), *time*(1), *vi*(1) in den *Kommandos*.

getnetpath(3N), in *CES, Leitfaden und Werkzeuge für die Programmierung mit C, Netzwerkschnittstellen*.

mm(1) im *DOCUMENTER'S WORKBENCH Software Technical Discussion and Reference Manual*.

BEZEICHNUNG

fcntl – Optionen für die Dateibehandlung

ÜBERSICHT

```
#include <fcntl.h>
```

BESCHREIBUNG

In der Datei <fcntl.h> sind die Anforderungskonstanten und die Argumente definiert, wie sie von den Funktionen fcntl (siehe fcntl(2)) und open (siehe open(2)) verwendet werden.

Werte für den *cmd* Parameter von fcntl (die folgenden Werte sind eindeutig):

F_DUPFD	Dateideskriptor duplizieren
F_GETFD	Dateideskriptorschalter lesen
F_SETFD	Dateideskriptorschalter setzen
F_GETFL	Dateistatus lesen
F_SETFL	Dateistatus setzen
F_GETLK	Information über Satzsperrern lesen
F_SETLK	Satzsperrern setzen
F_SETLKW	Satzsperrern setzen und bei Blockierung warten

Dateideskriptorschalter, wie sie von fcntl benutzt werden:

FD_CLOEXEC	Bei Überlagerung durch exec (siehe exec(2)) Dateideskriptor schließen
------------	--

Werte von *l_type* für fcntl zum Sperren von Sätzen
(die Werte sind eindeutig):

F_RDLCK	Sperre für Lesen oder gemeinsame Nutzung
F_UNLCK	Sperrfreigabe
F_WRLCK	Sperre für Schreiben oder Exklusivnutzung

Die drei folgenden Wertegruppen unterscheiden sich bitweise:

Werte von *oflag* für open:

O_CREAT	Bei Nichtexistenz Datei erzeugen
O_EXCL	Schalter für Exklusivnutzung
O_NOCTTY	Kein kontrollierendes Terminal zuweisen
O_TRUNC	Schalter für Datei entfernen

Datei-Status-Schalter für open und fcntl:

O_APPEND	Zugriffsart append
O_NDELAY	Blockierung aus
O_NONBLOCK	Blockierung aus (POSIX)
O_SYNC	Synchrones Schreiben

Maske für Zugriffsart:

O_ACCMODE

Maske für Zugriffsart

Zugriffsart für open und fcntl:

O_RDONLY

Nur zum Lesen öffnen

O_RDWR

Öffnen zum Lesen und Schreiben

O_WRONLY

Nur zum Schreiben öffnen

Die Art von Dateisperren ist durch die Struktur flock beschrieben, die wie folgt aufgebaut ist:

```

short  l_type;      /* Art der Sperre */
short  l_whence;    /* Schalter für Startadresse */
off_t  l_start;     /* Relative Distanz in Bytes */
off_t  l_len;       /* Größe; falls 0, bis Dateiende */
long   l_sysid;     /* wird bei Angabe von F_GETLK zurückgegeben */
pid_t  l_pid;       /* wird bei Angabe von F_GETLK zurückgegeben */

```

SIEHE AUCH

creat(2), exec(2), fcntl(2), open(2).

BEZEICHNUNG

iconv – Konvertierungstabellen für Zeichensätze

ÜBERSICHT

/usr/lib/iconv

BESCHREIBUNG

Folgende Zeichensatzkonvertierungen werden unterstützt:

unterstützte Zeichensatzkonvertierungen				
Code	Symbol	Ziel-Code	Symbol	Kommentar
ISO 646	646	ISO 8859-1	8859	US Ascii
ISO 646de	646de	ISO 8859-1	8859	Deutsch
ISO 646da	646da	ISO 8859-1	8859	Dänisch
ISO 646en	646en	ISO 8859-1	8859	Ascii Englisch
ISO 646es	646es	ISO 8859-1	8859	Spanisch
ISO 646fr	646fr	ISO 8859-1	8859	Französisch
ISO 646it	646it	ISO 8859-1	8859	Italienisch
ISO 646sv	646sv	ISO 8859-1	8859	Schwedisch
ISO 8859-1	8859	ISO 646	646	Ascii 7 Bit
ISO 8859-1	8859	ISO 646de	646de	Deutsch
ISO 8859-1	8859	ISO 646da	646da	Dänisch
ISO 8859-1	8859	ISO 646en	646en	Ascii Englisch
ISO 8859-1	8859	ISO 646es	646es	Spanisch
ISO 8859-1	8859	ISO 646fr	646fr	Französisch
ISO 8859-1	8859	ISO 646it	646it	Italienisch
ISO 8859-1	8859	ISO 646sv	646sv	Schwedisch

Die Konvertierungen werden gemäß den nachfolgenden Tabellen durchgeführt.
Die Tabellenwerte sind in Oktaldarstellung angegeben.

ISO 646 (US ASCII) nach ISO 8859-1

Für die Konvertierung von ISO 646 nach ISO 8859-1 können alle Zeichen aus ISO 646 unverändert auf ISO 8859-1 abgebildet werden.

iconv(5)

iconv(5)

ISO 646de (DEUTSCH) nach ISO 8859-1

Für die Konvertierung von ISO 646de nach ISO 8859-1 können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 646de	ISO 8859-1
100	247
133	304
134	326
135	334
173	344
174	366
175	374
176	337

ISO 646da (DÄNISCH) nach ISO 8859-1

Für die Konvertierung von ISO 646da nach ISO 8859-1 können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 646da	ISO 8859-1
133	306
134	330
135	305
173	346
174	370
175	345

ISO 646en (ASCII ENGLISCH) nach ISO 8859-1

Für die Konvertierung von ISO 646en nach ISO 8859-1 können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 646en	ISO 8859-1
043	243

iconv(5)

iconv(5)

ISO 646fr (FRANZÖSISCH) nach ISO 8859-1

Für die Konvertierung von ISO 646fr nach ISO 8859-1 können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 646fr	ISO 8859-1
043	243
100	340
133	260
134	347
135	247
173	351
174	371
175	350
176	250

ISO 646it (ITALIENISCH) nach ISO 8859-1

Für die Konvertierung von ISO 646it nach ISO 8859-1 können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 646it	ISO 8859-1
043	243
100	247
133	260
134	347
135	351
140	371
173	340
174	362
175	350
176	354

iconv(5)

iconv(5)

ISO 646es (SPANISCH) nach ISO 8859-1

Für die Konvertierung von ISO 646es nach ISO 8859-1 können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 646es	ISO 8859-1
100	247
133	241
134	321
135	277
173	260
174	361
175	347

ISO 646sv (SCHWEDISCH) nach ISO 8859-1

Für die Konvertierung von ISO 646sv nach ISO 8859-1 können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 646sv	ISO 8859-1
100	311
133	304
134	326
135	305
136	334
140	351
173	344
174	366
175	345
176	374

iconv(5)

iconv(5)

ISO 8859-1 nach ISO 646 (ASCII)

Für die Konvertierung von ISO 8859-1 nach ISO 646 können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertiert auf Unterstrich ' ' (137)							
200	201	202	203	204	205	206	207
210	211	212	213	214	215	216	217
220	221	222	223	224	225	226	227
230	231	232	233	234	235	236	237
240	241	242	243	244	245	246	247
250	251	252	253	254	255	256	257
260	261	262	263	264	265	266	267
270	271	272	273	274	275	276	277
300	301	302	303	304	305	306	307
310	311	312	313	314	315	316	317
320	321	322	323	324	325	326	327
330	331	332	333	334	335	336	337
340	341	342	343	344	345	346	347
350	351	352	353	354	355	356	357
360	361	362	363	364	365	366	367
370	371	372	373	374	375	376	377

ISO 8859-1 nach ISO 646de (DEUTSCH)

Für die Konvertierung von ISO 8859-1 nach ISO 646de können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 8859-1	ISO 646de
247	100
304	133
326	134
334	135
337	176
344	173
366	174
374	175

Konvertiert auf Unterstrich ' ' (137)							
100	133	134	135	173	174	175	176
200	201	202	203	204	205	206	207
210	211	212	213	214	215	216	217
220	221	222	223	224	225	226	227
230	231	232	233	234	235	236	237
240	241	242	243	244	245	246	
250	251	252	253	254	255	256	257
260	261	262	263	264	265	266	267
270	271	272	273	274	275	276	277
300	301	302	303		305	306	307
310	311	312	313	314	315	316	317
320	321	322	323	324	325		327
330	331	332	333		335	336	337
340	341	342	343		345	346	347
350	351	352	353	354	355	356	357
360	361	362	363	364	365		367
370	371	372	373		375	376	377

iconv(5)

iconv(5)

ISO 8859-1 nach ISO 646da (DÄNISCH)

Für die Konvertierung von ISO 8859-1 nach ISO 646da können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 8859-1	ISO 646da
305	135
306	133
330	134
345	175
346	173
370	174

Konvertiert auf Unterstrich ' ' (137)							
133 134 135 173 174 175							
200	201	202	203	204	205	206	207
210	211	212	213	214	215	216	217
220	221	222	223	224	225	226	227
230	231	232	233	234	235	236	237
240	241	242	243	244	245	246	247
250	251	252	253	254	255	256	257
260	261	262	263	264	265	266	267
270	271	272	273	274	275	276	277
300	301	302	303	304			307
310	311	312	313	314	315	316	317
320	321	322	323	324	325	326	327
	331	332	333	334	335	336	337
340	341	342	343	344			347
350	351	352	353	354	355	356	357
360	361	362	363	364	365	366	367
	371	372	373	374		376	377

iconv(5)

iconv(5)

ISO 8859-1 nach ISO 646en (ASCII ENGLISCH)

Für die Konvertierung von ISO 8859-1 nach ISO 646en können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 8859-1	ISO 646en
243	043

Konvertiert auf Unterstrich ' _ ' (137)							
043							
200	201	202	203	204	205	206	207
210	211	212	213	214	215	216	217
220	221	222	223	224	225	226	227
230	231	232	233	234	235	236	237
240	241	242		244	245	246	247
250	251	252	253	254	255	256	257
260	261	262	263	264	265	266	267
270	271	272	273	274	275	276	277
300	301	302	303	304	305	306	307
310	311	312	313	314	315	316	317
320	321	322	323	324	325	326	327
330	331	332	333	334	335	336	337
340	341	342	343	344	345	346	347
350	351	352	353	354	355	356	357
360	361	362	363	364	365	366	367
370	371	372	373	374	375	376	377

iconv(5)

iconv(5)

ISO 8859-1 nach ISO 646fr (FRANZÖSISCH)

Für die Konvertierung von ISO 8859-1 nach ISO 646fr können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 8859-1	ISO 646fr
243	043
247	135
250	176
260	133
340	100
347	134
350	175
351	173
371	174

Konvertiert auf Unterstrich ' _ ' (137)							
043							
100	133	134	135	173	174	175	176
200	201	202	203	204	205	206	207
210	211	212	213	214	215	216	217
220	221	222	223	224	225	226	227
230	231	232	233	234	235	236	237
240	241	242		244	245	246	
	251	252	253	254	255	256	257
	261	262	263	264	265	266	267
270	271	272	273	274	275	276	277
300	301	302	303	304	305	306	307
310	311	312	313	314	315	316	317
320	321	322	323	324	325	326	327
330	331	332	333	334	335	336	337
	341	342	343	344	345	346	
		352	353	354	355	356	357
360	361	362	363	364	365	366	367
370		372	373	374	375	376	377

ISO 8859-1 nach ISO 646it (ITALIENISCH)

Für die Konvertierung von ISO 8859-1 nach ISO 646it können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 8859-1	ISO 646it
243	043
247	100
260	133
340	173
347	134
350	175
351	135
354	176
362	174
371	140

Konvertiert auf Unterstrich ' ' (137)							
043							
100	133	134	135	173	174	175	176
200	201	202	203	204	205	206	207
210	211	212	213	214	215	216	217
220	221	222	223	224	225	226	227
230	231	232	233	234	235	236	237
240	241	242		244	245	246	
250	251	252	253	254	255	256	257
	261	262	263	264	265	266	267
270	271	272	273	274	275	276	277
300	301	302	303	304	305	306	307
310	311	312	313	314	315	316	317
320	321	322	323	324	325	326	327
330	331	332	333	334	335	336	337
	341	342	343	344	345	346	
		352	353	354	355	356	357
360	361		363	364	365	366	367
370		372	373	374	375	376	377

iconv(5)

iconv(5)

ISO 8859-1 nach ISO 646es (SPANISCH)

Für die Konvertierung von ISO 8859-1 nach ISO 646es können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 8859-1	ISO 646es
241	133
247	100
260	173
277	135
321	134
347	175
361	174

Konvertiert auf Unterstrich ' ' (137)							
100 133 134 135 173 174 175							
200	201	202	203	204	205	206	207
210	211	212	213	214	215	216	217
220	221	222	223	224	225	226	227
230	231	232	233	234	235	236	237
240		242	243	244	245	246	
250	251	252	253	254	255	256	257
	261	262	263	264	265	266	267
270	271	272	273	274	275	276	
300	301	302	303	304	305	306	307
310	311	312	313	314	315	316	317
320		322	323	324	325	326	327
330	331	332	333	334	335	336	337
340	341	342	343	344	345	346	
350	351	352	353	354	355	356	357
360		362	363	364	365	366	367
370	371	372	373	374	375	376	377

ISO 8859-1 nach ISO 646sv (SCHWEDISCH)

Für die Konvertierung von ISO 8859-1 nach ISO 646sv können alle Zeichen, bis auf die in der folgenden Tabelle, unverändert abgebildet werden.

Konvertierung	
ISO 8859-1	ISO 646sv
304	133
305	135
311	100
326	134
334	136
344	173
345	175
351	140
366	174
374	176

Konvertiert auf Unterstrich ' ' (137)							
100 133 134 135 136 140							
173 174 175 176							
200	201	202	203	204	205	206	207
210	211	212	213	214	215	216	217
220	221	222	223	224	225	226	227
230	231	232	233	234	235	236	237
240	241	242	243	244	245	246	247
250	251	252	253	254	255	256	257
260	261	262	263	264	265	266	267
270	271	272	273	274	275	276	277
300	301	302	303			306	307
310		312	313	314	315	316	317
320	321	322	323	324	325		327
330	331	332	333		335	336	337
340	341	342	343			346	347
350		352	353	354	355	356	357
360	361	362	363	364	365		367
370	371	372	373		375	376	377

DATEIEN

/usr/lib/iconv/iconv_data
/usr/lib/iconv/*.t

Liste der unterstützten Konvertierungen
Konvertierungstabellen

SIEHE AUCH

iconv(1) in den *Kommandos*.

BEZEICHNUNG

jagent – Steuerung von fensterfähigen Terminals durch den Host

ÜBERSICHT

```
#include <sys/jioctl.h>

int ioctl (int cntlfd, JAGENT, &arg);
```

BESCHREIBUNG

Ein Host-Programm, welches an ein fensterfähiges Terminal Information schicken möchte, bedient sich des Systemaufrufs `ioctl`, in dem die JAGENT Anforderung gesetzt ist, und der an ein `xt(7)` Gerät gerichtet wird.

Die drei Argumente von `ioctl` sind:

`cntlfd` der Dateideskriptor für den `xt`-Steuerkanal

JAGENT die `ioctl` Anforderung an `xt`, am fensterfähigen Terminal eine Behandlungsroutine aufzurufen.

`&arg` die Adresse einer Struktur vom Typ `bagent`, die folgendermaßen in `<sys/jioctl.h>` definiert ist:

```
struct bagent {
    long    size; /* Größenangabe für Senden & Empfangen */
    char    *src; /* Zeiger auf Zeichenkette für Senden */
    char    *dest; /* Zeiger auf Zeichenkette für Empfang */
};
```

Der Zeiger `src` muß mit der Adresse der Zeichenkette versorgt sein, die an das Terminal geschickt werden soll. JAGENT Zeichenketten, die von fensterfähigen Terminals erkannt werden, sind unter `layers(5)` beschrieben. Genauso muß der Zeiger `dest` mit der Adresse des Zielbereichs versorgt sein, der die von dem Terminal zurückgesandte Zeichenkette aufnehmen soll. Beim Aufruf von `ioctl` ist das Argument `size` mit der Länge der geschickten Zeichenkette versorgt. Beim Abschluß von `ioctl` enthält dieses Feld die Länge der zurückgeschickten Zeichenkette.

SIEHE AUCH

`ioctl(2)`, `libwindows(3X)`, `layers(5)`.
`xt(7)` in *CES, Leitfaden und Werkzeuge für die Programmierung mit C*.

EXIT-CODES

Im Fehlerfall wird der Exit-Code `-1` zurückgegeben. Ansonsten wird die Länge der gesandten Zeichenkette als nicht negativer Wert zurückgegeben.

BEZEICHNUNG

langinfo – Konstantendefinitionen für Sprachen

ÜBERSICHT

```
#include <langinfo.h>
```

BESCHREIBUNG

Diese Include-Datei enthält Konstanten, die Objekte in Sprachen definieren. Der Typ der Objekte ist in `nl_types` aufgeführt.

DAY_1	Nationale Äquivalenz für 'sunday'
DAY_2	Nationale Äquivalenz für 'monday'
DAY_3	Nationale Äquivalenz für 'tuesday'
DAY_4	Nationale Äquivalenz für 'wednesday'
DAY_5	Nationale Äquivalenz für 'thursday'
DAY_6	Nationale Äquivalenz für 'friday'
DAY_7	Nationale Äquivalenz für 'saturday'
ABDAY_1	Nationale Äquivalenz für 'sun'
ABDAY_2	Nationale Äquivalenz für 'mon'
ABDAY_3	Nationale Äquivalenz für 'tue'
ABDAY_4	Nationale Äquivalenz für 'wed'
ABDAY_5	Nationale Äquivalenz für 'thur'
ABDAY_6	Nationale Äquivalenz für 'fri'
ABDAY_7	Nationale Äquivalenz für 'sat'
MON_1	Nationale Äquivalenz für 'january'
MON_2	Nationale Äquivalenz für 'february'
MON_3	Nationale Äquivalenz für 'march'
MON_4	Nationale Äquivalenz für 'april'
MON_5	Nationale Äquivalenz für 'may'
MON_6	Nationale Äquivalenz für 'june'
MON_7	Nationale Äquivalenz für 'july'
MON_8	Nationale Äquivalenz für 'august'
MON_9	Nationale Äquivalenz für 'september'
MON_10	Nationale Äquivalenz für 'october'
MON_11	Nationale Äquivalenz für 'november'
MON_12	Nationale Äquivalenz für 'december'
ABMON_1	Nationale Äquivalenz für 'jan'

ABMON_2	Nationale Äquivalenz für 'feb'
ABMON_3	Nationale Äquivalenz für 'mar'
ABMON_4	Nationale Äquivalenz für 'apr'
ABMON_5	Nationale Äquivalenz für 'may'
ABMON_6	Nationale Äquivalenz für 'jun'
ABMON_7	Nationale Äquivalenz für 'jul'
ABMON_8	Nationale Äquivalenz für 'aug'
ABMON_9	Nationale Äquivalenz für 'sep'
ABMON_10	Nationale Äquivalenz für 'oct'
ABMON_11	Nationale Äquivalenz für 'nov'
ABMON_12	Nationale Äquivalenz für 'dec'
RADIXCHAR	Nationale Äquivalenz für '.'
THOUSEP	Nationale Äquivalenz für ','
YESSTR	Nationale Äquivalenz für 'yes'
NOSTR	Nationale Äquivalenz für 'no'
CRNCYSTR	Symbol für die Währung
D_T_FMT	Nationales Standardformat für Datum und Zeit
D_FMT	Nationales Standardformat für das Datum
T_FMT	Nationales Standardformat für die Zeit
AM_STR	Nationale Äquivalenz für 'AM'
PM_STR	Nationale Äquivalenz für 'PM'

Die Information hierüber wird aus `nl_langinfo` gewonnen.

Die Objekte `CRNCYSTR`, `RADIXCHAR` und `THOUSEP` werden aus den Feldern `currency_symbol`, `decimal_point` und `thousands_sep` der Struktur gewonnen, die `localeconv` zurückgibt.

Die Objekte `T_FMT`, `D_FMT`, `D_T_FMT`, `YESSTR` und `NOSTR` werden aus einem speziellen Meldungskatalog `Xopen_info` gewonnen, welcher für jedes unterstützte Land erzeugt und in dem entsprechenden Verzeichnis abgelegt werden sollte (siehe `gettext(3C)` und `mkmsgs(1M)`). Dieser Katalog sollte die folgenden Meldungen in der angegebenen Reihenfolge enthalten: `T_FMT`, `D_FMT`, `D_T_FMT`, `YESSTR` und `NOSTR`.

Alle anderen Objekte sind so beschaffen, wie sie von `strftime` zurückgegeben werden.

SIEHE AUCH

`gettext(3C)`, `localeconv(3C)`, `nl_langinfo(3C)`, `strftime(3C)`, `cftime(4)`, `nl_types(5)`, `chrtbl(1)`, `mkmsgs(1M)` im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

layers – Protokoll zwischen Host und fensterfähigem Terminal unter layers(1)

BESCHREIBUNG

layer steht für Schicht oder Fenster; damit werden asynchrone Fenster bezeichnet, die in fensterfähigen Terminals vom dortigen Betriebssystem unterstützt werden. Die Kommunikation zwischen Prozessen des UNIX-Systems und Terminalprozessen, die unter der Regie von layers(1) ablaufen, verläuft über Multiplexkanäle. Diese werden von dem jeweiligen Betriebssystem verwaltet, wobei ein Protokoll verwendet wird, wie es in xtpROTO(5) beschrieben ist.

Die zwischen den Prozessen des UNIX-Systems und einer Schicht ausgetauschten Pakete sind asymmetrischen Inhalts. Die vom UNIX-System an einen bestimmten Terminalprozeß geschickten Daten sind undifferenziert; es ist Sache des Terminalprozesses, den Paketinhalt zu interpretieren.

Steuerinformationen für Terminalprozesse werden über Kanal 0 übermittelt. Am fensterfähigen Terminal führt dann der Prozeß 0 die vorgesehenen Funktionen durch, indem er sich des Prozesses bedient, der den angegebenen Kanal bedient. Pakete dieser Art haben folgendes Format:

kommando, kanal

mit Ausnahme der beiden Angaben JTIMOM und JAGENT; in diesen Fällen sind die Pakete von folgender Form:

kommando, daten ...

Die Kommandos sind in den unteren acht Bits der folgenden Codes für ioctl(2) verschlüsselt:

JBOOT	Vorbereitung zum Laden eines neuen Terminalprogramms in das genannte Fenster.
JTERM	Beenden des ins Terminal hinuntergeladenen Programms und Wiederherstellung des Standard-Fensterprogramms.
JTIMOM	Parameter für die Zeitgrenzen des Protokolls setzen. Die Daten bestehen aus zwei Gruppen zu je zwei Bytes, dem Zeitlimit beim Empfang und dem beim Senden. Jedes Zeitlimit ist in Millisekunden angegeben und von den zwei Byte jeder Gruppe wird das niederwertige zuerst übertragen.
JZOMBOOT	Ähnlich wie JBOOT, allerdings wird das Programm nur geladen und noch nicht ausgeführt.
JAGENT	Schickt der Behandlungsroutine am Terminal eine Zeichenkette und wartet auf eine Antwortzeichenkette.

Die Daten sind gemäß der Struktur `bagent` angeordnet [siehe `jagent(5)`]. Das erste Byte enthält eine Längenangabe, es folgen der Kommandocode für die Behandlungsroutine (zwei Byte) und weitere Parameter. Bei ganzzahligen 2-Byte-Werten, als Teil eines Kommandos der Behandlungsroutine, wird das höherwertige Byte als erstes übertragen. Allgemein betrachtet erfolgt die Antwort vom Terminal identisch zum Kommandopakete. Der einzige Unterschied besteht darin, daß die beiden Bytes des Kommandocodes für den Return-Code benutzt

werden. Er ist im Fehlerfall -1, sonst 0. Es ist zu beachten, daß alle Bibliotheksroutinen aus `libwindows(3X)` Parameter in Strukturen vom Typ `agentrect` verpackt versenden. Es folgen die Kommandocodes und Parameter für die Behandlungsroutine:

<code>A_NEWLAYER</code>	gefolgt von einer 2-Byte Kanalnummer und vier Koordinaten zu je 2 Byte, die ein Rechteck definieren.
<code>A_CURRENT</code>	gefolgt von einer 2-Byte Kanalnummer.
<code>A_DELETE</code>	gefolgt von einer 2-Byte Kanalnummer.
<code>A_TOP</code>	gefolgt von einer 2-Byte Kanalnummer.
<code>A_BOTTOM</code>	gefolgt von einer 2-Byte Kanalnummer.
<code>A_MOVE</code>	gefolgt von einer 2-Byte Kanalnummer und zwei Koordinaten zu je 2 Byte, die einen Zielpunkt definieren.
<code>A_RESHAPE</code>	gefolgt von einer 2-Byte Kanalnummer und vier Koordinaten zu je 2 Byte, die das neue Rechteck definieren.
<code>A_NEW</code>	gefolgt von einer 2-Byte Kanalnummer und vier Koordinaten zu je 2 Byte, die ein Rechteck definieren.
<code>A_EXIT</code>	keine Parameter erforderlich.
<code>A_ROMVERSION</code>	keine Parameter erforderlich. Das Antwortpaket enthält ein Byte mit der Länge, zwei Bytes mit dem Rückgabewert, zwei nicht benutzte Bytes und den Parameteranteil aus der Zeichenkette, die das Terminal identifiziert (z.B. 8;7;3) .

`JXTPROTO` Legt den Typ des `xt`-Protokolls fest [siehe `xtproto(5)`]. Der Datenteil besteht aus einem Byte, welches die Maximalgröße des Datenanteils normaler `xt`-Pakete bestimmt, die vom Host an das Terminal gesendet werden. Diese Größenangabe kann kleiner sein als die von `A_XTPROTO` zurückgegebene Zahl, wenn geringere Übertragungsraten gelten oder beim Aufruf von `layers` die Option `-m` angegeben wurde. Eine Größenangabe von 1 legt das `xt`-Netzwerkprotokoll fest.

Pakete von fensterfähigen Terminals zum UNIX-System sind alle von folgender Form:

command, data ...

Es folgen die in einem Byte verschlüsselten Kommandos:

<code>C_SENDCHAR</code>	Nächstes Byte an den Prozeß im UNIX-System schicken.
<code>C_NEW</code>	Erzeugt im UNIX-System eine neue Prozeßgruppe für dieses Fenster. Notiert die Größenparameter dieses Fensters. Der Datenteil dieses Kommandos genügt der Strukturdefinition <code>jwinsize</code> . Die Fenstergröße wird durch zwei ganzzahlige Werte zu je 2 Bytes spezifiziert, wobei jeweils das niederwertige Byte als erstes übertragen wird.

layers(5)

C_UNBLK	Aufhebung der Übertragungssperre für dieses Fenster. Zu diesem Kommando gehören keine Daten.
C_DELETE	Löschen der mit diesem Fenster verbundenen Prozeßgruppe im UNIX-System. Zu diesem Kommando gehören keine Daten.
C_EXIT	Ende. Beendigung sämtlicher mit diesem Terminal verbundener Prozeßgruppen im UNIX-System und Beendigung der Sitzung. Zu diesem Kommando gehören keine Daten.
C_DEFUNCT	Das Fensterprogramm existiert nicht mehr. Den mit diesem Terminal verbundenen Prozeßgruppen am UNIX-System wird ein Signal zugestellt, sich zu beenden. Zu diesem Kommando gehören keine Daten.
C_SENDCNCHARS	Die nachfolgenden Daten sind Zeichen, die dem Prozeß am UNIX-System zugestellt werden sollen.
C_RESIZE	Die Fenstergröße ist verändert worden. Die Größenparameter für dieses Fenster müssen angepaßt werden. Die Daten sind von gleicher Form wie bei dem Kommando C_NEW. Außerdem wird dem Prozeß im Fenster das Signal SIGWINCH zugestellt, um ihn davon zu unterrichten, daß sich die Fenstergröße geändert hat und er die neuen Parameter bekommen kann.
C_NOFLOW	Deaktivieren der xt-Flußsteuerung im Netzwerk (siehe xtproto(5)).
C_YESFLOW	Aktivieren der xt-Flußsteuerung im Netzwerk (siehe xtproto(5)).

DATEIEN

/usr/include/windows.h
/usr/include/sys/jioctl.h

SIEHE AUCH

layers(1), libwindows(3X), jagent(5), xtproto(5).
xt(7) in *CES, Leitfaden und Werkzeuge für die Programmierung mit C*.

layers(5)

BEZEICHNUNG

nl_types – Datentypen für Sprachen

ÜBERSICHT

```
#include <nl_types.h>
```

BESCHREIBUNG

Diese Include-Datei enthält die folgenden Beschreibungen:

nl_catd	wird von den Katalogfunktionen catopen, catgets und catclose benutzt, um Kataloge zu identifizieren.
nl_item	wird von nl_langinfo benutzt, um einzelne Teile in der landessprachlichen Information aufzufinden. Die Werte für Objekte vom Typ nl_item sind in langinfo.h definiert.
NL_SETD	wird von gencat benutzt, wenn keine \$set Anweisung vorgegeben ist. Nachfolgende Aufrufe von catgets können diese Konstante als entsprechenden set-Parameter verwenden.
NL_MGSMAX	maximale Anzahl der Meldungen pro Einheit.
NL_SETMAX	maximale Anzahl der Einheiten pro Katalog.
NL_TEXTMAX	maximale Länge einer Meldung.
DEF_NLSPATH	Standardsuchpfad für Meldungskataloge.

SIEHE AUCH

catgets(3C), catopen(3C), nl_langinfo(3C), langinfo(5).
gencat(1M) im *Referenzhandbuch für Systemverwalter*.

BEZEICHNUNG

regexp: compile, step, advance – Operationen zum Kompilieren und Auswerten von regulären Ausdrücken

ÜBERSICHT

```
#define INIT declarations
#define GETC(void) getc code
#define PEEKC(void) peekc code
#define UNGETC(void) ungetc code
#define RETURN(ptr) return code
#define ERROR(val) error code

#include <regex.h>

char *compile(char *instring, char *expbuf, char *endbuf, int eof);
int step(char *string, char *expbuf);
int advance(char *string, char *expbuf);
extern char *loc1, *loc2, *locs;
```

BESCHREIBUNG

Diese Funktionen sind allgemein verwendbare Routinen für Programme, die reguläre Ausdrücke auswerten. Sie sind in der Include-Datei `<regex.h>` definiert.

Die beiden Funktionen `step` und `advance` werten eine Zeichenkette bzgl. regulärer Ausdrücke aus.

Die Funktion `compile` erwartet als Eingabe einen regulären Ausdruck, wie er weiter unten definiert ist, und liefert einen kompilierten Ausdruck, der von `step` oder `advance` verwendet werden kann.

Ein regulärer Ausdruck paßt zu einer bestimmten Menge von Zeichenreihen. Eine solche Zeichenreihe wird auch als Treffer zu diesem regulären Ausdruck bezeichnet. Manche Zeichen innerhalb von regulären Ausdrücken haben eine Sonderbedeutung; die anderen stehen für sich selbst.

Die regulären Ausdrücke für die `regex`-Funktionen haben folgenden Aufbau:

Ausdruck	Bedeutung
<i>c</i>	Das Zeichen <i>c</i> selbst, welches keine Sonderbedeutung hat.
<code>\c</code>	Ein beliebiges Zeichen <i>c</i> , außer den Ziffern im Bereich von 1-9.
<code>^</code>	Der Zeilenanfang der gerade zu vergleichenden Zeile.
<code>\$</code>	Das Zeilenende der gerade zu vergleichenden Zeile.
<code>.</code>	Ein beliebiges Zeichen in der Eingabe.
<code>[s]</code>	Jedes Einzelzeichen aus der Menge <i>s</i> ; dabei ist <i>s</i> eine Aufzählung von Zeichen oder ein Bereich von Zeichen, wie z.B. <code>[c-c]</code> .
<code>[^s]</code>	Jedes Einzelzeichen, welches in der Aufzählung <i>s</i> nicht enthalten ist. Dabei ist <i>s</i> wie oben definiert.

<code>r*</code>	Null oder mehr aufeinanderfolgende Vorkommen des regulären Ausdrucks <code>r</code> . Als Treffer gilt die längste äußerst linke Übereinstimmung.
<code>rx</code>	das Vorkommen des regulären Ausdrucks <code>r</code> , dem der reguläre Ausdruck <code>x</code> folgt (Verkettung).
<code>r\{m,n\}</code>	Eine beliebige Zahl von <code>m</code> bis <code>n</code> aufeinanderfolgenden Vorkommen des regulären Ausdrucks <code>r</code> . Zu dem regulären Ausdruck <code>r\{m\}</code> passen genau <code>m</code> Vorkommen von <code>r</code> und zu <code>r\{m,\}</code> paßt eine Zeichenreihe, die mindestens <code>m</code> Vorkommen von <code>r</code> aufweist.
<code>\(r\)</code>	Der reguläre Ausdruck <code>r</code> . Taucht <code>\n</code> (wobei <code>n</code> größer Null ist) in einem zusammengesetzten regulären Ausdruck auf, steht das für das <code>n-te</code> Vorkommen des in diesem zusammengesetzten Ausdruck vorher vorkommenden und mit <code>\(</code> und <code>\)</code> geklammerten regulären Ausdrucks. Beispielsweise stellt der Ausdruck <code>\(r\)x\ (y\)\ z\2</code> die Konkatenation der regulären Ausdrücke <code>rxzyz</code> dar.

Die Zeichen `.`, `*`, `[` und `\` haben eine Sonderbedeutung, sofern sie nicht in eckigen Klammern (`[ ]`) stehen oder durch einen vorangestellten Gegenschrägstrich entwertet sind. Andere Sonderzeichen, wie z.B. `$`, haben eine eingeschränkte Sonderbedeutung.

Das Zeichen `^` am Anfang eines Ausdrucks bewirkt, daß eine passende Zeichenkette nur dann als Treffer gilt, wenn sie direkt am Zeilenanfang steht, wohingegen sie bei einem `$` am Ende des Ausdrucks unmittelbar vor dem Zeilenende stehen muß.

Zwei Zeichen haben nur innerhalb von eckigen Klammern eine Sonderbedeutung. Der Bindestrich zwischen zwei Zeichen bezeichnet einen Zeichenbereich `[c-c]`, vorausgesetzt er ist nicht das erste oder das letzte Zeichen innerhalb der Klammern. In `[-c]` oder `[c-]` hat er also keine Sonderbedeutung. Das Zeichen `^` hat in eckigen Klammern eine Komplementärwirkung, allerdings nur, wenn es dort das erste Zeichen ist, und bedeutet beispielsweise in `[^c]` alle Zeichen außer `c`, wohingegen es in `[c^]` das gewöhnliche Zeichen `^` selbst bedeutet.

Die Sonderbedeutung des Entwertungszeichens `\` kann nur durch ein weiteres vorangestelltes `\` aufgehoben werden, also `\\`.

In Programmen müssen vor der Include-Anweisung `#include <regexp.h>` die folgenden fünf Makros deklariert sein, die von der Funktion `compile` verwendet werden. Die Makros `GETC`, `PEEKC` und `UNGETC` operieren auf regulären Ausdrücken, die der Funktion `compile` als Eingabe übergeben werden.

<code>GETC</code>	Dieses Makro gibt das nächste Zeichen (ein Byte) aus dem regulären Ausdruck zurück. Aufeinanderfolgende Aufrufe von <code>GETC</code> sollten sequentiell die Zeichen aus dem regulären Ausdruck abliefern.
<code>PEEKC</code>	Dieses Makro gibt das nächste Zeichen (ein Byte) aus dem regulären Ausdruck zurück. Unmittelbar darauffolgende Aufrufe von <code>PEEKC</code> sollten immer dasselbe Zeichen zurückliefern, das mit dem übereinstimmen sollte, welches <code>GETC</code> zurückliefert.

UNGETC	Dieses Makro sorgt dafür, daß sein Argument den nächsten Aufrufen von GETC und PEEKC wieder zur Verfügung steht. Es wird nie mehr als ein Zeichen benötigt, das auf den Stack rückgepusht wird, und dieses wird garantiert beim nächsten Aufruf von GETC gelesen. Der Rückgabewert von UNGETC(c) wird stets ignoriert.
RETURN(ptr)	Dieses Makro wird bei ordnungsgemäßigem Ablauf von compile für den Rücksprung verwendet. Der Inhalt des Arguments ptr ist ein Zeiger, der auf das Zeichen hinter dem letzten des berechneten regulären Ausdrucks zeigt. Das ist für Programme mit Speicher-verwaltung nützlich.
ERROR(val)	Dieses Makro wird bei fehlerhaftem Abarbeiten von compile für den Rücksprung verwendet. Das Argument val enthält eine Fehler- nummer, deren Bedeutung unter Exit-Codes aufgeführt ist. Aus diesem Makro sollte kein Rücksprung in compile möglich sein.

Die Funktion compile hat folgende Syntax:

```
compile(instring, expbuf, endbuf, eof)
```

Der Parameter *instring* wird von der Funktion *compile* selbst nicht benutzt, ist aber für Programme sinnvoll, die verschiedene Zeiger auf Eingabezeichen übergeben. Manchmal wird er in der Deklaration *INIT* benutzt (siehe unten). Programme, die Funktionen auf Eingabezeichen ansetzen oder diese Zeichen in externen Feldern gespeichert haben, können für diesen Parameter den Nullzeiger (char *)0 übergeben.

Der zweite Parameter *expbuf* ist ein Zeiger vom Typ char. Er zeigt auf den Speicherplatz, wo der berechnete reguläre Ausdruck abgelegt werden soll.

Der Parameter *endbuf* zeigt unmittelbar hinter den Speicherplatz, wo der berechnete reguläre Ausdruck abgelegt werden soll. Belegt der berechnete reguläre Ausdruck mehr als (endbuf-expbuf) Bytes, erfolgt ein Aufruf von ERROR(50).

Der Parameter *eof* enthält das Zeichen, welches das Ende des regulären Ausdrucks kennzeichnet. Es ist üblicherweise ein /.

Jedes Programm, das die Include-Datei <regexp.h> benutzt, muß eine #define Anweisung für *INIT* enthalten. Sie dient individuellen Deklarationen und Initialisierungen. Meist wird sie dafür verwendet, eine *register* Variable zu initialisieren, so daß sie auf den Anfang des regulären Ausdrucks zeigt. Diese Variable kann dann in den Deklarationen von GETC, PEEKC und UNGETC verwendet werden. Andererseits können damit externe Variable deklariert werden, die von GETC, PEEKC und UNGETC verwendet werden. (Siehe unten: BEISPIEL)

Der erste Parameter von *step* und *advance* zeigt auf eine Zeichenkette, die mit dem Nullzeichen abgeschlossen sein sollte und die auf einen Treffer untersucht wird.

Der zweite Parameter *expbuf* verweist auf den berechneten regulären Ausdruck, der von der Funktion *compile* geliefert wurde.

Die Funktion `step` gibt einen von Null verschiedenen Wert zurück, wenn in der Zeichenkette *string* eine Teilzeichenkette gefunden wird, die zu dem regulären Ausdruck in *expbuf* paßt. Andernfalls gibt sie Null zurück. Im Erfolgsfall werden durch den Aufruf von `step` zwei externe Zeiger über einen Seiteneffekt mit Adressen versorgt: Der Zeiger *loc1* verweist auf das erste Zeichen der gefundenen Teilzeichenkette, während *loc2* auf das erste Zeichen hinter der gefundenen Zeichenreihe zeigt. Für den Fall, daß die gesamte untersuchte Zeichenkette zu dem regulären Ausdruck paßt, gilt damit folgendes: *loc1* zeigt auf das erste Zeichen von *string* und *loc2* zeigt auf das abschließende Nullzeichen von *string*.

Die Funktion `advance` gibt einen von Null verschiedenen Wert zurück, falls die Anfangsteilzeichenkette von *string* zu dem regulären Ausdruck in *expbuf* paßt. In diesem Fall wird über einen Seiteneffekt ein externer Zeiger *loc2* so gesetzt, daß er auf das erste Zeichen hinter der gefundenen Teilzeichenkette zeigt.

Trifft die Funktion `advance` in dem regulären Ausdruck auf das Zeichen `*` oder auf die Sequenz `\{ \}`, versetzt sie ihren Zeiger auf die zu untersuchende Zeichenkette so weit als möglich nach vorn und ruft sich selbst iterativ auf, um zu prüfen, ob die restliche Zeichenreihe zum Rest des regulären Ausdrucks paßt. Solange keine Übereinstimmung gefunden wird, setzt `advance` rückwärts neu auf der Zeichenkette auf, bis eine Übereinstimmung auftritt oder der Punkt erreicht wird, ab dem die Teilzeichenkette zu `*` oder `\{ \}` gepaßt hat. Manchmal ist es wünschenswert, daß das Neuaufsetzen vor Erreichen dieses Punkts gestoppt wird. Wird während des Neuaufsetzens an einem Punkt festgestellt, daß der externe Zeiger *locs* auf diesen verweist, verläßt `advance` die Schleife und gibt Null zurück.

Die Namen der externen Variablen *circof*, *sed* und *nbra* sind reserviert.

EXIT-CODES

Bei fehlerfreiem Ablauf benutzt die Funktion `compile` das Makro `RETURN`, im Fehlerfall das Makro `ERROR` (siehe oben). Bei einem Treffer geben die beiden Funktionen `step` und `advance` einen Wert ungleich Null zurück, sonst Null. Fehlerfälle:

- 11 Bereich zu groß.
- 16 Falsche Zahl.
- 25 \ Ziffer ungültig.
- 36 Falsches oder fehlendes Trennzeichen.
- 41 Kein Suchstring gespeichert.
- 42 \ (\) unpaarig.
- 43 Zu viele \ (.
- 44 Mehr als zwei Zahlen in \{ \}.
- 45 } erwartet nach \.
- 46 Erste Zahl in \{ \} ist größer als zweite.
- 49 [] unpaarig.
- 50 Überlauf bei der Berechnung des regulären Ausdrucks.

BEISPIEL

Das folgende Beispiel zeigt, wie die Makros und Funktionsaufrufe zur Berechnung regulärer Ausdrücke in einem Programm definiert werden können:

```
#define INIT          register char *sp = instr;
#define GETC          (*sp++)
#define PEEKC         (*sp)
#define UNGETC(c)     (--sp)
#define RETURN(*c)    return;
#define ERROR(c)      regerr

#include <regexp.h>

...
(void) compile(*argv, expbuf, &expbuf[ESIZE], '\0');
...
if (step(linebuf, expbuf))
    succeed;
```

BEZEICHNUNG

siginfo – Information zur Signalerzeugung

ÜBERSICHT

```
#include <siginfo.h>
```

BESCHREIBUNG

Wenn ein Prozeß ein Signal bekommt, möchte er möglicherweise wissen, warum das System dieses Signal erzeugt hat (siehe `sigaction(2)`). Weiterhin kann ein Prozeß, der seine Kindprozesse überwacht, Informationen erhalten, die ihm Auskunft darüber geben, warum ein Kindprozeß seinen Zustand geändert hat (siehe `waitid(2)`). In beiden Fällen übergibt das System die entsprechende Information in einer Struktur vom Typ `siginfo_t` folgenden Inhalts:

```
int si_signo    /* Signalnummer */
int si_errno    /* Fehlernummer */
int si_code     /* Signalcode */
```

In `si_signo` steht die vom System erzeugte Signalnummer (für die Funktion `waitid(2)` ist `si_signo` stets gleich `SIGCHLD`).

Falls `si_errno` ungleich Null ist, steht hier die zu diesem Signal gehörige Fehlernummer, wie sie in `errno.h` definiert ist.

Das Feldelement `si_code` enthält einen Code, der den Grund zur Erzeugung des Signals enthält; ist sein Wert kleiner gleich Null, so wurde das Signal durch einen Benutzerprozeß erzeugt (siehe `kill(2)` und `sigsend(2)`). In diesem Fall enthält die `siginfo` Struktur zusätzlich die folgenden Informationen:

```
pid_t si_pid    /* Prozeßnummer des Senders */
uid_t si_uid    /* Benutzernummer des Senders */
```

Andernfalls steht in `si_code` einer der folgenden signalspezifischen Gründe zur Erzeugung des Signals:

Signal	Code	Grund
SIGILL	ILL_ILLOPC	ungültiger Befehl
	ILL_ILLOPN	ungültiger Operand
	ILL_ILLADR	ungültige Adressierungsart
	ILL_ILLTRP	ungültige Signalfalle (trap)
	ILL_PRVOPC	privilegierter Befehl
	ILL_PRVREG	privilegiertes Register
	ILL_COPROC	Koprozessorfehler
	ILL_BADSTK	interner Stapelfehler
SIGFPE	FPE_INTDIV	Ganzzahldivision durch Null
	FPE_INTOVF	Ganzzahl-Überlauf
	FPE_FLTDIV	Gleitkommadivision durch Null
	FPE_FLTOVF	Gleitkomma-Überlauf
	FPE_FLTUND	Gleitkomma-Unterlauf
	FPE_FLTRES	ungenaueres Gleitkommaergebnis
	FPE_FLTINV	ungültige Gleitkomma-Operation
	FPE_FLTSUB	Bereichsüberschreitung bei Indizes

Signal	Code	Grund
SIGSEGV	SEGV_MAPERR	ungültige Objektadresse
	SEGV_ACCERR	ungültige Zugriffsrechte für Abbildungsobjekt
SIGBUS	BUS_ADRALN	ungültige Adreßausrichtung
	BUS_ADRERR	physische Adresse existiert nicht
	BUS_OBJERR	objektspezifischer Hardwarefehler
SIGTRAP	TRAP_BRKPT	Prozeß-Haltepunkt
	TRAP_TRACE	Prozeßüberwachungs-Haltepunkt
SIGCHLD	CLD_EXITED	Kindprozeß hat sich beendet
	CLD_KILLED	Kindprozeß wurde gewaltsam beendet
	CLD_DUMPED	Kindprozeß wurde fehlerhaft beendet
	CLD_TRAPPED	überwachter Kindprozeß hat angehalten
	CLD_STOPPED	Kindprozeß hat angehalten
	CLD_CONTINUED	Kindprozeß hat Ausführung fortgesetzt
SIGPOLL	POLL_IN	Eingabedaten verfügbar
	POLL_OUT	Ausgabepuffer verfügbar
	POLL_MSG	Eingabenachricht verfügbar
	POLL_ERR	E/A-Fehler
	POLL_PRI	Eingabe hoher Priorität verfügbar
	POLL_HUP	keine Geräteverbindung

Außerdem stehen für Signale, die vom Systemkern erzeugt wurden, noch die folgenden Informationen zur Verfügung:

Signal	Feld	Wert
SIGILL	caddr_t si_addr	Adresse des fehlerverursachenden Befehls
SIGFPE		
SIGSEGV	caddr_t si_addr	Adresse, die Speicherfehler verursachte
SIGBUS		
SIGCHLD	pid_t si_pid	Prozeßnummer PID des Kindprozesses
	int si_status	Exit-Status oder Signal
SIGPOLL	long si_band	Leitungsereignis für POLL_IN, POLL_OUT oder POLL_MSG

SIEHE AUCH

sigaction(2), waitid(2), signal(5).

HINWEIS

Für SIGCHLD gilt folgendes: Ist si_code gleich CLD_EXITED, so enthält si_status den Exit-Status des Prozesses; andernfalls steht dort die Signalnummer des Signals, welches für die Zustandsänderung des Prozesses verantwortlich war. Bei manchen Implementierungen mag der genaue Wert von si_addr nicht verfügbar sein; in diesem Fall ist aber sichergestellt, daß si_addr auf die gleiche Speicherseite verweist, in der der fehlerhafte Befehl oder der Speicherfehler auftrat.

BEZEICHNUNG

signal – Signale

ÜBERSICHT

#include <signal.h>

BESCHREIBUNG

Ein Signal ist die Anzeige eines asynchronen Ereignisses. Man spricht von Signalerzeugung für (oder Signalzustellung an) einen Prozeß, die genau dann eintritt, wenn das diesem Signal zugeordnete Ereignis eintritt. Solche Ereignisse sind beispielsweise Hardwarefehler, Zeitablauf, Aktivitäten am Terminal und Systemaufrufe von `kill` oder `sigsend`. Unter bestimmten Umständen erzeugt ein Ereignis Signale für mehrere Prozesse gleichzeitig. Es ist möglich, daß Prozesse Einzelheiten über die Quelle und den Grund für die Erzeugung von Signalen anfordern (siehe `siginfo(5)`).

Jeder Prozeß kann als Reaktion auf ein Signal eine bestimmte Systemtätigkeit vereinbaren, die man Signalverarbeitung nennt. Die jeweils durch das System zu ergreifende Aktion ist durch den Vaterprozeß initialisiert. Eine einmal eingestellte Aktion für ein bestimmtes Signal bleibt gewöhnlich solange gültig, bis diese explizit mittels eines Aufrufs von `sigaction`, `signal` oder `sigset` durch eine neue ersetzt wird oder bis sich der Prozeß mittels `exec` überlagert (siehe `sigaction(2)` und `signal(2)`). Im letzteren Fall wird die Signalbehandlung aller abgefangenen Signale auf `SIG_DFL` zurückgesetzt. Darüberhinaus gibt es noch die Möglichkeit, das System dazu zu veranlassen, die Signalbehandlung eines abgefangenen Signals unmittelbar danach automatisch wieder auf `SIG_DFL` zu setzen (siehe `sigaction(2)` und `signal(2)`).

Ein Signal gilt als an den Prozeß übergeben, wenn die entsprechende Aktion für dieses Signal im Prozeß ausgeführt wird. Während des Zeitraums zwischen Signalerzeugung und seiner Zustellung bezeichnet man ein Signal als anstehend (siehe `sigpending(2)`). Gewöhnlich kann dieser Zeitraum von einer Applikation nicht bestimmt werden, jedoch ist es möglich, die Auslieferung eines Signals an den Prozeß zu blockieren (siehe `signal(2)` und `sigprocmask(2)`). Wird ein blockiertes Signal erzeugt und ist als dessen Signalbehandlung "nicht ignorieren" eingestellt, verharrt das Signal im Zustand des "Anstehens", bis entweder die Blockierung aufgehoben wird oder aber die Signalbehandlung auf "ignorieren" umgestellt wird. Ist für ein blockiertes Signal als Signalbehandlung "ignorieren" eingestellt, wird das Signal bei der Erzeugung unverzüglich verworfen.

Für jeden Prozeß existiert eine Signalmaske, die die blockierten Signale ausblendet, so daß diese nicht zugestellt werden (siehe `sigprocmask(2)`). Diese Signalmaske ist durch den Vaterprozeß initialisiert.

Die Art der Reaktion auf ein Signal wird zum Zeitpunkt der Auslieferung bestimmt, so daß zwischen Erzeugung und Auslieferung jederzeit noch Änderungen möglich sind, und zwar unabhängig von den ursprünglichen Ursachen der Signalerzeugung.

Es folgen die aktuell gültigen Signale, wie sie in `signal.h` definiert sind:

Name	Wert	Standard	Ereignis
SIGHUP	1	Exit	Hängezustand (siehe <code>termio(7)</code>)
SIGINT	2	Exit	Unterbrechung (siehe <code>termio(7)</code>)
SIGQUIT	3	Core	Abbruch (siehe <code>termio(7)</code>)
SIGILL	4	Core	ungültiger Befehl
SIGTRAP	5	Core	Überwachungshaltepunkt
SIGIOT	6	Core	Abbruch
SIGEMT	7	Core	Emulationsunterbrechung (Trap)
SIGFPE	8	Core	Ausnahmebedingung Arithmetik
SIGKILL	9	Exit	unbedingter Abbruch
SIGBUS	10	Core	Busfehler
SIGSEGV	11	Core	Adreßfehler
SIGSYS	12	Core	falscher Systemaufruf
SIGPIPE	13	Exit	unterbrochene Pipe
SIGALRM	14	Exit	Alarm
SIGTERM	15	Exit	Beendigung
SIGUSR1	16	Exit	Benutzersignal 1
SIGUSR2	17	Exit	Benutzersignal 2
SIGCHLD	18	Ignore	Kindprozeßstatus (POSIX-konformes Alias)
SIGPWR	19	Ignore	Stromausfall/Neustart
SIGWINCH	20	Ignore	Änderung der Fenstergröße
SIGURG	21	Ignore	dringende Socket-Bedingung
SIGIO	22	Ignore	Socket E/A möglich (SIGPOLL Alias)
SIGSTOP	23	Stop	gestoppt durch Signal
SIGTSTP	24	Stop	gestoppt durch Benutzer (siehe <code>termio(7)</code>)
SIGCONT	25	Ignore	fortgesetzt
SIGTTIN	26	Stop	tty-Eingabe gestoppt (siehe <code>termio(7)</code>)
SIGTTOU	27	Stop	tty-Ausgabe gestoppt (siehe <code>termio(7)</code>)
SIGVTALRM	28	Exit	Erreichen einer virtuellen Zeitschranke
SIGPROF	29	Exit	Erreichen einer Zeitschranke für Auswertung
SIGXCPU	30	Core	Überschreitung der CPU-Zeit (siehe <code>getrlimit(2)</code>)
SIGXFSZ	31	Core	Überschreitung der Dateigröße (siehe <code>getrlimit(2)</code>)

Ein Prozeß kann mit den Systemaufrufen `signal`, `sigset` oder `sigaction` eine der drei folgenden Signalbehandlungen einstellen: entweder die Voreinstellung benutzen, das Signal ignorieren oder das Signal abfangen.

Voreinstellung: SIG_DFL

Die Angabe von SIG_DFL bewirkt die Standardeinstellung für die Signalbehandlung, wie sie in der oben aufgeführten Liste angegeben ist. Dabei bedeuten die einzelnen Angaben folgendes:

- Exit Bei Zustellung des Signals wird der Prozeß mit allen unter exit aufgeführten Konsequenzen beendet.
- Core Bei Zustellung des Signals wird der Prozeß mit allen unter exit aufgeführten Konsequenzen beendet. Außerdem wird im aktuellen Verzeichnis ein Speicherabzug (core) des Prozesses erzeugt.
- Stop Beim Empfang des Signals hat der betroffene Prozeß anzuhalten.
- Ignore Der Prozeß ignoriert das Signal. Gleichbedeutend dazu ist die Einstellung der Signalbehandlung auf SIG_IGN.

Signal ignorieren: SIG_IGN

Die Einstellung der Signalbehandlung auf SIG_IGN bewirkt, daß das Signal ignoriert wird.

Signal abfangen: Funktionsadresse

Die Einstellung der Signalbehandlung auf eine Funktionsadresse bedeutet, daß beim Eintreffen des Signals die dadurch angegebene Funktion als Signalbehandlungsroutine ausgeführt werden soll. Normalerweise ist die Signalnummer das einzige Argument für die Signalbehandlungsroutine. Es können allerdings mehr Argumente nötig sein, falls die Behandlungsroutine mittels sigaction umgestellt wurde (siehe sigaction(2)). Nach der Rückkehr aus der Signalbehandlungsroutine setzt der Prozeß seine Ausführung an dem Punkt fort, an dem er unterbrochen wurde, es sei denn, daß diese Routine etwas anderes vorsieht. Bezeichnet die angegebene Adresse keine gültige Funktion, ist das resultierende Verhalten undefiniert.

Falls die Signalbehandlung mittels sigset oder sigaction eingestellt wurde, wird das entsprechende Signal automatisch durch das System blockiert, solange die Abfangroutine abläuft. Wird die Abfangroutine über den Systemaufruf longjmp verlassen (siehe setjmp(3C)), muß das Signal explizit durch den Benutzer freigegeben werden (siehe signal(2) und sigprocmask(2)).

Falls durch die Ausführung einer Signalbehandlungsroutine ein blockierter Systemaufruf unterbrochen wird, gibt dieser den Wert -1 zurück und setzt errno auf EINTR. Ist jedoch der Schalter SA_RESTART gesetzt, so setzt der Systemaufruf in transparenter Weise neu auf.

HINWEIS

Die Signalbehandlung für die Signale SIGKILL und SIGSTOP kann nicht verändert werden. Jeder Versuch in dieser Richtung wird vom System mit einem Fehler quittiert.

Die Signale SIGKILL und SIGSTOP können nicht blockiert werden. Dafür sorgt stillschweigend das System.

Empfängt ein Prozeß eines der Signale SIGSTOP, SIGTSTP, SIGTTIN oder SIGTTOU, werden ungeachtet der aktuellen Einstellung anstehende SIGCONT Signale verworfen.

Sowie ein Prozeß ein SIGCONT Signal erhält, werden ungeachtet der aktuellen Einstellung alle anstehenden Signale der Art SIGSTOP, SIGTSTP, SIGTTIN und SIGTTOU verworfen und der Prozeß wird fortgesetzt, falls er angehalten war.

SIGPOLL ist ein Signal, das für einen STREAM im Fall eines anstehenden "auswählbaren" Ereignisses erzeugt wird (siehe intro(2)). Allerdings muß ein Prozeß dieses Signal explizit über einen `I_SETSIG` `ioctl` Aufruf anfordern, weil sonst das Signal SIGPOLL nicht zugestellt wird.

Das Signal SIGCHLD wird einem Prozeß nur bei der Beendigung von Kindprozessen zugestellt, wenn dieser die Behandlungsroutine mittels `signal`, `sigset` oder `sigaction` eingestellt hat und der Schalter `SA_NOCLDSTOP` gesetzt ist. Andernfalls erhält er das Signal auch beim Anhalten und bei der Fortsetzung von Kindprozessen gemäß Auftragssteuerung.

Die Bezeichnung SIGCLD ist neben SIGCHLD ebenfalls in der Include-Datei aufgeführt und bedeutet dasselbe Signal. Dies ist aus Gründen der Abwärtskompatibilität der Fall. Neue Anwendungen sollten allerdings ausschließlich SIGCHLD verwenden.

Signalbehandlungen, die mit der Einstellung `SIG_IGN` geerbt wurden, sollten nicht verändert werden.

SIEHE AUCH

`exit(2)`, `getrlimit(2)`, `intro(2)`, `kill(2)`, `pause(2)`, `sigaction(2)`, `sigaltstack(2)`, `signal(2)`, `sigprocmask(2)`, `sigsend(2)`, `sigsuspend(2)`, `wait(2)`, `sigsetops(3C)`, `siginfo(5)`, `ucontext(5)`

BEZEICHNUNG

stat – vom Systemaufruf stat zurückgelieferte Daten

ÜBERSICHT

```
#include <sys/types.h>
#include <sys/stat.h>
```

BESCHREIBUNG

Die Systemaufrufe stat, lstat und fstat liefern ihre Daten in einer Struktur vom Typ stat zurück, die in stat.h definiert ist.

Die gleiche Datei enthält auch die Konstantendefinitionen für das Feld st_mode:

```
#define S_IFMT /* Dateityp */
#define S_IAMB /* Zugriffsrechte */
#define S_IFIFO /* fifo (Pipe) */
#define S_IFCHR /* zeichenorientiertes Gerät */
#define S_IFDIR /* Verzeichnis */
#define S_IFNAM /* XENIX Gerätename */
#define S_INSEM /* XENIX Semaphore - Untertyp von IFNAM */
#define S_INSHD /* XENIX shared data - Untertyp von IFNAM */
#define S_IFBLK /* blockorientiertes Gerät */
#define S_IFREG /* Normaldatei */
#define S_IFLNK /* symbolischer Verweis */
#define S_ISUID /* s-Bit (Eigentümer) */
#define S_ISGID /* s-Bit (Gruppe) */
#define S_ISVTX /* t-Bit (Codesegment bewahren) */
#define S_IREAD /* Lese-Erlaubnis, Eigentümer */
#define S_IWRITE /* Schreib-Erlaubnis, Eigentümer */
#define S_IEXEC /* Ausführ/Durchsuch-Erlaubnis, Eigentümer */
#define S_ENFMT /* Schalter für Satzsperrung */
#define S_IRWXU /* rwx (lesen, schreiben, ausführen): Eigentümer */
#define S_IRUSR /* Lese-Erlaubnis: Eigentümer */
#define S_IWUSR /* Schreib-Erlaubnis: Eigentümer */
#define S_IXUSR /* Ausführ-Erlaubnis: Eigentümer */
#define S_IRWXG /* rwx: Gruppe */
#define S_IRGRP /* Lese-Erlaubnis: Gruppe */
#define S_IWGRP /* Schreib-Erlaubnis: Gruppe */
#define S_IXGRP /* Ausführ-Erlaubnis: Gruppe */
#define S_IRWXO /* rwx: andere Benutzer */
#define S_IROTH /* Lese-Erlaubnis: andere Benutzer */
#define S_IWOTH /* Schreib-Erlaubnis: andere Benutzer */
#define S_IXOTH /* Ausführ-Erlaubnis: andere Benutzer */
```

Die folgenden Makros dienen der Konformität zu POSIX:

#define	S_ISBLK(mode)	blockorientierte Gerätedatei
#define	S_ISCHR(mode)	zeichenorientierte Gerätedatei
#define	S_ISDIR(mode)	Verzeichnis
#define	S_ISFIFO(mode)	(benannte) Pipe
#define	S_ISREG(mode)	normale Datei

SIEHE AUCH

stat(2), types(5).

BEZEICHNUNG

`term` – übliche Terminalbezeichnungen

BESCHREIBUNG

Terminalbezeichnungen werden als Teil der Shell-Umgebung in der Umgebungsvariablen `TERM` gehalten (siehe `sh(1)`, `profile(4)` und `environ(5)`). Diese Bezeichnungen werden von bestimmten Kommandos (z.B. `tabs`, `tput` und `vi`) und Funktionen (vgl. `curses(3X)`) benutzt.

Die Dateien in `/usr/share/lib/terminfo` dienen der Beschreibung von Terminals und deren Benennung. Die Dateien besitzen das in `terminfo(4)` beschriebene Format. Einträge in Quelldateien für `terminfo` bestehen aus einer Anzahl von Feldern, die durch Kommata getrennt sind. Leerzeichen hinter den Kommata bleiben unbeachtet. Zur Ausgabe der Beschreibung eines Terminals `term` dient das Kommando `infocmp -I term` (siehe `infocmp(1M)`). Für jede Terminalbeschreibung in der Datenbank `terminfo` enthält jeweils die erste Zeile die Bezeichnungen des Terminals, wie sie `terminfo` kennt. Die Namen sind durch einen senkrechten Strich (`|`) voneinander getrennt. Der erste Name in dieser Zeile ist die meistbenutzte Abkürzung für dieses Terminal und genau dieser Name sollte auch benutzt werden, um die Umgebungsvariable `TERM` in der Datei `$HOME/.profile` damit zu belegen (siehe `profile(4)`). Der letzte angegebene Name sollte eine ausführliche und genaue Bezeichnung des Terminals sein und kann aus Gründen der Lesbarkeit Leerzeichen enthalten. Die Namen in dieser Zeile werden alle als Synonyme für dieses Terminal betrachtet. Sie sollten, mit Ausnahme des letzten, keine Leerzeichen enthalten und müssen in den ersten 14 Zeichen eindeutig sein.

Außer der letzten ausführlichen Bezeichnung sollten bei der Vergabe von Terminalnamen die folgenden Konventionen beachtet werden. Im Basisnamen des Terminals sollte ein Hinweis auf die besondere Hardware des Terminals auftauchen, beispielsweise `97801` für das Terminal 97801 von SNI. Dieser Name sollte keine Bindestriche enthalten, es sei denn, daß die gewählten Synonyme zu keiner Konfliktsituation mit anderen Namen führen. Der Basisname eines Terminals besteht aus bis zu acht alphanumerischen Zeichen (den Kleinbuchstaben von `a` bis `z` und den Ziffern von `0` bis `9`). Im allgemeinen sollten die Basisnamen eher auf die Originalhersteller als auf örtliche Händler verweisen. Ein von einem Hersteller erworbenes Terminal sollte keine verschiedenen Basisnamen haben. Gleiche Terminals unterschiedlicher Ausbaustufe oder unterschiedlicher Funktionalität, die hardwarebedingt oder benutzergesteuert ist, zeigen dies in ihrem Namen durch einen Bindestrich an, dem ein Hinweis auf den entsprechenden Modus folgt. So wird zum Beispiel ein AT&T 4425 Terminal mit 132 Spalten durch `att4425-w` bezeichnet. Wo es angebracht ist, sollten die folgenden Suffixe verwendet werden:

Suffix	Bedeutung	Beispiel
-w	Modus breit (mehr als 80 Spalten)	att4425-w
-am	Mit auto. Rändern (normalerweise Standard)	vt100-am
-nam	Ohne auto. Ränder	vt100-nam
-n	Anzahl Bildschirmzeilen	aaa-60
-na	Keine Pfeiltasten (lokale Bearb.)	c100-na
-np	Anzahl der Speicherseiten	c100-4p
-rv	inverse Darstellung	att4415-rv

Um bei der Beschreibung unterschiedlicher Modi (z.B. -w) von Terminals Konflikte bezüglich der Namenskonventionen zu vermeiden, sollten in den Basisnamen von Terminals keine Bindestriche stehen. Weiterhin hat es sich als praktisch erwiesen, sämtliche Namen in der Datenbank terminfo(4) so zu wählen, daß sie eindeutig sind. Terminaleinträge, die mittels der use=-Anweisung lediglich der Einschleusung in andere Einträge dienen, sollten ein '+' (wie in 4415+nl) in ihrem Namen haben.

Es folgt eine Reihe bekannter Terminalnamen: (Um eine komplette Liste zu erhalten, kann man das Kommando `ls -C /usr/share/lib/terminfo/?` eingeben.)

2621, hp2621	Hewlett-Packard Serie 2621
2631	Hewlett-Packard 2631 Zeilendrucker
2631-c	Hewlett-Packard 2631 Zeilendrucker, komprimierter Modus
2631-e	Hewlett-Packard 2631 Zeilendrucker, erweiterter Modus
2640, hp2640	Hewlett-Packard Serie 2640
2645, hp2645	Hewlett-Packard Serie 2645
3270	IBM Modell 3270
33, tty33	AT&T Teletype Modell 33 KSR
35, tty35	AT&T Teletype Modell 35 KSR
37, tty37	AT&T Teletype Modell 37 KSR
4000a	Trendata 4000a
4014, tek4014	TEKTRONIX 4014
40, tty40	AT&T Teletype Dataspeed 40/2
43, tty43	AT&T Teletype Modell 43 KSR
4410, 5410	AT&T 4410/5410 80 Spalte, Version 2
4410-nfk, 5410-nfk	AT&T 4410/5410 ohne Funktionstasten, Version 1
4410-ns1, 5410-ns1	AT&T 4410/5410 pln nicht definiert
4410-w, 5410-w	AT&T 4410/5410 132 Spalten
4410v1, 5410v1	AT&T 4410/5410 80 Spalten, Version 1
4410v1-w, 5410v1-w	AT&T 4410/5410 132 Spalten, Version 1
4415, 5420	AT&T 4415/5420 80 Spalten

term(5)**term(5)**

4415-nl, 5420-nl	AT&T 4415/5420 keine Änderung von Marken
4415-rv, 5420-rv	AT&T 4415/5420 80 Spalten, Inversdarstellung
4415-rv-nl, 5420-rv-nl	AT&T 4415/5420 Inversdarstellung, keine Änderung von Marken
4415-w, 5420-w	AT&T 4415/5420 132 Spalten
4415-w-nl, 5420-w-nl	AT&T 4415/5420 132 Spalten, keine Änderung von Marken
4415-w-rv, 5420-w-rv	AT&T 4415/5420 132 Spalten, Inversdarstellung
4418, 5418	AT&T 5418 80 Spalten
4418-w, 5418-w	AT&T 5418 132 Spalten
4420	AT&T Teletype Modell 4420
4424	AT&T Teletype Modell 4424
4424-2	AT&T Teletype Modell 4424 in Funktionsgruppe ii
4425, 5425	AT&T 4425/5425
4425-fk, 5425-fk	AT&T 4425/5425 keine Funktionstasten
4425-nl, 5425-nl	AT&T 4425/5425 80 Spalten, keine Änderung von Marken
4425-w, 5425-w	AT&T 4425/5425 132 Spalten
4425-w-fk, 5425-w-fk	AT&T 4425/5425 132 Spalten, keine Funktionstasten
4425-nl-w, 5425-nl-w	AT&T 4425/5425 132 Spalten, keine Änderung von Marken
4426	AT&T Teletype Modell 4426S
450	DASI 450 (gleich zu Diablo 1620)
450-12	DASI 450 12-pitch Modus
500, att500	AT&T-IS 500 Terminal
510, 510a	AT&T 510/510a 80 Spalten
513bct, att513	AT&T 513 bct Terminal
5320	AT&T 5320 Hardcopy Terminal
5420_2	AT&T 5420 Modell 2, 80 Spalten
5420_2-w	AT&T 5420 Modell 2, 132 Spalten
5620, dmd	AT&T 5620 Terminal, 88 Spalten
5620-24, dmd-24	AT&T Teletype Modell DMD 5620 in einem 24x80 Fenster
5620-34, dmd-34	AT&T Teletype Modell DMD 5620 in einem 34x80 Fenster
610, 610bct	AT&T 610 bct Terminal, 80 Spalten

term(5)

610-w, 610bct-w	AT&T 610 bct Terminal, 132 Spalten
630, 630MTG	AT&T 630 Terminal f. Multi-Tasking und Grafik
7300, pc7300, unix_pc	AT&T UNIX PC Modell 7300
735, ti	Texas Instruments TI735 und TI725
745	Texas Instruments TI745
97801, 97807	Siemens Nixdorf Informationssysteme
dumb	generischer Name für Terminals, denen spezielle Escape-Sequenzen und Spezialfunktionen fehlen
hp	Hewlett-Packard (gleich zu 2645)
lp	generischer Name für Zeilendrucker
pt505	AT&T Personal Terminal 505 (22 Zeilen)
pt505-24	AT&T Personal Terminal 505 (24 Zeilen)
sync	generischer Name für synchrone Terminals, die zu Teletype Modell 4540 kompatibel sind

Kommandos, deren Verhalten vom Terminaltyp abhängt, sollten Argumente der Form `-T term` akzeptieren, wobei *term* einer der oben angegebenen Namen ist. Fehlt ein solches Argument, sollten sich diese Kommandos den Terminaltyp aus der Umgebungsvariablen `TERM` besorgen, die wiederum natürlich *term* enthalten sollte.

DATEIEN

`/usr/share/lib/terminfo/?/*`

Datenbank mit Binärversionen der Terminalbeschreibungen

SIEHE AUCH

`infocmp(1M)`, `profile(4)`, `terminfo(4)`, `environ(5)` im *Referenzhandbuch für Systemverwalter*.

`sh(1)`, `stty(1)`, `tabs(1)`, `tput(1)`, `vi(1)` in den *Kommandos*.

`curses(3X)`.

term(5)

BEZEICHNUNG

xtpROTO – Protokoll des xt-Treibers für Multiplex-Kanäle

BESCHREIBUNG

Das xt-Protokoll dient der Kommunikation zwischen Host-Prozessen am UNIX-System und fensterfähigen Terminals von AT&T, die unter dem Kommando `layers` laufen (siehe xt(7)). Es handelt sich um ein Multiplex-Protokoll, das den Datenverkehr zwischen Host-Prozessen und fensterfähigen Terminals regelt und mehrere virtuelle Terminalsitzungen über eine einzige Leitung ermöglicht. Das Protokoll ist am Host durch den xt-Treiber und am Terminal durch entsprechende Firmware implementiert.

Der xt-Treiber stellt zwei unterschiedliche Protokolle der unteren Ebene zur Verfügung. Welches der beiden benutzt wird, hängt von der Zuverlässigkeit der Leitungsverbindung zum Kommunikationsaustausch mit dem Terminal ab. Bei unzuverlässigen Verbindungen, wie z.B. RS-232, wird das normale xt-Protokoll verwendet. Dieses sorgt für die Flußsteuerung und Fehlerkorrektur und stellt so eine fehlerfreie Datenübermittlung sicher. Bei zuverlässigen Verbindungen, wie z.B. einem lokalen Netzwerk (LAN), wird das xt-Netzwerkprotokoll benutzt. Mit dem Ziel eines maximal möglichen Durchsatzes verläßt sich das xt-Netzwerkprotokoll darauf, daß das zugrundeliegende Netzwerk die Flußsteuerung und die Fehlerkorrektur übernimmt.

Das Kommando `layers` fragt mittels eines Systemaufrufs der Art `A_XTPROTO` JAGENT `ioctl` bei dem fensterfähigen Terminal an, welches Protokoll zu benutzen ist [siehe `layers`(5)]. Aufgrund des Rückgabewertes in `A_XTPROTO`, der Übertragungsgeschwindigkeit und der Option `-m` beim Aufruf entscheidet dann das Kommando `layers`, ob das normale oder das Netzwerkprotokoll von xt zu verwenden ist.

Das normale xt-Protokoll verwendet Pakete mit einem Paketkopf von 2 Bytes, worin eine 3-Bit Sequenznummer, eine 3-Bit Kanalnummer, ein Schalter zur Steuerung und ein Byte für die Länge des Datenteils verschlüsselt sind. Der Anteil der Daten im Paket darf nicht mehr als 252 Bytes belegen. Das festgelegte Maximum kann bei niedrigeren Baudraten oder bei Angabe der Option `-m` beim Aufruf von `layers` kleiner als 252 sein. Weiter gilt, daß bei einigen der früheren fensterfähigen Terminals das Maximum des Datenanteils auf 32 Bytes festgelegt ist. Für die Übertragungsrichtung vom Terminal zum Host gilt generell eine Beschränkung auf 32 Bytes. Der Paketanhang enthält in 2 Bytes die Prüfsumme CRC-16. Die Kanäle sind alle doppelt gepuffert.

Normale xt-Pakete, die in der richtigen Reihenfolge fehlerfrei empfangen wurden, werden mit einem Steuerpaket quittiert, welches ein ACK enthält. Pakete mit einer falschen Sequenznummer verursachen die Rückübermittlung eines Steuerpakets, welches ein NAK enthält. Daraufhin werden alle nicht positiv quittierten Pakete der Reihe nach neu übertragen.

Wird eine Zeitschranke erreicht, die von der Übertragungsgeschwindigkeit abhängt, werden normale xt-Pakete, die bis dahin nicht positiv quittiert sind, ebenfalls neu übertragen. Eine weitere Zeitschranke legt fest, innerhalb welcher Zeitspanne unvollständig empfangene Pakete verworfen werden.

Das xt-Netzwerkprotokoll verwendet einen 3-Byte Paketkopf, in dem eine 3-Bit Kanalnummer, verschiedene Steuerbits und ein 2-Byte Wert für die Größe des Datenteils verschlüsselt sind. Für die Übertragungsrichtung vom Host zum Terminal gibt es keine Beschränkung bezüglich der Größe des Datenanteils, in der umgekehrten Richtung darf er nicht mehr als 1025 Bytes betragen.

Bei der Verwendung des xt-Netzwerkprotokolls braucht es keine Prüfsummen oder Zeitschranken, da das zugrundeliegende lokale Netzwerk für die fehlerfreie Datenübermittlung sorgt.

Das xt-Netzwerkprotokoll benutzt einen einfachen Mechanismus zur Flußsteuerung, um die Datenmenge, die an ein Fenster in einem Terminal gesandt wird, zu begrenzen, bis eine positive Quittung `NETWORK XT ACK` vom Host empfangen wurde. Wenn beispielsweise ein Benutzer die Bildschirmausgabe angehalten hat, liest das Terminal keine übermittelten Daten. Vor diesem Hintergrund ist die Begrenzung der Datenmenge sinnvoll und notwendig, um nicht Daten wiederholt senden zu müssen und damit die Übermittlung von Daten an andere Fenster zu blockieren. Prozesse im Terminal, die ihre Eingabe stets sehr rasch abholen, können die Flußsteuerung des xt Netzwerkprotokolls ausschalten und so insgesamt einen besseren Durchsatz bewirken.

DATEIEN

`/usr/include/sys/xtpROTO.h` Protokolldefinitionen für Multiplex-Kanäle

SIEHE AUCH

`jagent(5)`, `layers(5)`.

`layers(1)` in den *Kommandos*.

`xt(7)` in *CES, Leitfaden und Werkzeuge für die Programmierung mit C*.

7

Gerätedateien

Gerätedateien

7-1

BEZEICHNUNG

`intro` – Einführung zum Kapitel Gerätedateien

BESCHREIBUNG

Dieses Kapitel beschreibt verschiedene Gerätedateien, die bestimmten Hardware-Peripheriegeräten zugeordnet sind, sowie SINIX Systemgeräte-Treiber. STREAMS (siehe `intro(2)`) Software-Treiber werden ebenfalls beschrieben.

Im Gegensatz zu den Namen der Gerätedateien werden die Namen der Einträge für hardware-orientierte Dateien im allgemeinen aus den Namen der Hardware gebildet. Die Eigenschaften sowohl der Hardware-Geräte als auch der entsprechenden SINIX Systemgeräte-Treiber werden im Bedarfsfall beschrieben.

Dateinamen für Plattengeräte haben z.B. folgendes Format:

```
/dev/dsk/shdXtYsZ, ...
```

```
/dev/rdsk/shdXtYsZ, ....
```

Dabei bezeichnet `r` (in `rdsk`) die zeichenorientierte Schnittstelle zu der Platte oder das im Raw-Modus betriebene Gerät. Die Angabe ohne `r` (`dsk`) bezeichnet die blockorientierte Schnittstelle zu der Platte oder das im Block-Modus betriebene Gerät. `X` steht hier für die Nummer des Controllers und `Y` für die SCSI-Targetnummer. `Z` bezeichnet die Slicenummer.

acc(7)

acc(7)

BEZEICHNUNG

`acc` – SCSI-Treiber für Worm-Plattenwechsler (Jukeboxes)

BESCHREIBUNG

Der Treiber `acc` unterstützt den Plattenwechsler (Jukebox) für optische Platten (Worm-Platten) Siemens-Nixdorf 9082 Jukebox mit LMS LD 510 Laufwerken.

Die Gerätenummer (minor device number) ist so aufgebaut:

- Bits 0-3 sind reserviert,
- Bits 4-6 beschreiben die logische Einheit (LUN, logical unit) und sind bei der 9082 Jukebox immer 0,
- Bits 7-9 beschreiben die SCSI-Targetadresse (SCSI target address),
- Bits 10-12 geben die Nummer des Host Adapter (ADP32 Controller) an einem MB-I bzw. die Nummer des Host Adapter AHA1542B an einem AT-Bus an.

ioctl-Aufrufe

Die verfügbaren `ioctl`-Aufrufe und Datenstrukturen sind in `sys/acciocctl.h` definiert.

DATEIEN

Die Gerätedateien haben folgende Namen:

`/dev/accYtZsO`

Dabei steht `Y` für die Nummer des Controllers, `Z` für die SCSI-Targetnummer und `O` für eine Option (immer 0). Für die MX300 und die MX500 gilt: Eine 1 für `Y` bezeichnet den ersten ADP32 Controller (`adp32.0`), eine 2 den zweiten ADP32 Controller (`adp32.1`) usw. Bei der WX200 bezeichnet eine 1 für `Y` den ersten AHA1542B Controller (`stos.0`), eine 2 den zweiten AHA1542B Controller (`stos.1`) usw.

Die Gerätedateien (Geräteknoten, device nodes) werden erzeugt, wenn ein neuer Kern generiert wurde und das System danach erstmals hochgefahren wird (z.B. bei der Installation). Damit die Gerätedateien korrekt erzeugt werden können, müssen die jeweiligen Geräte angeschaltet und mit dem System verbunden sein.

SIEHE AUCH

`adp32(7)`, `lad(7)`.

FEHLERMELDUNGEN

Der Treiber liefert Fehlermeldungen der Art:

`accYtZ: error text`

Dabei bezeichnet `Y` die Nummer des Controllers, `Z` die SCSI-Targetnummer und `error text` den Fehler.

BEZEICHNUNG

adp32 – NCR Host Adapter (ADP32 Controller) am MB-I

BESCHREIBUNG

Der NCR SCSI Host Adapter (ADP32 Controller) unterstützt bis zu 8 Targets am SCSI-Bus; davon ist ein Target für den Host Adapter reserviert, die anderen 7 stehen angeschlossenen Targets zur Verfügung. Pro Target unterstützt der Adapter maximal 8 logische Untereinheiten (logical subunit, LUN). An den Controllern können insgesamt maximal 90 Geräte angeschlossen sein.

Folgende SCSI-Geräte sind anschließbar:

- Festplatten (*shd*),
- Bandlaufwerke (*exa*),
- optische Platten oder Worm-Platten (*lad*),
- Worm-Plattenwechsler (Jukeboxes) (*acc*) und
- CD-ROM-Laufwerke (*cdr*) (nicht in SINIX V5.40).

Der Treiber *adp32* unterstützt maximal zwei (MX300) bzw. vier (MX500) Controller am MB-I (MBAD). Die jeweils verwendbare Anzahl von Controllern hängt von den *adp32*-Einträgen in der Konfigurationsdatei *sdevice* ab.

ioctl-Aufrufe

Die verfügbaren *ioctl*-Aufrufe und Datenstrukturen sind in *sys/adp32ioctl.h* definiert. *ioctl*-Aufrufe können nur von der *root*-Kennung aus erfolgen.

SIEHE AUCH

acc(7), *cdr(7)*, *exa(7)*, *lad(7)*, *shd(7)*.

FEHLERMELDUNGEN

Der Treiber liefert folgende Fehlermeldungen:

adp32.Y: error text oder

sadmY: error text

Dabei bezeichnet *Y* die Nummer des Controllers, *error text* den Fehler.

BEZEICHNUNG

aha – Host Adapter (AHA1542B) am AT-Bus

BESCHREIBUNG

Der SCSI Host Adapter (AHA1542B) unterstützt bis zu 8 Targets am SCSI-Bus; davon ist ein Target für den Host Adapter reserviert, die anderen 7 stehen angeschlossenen Targets zur Verfügung. Pro Target unterstützt der Adapter maximal 8 logische Untereinheiten (logical subunit, LUN). An den Controllern können insgesamt maximal 90 Geräte angeschlossen sein.

Folgende SCSI-Geräte sind anschließbar:

- Festplatten (shd),
- Bandlaufwerke (exa),
- optische Platten oder Worm-Platten (lad),
- Worm-Plattenwechsler (Jukeboxes) (acc),
- CD-ROM-Laufwerke (cdr).

Der Treiber aha unterstützt maximal zwei Controller am AT-Bus. Die jeweils verwendbare Anzahl von Controllern hängt von den aha-Einträgen in der Konfigurationsdatei sdevice ab.

ioctl-Aufrufe

Die verfügbaren ioctl-Aufrufe und Datenstrukturen sind in `sys/ahaioctl.h` definiert. ioctl-Aufrufe können nur von der root-Kennung aus erfolgen.

SIEHE AUCH

acc(7), cdr(7), exa(7), lad(7), shd(7).

FEHLERMELDUNGEN

Der Treiber liefert folgende Fehlermeldungen:

aha.Y: *error text* oder

sadmY: *error text*

Dabei bezeichnet Y die Nummer des Controllers, *error text* den Fehler.

BEZEICHNUNG

ARP – Adreßauflösungsprotokoll

ÜBERSICHT

```
#include <sys/socket.h>
#include <net/if_arp.h>
#include <netinet/in.h>

s = socket(AF_INET, SOCK_DGRAM, 0);
d = open ("/dev/arp", O_RDWR);
```

BESCHREIBUNG

ARP ist ein Protokoll, das zur dynamischen Umsetzung von Internet Protocol (IP) Adressen auf 10Mb/s Ethernet-Adressen benutzt wird. Es wird von allen 10Mb/s Ethernet-Verbindungsvorrichtungen (Schnittstellentreibern) verwendet. Es ist nicht speziell auf die Anwendung für Internet Protocol oder 10Mb/s Ethernet beschränkt, aber diese Implementierung unterstützt derzeit nur diese Kombination. Das STREAMS-Gerät /dev/arp ist keine TLI Transportvorrichtung (Transport Level Interface) und darf nicht zusammen mit der Schnittstelle TLI benutzt werden.

ARP puffert die IP-Ethernet-Adreßumsetzung. Wenn eine Schnittstelle eine Adreßumsetzung erfordert, die nicht im Systempuffer vorhanden ist, reiht ARP die Nachricht, die die Umsetzung benötigt, in die Warteschlange ein und gibt eine entsprechende Meldung auf dem zugehörigen Netzwerk aus und fordert damit die Adreßumsetzung an. Wenn eine Antwort erfolgt, wird die neue Umsetzung gepuffert und jedwede anstehende Nachricht übertragen. ARP kann maximal ein Paket in die Warteschlange einreihen, während es auf die Antwort einer Umsetzungsanforderung wartet. Es wird immer nur das letzte übertragene Paket aufgehoben.

Um die Kommunikation mit Systemen zu erleichtern, die keine ARP `ioctl()` Aufrufe verwenden, sind Eingabeaufforderungen vorgesehen, mit denen Einträge in den IP-Ethernet-Umsetzungstabellen abgelegt oder gelöscht werden können.

ANWENDUNG

```
#include <sys/sockio.h>
#include <sys/socket.h>
#include <net/if.h>
#include <net/if_arp.h>
struct arpreq arpreq;
ioctl(s, SIOCSARP, (caddr_t)&arpreq);
ioctl(s, SIOCGARP, (caddr_t)&arpreq);
ioctl(s, SIOCDEARP, (caddr_t)&arpreq);
```

Jede `ioctl()` Eingabeaufforderung benutzt die gleiche Struktur als Argument. `SIOCSARP` setzt einen ARP-Eintrag, `SIOCGARP` greift auf einen ARP-Eintrag zu und `SIOCDEARP` löscht einen ARP-Eintrag. Diese `ioctl()` Eingabeaufforderungen können für jeden Socketdeskriptor `s` der Internet-Familie oder für den Deskriptor eines ARP-Geräts angegeben werden, allerdings nur durch einen privilegierten Benutzer.

Die `arpreq`-Struktur enthält folgende Information:

```
/*
 * ARP ioctl request
 */
struct arpreq {
    struct sockaddr arp_pa;    /* Protokolladresse */
    struct sockaddr arp_ha;    /* Hardware Adresse */
    int arp_flags;             /* Schalter */
};
/* Werte für arp_Schalter */
#define ATF_COM 0x2    /* vollst. Eintrag (arp_ha valid) */
#define ATF_PERM 0x4   /* permanenter Eintrag */
#define ATF_PUBL 0x8   /* (Antwort für anderen Host) */
#define ATF_USETRAILERS 0x10 /* Trailer-Pakete an Host */
```

Die Adreßfamilie für die `arp_pa` Socketadresse muß `AF_INET` sein; für die `arp_ha` Socketadresse muß sie `AF_UNSPEC` sein. Die einzigen Markierungsbits, die angegeben werden dürfen, sind `ATF_PERM`, `ATF_PUBL` und `ATF_USETRAILERS`. `ATF_PERM` macht den Eintrag permanent, wenn die `ioctl()` Anforderung erfolgreich war. Aufgrund der besonderen Natur der ARP-Tabellen kann die `ioctl()` Anforderung abgewiesen werden, wenn zuviele permanente IP-Adressen auf die gleiche Position verweisen. `ATF_PUBL` gibt an, daß der ARP-Code den ARP-Anforderungen des angezeigten lokalen Rechners entsprechen sollte, falls sie von anderen Maschinen kommen. Dies ermöglicht es einem lokalen Rechner, sich wie ein ARP Server zu verhalten, was dann besonders nützlich sein kann, wenn eine Verbindung von einem nur-ARP-Gerät zu einem nicht-ARP-Gerät hergestellt werden soll.

ARP wird außerdem verwendet, um die Verwendung von IP-Trailer-Einbettungen zu ermöglichen. Trailer sind eine alternative Einbettung, die eine effiziente Paketausrichtung für große Pakete ermöglicht, auch wenn diese Pakete variabel lange Anfangsetiketten haben. Rechner, die IP-Trailer-Einbettung empfangen wollen, zeigen dies an, indem sie zusätzliche ARP-Übersetzungsantworten zusammen mit den Antworten auf IP-Anforderungen abschicken; sie werden auch als Erwiderung auf IP-Übersetzungsantworten geschickt. Die Übermittlung ist auf diese Weise völlig symmetrisch, indem sowohl einer der beiden Rechner als auch beide Rechner Trailer anfordern können. Die `ATF_USETRAILERS` Marke wird verwendet, um den Eingang einer solchen Antwort aufzuzeichnen, und ermöglicht auf diese Weise die Übertragung von nachfolgenden Paketen an diesen Rechner.

ARP wartet passiv auf Rechner, die einen lokalen Rechner darstellen, d.h. die auf eine Umsetzungsanforderung für die Adresse eines lokalen Rechners antworten.

SIEHE AUCH

arp(1M), ifconfig(1M), if(3N), inet(7).

Plummer, Dave, *"An Ethernet Address Resolution Protocol -or- Converting Network Protocol Addresses to 48.bit Ethernet Addresses for Transmission on Ethernet Hardware,"* RFC 826, Network Information Center, SRI International, Menlo Park, Calif., November 1982.

Leffler, Sam, and Michael Karels, *"Trailer Encapsulations,"* RFC 893, Network Information Center, SRI International, Menlo Park, Calif., April 1984.

BEZEICHNUNG

`cdr` – SCSI-Treiber für CD-ROM

BESCHREIBUNG

Der Treiber `cdr` unterstützt das CD-ROM-Laufwerk SNI CDU-541. Andere SCSI-CD-ROM-Laufwerke, die der SCSI-Spezifikation genügen, sollten mit dem Treiber arbeiten können. Doch verhalten sich die Kommandos `get/set mode` nicht auf allen anderen Laufwerken gleich.

Der Hersteller schreibt die Daten auf die CD-ROMs. Der Benutzer kann die Platte nur lesen. CD-ROMs sind blockorientierte Medien. Man kann deshalb nur auf ganze Blöcke zugreifen und muß in Einheiten der Blockgröße lesen. Die Blockgröße beträgt normalerweise 2048 Bytes. Diese Größe wird standardmäßig vom Treiber verwendet.

Die Gerätenummer (minor device number) ist so aufgebaut:

- Bits 0-3 sind für eine Partitionnummer reserviert, sie werden nicht benutzt.
- Bits 4-6 beschreiben eine logische Einheit (LUN, logical unit) und werden gegenwärtig nicht benutzt. Sie sind deshalb 0.
- Bits 7-9 bezeichnen die SCSI-Targetadresse (SCSI target address).
- Bits 10-12 geben die Nummer des Host Adapters (ADP32 Controllers bzw. AHA 1542B) an.

ioctl-Aufrufe

Die verfügbaren `ioctl`-Aufrufe und Datenstrukturen sind in `sys/cdrioc1.h` definiert.

DATEIEN

Die Gerätedateien haben folgende Namen:

<code>/dev/rdisk/cdrYtZsO</code>	Gerätedatei (Raw-Modus)
<code>/dev/dsk/cdrYtZsO</code>	Gerätedatei (Block-Modus)

Dabei steht `Y` für die Nummer des Controllers, `Z` für die SCSI-Targetnummer und `O` für eine Option (immer 0). Für die MX300 gilt: Eine 1 für `Y` bezeichnet den ersten ADP32 Controller (`adp32.0`), eine 2 den zweiten ADP32 Controller (`adp32.1`) usw. Bei der WX200 bezeichnet eine 1 für `Y` den ersten AHA1542B Controller (`stos.0`), eine 2 den zweiten AHA1542B Controller (`stos.1`) usw.

Die Gerätedateien (Geräteknotten, device nodes) werden erzeugt, wenn ein neuer Kern generiert wurde und das System danach erstmals hochgefahren wird (z.B. bei der Installation). Damit die Gerätedateien korrekt erzeugt werden können, müssen die jeweiligen Geräte angeschaltet und mit dem System verbunden sein.

SIEHE AUCH

`adp32(7)`

FEHLERMELDUNGEN

Der Treiber liefert Fehlermeldungen der Art:

`cdrYtZ: error text`

Dabei bezeichnet `Y` die Nummer des Controllers, `Z` die SCSI-Targetnummer und `error text` den Fehler.

BEZEICHNUNG

cram – CMOS RAM Schnittstelle

BESCHREIBUNG

Der `cram`-Treiber stellt eine Schnittstelle zu den 64 Bytes des batteriegepufferten RAM zur Verfügung. Dieser Speicher enthält Informationen wie Diagnosemeldungen und Konfigurationsdaten.

ioctl-Aufrufe**CMOSREAD**

Dieser Aufruf wird benutzt, um den Inhalt einer der CMOS RAM Positionen zu lesen. Das Argument für den `ioctl`-Aufruf ist die Adresse eines zwei Zeichen langen Puffers vom Typ `unsigned char`, wovon das erste Zeichen die zu lesende Adresse enthält. Im zweiten Byte trägt der `ioctl`-Aufruf die Daten ein. Eine Adresse kleiner 0 oder größer 63 führt zu einem Fehler, `errno` wird in diesem Fall auf `ENXIO` gesetzt.

CMOSWRITE

Dieser Aufruf wird benutzt, um einen Wert in eine der CMOS RAM Positionen einzutragen. Das Argument für den `ioctl`-Aufruf ist die Adresse eines zwei Byte langen Puffers vom Typ `unsigned char`; das erste Byte enthält die Adresse, das zweite Byte enthält den Wert, der an diese Adresse geschrieben werden soll. Eine Adresse kleiner 0 oder größer 63 führt zu einem Fehler, `errno` wird in diesem Fall auf `ENXIO` gesetzt. Es ist zu beachten, daß das CMOS RAM Gerät nur von einem privilegierten Benutzer für Schreibzugriffe geöffnet werden darf und daß dieser Aufruf für alle anderen Benutzer abgewiesen wird.

DATEIEN

`/dev/cram`

BEZEICHNUNG

ct – SCSI-Controller für Bandlaufwerke

BESCHREIBUNG

Dieser Gerätetreiber unterstützt SCSI-Adapter für Streaming-Bandlaufwerke. Pro SCED-Board wird ein Bandlaufwerk unterstützt.

Abhängig vom Namen der Gerätedatei gilt:

<code>/dev/rmt/c0s0n</code>	kein Rückspulen beim Schließen, keine Bandspannung beim Öffnen
<code>/dev/rmt/c0s0</code>	Rückspulen beim Schließen, keine Bandspannung beim Öffnen
<code>/dev/rmt/c0s0nr</code>	kein Rückspulen beim Schließen, Bandspannung beim Öffnen
<code>/dev/rmt/c0s0r</code>	Rückspulen beim Schließen, Bandspannung beim Öffnen

AUSWIRKUNGEN VON STREAMING

Da der Treiber ct Streaming-Laufwerke unterstützt, unterscheidet sich die Schnittstelle zum Treiber von der bei Standard-Bandlaufwerken üblichen:

Block-Schnittstelle

Die Block-Schnittstelle wird überhaupt nicht unterstützt. Versuche, die Block-Schnittstelle zu benutzen, liefern ENXIO.

Raw-Schnittstelle

Die Raw-Schnittstelle des Treibers unterscheidet sich in einigen Punkten von Standard-Bandlaufwerken:

Um angemessenen Durchsatz und Lebensdauer des Laufwerks zu gewährleisten, müssen die Daten vom und zum Band in großen Blöcken übertragen werden. Der Treiber erreicht dies durch interne Pufferung der Daten. Schreibfehler werden deshalb erst lange Zeit nach der Übertragung zum Treiber gemeldet. Einem Anwendungsprogramm werden Schreibfehler erst beim nächsten write-Systemaufruf oder bei einem folgenden close-Systemaufruf gemeldet.

Die Raw-Schnittstelle unterstützt Blockgrößen nicht wie bei normalen Bandlaufwerken. Das Bandlaufwerk ct verwendet eine physikalische Blockgröße von 512 Bytes. Datenblöcke werden beim Schreiben in diese physikalischen Blöcke gepackt und beim Lesen aus diesen Blöcken extrahiert. Beim Lesen erhält der Benutzer daher die gewünschte Anzahl Bytes, solange nicht das Dateiende erreicht wird. Falls kein Vielfaches von 512 Bytes geschrieben wurde, wird die Dateigröße auf ein Vielfaches von 512 Bytes aufgerundet. Die zusätzlich eingefügten Bytes enthalten zufällige Werte.

Folgende Technik funktioniert nicht wie erwartet: Teilweises Lesen einer Datei auf einem Gerät ohne Zurückspulen (no-rewind device), Schließen und erneutes Öffnen der Gerätedatei. Beim Schließen wird das Band immer zum Anfang der nächsten Datei weitertransportiert.

ioctl-Aufrufe

Die verfügbaren ioctl-Aufrufe und Datenstrukturen sind in `sys/io.ct.h` definiert.

INFORMATIONEN FÜR DEN BETRIEB

Bei einer Konfiguration mit `CTC_AUTORET` wird ein Band immer gestrafft, wenn es zum ersten Mal für einen `read`- oder `ioctl`-Systemaufruf geöffnet wird. Es wird gelöscht, wenn es zum ersten Mal zum Schreiben geöffnet wird. Wenn die erste Operation nach Einlegen des Bands (oder Hochfahren der Maschine) mit `reposition` ist, wird das Band daher zweimal gestrafft. Das automatische Straffen soll eine sichere Datenübertragung, vor allem beim Schreiben, gewährleisten.

Die Datenkapazität eines Bands nimmt mit der Anzahl von Software-Fehlern auf dem Band und der Anzahl der Repositionierungen zwischen Schreibvorgängen ab, da diese Platz auf dem Band verbrauchen. Ein Band mit 450 Fuß hat normalerweise eine Kapazität von etwa 44 MBytes.

Die Anzahl der Repositionierungen während der Lese- und Schreibvorgänge beeinflussen ebenfalls die Fehleranfälligkeit des Bandlaufwerks. Das Band wird gleichmäßiger gespult, wenn der Zugriff bei geringer Systemauslastung erfolgt.

KONFIGURATION

Das Feld `ct_bconf` in der binären Konfigurationsdatei des Gerätetreibers (`/etc/conf/patch.d/ct/space.c`) muß für jedes konfigurierte Laufwerk einen Eintrag enthalten. Standardmäßig enthält `ct_bconf` Einträge für vier Laufwerke. Überzählige Einträge werden ignoriert.

EINSTELLBARE PARAMETER

Für jedes Laufwerk definiert man die folgenden einstellbaren Parameter im Feld `ct_bconf` der binären Konfigurationsdatei:

- (1) Die Puffergröße (in Kilobytes) der Datenpuffer, die zur Ein-/Ausgabe benutzt werden. Zwei Puffer dieser Größe werden aus dem Free Page Pool des Kerns allokiert, wenn die Gerätedatei geöffnet wird. Diese Puffer werden benutzt, um den Durchsatz zu verbessern. Sie werden wieder freigegeben, wenn die Gerätedatei geschlossen wird. Die Puffergröße muß ein Vielfaches der Seitengröße (4 KB) sein.

- (2) Optionen für konfigurierbare Funktionen. Folgende Optionen sind möglich:

`CTC_OPENFAIL`

Mit dieser Option schlägt das Öffnen der Gerätedatei fehl, falls kein physikalischer Speicher für die Puffer vorhanden ist. Ohne die Option, wird der Speicher für die Puffer immer allokiert. Der Prozeß wird dann allerdings während der Allokierung blockiert und andere Prozesse werden ausgelagert, um Speicher freizugeben.

`CTC_AUTORET`

Falls diese Option gesetzt ist, wird ein Band beim ersten Zugriff nach dem Bandwechsel automatisch gestrafft. Falls die Optionen nicht gesetzt ist, wird nicht automatisch gestrafft.

`CTC_PRSENSE`

Falls diese Option gesetzt ist, werden alle Sense-Puffer von SCSI-Anforderungen zusammen mit dem Kommando, das die Bedingung auslöste, ausgegeben. Andernfalls werden nur Sense-Puffer ausgegeben, die irgendwie ungewöhnlich erscheinen.

- (3) Bits: Beim Target-Adapter ADSI und Emulex sowie beim Embedded Controller Tandberg TDS 3620/TDS 3660 sollte hier immer 1 stehen, um ein ordnungsgemäßes Arbeiten des Geräts zu gewährleisten.

FEHLERMELDUNGEN

ct: error writing buffer to tape
Ein Ein-/Ausgabefehler trat auf, während Daten auf das Band geschrieben wurden.

ct: ctclose: error writing file mark on close
Ein Ein-/Ausgabefehler trat auf, während eine Dateimarke auf das Band geschrieben wurde. Diese Meldung folgt häufig auf die vorige Meldung.

ct: illegal read after write
Der Prozeß versuchte nach einem erfolgreich abgeschlossenen Schreibvorgang Daten vom Band zu lesen, ohne vorher das Band zurückzuspulen oder zu straffen.

ct: ctioctl: bad operation after write
Der Prozeß versuchte nach einem erfolgreich abgeschlossenen Schreibvorgang, einen ungültigen ioctl-Systemaufruf abzusetzen. Nach einem Schreibvorgang sind nur Zurückspulen und Straffen zulässig.

ct: *error* on command *command*
Viele Fehlermeldungen haben diese Form.
Error ist eine der folgenden Meldungen:

Aborted command
Hardware error
Illegal request
Media error
Media is protected
Tape not ready
Unknown error
Vendor unique error
Volume overflow

Command ist eines der folgenden Kommandos:

erase
mode select
read
request sense
retension
rewind
space
test unit ready
write
write file marks

Bei einigen Fehlermeldungen werden zusätzlich der Inhalt des Kommandopuffers und des Sense-Puffers zum Zeitpunkt des Fehlers ausgegeben.

ct (7)

(MX500)

ct (7)

FEHLER

Eine doppelte Dateimarke am Ende eines Bandes wird nicht unterstützt.

DATEIEN

/dev/rmt/*

Raw-Dateien

/etc/conf/pack.d/ct/space.c

binäre Konfigurationsdatei

SIEHE AUCH

mt(1), sd(7)

BEZEICHNUNG

digi – Intelligente serielle E/A-Erweiterung DigiBoard

BESCHREIBUNG

Der DigiBoard-Treiber unterstützt bis zu vier installierte DigiBoards DigiPC/Xe. Die vier Anschlüsse pro Board können bzgl. Geschwindigkeit (50-38400 Baud), Zeichenlänge und Parity-Bit programmiert werden. Die Ausgabegeschwindigkeit entspricht jeweils der Eingabegeschwindigkeit. Der Anschluß verhält sich wie in `termio(7)` beschrieben.

Ein DigiBoard arbeitet mit Puffern sowohl bei der Eingabe wie auch bei der Ausgabe. Da ein DigiBoard mit einem eigenen Prozessor ausgerüstet ist, wird die CPU bei der Kontrolle des Anschlusses wesentlich entlastet. Dadurch erhöht sich die Übertragungsgeschwindigkeit. Außerdem verbessert sich die Datenübertragung mittels Datenübertragungsprogrammen, wie z.B. `uucp(1)` und `slip`, bis zu einer Geschwindigkeit von 19200 Baud.

Die Übertragungsgeschwindigkeiten des programmierbaren Baudraten-Generators beim seriellen Adapter stimmen nicht genau mit den Übertragungsgeschwindigkeiten des Systems überein. Speziell das Setzen von `B0` bewirkt einen Leitungsabbau. Das Setzen von `EXTA` bewirkt eine Baudrate von 19200 und das Setzen von `EXTB` eine Baud-Rate von 38400. Es ist nicht möglich, eine Baud-Rate von 2000, 3600 oder 7200 direkt zu setzen.

DATEIEN

`/dev/term/tty01*`

SIEHE AUCH

`signal(2)`, `termio(7)`.

BEZEICHNUNG

disk – Speichermedium für Direktzugriff

BESCHREIBUNG

Die sekundären Speichergeräte, die das System benutzt, sind Festplatten und Disketten. Festplatten sind rotierende Magnetspeicher mit hoher Umdrehungsgeschwindigkeit. Sie werden als eine Ansammlung konzentrischer Ringe angesehen, die man als Spuren bezeichnet. Es gibt verschiedene Scheiben auf der Festplatte (ihre Anzahl wird mit n angegeben). Dabei sind bis zu zwei Oberflächen pro Scheibe (oder eine Gesamtsumme von $2n$ Oberflächen) vorgesehen. Jede Sammlung von bis zu $2n$ parallelen Spuren auf diesen Oberflächen wird als *Zylinder* bezeichnet. Jede Spur ist in verschiedene *Sektoren* unterteilt. Ein Sektor ist gewöhnlich die kleinste Einheit, die von oder zu einer Festplatte übertragen werden kann. Die Treiber lassen allerdings *Lese-* oder *Schreib-* Operationen von beliebiger Größe und von einer beliebigen Position auf der Platte aus zu; ausgenommen sind Platten, die im Raw-Modus betrieben werden.

Logische Platten

Oft ist es nützlich, physikalische Festplatten in kleinere Bereiche zu unterteilen. Diese Aufteilung erfolgt in zwei Stufen:

Zunächst wird eine physikalische Platte in eine oder mehrere *logische Platten* oder *Partitionen* (`fdisk(1M)`) unterteilt. Eine dieser logischen Platten (und nur eine) wird als die *aktive UNIX Partition* gekennzeichnet (`fdisk(1M)`).

`fdisk` erlaubt nicht nur die anfängliche Aufteilung einer Platte in logische Laufwerke (Platten, Partitions), sondern informiert auch über die momentane Aufteilung, kann diese ändern und kann die aktive UNIX Partition auswählen. Jede Änderung der Aufteilung einer physikalischen Platte in logische Platten wird vom Plattentreiber aber erst dann aktiviert, wenn das erste `open` auf diese physikalische Platte erfolgt, nachdem alle Zugriffe auf die Platte beendet sind. Für eine Systemplatte heißt dies, daß eine Änderung der Plattenaufteilung erst dann wirksam wird, wenn das System das nächste Mal wieder hochgefahren wird.

Jeder weitere Zugriff auf eine physikalische Platte erfolgt über die aktive UNIX Partition.

Mit `disksetup(1M)` kann die aktive UNIX Partition wiederum in einzelne Slices aufgeteilt werden. Jede dieser Slices kann ein Dateisystem enthalten.

Zugriffe außerhalb der aktiven UNIX Partition können nur über die `ioctl`-Schnittstelle des Plattentreibers erfolgen, nicht aber über normale Lese-/Schreibzugriffe. Eine typische Unterteilung einer Platte in logische Partitionen könnte folgendermaßen aussehen:

```
/dev/dsk/c0d0s0
    aktive UNIX Partition der ersten Platte am ersten Controller

/dev/dsk/c0d0s1
    erste Slice der aktiven UNIX Partition der ersten Platte am ersten Controller

/dev/dsk/c0d1s0
    aktive UNIX Partition der zweiten Platte am ersten Controller
```

disk(7)

disk(7)

`/dev/dsk/c0d1s1`

erste Slice der aktiven UNIX Partition der zweiten Platte am ersten Controller

Für logische Platten, die im Raw-Modus betrieben werden, gilt entsprechend:

`/dev/rdisk/c0d0s0`, `/dev/rdisk/c0d1s0` etc.

Tatsächlich werden oft komplexere Unterteilungen vorgenommen. Denn oft ist es wünschenswert, logische Platten unterschiedlicher Größe entsprechend ihrem Verwendungszweck zu haben. Oder es ist wünschenswert, eine einzige Platte auf verschiedene Weise zu unterteilen. Anweisungen für die Unterteilung von Platten sind im *Leitfaden für Systemverwalter* beschrieben.

SIEHE AUCH

`addbad(1M)`, `fdisk(1M)`, `install(1M)`, `mkpart(1M)`, `fd(7)`, `hd(7)`, `intro(7)`.

BEZEICHNUNG

display – Bildschirmausgabe auf Systemkonsole

BESCHREIBUNG

Die Systemkonsole (und das Benutzerterminal) bestehen aus zwei getrennten Teilen: der Tastatur (siehe `keyboard(7)`) und dem Bildschirm. Aufgrund ihrer Komplexität und der Tatsache, daß es mehrere mögliche Bildschirmschnittstellen gibt (z.B. VGA, WD8514, PG1281, monochrome Adapter, Grafikadapter für Farbe und erweiterte Grafikadapter), werden sie in unterschiedlichen Handbuch-Einträgen beschrieben.

Der Bildschirm besteht normalerweise aus 25 Zeilen mit jeweils 80 Spalten; der Grafikadapter für Farbe unterstützt außerdem Zeilen mit 40 Spalten und der erweiterte Grafikadapter unterstützt 43 Zeilen mit je 80 Spalten. Die Wirkung von Zeichen, die an der Konsole oder an einem der virtuellen Bildschirme (`/dev/console` oder `/dev/vtxx`) eingegeben werden, ist abhängig von der Art der Zeichen. Alle Zeichen, die auf `/dev/console` eingegeben werden, werden zunächst von der Terminalschnittstelle (siehe `termio(7)`) bearbeitet. So werden z.B. Neue-Zeile-Zeichen auf Wagenrücklauf plus Neue-Zeile-Zeichen abgebildet und Tabulatorzeichen werden mit Leerzeichen expandiert, bevor die folgende Verarbeitung stattfindet:

<code>x</code>	Wenn <code>x</code> nicht eine der folgenden Angaben darstellt, wird <code>x</code> ausgegeben.
<code>BEL</code>	Erzeugt ein akustisches Signal (hörbarer Ton, keine Modulation).
<code>CR</code>	Setzt die Schreibmarke auf Spalte 1 der aktuellen Zeile.
<code>LF, VT</code>	Setzt die Schreibmarke auf die gleiche Spalte in der nächsten Zeile (blättert weiter, wenn die aktuelle Zeile die 25. Zeile ist).
<code>FF</code>	Löscht den Bildschirm und setzt die Schreibmarke auf Zeile 1 und Spalte 1.
<code>S</code>	Ist abhängig vom vorhergehenden Zeichen: falls es sich dabei um <code>_</code> (Unterstrich) handelt, siehe unten. Ansonsten wird die Schreibmarke innerhalb der gleichen Zeile um eine Spalte nach links bewegt, außer die Schreibmarke steht auf Spalte 1. Steht die Schreibmarke auf Spalte 1, nicht aber auf Zeile 1, so wird sie auf die Spalte 79 der vorhergehenden Zeile gesetzt. Steht die Schreibmarke auf Spalte 1 in Zeile 1, so wird sie nicht verändert.
<code>_Sx</code>	Setzt das Attribut <i>Unterstreich</i> für das auszugebende Zeichen <code>x</code> . Das Attribut <i>Unterstreich</i> für Grafikadapter für Farbe ist eine rote Hintergrundfarbe mit weißer Vordergrundfarbe.
<code>ESC x</code>	Wenn <code>x</code> einer der möglichen 256 Codes ist (außer <code>c</code> and <code>[]</code>), wird der Wert unverändert ausgegeben. Dies dient dazu, die für die Bildschirmausgabe zur Verfügung stehenden grafischen Möglichkeiten in vollem Umfang zu nutzen. Es ist zu beachten, daß die Zeichen vor der Escape-Folge von der Terminal-Schnittstelle bearbeitet werden. Daher ist es für manche der 256 Zeichen notwendig, daß sie nicht nachbearbeitet werden. Der einfachste Weg, um dies zu erreichen, ist das Ausschalten von <code>OPOST</code> im Feld <code>c_oflag</code> (siehe <code>termio(7)</code>); dies kann jedoch andere Nebenwirkungen haben.

Die Bildschirmausgabe kann über ANSI X3.64 *Escape-Sequenzen* gesteuert werden. Dies sind spezielle Zeichenketten, denen das ASCII-Zeichen ESC vorangestellt ist. Die folgenden Escape-Sequenzen gelten jeweils für den monochromen Adapter, den Grafikadapter für Farbe oder den erweiterten Grafikadapter:

- ESCc Löscht den Bildschirm und setzt die Schreibmarke auf Zeile 1 und Spalte 1.
- ESC Q *n* '*zeichenkette*' Definiert die Funktionstaste *n* mit *zeichenkette*. Der Begrenzer der Zeichenkette kann jedes Zeichen sein, das nicht in *zeichenkette* vorkommt. Die Funktionstasten sind von 0 bis 11 numeriert (F1 = 0, F2 = 1, usw.)
- ESC[*n* @ Zeichen einfügen—fügt *n* Leerzeichen an der aktuellen Schreibmarkenposition ein.
- ESC[*n* ` Absolute waagrechte Positionierung—positioniert auf die in *n* angegebene Spalte.
- ESC[*n* A Schreibmarke hochsetzen—bewegt die Schreibmarke *n* Zeilen nach oben (Vorgabewert: *n*=1).
- ESC[*n* a Relative waagrechte Positionierung—setzt die aktive Position um *n* Spalten nach rechts (Vorgabewert: *n*=1).
- ESC[*n* B Schreibmarke heruntersetzen—bewegt die Schreibmarke um *n* Zeilen abwärts (Vorgabewert: *n*=1).
- ESC[*n* C Schreibmarke nach rechts—bewegt die Schreibmarke um *n* Spalten nach rechts (Vorgabewert: *n*=1).
- ESC[*n* c *n* kann 0 (unterstrichene Schreibmarke), 1 (Blockschreibmarke), oder 2 (keine Schreibmarke) sein. 0 ist der Vorgabewert für *n*.
- ESC[*n* D Schreibmarke nach links—bewegt die Schreibmarke um *n* Spalten nach links (Vorgabewert: *n*=1).
- ESC[*n* d Absolute senkrechte Positionierung—setzt die aktive Position auf die in *n* angegebene Zeile.
- ESC[*n* E Schreibmarke nächste Zeile—setzt die Schreibmarke zunächst auf Spalte 1 der nächsten Zeile und danach um *n*-1 Zeilen weiter (Vorgabewert: *n*=1).
- ESC[*n* e Relative senkrechte Positionierung—setzt die aktive Position um *n* Zeilen abwärts (Vorgabewert: *n*=1).
- ESC[*n* F Schreibmarke vorhergehende Zeile—setzt die Schreibmarke auf Spalte 1 der aktuellen Zeile und danach um *n* Zeilen nach oben (Vorgabewert: *n*=1).
- ESC[*n* G Waagrechte Schreibmarkenpositionierung—setzt die Schreibmarke auf Spalte *n* der aktuellen Zeile (Vorgabewert: *n*=1).
- ESC[*n* ; *m* H Schreibmarkenpositionierung—setzt die Schreibmarke auf Spalte *m* in Zeile *n* (Vorgabewert: *n*=1, *m*=1).

- ESC[*n* ; *m* f Schreibmarkenpositionierung—setzt die Schreibmarke auf Spalte *m* in Zeile *n* (Vorgabewert: *n*=1, *m*=1).
- ESC[*n* J Fenster löschen—falls *n*=0, wird von der aktuellen Schreibmarkenposition bis zum Ende des Fensters gelöscht; falls *n*=1, wird vom Anfang des Fensters bis zur aktuellen Schreibmarkenposition gelöscht und falls *n*=2, wird das gesamte Fenster gelöscht (Vorgabewert: *n*=0).
- ESC[*n* K Zeile löschen—falls *n*=0, wird die Zeile von der aktuellen Schreibmarkenposition bis zum Zeilenende gelöscht; falls *n*=1, wird die Zeile vom Zeilenanfang bis zur aktuellen Schreibmarkenposition gelöscht; falls *n*=2, wird die gesamte Zeile gelöscht (Vorgabewert: *n*=0).
- ESC[*n* L Zeile einfügen—fügt *n* Zeilen an der aktuellen Schreibmarkenposition ein (Vorgabewert: *n*=1).
- ESC[*n* M Zeile löschen —löscht *n* Zeilen beginnend mit der aktuellen Schreibmarkenposition (Vorgabewert: *n*=1).
- ESC[*n* P Zeichen löschen—löscht *n* Zeichen einer Zeile beginnend mit der aktuellen Schreibmarkenposition (Vorgabewert: *n*=1).
- ESC[*n* S Verschieben nach oben—verschiebt die Zeichen im aktuellen Fenster um *n* Zeilen nach oben. Die unteren *n* Zeilen werden mit Leerzeichen überschrieben (Vorgabewert *n*=1).
- ESC[*n* T Verschieben nach unten—verschiebt die Zeichen im aktuellen Fenster um *n* Zeilen nach unten. Die oberen *n* Zeilen werden mit Leerzeichen überschrieben (Vorgabewert *n*=1).
- ESC[*n* X Zeichen löschen—löscht *n* Zeichen beginnend an der aktuellen Schreibmarkenposition (Vorgabewert *n*=1).
- ESC[*n* Z Schreibmarke auf Tabulatorposition zurücksetzen—setzt die aktuelle Position um *n* Tabulatorpositionen zurück.
- ESC[2h Sperrt die Tastatur und ignoriert Tastatureingaben, bis die Sperrung aufgehoben wird. Eingegebene Zeichen werden nicht gespeichert.
- ESC[2i Schickt den Bildschirminhalt an den Host. Der aktuelle Bildschirminhalt wird an die Anwendung geschickt.
- ESC[2l Entsperrt die Tastatur. Ermöglicht wieder Tastatureingabe.
- ESC[*Ps* ; *Ps* ; *m* Zeichenattribute—jedes *Ps* steht für eines der folgenden Zeichen; mehrere Zeichen werden durch Semikolon getrennt. Diese Parameter beziehen sich in additiver Weise auf die Ausgabe nachfolgender Zeichen, (z.B. kann sowohl Hervorhebung als auch Unterstreichung ausgewählt werden). Für den monochromen Adapter gelten nur die Parameter bis 7, für den Grafikadapter für Farbe und den erweiterten Grafikadapter sind alle Parameter möglich. (Vorgabewert: *Ps*=0).

<i>Ps</i>	Bedeutung		
0	alle Attribute ausgeschaltet (normale Bildschirmausgabe) (weißer Vordergrund auf schwarzem Hintergrund)		
1	intensive Hervorhebung		
4	Unterstreichnung eingeschaltet (weißer Vordergrund auf rotem Hintergrund bei farbigen Adaptern)		
5	blinkende Darstellung eingeschaltet		
7	inverse Darstellung eingeschaltet		
30	schwarz	(grau)	Vordergrund
31	rot	(hellrot)	Vordergrund
32	grün	(hellgrün)	Vordergrund
33	braun	(gelb)	Vordergrund
34	blau	(hellblau)	Vordergrund
35	magenta	(hellmagenta)	Vordergrund
36	zyanblau	(hellzyanblau)	Vordergrund
37	weiß	(leuchtendweiß)	Vordergrund
40	schwarz	(grau)	Hintergrund
41	rot	(hellrot)	Hintergrund
42	grün	(hellgrün)	Hintergrund
43	braun	(gelb)	Hintergrund
44	blau	(hellblau)	Hintergrund
45	magenta	(hellmagenta)	Hintergrund
46	zyanblau	(hellzyanblau)	Hintergrund
47	weiß	(leuchtendweiß)	Hintergrund
ESC[8 m	setzt Leerzeichen (keine Ausgabe)		
ESC[10 m	wählt den primären Zeichensatz aus		
ESC[11 m	wählt den ersten alternativen Zeichensatz aus; bewirkt, daß ASCII-Zeichen kleiner als 32 als ROM- Zeichen ausgegeben werden.		
ESC[12 m	wählt den zweiten alternativen Zeichensatz aus; schal- tet das höchstwertige Bit des erweiterten ASCII-Codes aus, bevor ROM-Zeichen ausgegeben werden.		
ESC[38 m	ermöglicht die Unterstreichungsoption; weißer Vor- dergrund mit weißen Unterstreichungen (siehe HINWEIS)		
ESC[39 m	setzt die Unterstreichungsoption außer Kraft (siehe HINWEIS)		

Es ist zu beachten, daß für die Attribute 30-37 die ausgewählte Vordergrundfarbe davon abhängt, ob das Attribut *intensive Hervorhebung* (1) eingeschaltet ist. Ist das Attribut nicht eingeschaltet, wird die zuerst aufgeführte Farbe verwendet; im anderen Fall wird die zweite aufgeführte Farbe verwendet.

Entsprechend hängt für die Attribute 40-47 die ausgewählte Hintergrundfarbe davon ab, ob das Attribut *blinkende Darstellung* (5) eingeschaltet ist. Ist das Attribut *blinkende Darstellung* nicht eingeschaltet, wird die erste aufgeführte Farbe verwendet. Ist das

Attribut *blink* eingeschaltet, wird die zweite aufgeführte Farbe verwendet.

ESC[*Pn* d CSI*Pn* d

Setzt die Schreibmarke auf die in *Pn* angegebene Zeile.

ioctl-Aufrufe

Die folgenden *ioctl*-Aufrufe können dazu benutzt werden, die Bildschirmausgabe für den Video-Monitor zu ändern.

Achtung: Alle folgenden *ioctl*-Aufrufe werden entweder für den Dateideskriptor eines virtuellen Terminals oder für die Gerätedatei `/dev/video` ausgeführt. *ioctl*-Aufrufe für `/dev/video` sind mit einem Stern (*) gekennzeichnet. Damit *ioctl*-Aufrufe für `/dev/video` ausgeführt werden können, muß das kontrollierende *tty* für den Prozeß das virtuelle Terminal sein, für das die Operation durchgeführt werden soll. Falls *tty* kein virtuelles Terminal ist, wird der Rückgabewert auf -1 und *errno* auf `EINVAL` gesetzt.

SWAPMONO (*)

Dieser Aufruf wählt den monochromen Adapter als Ausgabegerät für die Systemkonsole aus.

SWAPCGA (*)

Dieser Aufruf wählt den Grafik-Adapter für Farbe als Ausgabegerät für die Systemkonsole aus.

SWAPEGA

Dieser Aufruf wählt den erweiterten Grafik-Adapter als Ausgabegerät für die Systemkonsole aus.

Der folgende *ioctl*-Aufruf dient dazu, mehr Information über den Bildschirm-Adapter zu erhalten, der gerade an den Video-Monitor angeschlossen ist.

CONS_CURRENT (*)

Dieser Aufruf meldet den Typ des Bildschirm-Adapters zurück, der gerade an den Video-Monitor angeschlossen ist. Der Rückgabewert kann einer der folgenden sein: MONO, CGA oder EGA.

Die folgenden *ioctl*-Aufrufe können dazu verwendet werden, um die Darstellungsmodi auf den verschiedenen Video-Adaptoren umzuschalten:

SW_B40x25 (*)

Dieser Aufruf wählt den 40x25 (40 Spalten bei 25 Zeilen) schwarz-weißen Darstellungsmodus für Text aus. Er ist nur für CGA und EGA Geräte zulässig.

SW_C40x25 (*)

Dieser Aufruf wählt den 40x25 (40 Spalten bei 25 Zeilen) farbigen Darstellungsmodus für Text aus.

SW_B80x25 (*)

Dieser Aufruf wählt den 80x25 (80 Spalten auf 25 Zeilen) schwarz-weißen Darstellungsmodus für Text aus. Er ist nur für CGA und EGA Geräte zulässig.

SW_C80x25 (*)

Dieser Aufruf wählt den 80x25 (80 Spalten auf 25 Zeilen) farbigen Darstellungsmodus für Text aus. Er ist nur für CGA und EGA Geräte zulässig.

SW_BG320 (*)

Dieser Aufruf wählt den 320x200 schwarz-weißen grafischen Darstellungsmodus aus. Er ist nur für CGA und EGA Geräte zulässig.

SW_CG320 (*)

Dieser Aufruf wählt den 320x200 farbigen grafischen Darstellungsmodus aus. Er ist nur für CGA und EGA Geräte zulässig.

SW_BG640 (*)

Dieser Aufruf wählt den 640x200 schwarz-weißen grafischen Darstellungsmodus aus. Er ist nur für CGA und EGA Geräte zulässig.

SW_CG320_D (*)

Dieser Aufruf wählt EGA Unterstützung für den 320x200 grafischen Darstellungsmodus aus (EGA Modus D). Er ist nur für EGA Geräte zulässig.

SW_CG640_E (*)

Dieser Aufruf wählt EGA Unterstützung für den 620x200 grafischen Darstellungsmodus aus (EGA Modus E). Er ist nur für EGA Geräte zulässig.

SW_EGAMONOAPA (*)

Dieser Aufruf wählt EGA Unterstützung für den 640x350 grafischen Darstellungsmodus aus (EGA Modus F). Er ist nur für EGA Geräte zulässig.

SW_ENHMONOAPA2 (*)

Dieser Aufruf wählt EGA Unterstützung für den 640x350 grafischen Darstellungsmodus mit erweitertem Speicherplatz aus (EGA Modus F*). Er ist nur für EGA Geräte zulässig.

SW_CG640x350 (*)

Dieser Aufruf wählt EGA Unterstützung für den 640x350 grafischen Darstellungsmodus aus (EGA Modus 10). Er ist nur für EGA Geräte zulässig.

SW_ENH_CG640 (*)

Dieser Aufruf wählt EGA Unterstützung für den 640x350 grafischen Darstellungsmodus mit erweitertem Speicherplatz aus (EGA Modus 10*). Er ist nur für EGA Geräte zulässig.

SW_EGAMONO80x25 (*)

Dieser Aufruf wählt den EGA monochromen Darstellungsmodus für Text aus (EGA Modus 7), der Unterstützung durch den monochromen Adapter emuliert. Er ist nur für EGA Geräte zulässig.

SW_ENHB40x25 (*)

Dieser Aufruf wählt den erweiterten 40x25 schwarz-weißen Darstellungsmodus für Text aus. Er ist nur für EGA Geräte zulässig.

SW_ENHC40x25 (*)

Dieser Aufruf wählt den erweiterten 40x25 farbigen Darstellungsmodus für Text aus. Er ist nur für EGA Geräte zulässig.

SW_ENHB80x25 (*)

Dieser Aufruf wählt den erweiterten 80x25 schwarz-weißen Darstellungsmodus aus. Er ist nur für EGA Geräte zulässig.

SW_ENHC80x25 (*)

Dieser Aufruf wählt den erweiterten 80x25 farbigen Darstellungsmodus für Text aus. Er ist nur für EGA Geräte zulässig.

SW_ENHB80x43 (*)

Dieser Aufruf wählt den erweiterten 80x43 schwarz-weißen Darstellungsmodus für Text aus. Er ist nur für EGA Geräte zulässig.

SW_ENHC80x43 (*)

Dieser Aufruf wählt den erweiterten 80x43 farbigen Darstellungsmodus für Text aus. Er ist nur für EGA Geräte zulässig.

SW_MCAMODE (*)

Dieser Aufruf reinitialisiert den monochromen Adapter. Er ist nur für den monochromen Adapter zulässig.

SW_ATT_640 (*)

Dieser Aufruf wählt den 640x400 16 Farbmodus aus, wenn eine AT&T Super-Vu Video-Steuerung angeschlossen ist.

Das Umschalten in einen unzulässigen Darstellungsmodus für ein Bildschirmgerät führt zu einem Fehler.

Die folgenden `ioctl`-Aufrufe können dazu verwendet werden, Informationen über den aktuellen Darstellungsmodus zu erhalten.

CONS_GET (*)

Dieser Aufruf meldet die aktuelle Einstellung des Darstellungsmodus für den jeweils verwendeten Bildschirm-Adapter zurück. Folgende Rückgabewerte sind möglich:

M_B40x25 (0), schwarz-weiß 40 Spalten. Nur CGA und EGA.

M_C40x25 (1), farbig 40 Spalten. Nur CGA und EGA.

M_B80x25 (2), schwarz-weiß 80 Spalten. Nur CGA und EGA.

M_C80x25 (3), farbig 80 Spalten. Nur CGA und EGA.

M_BG320 (4), schwarz-weiß Grafik 320x200. Nur CGA und EGA.

M_CG320 (5), Farbgrafik 320x200. Nur CGA und EGA.

M_BG640 (6), schwarz-weiß Grafik 640x200 mit hoher Auflösung. Nur CGA und EGA.

M_EGAMONO80x25 (7), EGA mono 80x25. Nur EGA.

M_CG320_D (13), EGA Modus D.

M_CG640_E (14), EGA Modus E.

M_EFAMONOAPA (15), EGA Modus F.

M_CG640x350 (16), EGA Modus 10.

M_ENHMONOAPA2 (17), EGA Modus F mit erweitertem Speicherplatz.

M_ENH_CG640 (18), EGA Modus 10*.

M_ENH_B40x25 (19), EGA erweitert schwarz-weiß 40 Spalten.

M_ENH_C40x25 (20), EGA erweitert farbig 40 Spalten.

M_ENH_B80x25 (21), EGA erweitert schwarz-weiß 80 Spalten.

M_ENH_C80x25 (22), EGA erweitert farbig 80 Spalten.

M_ENH_B80x43 (0x70), EGA schwarz-weiß 80x43.

M_ENH_C80x43 (0x71), EGA farbig 80x43.

M_MCA_MODE (0xff), monochromer Adapter-Modus.

MCA_GET (*)

Dieser Aufruf meldet die aktuelle Einstellung des Darstellungsmodus für den monochromen Adapter zurück. Siehe `CONS_GET` für eine Liste der Rückgabewerte. Falls der monochrome Adapter nicht angeschlossen ist, wird der Aufruf abgewiesen und `errno` auf 22 gesetzt (EINVAL).

CGA_GET (*)

Dieser Aufruf meldet die aktuelle Einstellung des Darstellungsmodus für den Grafik-Adapter für Farbe zurück. Siehe `CONS_GET` für eine Liste der Rückgabewerte. Falls der Grafik-Adapter für Farbe nicht angeschlossen ist, wird der Aufruf abgewiesen und `errno` auf 22 gesetzt (EINVAL).

EGA_GET (*)

Dieser Aufruf meldet die aktuelle Einstellung des Darstellungsmodus für den erweiterten Grafik-Adapter zurück. Siehe `CONS_GET` für eine Liste der Rückgabewerte. Falls der erweiterte Grafik-Adapter nicht angeschlossen ist, wird der Aufruf abgewiesen und `errno` auf 22 gesetzt (EINVAL).

Die folgenden `ioctl`-Aufrufe können dazu benutzt werden, den Speicher des Video-Adapters auf den Benutzerdatenraum abzubilden.

MAPCONS (*)

Dieser Aufruf bildet den Bildschirmspeicher des derzeit verwendeten Adapters auf den Benutzerdatenraum ab.

MAPMONO (*)

Dieser Aufruf bildet den Bildschirmspeicher des monochromen Adapters auf den Benutzerdatenraum ab.

MAPCGA (*)

Dieser Aufruf bildet den Bildschirmspeicher des farbigen/grafischen Adapters auf den Benutzerdatenraum ab.

MAPEGA (*)

Dieser Aufruf bildet den Bildschirmspeicher des erweiterten grafischen Adapters auf den Benutzerdatenraum ab.

Die `ioctl` Aufrufe können dazu benutzt werden, ein Byte über den Anschluß des Grafik-Adapters einzulesen oder ein Byte an den Anschluß des Grafik-Adapters auszugeben. Das Argument für `ioctl` verwendet die `port_io_arg` Datenstruktur:

```
struct port_io_arg {
    struct port_io_struct_args[4];
};
```

Wie aus dem vorangegangenen Beispiel ersichtlich ist, zeigt die `port_io_arg` Struktur auf einen Bereich mit vier `port_io_struct` Datenstrukturen. Die `port_io_struct` Datenstruktur hat folgendes Format:

```
struct port_io_struct {
    char    dir;           /*direction flag (in vs. out)*/
    unsigned short port;  /*port address*/
    char    data;         /*byte of data*/
};
```

In dem Bereich für einen `ioctl`-Aufruf können ein, zwei, drei oder vier der `port_io_struct` Strukturen angegeben werden. Der Wert von `dir` kann entweder `IN_ON_PORT` sein (damit wird ein Byte spezifiziert, das über den Anschluß des grafischen Adapters eingegeben wird) oder `OUT_ON_PORT` (damit wird ein Byte spezifiziert, das über den Anschluß des grafischen Adapters ausgegeben wird). Der Wert für `port` ist eine Ganzzahl, die die Anschlußadresse des gewünschten Grafik-Adapters angibt. In `data` steht das ein- oder auszugebende Datenbyte. Falls keine der `port_io_struct` Strukturen verwendet werden, wird in `port` 0 eingetragen und die nicht verwendeten Strukturen werden am Ende des Bereichs gelassen. Bitte schlagen Sie in Ihrem Hardware-Manual die Adressen und Funktionen für die verschiedenen Adapter nach.

Die folgenden `ioctl`-Aufrufe können zur Ein- und Ausgabe von Bytes am Anschluß des Grafik-Adapters verwendet werden:

MCAIO Ein- oder Ausgabe eines Bytes am angegebenen monochromen Adapteranschluß ein oder aus.

CGAIO Ein- oder Ausgabe eines Bytes am angegebenen Anschluß des Grafik-Adapters ein oder aus.

EGAIO Ein- oder Ausgabe eines Bytes am angegebenen Anschluß des erweiterten Grafik-Adapters.

Um ein Byte auf einem der grafischen Adapteranschlüsse einzugeben, müssen Sie `dir` mit `IN_ON_PORT` und `port` mit der Anschlußadresse des grafischen Adapters versorgen. Das vom grafischen Adapteranschluß eingegebene Byte wird in `data` zurückgemeldet.

Um ein Byte auszugeben, müssen Sie `dir` mit `OUT_ON_PORT`, `port` mit der Anschlußadresse des Grafik-Adapters und `data` mit dem Byte versorgen, das Sie am grafischen Adapteranschluß ausgeben wollen.

Die folgenden `ioctl`-Aufrufe können sowohl für monochrome Adapter als auch für Grafik-Adapter für Farbe oder erweiterte Grafik-Adapter verwendet werden:

KDDISPTYPE (*)

Dieser Aufruf gibt dem Benutzer Bildschirminformation zurück. Das erwartete Argument ist die Pufferadresse einer Struktur vom Typ `kd_disparam`, in der die Bildschirminformation an den Benutzer zurückgegeben wird. Die `kd_disparam` Struktur ist folgendermaßen definiert:

```
struct kd_disparam {
    long type;                /*display type*/
    char *addr;               /*display memory address*/
    ushort ioaddr[MKDIOADDR]; /*valid I/O addresses*/
}
```

Folgende Werte für das Typfeld sind möglich:

`KD_MONO` (0x01), für den IBM monochromen Bildschirm-Adapter.

`KD_HERCULES` (0x02), für den Hercules monochromen Grafik-Adapter.

`KD_CGA` (0x03), für den IBM Farbgrafik-Adapter.

`KD_EGA` (0x04), für den IBM erweiterten Grafik-Adapter.

KIOCSOUND

Tonerzeugung starten. Ton anschalten. In "arg" steht die gewünschte Frequenz. Die Frequenz 0 schaltet den Ton aus.

KDGETLED

LED Status der Tastatur holen. Das Argument ist ein Zeiger auf eine Variable vom Typ `char`. Die Variable wird mit einer booleschen Verknüpfung versorgt, die folgende Werte annehmen kann:

<code>LED_SCR</code>	0x01	(Markierungsbit für Verschiebesperrung)
<code>LED_CAP</code>	0x04	(Markierungsbit für caps Sperrung)
<code>LED_NUM</code>	0x02	(Markierungsbit für num Sperrung)

KDSETLED

LED Status der Tastatur setzen. Das Argument ist ein Zeichen, dessen Wert die boolesche Verknüpfung der unter `KDGETLED` aufgeführten Werte darstellt.

KDMKTONE

Ton fester Länge erzeugen. Das Argument ist ein 32-Bit-Wert, bei dem die 16 niederwertigen Bits die Frequenz und die 16 höherwertigen Bits die Dauer (in Millisekunden) angeben.

KDGBKTYPE

Tastaturtyp holen. Das Argument ist ein Zeiger auf eine Variable vom Typ `char`. Die Variable wird mit einem der folgenden Werte zurückgegeben:

<code>KB_84</code>	0x01	(Tastatur mit 84 Tasten)
<code>KB_101</code>	0x02	(Tastatur mit 101 Tasten)
<code>KB_OTHER</code>	0x03	

KDADDIO (*)

I/O Anschlußadresse zur Liste der gültigen Video-Adapter-Adressen hinzufügen. Das Argument ist ein Zeiger auf eine Variable vom Typ `unsigned short`, die eine gültige Anschlußadresse für den angeschlossenen Video-Adapter enthalten sollte.

KDDELIO (*)

I/O Anschlußadresse aus der Liste der gültigen Video-Adapter-Adressen löschen. Das Argument ist ein Zeiger auf eine Variable vom Typ `unsigned short`, die eine gültige Anschlußadresse für den angeschlossenen Video-Adapter enthalten sollte.

KDENABIO (*)

Ein- und Ausgaben auf dem Video-Adapter-Anschluß einschalten. Ohne Argument.

KDDISABIO (*)

Ein- und Ausgaben auf dem Video-Adapter-Anschluß ausschalten. Ohne Argument.

KDQUEMODE (*)

Speziellen Warteschlangenmodus einschalten/ausschalten. Der Warteschlangenmodus wird von der X-Windows-Software verwendet, um eine mehrbenutzbare Warteschlange für Tastaturzugriffe und Maus-Ereignisse einzurichten. Das Argument ist ein Zeiger auf die Struktur `kd_quemode`. Falls ein `NULL`-Zeiger als Argument angegeben wurde, wird die Warteschlange geschlossen und der Modus ausgeschaltet. Die Struktur `kd_quemode` sieht folgendermaßen aus:

```
struct kd_quemode {
    int    qsize; /* gewünschte # des Elements in der
                  Warteschlange*/
    int    signo; /* zu sendende Signal-Nummer, wenn die
                  Warteschlange nicht leer ist*/
    char   *qaddr; /* virtuelle Benutzeradresse der Warte-
                   schlange (vom Treiber gesetzt)*/
};
```

KDSBORDER (*)

Farbige Bildschirmbegrenzung im EGA Textmodus setzen. Das Argument ist vom Typ `char`. Jede Bitposition entspricht einer Farbauswahl. Von Bitposition 0 bis zu Bitposition 6 sind die entsprechenden Farbwerte angegeben, Blau, Grün, Rot, Sekundärblau, Sekundärgrün und Sekundärrot. Setzt man an der entsprechenden Bitposition eine logische 1, wird die gewünschte Farbe bzw. werden die gewünschten Farben ausgewählt. Siehe auch die unten angeführten Bemerkungen unter HINWEIS.

KDSETMODE (*)

Konsole in den Text- oder Grafikmodus setzen. Das Argument ist ganzzahlig und sollte einen der folgenden Werte enthalten.

```

KD_TEXT      0x00    ( setzt den Konsolmodus auf Textmodus )
KD_GRAPHICS  0x01    ( setzt den Konsolmodus auf Grafikmodus )

```

Es ist zu beachten, daß es in der Verantwortung des Benutzers liegt, die Register für den Farbgrafik-Adapter entsprechend den grafischen Möglichkeiten zu versorgen.

KDGETMODE

Aktuellen Konsolmodus abfragen. Meldet ein ganzzahliges Argument zurück, das entweder KD_TEXT oder KD_GRAPHICS enthält, je nachdem, was in der KDSETMODE ioctl Beschreibung angegeben wurde.

KDMAPDISP (*)

Bildet den Bildschirmspeicher auf den Prozeßadreßraum des Benutzers ab. Das Argument ist ein Zeiger auf eine Struktur vom Type kd_memloc. Die Strukturdefinition sieht folgendermaßen aus:

```

struct kd_memloc {
    char    *vaddr;      /* virtuelle Empfangsadresse */
    char    *physaddr;   /* physikalische Sendeadresse */
    long    length;      /* Anzahl der zu übertragenden Bytes */
    long    ioflg;       /* i/o Adressen verfügbar machen,
                        falls gesetzt */
}

```

KDUNMAPDISP (*)

Bildschirmspeicher aus dem Prozeßadreßraum des Benutzers zurückübertragen. Kein Argument nötig.

VT_OPENQRY

Ein verfügbares virtuelles Terminal finden. Das Argument ist ein Zeiger auf eine Variable vom Datentyp long. Diese wird mit der Nummer des ersten verfügbaren "VT" versorgt, das von keinem anderen Prozeß geöffnet wurde. Ist keines vorhanden, wird diese Variable auf -1 gesetzt.

VT_GETMODE (*)

Feststellen, ob sich das aktive virtuelle Terminal derzeit im Modus VT_AUTO oder VT_PROCESS befindet. Das Argument für den ioctl-Aufruf ist die Adresse einer Struktur folgenden Typs:

```

struct vt_mode {
    char    mode;        /* VT Modus */
    char    waitv;       /* wenn gesetzt, Schreib-Ops anhalten,
                        bis aktiv */
    short   relsig;       /* Signal für Freigabeaufforderung */
    short   acqsig;       /* Signal für zugeordneten Bildschirm */
    short   frsig;        /* ist 0, wird nicht verwendet */
}

#define VT_AUTO      0x00    /* automatisches VT Umschalten */
#define VT_PROCESS   0x01    /* Umschalten der
                        Steuerprozesse */

```

In der `vt_mode` Struktur wird jedes Feld mit dem jeweilig aktuellen Wert versorgt.

VT_GETSTATE (*)

Der `VT_GETSTATE` ioctl-Aufruf meldet globale Statusinformationen über virtuelle Terminals zurück. Er meldet im `v_active` Feld das aktive virtuelle Terminal und im `vt_state` Feld die Nummer des aktiven virtuellen Terminals sowie eine Bitmaske mit dem globalen Status zurück. Dabei ist Bit `x` der Status von `vt x` (1 gibt an, daß das virtuelle Terminal geöffnet ist).

VT_SETMODE (*)

Modus für virtuelles Terminal setzen. Das Argument zeigt auf eine `vt_mode` Struktur entsprechend der oben definierten.

VT_SENDSIG (*)

Der `VT_SENDSIG` ioctl-Aufruf spezifiziert ein Signal (in `vt_signal`), das an eine Bitmaske für virtuelle Terminals (in `vt_state`) gesendet wird.

Die von den `VT_GETSTATE` und `VT_SENDSIG` ioctl-Aufrufen verwendete Datenstruktur ist:

```
struct vt_stat {
    ushort v_active; /* active vt */
    ushort v_signal; /* signal to send (VT_SENDSIG)
    */
    ushort v_state; /* vt bit mask (VT_SENDSIG and
                    VT_GETSTATE) */
};
```

und ist in `/usr/include/sys/vt.h` definiert.

VT_RELDISP (*)

Wird verwendet, um der virtuellen Terminalverwaltung mitzuteilen, ob der Prozeß den Bildschirm freigegeben hat oder nicht. Ein Argument ungleich Null zeigt an, daß der Bildschirm freigegeben wurde; ein Argument Null zeigt an, daß der Bildschirm nicht freigegeben ist.

VT_ACTIVATE

Macht das virtuelle Terminal mit der im Aufruf angegebenen Nummer zum aktiven "VT". Die "VT"-Verwaltung setzt einen Schalter, ganz so, als ob eine Unterbrechungsfolge an der Tastatur eingegeben worden wäre. Falls das angegebene "VT" nicht geöffnet ist oder nicht existiert, wird der Aufruf zurückgewiesen und `errno` auf `ENXIO` gesetzt.

KIOCFINFO

Dieser Aufruf teilt dem Benutzer mit, um welches Gerät es sich handelt.

GIO_SCRNMAP (*)

Dieser Aufruf fordert die Bildschirm-Umsetzungstabelle vom Systemkern an.

GIO_ATTR

Dieser Aufruf meldet die aktuellen Bildschirmattribute zurück. Die Bits werden folgendermaßen interpretiert:

Bit 0 legt Unterstreichung für schwarz-weiße Monitore fest (1= Unterstreichung eingeschaltet).

Bits 0-2 gelten nur für Farbmonitore und wählen die Vordergrundfarbe aus. Die folgende Liste zeigt an, welche Farben aufgrund der angegebenen Werte ausgewählt werden.

- Der Wert 0 bedeutet schwarz.
- Der Wert 1 bedeutet rot.
- Der Wert 2 bedeutet grün.
- Der Wert 4 bedeutet blau.
- Der Wert 5 bedeutet magenta.
- Der Wert 6 bedeutet cyanblau.
- Der Wert 7 bedeutet weiß.

Bit 3 ist das Bit für die Helligkeit (1=blinkende Darstellung eingeschaltet).

Bits 4-6 gelten nur für Farbmonitore und wählen die Hintergrundfarbe aus. Die Liste der Farben und der dazugehörigen Werte entspricht der Liste für Vordergrundfarben.

Bit 7 gibt blinkende Darstellung an (1=blinkende Darstellung eingeschaltet).

GIO_COLOR (*)

Dieser Aufruf meldet einen Wert ungleich Null zurück, wenn das aktuelle Bildschirmgerät ein Farbmonitor ist; andernfalls wird der Wert Null zurückgemeldet.

PIO_SCRNMAP

Dieser Aufruf übergibt die Bildschirm-Umsetzungstabelle an den Systemkern.

Die Bildschirm-Umsetzungstabelle bildet erweiterte ASCII-Zeichen (8-Bit) auf ROM-Zeichen ab. Sie ist ein 256 Byte langes Feld (array [256]) vom Typ char (typedef scrnmap_t); der Zugriff erfolgt über Indexwerte im erweiterten ASCII-Code. Die Werte der einzelnen Tabellenelemente sind die auszugebenden ROM-Zeichen.

BEISPIEL

Das folgende Beispiel zeigt, wie das ASCII-Zeichen '#' in das auszugebende englische Pfund-Zeichen geändert wird.

```
#include <sys/console.h>
change_pound() {
    scrnmap_t scrntab;
    /* Bildschirm-Umsetztabelle holen */
    if (ioctl(0, GIO_SCRNMAP, scrntab) == -1)
    {
        perror("screenmap read");
        exit(-1);
    }
    /* 156 ist der ROM Wert des englischen Pfundzeichens;
       der ASCII Wert von '#' ist 30.
    */
    scrntab[30] = 156;
    if (ioctl(0, PIO_SCRNMAP, scrntab) == -1)
    {
        perror("screenmap write");
        exit(-1);
    }
}
```

DATEIEN

```
/dev/console
/dev/vt00-n
/dev/video
/usr/include/sys/kd.h
```

SIEHE AUCH

stty(1), console(7), keyboard(7), termio(7).
ioctl(2) im *Referenzhandbuch für Programmierer*.

HINWEIS

Obwohl es möglich wäre, Zeichenfolgen anzugeben, die wahlweise Bits für die Bildschirmbedienung in jedem der drei grafischen Modi setzen, wird dieses Verfahren derzeit nicht unterstützt.

Das Ein- und Ausschalten der Unterstreichungs-Option mit "ESC [38 m" und "4ESC [39 m" funktioniert nur, wenn der Grafik-Adapter für Farbe des AT&T Rite-Vu Bildschirms installiert ist. Andernfalls wird die Option nicht unterstützt.

Es ist derzeit nicht möglich, auf das Register mit der 6845 Start-Adresse zuzugreifen. Damit ist es unmöglich, den Anfang des Bildschirmspeichers für den Farbmonitor festzulegen.

Das Bit im Farbauswahlregister, das eine alternierende Hintergrundfarbe angibt (Bit 4), verändert die Hintergrundfarbe im alphanumerischen Modus nicht.

`KDSBORDER ioctl` Aufrufe funktionieren nicht für den erweiterten Grafik-Adapter für Farbe am AT&T Super-Vu Bildschirm. Dagegen können die IBM EGA Karte und andere EGA-kompatible Bildschirm-Adapter mit diesen Aufrufen arbeiten.

Im Grafik-Modus für niedrige Zeichenauflösung werden 100 Zeilen zu 80 Spalten dargestellt.

BEZEICHNUNG

exa – SCSI-Band auf dem NCR Host Adapter (ADP32 Controller)

BESCHREIBUNG

Der Treiber exa unterstützt das 8 mm Video-Bandlaufwerk EXABYTE EXB-8200 am SCSI-Bus.

Struktur der Gerätenummer (minor device number)

Einheitennummer (unit number)

Bits 0-2:

SCSI-Targetnummer des Geräts.

Bit 3: immer 1.

Bit 10-13:

Nummer des Host Adapters.

Bedeutung der weiteren Bits:

Bit 4: Falls die letzte Operation ein Schreibvorgang war, wird beim Schließen eine Dateimärke geschrieben. Falls die letzte Operation ein Lesevorgang war, wird "skip 1/skip on filemark" an das Gerät gesandt.

Bit 5: Das Band wird beim Schließen zurückgespult.

Bit 6: Das Band wird beim Schließen entladen (ausgeworfen).

Bit 7: Beim Öffnen wird der Raw-Modus ausgewählt. Der Raw-Modus ist satzorientiert (record-oriented). Datensätze müssen mit der gleichen Puffergröße gelesen werden, mit der sie geschrieben wurden. Die Abarbeitung eines Auftrags im Raw-Modus dauert länger, da es schwierig ist, das Band im "streaming mode" zu halten.

ioctl-Aufrufe

Verschiedene ioctl-Aufrufe werden unterstützt. Die für Magnetbänder üblichen Aufrufe sind in `sys/mtio.h` und `sys/tape.h` beschrieben. Spezielle ioctl-Aufrufe und Datenstrukturen sind in `sys/exaioctl.h` definiert.

DATEIEN

Die Gerätedateien haben folgende Namen:

```
/dev/rmt/exaYtZ,  
/dev/rmt/exaYtZn,  
/dev/rmt/exaYtZv,  
/dev/rmt/exaYtZvn
```

Dabei bezeichnet Y die Nummer des Controllers und Z die SCSI-Targetnummer. Eine 1 für Y bezeichnet den ersten ADP32 Controller (`adp32.0`), eine 2 den zweiten ADP32 Controller (`adp32.1`) usw.

Die Gerätedateien (Geräteknoten, device nodes) für Controller 1 und Target 0 werden bei der Installation angelegt. Falls Geräte an einem anderen Controller oder an einer anderen Target-Adresse betrieben werden, müssen die entsprechenden Gerätedateien mit `/etc/init.d/scsigen exa` erzeugt werden. Dabei müssen die Geräte angeschaltet mit dem System verbunden sein und mit einer Bandkassette geladen sein.

SIEHE AUCH

adp32(7), tape.h(4), mtio.h(4), exaioc1.h(4).

FEHLERMELDUNGEN

Der Treiber gibt folgende Fehlermeldungen aus:

exaYtZ: no cartridge loaded.

exaYtZ: data cartridge write protected.

exaYtZ: new cartridge!.

exaYtZ: No write after LEOT allowed.

exaYtZ: No WriteFilemark after LEOT allowed.

exaYtZ: No Erase after LEOT allowed.

exaYtZ: No WriteVar after LEOT allowed.

exaYtZ: LEOT (logical end of tape) encountered.

exaYtZ: PEOT (physical end of tape) reached.

exaYtZ: VOLUME OVERFLOW.

exaYtZ: bad cartridge with unrecoverable medium error.

exaYtZ: illegal tape position.

Dabei bezeichnet Y die Nummer des Controllers und Z die SCSI-Targetnummer.

HINWEIS

Findet ein Reset auf dem SCSI-Bus statt, an dem das Exabyte-Gerät betrieben wird, dann führt das Exabyte-Gerät ein Rewind durch und positioniert das Band auf den Bandanfang (BOT). Die bisherige Position des Bandes geht verloren. Weitere Aufträge werden am Bandanfang ausgeführt. Bereits geschriebene Daten können dadurch überschrieben werden. Ein Sicherungslauf, der gerade stattgefunden hat, muß z.B. wiederholt werden.

BEZEICHNUNG

exa – SCSI-Band auf dem Host Adapter AHA1542B

BESCHREIBUNG

Der Treiber `exa` unterstützt die 8 mm Video-Bandlaufwerke EXABYTE EXB-8200 und Tandberg TDC 3660 am SCSI-Bus.

Struktur der Gerätenummer (minor device number)

Einheitennummer (unit number)

Bits 0-2:

SCSI-Targetnummer des Geräts.

Bit 3: immer 1.

Bit 10-13:

Nummer des Host Adapters.

Bedeutung der weiteren Bits:

Bit 4: Falls die letzte Operation ein Schreibvorgang war, wird beim Schließen eine Dateimarke geschrieben. Falls die letzte Operation ein Lesevorgang war, wird "skip 1/skip on filemark" an das Gerät gesandt.

Bit 5: Das Band wird beim Schließen zurückgespult.

Bit 6: Das Band wird beim Schließen entladen (ausgeworfen).

Bit 7: Beim Öffnen wird der Raw-Modus ausgewählt. Der Raw-Modus ist satzorientiert (record-oriented). Datensätze müssen mit der gleichen Puffergröße gelesen werden, mit der sie geschrieben wurden. Die Abarbeitung eines Auftrags im Raw-Modus dauert länger, da es schwierig ist, das Band im "streaming mode" zu halten.

ioctl-Aufrufe

Verschiedene `ioctl`-Aufrufe werden unterstützt. Die für Magnetbänder üblichen Aufrufe sind in `sys/mtio.h` und `sys/tape.h` beschrieben. Spezielle `ioctl`-Aufrufe und Datenstrukturen sind in `sys/exaioctl.h` definiert.

DATEIEN

Die Gerätedateien haben folgende Namen:

```
/dev/rmt/exaYtZ,
/dev/rmt/cYsZ für Tandberg TDC 3660,
/dev/rmt/exaYtZn,
/dev/rmt/exaYtZv,
/dev/rmt/exaYtZvn,
/dev/rmt/cYsZn für Tandberg TDC 3660,
/dev/rmt/cYsZnr für Tandberg TDC 3660.
```

Dabei bezeichnet `Y` die Nummer des Controllers und `Z` die SCSI-Targetnummer. Eine 1 für `Y` bezeichnet den ersten Controller (`stos.0`), eine 2 den zweiten Controller (`stos.1`) usw.

Die Gerätedateien (Geräteknotten, device nodes) für Controller 1 und Target 0 werden bei der Installation angelegt. Falls Geräte an einem anderen Controller oder an einer anderen Target-Adresse betrieben werden, müssen die entsprechenden Gerätedateien mit `/etc/init.d/scsigen exa` erzeugt werden. Dabei müssen die

Geräte angeschaltet mit dem System verbunden sein und mit einer Bandkassette geladen sein.

SIEHE AUCH

aha(7), tape.h(4), mtio.h(4), exaioc1.h(4).

FEHLERMELDUNGEN

Der Treiber gibt folgende Fehlermeldungen aus:

exaYtZ: no cartridge loaded.

exaYtZ: data cartridge write protected.

exaYtZ: new cartridge!.

exaYtZ: No write after LEOT allowed.

exaYtZ: No WriteFilemark after LEOT allowed.

exaYtZ: No Erase after LEOT allowed.

exaYtZ: No WriteVar after LEOT allowed.

exaYtZ: LEOT (logical end of tape) encountered.

exaYtZ: PEOT (physical end of tape) reached.

exaYtZ: VOLUME OVERFLOW.

exaYtZ: bad cartridge with unrecoverable medium error.

exaYtZ: illegal tape position.

Dabei bezeichnet Y die Nummer des Controllers und Z die SCSI-Targetnummer.

HINWEIS

Findet ein Reset auf dem SCSI-Bus statt, an dem das Exabyte-Gerät betrieben wird, dann führt das Exabyte-Gerät ein Rewind durch und positioniert das Band auf den Bandanfang (BOT). Die bisherige Position des Bandes geht verloren. Weitere Aufträge werden am Bandanfang ausgeführt. Bereits geschriebene Daten können dadurch überschrieben werden. Ein Sicherungslauf, der gerade stattgefunden hat, muß z.B. wiederholt werden.

BEZEICHNUNG

fas – asynchroner serieller Anschluß

BESCHREIBUNG

Der fas-Treiber unterstützt simultan sowohl den seriellen Anschluß der Systembaugruppe als auch einen zusätzlichen seriellen Adapter. Es werden bis zu zwei serielle Anschlüsse unterstützt. Ist ein Adapter für einen Anschluß nicht installiert, kann er nicht geöffnet werden. Der Anschluß kann bzgl. Geschwindigkeit (50-19200 Baud), Zeichenlänge und Parity-Bit programmiert werden. Die Ausgabegeschwindigkeit entspricht jeweils der Eingabegeschwindigkeit. Der Anschluß verhält sich wie in `termio(7)` beschrieben.

Der asynchrone Anschluß ist ein sowohl für Eingabe als auch für Ausgabe zeichenorientiertes Gerät. Dies schränkt einerseits die Bandbreite ein, die innerhalb einer Leitung erreicht werden kann, andererseits vergrößert sich die Unterbrechungslast für die Zentraleinheit. Besonders Datenübertragungsprogramme wie `uucp(1)` laufen u.U. bei Geschwindigkeiten von mehr als 4800 Baud nicht einwandfrei ab.

Die Übertragungsgeschwindigkeiten des programmierbaren Baudraten-Generators beim seriellen Adapter stimmen nicht genau mit den Übertragungsgeschwindigkeiten des Systems überein. Speziell das Setzen von B0 bewirkt einen Leitungsabbau. Das Setzen von EXTA bewirkt eine Baudrate von 19200 und das Setzen von EXTB eine Baud-Rate von 38400. Es ist nicht möglich, eine Baud-Rate von 2000, 3600 oder 7200 direkt zu setzen.

DATEIEN

`/dev/term/tty00*1`

SIEHE AUCH

`signal(2)`, `termio(7)`.

BEZEICHNUNG

fd – Diskette (floppy disk)

BESCHREIBUNG

Der Treiber für Diskettenlaufwerke ermöglicht sowohl block- als auch zeichenorientierten Diskettenzugriff. Disketten müssen vor Gebrauch formatiert werden (siehe format (1)). Sowohl das Diskettenformat 5¼" als auch 3½" wird unterstützt. Der Treiber steuert bis zu zwei Diskettenlaufwerke. Die Gerätenummer (minor device number) gibt die Nummer des Laufwerks, das Format der Diskette und die Nummer des logischen Plattenlaufwerks an.

Dateinamen für Diskettengeräte beziehen sich jeweils auf eine Geräteklasse und auf ein Gerät und werden gemäß folgendem Format vergeben:

```
/dev/{r}dsk/f{0,1}{5h,5d9,5d8,5d4,5d16,5q,3h,3d}[t]
/dev/{r}dsk/tffl2[h]
```

Dabei bedeutet *r* eine Schnittstelle im Raw-Modus (zeichenorientierte Verarbeitung). *rdsk* wählt die Geräteschnittstelle im Raw-Modus aus und *dsk* wählt die Geräteschnittstelle im Block-Modus aus. *0* oder *1* legen das Laufwerk fest, auf das zugegriffen werden soll: *f0* wählt das Laufwerk 0 aus, während *f1* das Laufwerk 1 auswählt. Die folgende Liste beschreibt das zu verwendende Format:

5h	5¼" Diskette hoher Schreibdichte (high density) (1.2 MB).
5d9	5¼" Diskette doppelter Schreibdichte (double density), 9 Sektoren pro Spur (360 KB).
5d8	5¼" Diskette doppelter Schreibdichte (double density), 8 Sektoren pro Spur (320 KB).
5d4	5¼" Diskette doppelter Schreibdichte (double density), 4 Sektoren pro Spur (320 KB).
5d16	5¼" Diskette doppelter Schreibdichte (double density), 16 Sektoren pro Spur (320 KB).
5q	5¼" Diskette vierfacher Schreibdichte (quad density) (720 KB).
3h	3½" Diskette hoher Schreibdichte (high density) (1.44 MB).
3d	3½" Diskette doppelter Schreibdichte (double density) (720 KB).

Die Angabe des Formats ist obligatorisch, wenn ein Gerät für die Formatierung geöffnet werden soll. Bei anderen Zugriffen auf die Diskette (Lese- oder Schreiboperationen) kann das Feld für die Formatangabe weggelassen werden. In diesem Fall bestimmt der Diskettentreiber automatisch das Format, mit dem die Diskette formatiert wurde und führt dann die gewünschte Operation aus, z.B.

```
cpio -itvB</dev/rsdk/f1.
```

Der letzte Parameter *t* wählt das logische Laufwerk aus, auf das zugegriffen werden soll. *t* bedeutet die gesamte Diskette. Ist *t* nicht angegeben, wird die gesamte Diskette mit Ausnahme von Zylinder 0 ausgewählt.

Neben den oben beschriebenen Konventionen für Dateinamen existieren für einige Formate Alias-Namen, die auf frühere Freigaben zurückgehen. Die folgende Liste beschreibt die Formate, die Alias-Namen haben:

Format	Alias-Name
5h	q15d
5d8	d8d
5d9	d9d

So ist z.B. die Gerätedatei `/dev/rdisk/f0q15dt` gleichbedeutend mit `/dev/rdisk/f05ht`.

Für den Zugriff auf bereits formatierte Disketten existieren noch "autodetection devices": `f{0,1}[t]`. Diese erkennen das Format der eingelegten Diskette automatisch, sofern es einem der bekannten Standardformate entspricht.

Die Gerätedateien `tffl2` und `tffl2h` ermöglichen das Lesen von SINIX-Disketten, die mit `tar cvf /dev/fl2 ...` beschrieben wurden. Sie können allerdings nur auf den `fl2`-Bereich (Datenbereich) einer solchen Diskette zugreifen, nicht auf den `fl0`-Bereich (Labelbereich) und nicht auf den `fl1`-Bereich (Bootbereich). Sie können solche Disketten nicht beschreiben, sondern nur lesen. `tffl2h` ist die Gerätedatei für Disketten mit hoher Schreibdichte (high density). Sie können auf die Gerätedateien `tffl2` und `tffl2h` nicht über die "autodetection devices" zugreifen.

ioctl-Aufrufe

Die verfügbaren `ioctl`-Aufrufe und Datenstrukturen sind in `sys/fd.h` und `sys/vtoc.h` definiert.

DATEIEN

```

/dev/dsk/f0, /dev/rdisk/f0, ...
/dev/dsk/f0t, /dev/rdisk/f0t, ...
/dev/dsk/f05h, /dev/rdisk/f05h, ...
/dev/dsk/f05ht, /dev/rdisk/f05ht, ...
/dev/dsk/f05d9, /dev/rdisk/f05d9, ...
/dev/dsk/f05d9t, /dev/rdisk/f05d9t, ...
/dev/dsk/f05d8, /dev/rdisk/f05d8, ...
/dev/dsk/f05d8t, /dev/rdisk/f05d8t, ...
/dev/dsk/f05d4, /dev/rdisk/f05d4, ...
/dev/dsk/f05d4t, /dev/rdisk/f05d4t, ...
/dev/dsk/f05d16, /dev/rdisk/f05d16, ...
/dev/dsk/f05d16t, /dev/rdisk/f05d16t, ...
/dev/dsk/f05q, /dev/rdisk/f05q, ...
/dev/dsk/f05qt, /dev/rdisk/f05qt, ...
/dev/dsk/f03h, /dev/rdisk/f03h, ...
/dev/dsk/f03ht, /dev/rdisk/f03ht, ...
/dev/dsk/f03d, /dev/rdisk/f03d, ...
/dev/dsk/f03dt, /dev/rdisk/f03dt, ...
/dev/dsk/tffl2, /dev/rdisk/tffl2,
/dev/dsk/tffl2h, /dev/rdisk/tffl2h,
```

fd(7)

fd(7)

SIEHE AUCH

format(1), ioctl(2), hd(7).

FEHLERMELDUNGEN

Der Treiber wiederholt eine fehlerhafte Übertragung mehrmals. Kommt danach immer noch keine erfolgreiche Datenübertragung zustande, so gibt der Treiber eine Fehlermeldung aus.

Schreibversuche auf eine schreibgeschützte Diskette werden ebenfalls gemeldet.

Falls eine Ein-Ausgabe-Operation wiederholt werden mußte, bevor sie schließlich erfolgreich abgeschlossen werden konnte, gibt der Treiber eine Meldung aus, die ein Altern der verwendeten Diskette oder des Diskettenlaufwerks anzeigt.

BEZEICHNUNG

filesystem – Organisation des Dateisystems

ÜBERSICHT

```
/
/usr
```

BESCHREIBUNG

Der Dateisystembaum ist für verwaltungstechnische Zwecke organisiert. Bestimmte Bereiche innerhalb des Dateisystembaums sind für Dateien vorgesehen, die nur zu einem Rechner gehören, für Dateien, die von mehreren Rechnern gleicher Architektur gemeinsam benutzt werden können, für Dateien, die von allen Rechnern gemeinsam benutzt werden können und für Home-Verzeichnisse. Diese Organisation ermöglicht es, mehrbenutzbare Dateien auf einem Rechner zu speichern, aber von mehreren Rechnern aus darauf zuzugreifen, indem eine ferne Dateizugriffsmethode wie NFS benutzt wird. Das Zusammenfassen von gleichartigen Dateien macht es leichter, den Dateisystembaum zu aktualisieren und zu handhaben.

Der Dateisystembaum besteht aus einem Wurzeldateisystem (root) und einer Reihe von einhängbaren Dateisystemen. Das Programm `mount(2)` fügt einhängbare Dateisysteme in den Dateisystembaum an bestimmten Einhängpunkten (Verzeichniseinträgen) im Wurzeldateisystem oder anderen zuvor eingehängten Dateisystemen ein. Zwei Dateisysteme `/` (root) und `/usr` müssen eingehängt sein, um das vollständige Funktionieren des Systems zu garantieren. Das Wurzeldateisystem wird zur Ladezeit vom Systemkern automatisch eingehängt; das `/usr` Dateisystem wird von der `/etc/rc.boot` Prozedur eingehängt, die als ein Teil des Ladeprozesses abläuft.

Das Wurzeldateisystem enthält Dateien, die nur einmal für jeden Rechner vorhanden sind. Es enthält folgende Verzeichnisse:

```
/dev      Zeichen- und blockorientierte Gerätedateien. Diese Gerätedateien stellen
          Verbindungen zu Hardware-Geräten oder Betriebssystemroutinen zur
          Verfügung. Gerätedateien sind typischerweise so aufgebaut, daß sie
          dem Systemkern und der Hardware-Konfiguration des Rechners
          entsprechen.

/dev/term  Bildschirmgeräte.

/dev/pts   Pseudo-Bildschirmgeräte.

/dev/xt    Geräte, die von layers benutzt werden.

/dev/sxt   Gerätedateien der Shell-Fenster, die von sh1 benutzt werden.

/etc       Konfigurationsdateien für rechnerspezifische Verwaltung und Daten-
          banken der Systemverwaltung. /etc kann als das Home-Verzeichnis
          des Rechners angesehen werden; dieses Verzeichnis bezeichnet in
          gewissem Sinne die Identität des Rechners. Ausführbare Programme
          werden nicht mehr in /etc gespeichert.
```

/home	Wurzel eines Unterbaums für Benutzerverzeichnisse.
/mnt	Temporärer Einhängpunkt für Dateisysteme. /mnt ist ein leeres Verzeichnis, in das Dateisysteme temporär eingehängt werden können.
/opt	Wurzel eines Unterbaums für zusätzliche Anwendungspakete.
/proc	Wurzel eines Unterbaums für das Prozeßdateisystem.
/sbin	Notwendige ausführbare Programme, die beim Ladeprozeß und bei der manuellen Systemwiederherstellung benutzt werden. Der volle Umfang der Dienstprogramme steht erst zur Verfügung, nachdem /usr eingehängt ist.
/tmp	Temporäre Dateien, die während des Ladeprozesses gelöscht werden.
/var	Wurzel eines Unterbaums für variable Dateien. Variable Dateien sind Dateien, die nur einmal für einen Rechner vorhanden sind, die aber auf eine beliebige Größe anwachsen können, d.h. variabel sind. Ein Beispiel dafür ist eine Protokoll-Datei.
/var/adm	Protokoll- und Abrechnungsdateien des Systems.
/var/cron	Protokoll-Datei für cron.
/var/mail	Ablage der Benutzerpost.
/var/opt	Oberstes Verzeichnis, das von Anwendungspaketen benutzt wird.
/var/preserve	Sicherungsdateien für vi(1) und ex(1).
/var/spool	Unterverzeichnis für Dateien, die für den Drucker-Spool, für Postübermittlung, für cron(1), at(1), usw. verwendet werden.
/var/tmp	Temporäre Dateien, die während des Ladeprozesses gelöscht werden.

Da es wünschenswert ist, das Wurzeldateisystem möglichst klein und beständig zu halten, werden größere Dateisysteme in plattenorientierten Systemen oft in /home, /opt, /usr und /var eingehängt.

Das Dateisystem, das unter /usr aufgehängt ist, enthält sowohl architekturabhängige als auch architekturunabhängige mehrbenutzbare Dateien. Der zu /usr/share gehörige Unterbaum enthält architekturunabhängige mehrbenutzbare Dateien, der Rest des /usr Baumes enthält architekturabhängige Dateien. Durch das Einhängen eines gemeinsamen fernen Dateisystems kann eine Gruppe von Rechnern mit allgemeiner Architektur ein einzelnes /usr-Dateisystem gemeinschaftlich benutzen. Ein einzelnes /usr/share-Dateisystem kann von Rechnern jedweder Architektur gemeinsam benutzt werden. Ein Rechner, der als Datei-Server dient, kann viele verschiedene /usr-Dateisysteme exportieren und somit verschiedene Architekturen und Betriebssystemfreigaben unterstützen. Kunden hängen /usr üblicherweise schreibgeschützt ein, so daß mehrbenutzbare Dateien nicht irrtümlich geändert werden. Das /usr-Dateisystem enthält die folgenden Unterverzeichnisse:

<code>/usr/bin</code>	Die meisten Systemdienstprogramme.
<code>/usr/sbin</code>	Ausführbare Dateien für die Systemverwaltung.
<code>/usr/games</code>	Binärdateien und Daten für Spiele.
<code>/usr/include</code>	Include-Dateien (für C-Programme usw.)
<code>/usr/lib</code>	Programmbibliotheken, verschiedene architekturabhängige Datenbanken und ausführbare Dateien, die nicht direkt vom Benutzer aufgerufen werden (System-Dämonen etc).
<code>/usr/share</code>	Unterbaum für architekturunabhängige mehrbenutzbare Dateien.
<code>/usr/share/man</code>	Unterverzeichnisse für Online-Referenzhandbuchseiten (falls vorhanden).
<code>/usr/share/lib</code>	Architekturunabhängige Datenbanken.
<code>/usr/src</code>	Quellcode für Dienstprogramme und Bibliotheken.
<code>/usr/uch</code>	Binärwerte des Berkeley Kompatibilitätspakets.
<code>/usr/ucbinclude</code>	Include-Dateien des Berkeley Kompatibilitätspakets.
<code>/usr/ucblib</code>	Bibliotheken des Berkeley Kompatibilitätspakets.

Ein Rechner mit Platten kann Wurzeldateisysteme, Swap-Dateien und `/usr-` Dateisysteme sowohl auf Rechner ohne Platten exportieren als auch auf Rechner, die nur teilweise mit Platten arbeiten. In beiden Fällen werden die `/usr-` Dateisysteme in die Hierarchie des Standarddateisystems eingehängt. Der standardmäßige Verzeichnisbaum, mit dem diese Dateisysteme für einen Server mehrbenutzbar gemacht werden, ist:

<code>/export</code>	Die voreingestellte Wurzel des exportierten Dateisystembaums.
<code>/export/exec/<i>architecture-name</i></code>	Das exportierte <code>/usr-</code> Dateisystem, das <i>architecture-name</i> für die aktuelle Freigabe unterstützt.
<code>/export/exec/<i>architecture-name.release-name</i></code>	Das exportierte <code>/usr-</code> Dateisystem, das <i>architecture-name</i> für <i>release-name</i> unterstützt.
<code>/export/exec/share</code>	Der exportierte gemeinsame Verzeichnisbaum <code>/usr/share</code> .
<code>/export/exec/share.<i>release-name</i></code>	Der exportierte gemeinsame Verzeichnisbaum <code>/usr/share</code> für <i>release-name</i> .

filesystem (7)

filesystem (7)

`/export/root/hostname` Das exportierte Wurzeldateisystem für *hostname*.
`/export/swap/hostname` Die exportierte Swap-Datei für *hostname*.
`/export/var/hostname` Der exportierte Verzeichnisbaum `/var` für *hostname*.

SIEHE AUCH

`at(1)`, `sh(1)`, `vi(1)`, `intro(4)`, `init(1M)`, `mknoð(1M)`, `mount(1M)`, `fsck(1M)`.

BEZEICHNUNG

hd – Festplatte (hard disk)

BESCHREIBUNG

Standardmäßig werden bis zu 3 Storager Festplatten Controller ausgeliefert. Pro Controller können bis zu zwei Festplattenlaufwerke unterstützt werden. Die Eigenschaften der vorhandenen Festplattenlaufwerke werden zum Zeitpunkt des Systemstarts während der Autokonfiguration individuell für die einzelnen Laufwerke bestimmt. Dazu wird Information verwendet, die sich auf den Laufwerken selbst befindet. Entspricht eine Platte nicht diesen Angaben, so wird sie vom System nicht zur Verfügung gestellt und man kann auch nicht mehr auf sie zugreifen.

Ist eine Platte bereits initialisiert, d.h. trägt sie bereits eine Partitionierung (`fdisk(1M)`), so wird während der Autokonfiguration von dieser Partitionierung die aktive UNIX Partition ausgewählt und zur Verfügung gestellt. Alle weiteren Zugriffe auf diese Platte sind dann ausschließlich über diese Partition und die Aufteilung der Partition in Slices möglich (siehe `disksetup(1M)`).

Eine Änderung der aktiven UNIX Partition ist mit `fdisk(1M)` möglich, wird aber erst dann wirksam, wenn die Platte nach der Beendigung aller Zugriffe wieder erstmalig geöffnet wird. Für die Systemplatte heißt das, daß eine solche Änderung erst beim nächsten Wiederhochfahren des Systems wirksam wird.

Der Treiber unterstützt sowohl den block- als auch den zeichenorientierten Zugriff auf die Slices einer Platte. Dabei gibt die Gerätenummer (minor device number) der verwendeten Gerätedatei an, welche Slice auf welcher Platte gemeint ist. Diese Gerätedateien stehen im Dateiverzeichnis `/dev/[r]disk`. Die Gerätenummer von Plattenlaufwerken am Storager Controller hat folgende Struktur:

```

Bit:   16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit 0 - 5: Slicennummer (siehe auch VTOC)

Bit 6 - 9: `fdisk`-Index; wird nicht verwendet; während der Autokonfiguration wird die aktive UNIX Partition ausgewählt

Bit 10 - 13: Laufwerksnummer der Platte am Controller

Bit 14 - 16: Nummer des Controllers in der Maschine

Die Aufteilung von Plattenlaufwerken erfolgt in zwei Stufen:

Partitionen `fdisk`-Tabelle im physikalischen Block 0 der Platte.

Slices VTOC im Block 0 der aktiven UNIX Partition.

Der physikalische Zylinder 0 einer Platte wird von keiner Partition und damit auch von keiner Slice überdeckt. Dies wird durch `fdisk(1M)` garantiert. Dieser Zylinder enthält außer der `fdisk`-Tabelle in Block 0 auch noch Information über ausgelagerte Sektoren und Spuren (`mkpart(1M)`) und im Fall der Systemplatte das Boot-Programm.

Ebenfalls außerhalb jeder Partition und jeder Slice liegen die letzten 5 Zylinder jedes Plattenlaufwerks am Storage Controller. Diese enthalten die "Manufacturer's Bad Block List" und die Ausweichspuren zum Ersetzen von fehlerhaften Blöcken.

Der physikalische Block 0 enthält neben der `fdisk`-Tabelle auch noch die notwendige Information über die Platte selbst, die bei der Autokonfiguration ausgewertet wird.

DATEIEN

MX300:

```
/dev/dsk/c0d0s0, ...
/dev/rdisk/c0d0s0, ...
```

MX500:

```
/dev/dsk/c8d0s0, ...
/dev/rdisk/c8d0s0, ...
```

SIEHE AUCH

`fdisk(1M)`, `mkpart(1M)`, `ioctl(2)`, `fs(4)`, `fd(7)`.

FEHLERMELDUNGEN

Der Treiber wiederholt eine fehlerhafte Übertragung bis zu dreimal. Kommt nach drei Versuchen immer noch keine erfolgreiche Datenübertragung zustande, so gibt der Treiber eine Fehlermeldung aus:

```
cx dysz: HARD ERROR: blkno = physikalische Sektornummer
      disk sector (cyl, head, sector) sector count
      error = error code
iopb: hexadez. dump des verursachenden I/O parameter blocks
```

Falls eine E/A-Operation wiederholt werden mußte, bevor sie schließlich erfolgreich abgeschlossen werden konnte, gibt der Treiber eine Meldung aus, die ein Altern des verwendeten Plattenlaufwerks anzeigt:

```
NOTICE: cx dysz: abs sector physikalische Sektornummer
      n retries [ECC | Move Bad Data | Restore]
```

Bei diesen Meldungen bezeichnet `cx dysz` die Slicenummer *z* der Platte *y* am Controller *x*. *physikalische Sektornummer* kann als Eingabe für `mkpart(1M)` verwendet werden um einen defekten Sektor auszulagern.

HINWEIS

Das VTOC und das Ersetzungsverfahren für fehlerhafte Sektoren erfordern, daß keine fehlerhaften Sektoren in Zylinder 0 auftreten. `mkpart` meldet einen unbehebaren Fehler, wenn während der Konfiguration einer Platte fehlerhafte Sektoren im ersten Zylinder auftreten. Da das Boot-Programm zweiter Stufe auf der ersten Spur der ladbaren Partition installiert werden muß, darf auch diese Spur keine fehlerhaften Sektoren enthalten.

BEZEICHNUNG

hd – Festplatte (hard disk)

BESCHREIBUNG

Der Festplattentreiber bedient bis zu zwei Festplattenlaufwerke und bis zu zwei Diskettenlaufwerke. Die Laufwerkeigenschaften werden zur Ladezeit vom CMOS RAM eingelesen; diese Eigenschaften werden zum Zeitpunkt der Systemgenerierung unter Verwendung des `setup`-Programms auf der AT Wartungsdiskette definiert. Der Treiber bestimmt den Aufbau der Platte in der nachfolgend beschriebenen Weise. Er stellt block- und zeichenorientierten (raw) Zugriff sowohl auf die individuellen Partitionen der Platte als auch auf die gesamte physikalische Platte zur Verfügung.

Die Gerätenummer des zu bedienenden Gerätes gibt an, in welcher Weise das Laufwerk behandelt wird: Die niederwertigen 4 Bits bestimmen die Partition (0 bis 15) und das fünfte Bit gibt die Laufwerknummer an (0 oder 1). Partition 0 repräsentiert die gesamte UNIX-Systempartition (so wie sie in der Tabelle `fdisk` festgelegt ist). Slices werden durch Informationen festgelegt, die im Inhaltsverzeichnis des Datenträgers (VTOC) stehen. Wird auf Partition 0 zugegriffen, so werden die Grenzen anderer Partitionen ignoriert und keine Umsetzung von fehlerhaften Blöcken durchgeführt. Der Benutzer muß in diesem Fall selbst dafür sorgen, daß fehlerhafte Blöcke ersetzt werden, wenn er die Platte in dieser Weise benutzt.

Die Aufteilung der gesamten Festplatte erfolgt in zwei Stufen: Zum einen werden die Bereiche der Platte, die von unterschiedlichen Betriebssystemen benutzt werden, in der Tabelle `fdisk` im ersten Block der Platte beschrieben. Zum anderen werden die UNIX-Systembereiche anhand von Informationen im VTOC weiter unterteilt. Das VTOC kann in einem beliebigen Block liegen. Derzeit liegt das VTOC im 30. Sektor der UNIX System-Partition. Das VTOC enthält außerdem Informationen über die Nicht-UNIX Systempartitionen, die in der Tabelle `fdisk` beschrieben sind. Wenn das Plattengerät geöffnet wird, wird das VTOC vom Treiber gelesen und die gelesene Information dazu verwendet, die Tabellen der logischen Platten aufzubauen, die dieser Gerätenummer zugeordnet sind.

Jede Partition in der `fdisk`-Tabelle wird anhand ihres Typs spezifiziert (z.B. DOS, UNIX oder andere). Eine Partition (Dateisystem) kann vom UNIX-System nur benutzt werden, wenn seine Typ-Angabe richtig ist. Eine DOS Partition kann nicht vom UNIX-System benutzt werden, es sei denn als zeichenorientiertes Gerät ohne Dateisystem.

Auf jeder Platte enthält Sektor 0 den primären Urlader und die `fdisk`-Tabelle. Die ersten 29 Sektoren der UNIX Systempartition enthalten den Urlader der ersten und zweiten Stufe. Der 30. Sektor enthält die `pdinfo`- und die VTOC-Tabelle. Die Sektoren 31 - 34 enthalten die Umsetztabelle für fehlerhafte Blöcke und Spuren. Ab dem Sektor 35 werden so viele Sektoren wie nötig für die Ersatzspuren und Ersatzsektoren zur Verfügung gestellt.

Die `fdisk`-Tabelle zeigt an, welche der Partitionen aktive oder ladbare Partitionen sind. Wenn das System geladen wird, sucht der Urlader-Code in der `fdisk`-Tabelle nach einer aktiven Partition und springt zum Sektor 0 dieser Partition. Dort steht der Urlader erster Stufe. Ist er länger als ein Sektor, muß das der Urlader selbst berücksichtigen. Es ist zu beachten, daß sowohl auf den ersten Zylinder (der die `fdisk`-Tabelle, den primären Urlader, das VTOC und Ersatzsektoren enthält) als

auch auf die erste Spur der aktiven Partition (die den Urlader enthält), nur über Partition 0 zugegriffen werden kann, da diese Spuren vom VTOC normalerweise nicht als Teil irgendeiner Partition betrachtet werden.

Fehlerhafte Sektoren werden vom Treiber folgendermaßen behandelt: Die Tabelle der fehlerhaften Blöcke wird beim Öffnen des Geräts vom Treiber gelesen. Die Tabelle ist ein Bereich mit paarweisen Zahlenwerten, die jeweils den fehlerhaften Sektor und den zugewiesenen Ersatzsektor bezeichnen. Dabei wird jede Sektornummer absolut angegeben, d.h. der erste Sektor auf der Platte ist Sektor 0 usw.

Vor jeder Ein-/Ausgabeoperation durchsucht der Treiber diese Tabelle, um festzustellen, ob ein zu übertragender Sektor fehlerhaft ist. Findet er für die Ein-/Ausgabe eine fehlerhafte Sektorangabe, wird zunächst die gesamte Ein-/Ausgabe bis zum fehlerhaften Sektor durchgeführt, danach wird der fehlerhafte Sektor neu zugewiesen und schließlich die auf den fehlerhaften Sektor folgende Ein-/Ausgabe durchgeführt.

Es ist zu beachten, daß dieses Verfahren die Ausführung des Programms `mkpart` (1M) erfordert, bevor das System das erste Mal von der Festplatte geladen wird. `mkpart` versucht, jeden Sektor auf der Platte einmal zu schreiben und zu lesen, um so alle Sektoren aufzufinden, die fehlerhaft sind. Alle fehlerhaften Sektoren werden in die Umsetztabelle eingetragen, die von `mkpart` aufgebaut und zum gleichen Zeitpunkt wie das VTOC auf der Platte eingerichtet wird. Auch wenn dieser Prüfdurchlauf nicht stattfindet, kann das System arbeiten. Da der Treiber feststellt, daß die Tabelle leer ist, unternimmt er keinen Versuch, fehlerhafte Sektoren zu ersetzen.

Für den Fall, daß sich während des Systemablaufs fehlerhafte Blöcke ergeben, kann man `mkpart` (mit der `-A` Option) starten, um die neuen fehlerhaften Blöcke in die Tabelle einzutragen. Allerdings muß der Benutzer u. U. das gesamte Dateisystem von der letzten Gesamtsicherung wiederherstellen, je nachdem wo der fehlerhafte Block aufgetreten ist.

Partitionen

Die `fdisk`-Tabelle erlaubt nur die Zuweisung von Partitionen auf Zylindergrenze; das VTOC läßt dagegen auch Partitionen zu, die auf Spurgrenze anfangen. Diese Möglichkeit wird für die ladbare UNIX Systempartition genutzt, damit die erste Spur (die den Urlader-Code enthält) nicht unbedingt Teil dieser Partition sein muß. Die `fdisk`-Tabelle läßt maximal vier Partitionen auf der Festplatte zu, das VTOC erlaubt jedoch, daß der UNIX-Systembereich in maximal 16 Partitionen unterteilt wird. Jede Partition ist durch eine Gerätenummer gekennzeichnet. Die Zuordnung der Sektoren zu den Gerätenummern geschieht anhand der Datei `/etc/partitions` zu dem Zeitpunkt, an dem zum erstenmal auf die Platte zugegriffen wird. Diese Zuordnung bleibt solange bestehen, bis die Datei `/etc/partitions` geändert und das Programm `mkpart` nochmals aufgerufen wird.

Alle Versuche, Dateisysteme zu öffnen, für die keine Partitionen existieren, führen zu Fehlern (nicht existierendes Gerät). Ebenso verlaufen alle Versuche fehlerhaft, Partitionen einzuhängen, die keine UNIX Systeme enthalten (siehe `mount` (8)).

DATEIEN

/dev/dsk/c0d0s0, ...
 /dev/rdisk/c0d0s0, ...

SIEHE AUCH

fdisk(1M), mkpart(1M), ioctl(2), fs(4), fd(7).

FEHLERMELDUNGEN

Der Treiber wiederholt fehlerhafte Übertragungen bis zu zehnmal, je nach der Fehlerart. Für einige Fehler ist keine Wiederholung möglich. Beim Auftreten eines Fehlers während der Übertragung gibt der Treiber eine entsprechende Meldung aus. Die Fehlerarten, für die eine Wiederholung möglich ist, sind in der nachfolgenden Tabelle aufgeführt. Fehler der Festplattensteuerung werden in folgender Weise ausgegeben:

HD error: drive *n*, cyl *c*, head *h*, sector *s*: *drive error message*
 HD controller: *controller error message*

Die erste Meldung gilt für einen Fehler, der nach dem Beginn der Übertragung auftrat. Dabei bezeichnet *n* das Plattenlaufwerk, für das der Fehler auftrat, *c* bezeichnet den Zylinder, *h* die Spur und *s* den Sektor, der gelesen oder geschrieben werden soll. *drive error message* ist eine der Meldungen, die in der folgenden Liste vorkommen:

Track 0 not found

Die Platte ist möglicherweise nicht ordnungsgemäß formatiert.

Uncorrectable data read error

Bei dem Versuch, den geforderten Block zu lesen, bemerkte die Steuerung einen CRC Fehler.

Data address mark not found

Die Platte ist möglicherweise nicht ordnungsgemäß formatiert.

Sector not found

Die Platte ist möglicherweise nicht ordnungsgemäß formatiert.

Command aborted

Die Steuerung konnte ein Kommando nicht vollständig ausführen.

Bad track flag detected

Der angeforderte Block ist als fehlerhaft gekennzeichnet, erscheint aber nicht in der Umsetztabelle für fehlerhafte Blöcke.

Die zweite Meldung erscheint, wenn ein Steuerungsfehler während des Aufbaus oder der tatsächlichen Übertragung eines Blockes auftritt. Die *controller error message* ist eine der Meldungen, die in der nachfolgenden Liste aufgeführt sind.

command aborted

Die Steuerung konnte das geforderte Kommando nicht vollständig ausführen.

write fault

Die Steuerung entdeckte einen Fehler am Festplattenlaufwerk.

stays busy

Beim Versuch, auf die Steuerung zuzugreifen, trat eine Zeitüberschreitung auf.

Es gibt eine zusätzliche Meldung, die anzeigt, daß ein Fehler auftrat, der von der Steuerung korrigiert werden kann:

NOTE: Soft read error corrected by ECC algorithm: unit *n*, sector *s*

Dabei bezeichnet *n* die Plattennummer, für die der Fehler auftrat, und *s* bezeichnet den Sektor, der gerade gelesen wurde. Diese Warnungsmeldung gibt an, daß der Fehlerbehebungsalgorithmus der Steuerung diesen Fehler erfolgreich beheben konnte. Dies kann ein Zeichen dafür sein, daß ein Sektor gerade dabei ist, fehlerhaft zu werden. Falls diese Meldung mehrmals für den gleichen Sektor erscheint, sollte dieser Sektor vermutlich als fehlerhaft gekennzeichnet werden.

HINWEIS

Das VTOC und das Ersetzungsverfahren für fehlerhafte Sektoren erfordern, daß keine fehlerhaften Sektoren in Zylinder 0 auftreten. `mkpart` meldet einen unbeheb-
baren Fehler, wenn während der Konfiguration einer Platte fehlerhafte Sektoren im
ersten Zylinder auftreten. Da der Urlader zweiter Stufe auf der ersten Spur der lad-
baren Partition installiert werden muß, darf auch diese Spur keine fehlerhaften Sek-
toren enthalten.

BEZEICHNUNG

ICMP – Internet-Protokoll der Steuerungsmeldungen

ÜBERBLICK

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/ip_icmp.h>

s = socket(AF_INET, SOCK_RAW, proto);
t = t_open("/dev/icmp", O_RDWR);
```

BESCHREIBUNG

ICMP ist das Protokoll der Fehler- und Steuerungsmeldungen, das von der Internet-Protokollfamilie benutzt wird. Es wird vom Kern verwendet, um Fehler bei der Protokollverarbeitung zu bearbeiten und zu melden. Es kann auch von Programmen, die die Socket-Schnittstelle oder das Transport-Level-Interface (TLI) zur Netzwerksteuerung und für Diagnosefunktionen verwenden, benutzt werden. Bei der Verwendung über die Socket-Schnittstelle wird der Typ "raw socket" benutzt. Die Protokollnummer für ICMP, die im Parameter *proto* des Socket-Aufrufs verwendet wird, können Sie mit `getprotobyname()` erhalten (siehe `getprotoent(3N)`). ICMP-Dateideskriptoren und -Sockets sind verbindungslos und werden normalerweise mit `t_sndudata` / `t_rcvudata` und den Aufrufen `sendto()` / `recvfrom()` benutzt.

Ausgesendeten Paketen wird automatisch ein Anfangsetikett des Internet-Protokolls (IP) vorangestellt. Eingehende Pakete werden dem Benutzer mit unverändertem IP-Anfangsetikett und Optionen zur Verfügung gestellt.

ICMP ist ein Datagramm-Protokoll, das oberhalb IP angesiedelt ist. Es wird intern vom Protokoll-Code für verschiedene Zwecke benutzt wie z.B. zur Wegewahl, Fehler-Eingrenzung und Überlastungskontrolle. Der Empfang einer ICMP "redirect" Meldung bewirkt einen neuen Eintrag in der Routing-Tabelle oder verändert einen bereits existierenden Eintrag. ICMP-Meldungen werden routinemäßig vom Protokoll-Code abgeschickt. Empfangene ICMP-Meldungen können an Benutzer von Protokollen höherer Stufe, wie z.B. TCP oder UDP, als Fehler-Rückmeldungen von Systemaufrufen zurückgegeben werden. Eine Kopie aller vom System empfangenen ICMP-Meldungen, werden jedem Benutzer eines offenen ICMP-Sockets oder TLI-Deskriptors zur Verfügung gestellt.

SIEHE AUCH

`send(2)`, `getprotoent(3N)`, `recvfrom(3N)`, `t_rcvudata(3N)`, `t_sndudata(3N)`, `routing(4)`, `inet(7)`, `ip(7)`.

Postel, Jon, *Internet Control Message Protocol — DARPA Internet Program Protocol Specification*, RFC 792, Network Information Center, SRI International, Menlo Park, Calif., September 1981.

FEHLERMELDUNGEN

Bei einer Socket-Operation kann ein Fehler auftreten, wobei eine der folgenden Fehlermeldungen ausgegeben wird:

EISCONN	Es wurde versucht, für einen Socket eine Verbindung aufzubauen, für den bereits ein Anschluß existiert oder es wurde versucht, ein Datagramm an die Zieladresse des Sockets zu senden und der Socket ist bereits angeschlossen.
ENOTCONN	Es wurde versucht, ein Datagramm ohne Angabe einer Zieladresse abzuschicken, der Socket war jedoch nicht angeschlossen.
ENOBUFS	Das System hat nicht genug Speicherplatz für eine interne Datenstruktur.
EADDRNOTAVAIL	Es wurde versucht, einen Socket mit einer Netzwerkadresse aufzubauen, für den keine Netzwerkschnittstelle existiert.

HINWEIS

Antworten auf ICMP-Rückmeldungen werden nicht auf dem gleichen Wege an die Quelladresse zurückgesendet, sondern meist mittels der normalen Routing-Mechanismen.

BEZEICHNUNG

inet – Internet-Protokollfamilie

ÜBERBLICK

```
#include <sys/types.h>
#include <netinet/in.h>
```

BESCHREIBUNG

Die Internet-Protokollfamilie führt eine Reihe von Protokollen aus, in deren Mittelpunkt das *Internet-Protokoll* (IP) steht und die ein gemeinsames Adreßformat benutzen. Auf die Protokolle der Internetfamilie kann über die Socket-Schnittstelle zugegriffen werden, wo sie die Socket-Typen `SOCK_STREAM`, `SOCK_DGRAM` und `SOCK_RAW` unterstützen. Oder es kann über das Transport-Level-Interface (TLI) zugegriffen werden; hierbei werden die verbindungslosen (`T_CLTS`) und die verbindungsorientierten (`T_COTS_ORD`) Service-Typen unterstützt.

PROTOKOLLE

Die Internet-Protokollfamilie enthält das Internet-Protokoll (IP), das Adreßauflösungsprotokoll (ARP), das Internet-Protokoll der Steuerungsmeldungen (ICMP), das Protokoll der Übertragungssteuerung (TCP) und das Benutzer-Datagrammprotokoll (UDP).

TCP unterstützt die `SOCK_STREAM` Abstraktion der Socket-Schnittstellen und den TLI-Service-Typ `T_COTS_ORD`. UDP unterstützt die Socketabstraktion `SOCK_DGRAM` und den TLI Service-Typ `T_CLTS` (siehe `tcp(7)` und `udp(7)`). Eine direkte Schnittstelle zu IP steht sowohl über TLI als auch über die Socket-Schnittstelle zur Verfügung (siehe `ip(7)`). ICMP wird vom Kern benutzt, um Fehler bei der Protokollverarbeitung zu bearbeiten und zu melden. Benutzerprogramme können ebenfalls mit ICMP arbeiten (siehe `icmp(7)`). ARP wird benutzt, um 32-Bit IP-Adressen in 48-Bit Ethernet-Adressen zu übersetzen (siehe `arp(7)`).

Die 32-Bit IP-Adresse besteht aus einem Netzwerknúmernteil und einem Host-númernteil. Die IP-Adressen sind in Klassen unterteilt. Diese Klassen legen fest, welcher Teil der Netzwerknúmernteil ist. Bei Adressen der Klasse A ist das höchstwertige Bit Null, wobei die 8 höherwertigen Bits die Netzwerknúmer darstellen. Bei Adressen der Klasse B sind die beiden höchstwertigen Bits auf 10 gesetzt; in diesem Fall dienen die 16 höherwertigen Bits als Netzwerknúmerfeld. Bei Adressen der Klasse C ist der Netzwerknúmernteil 24 Bit lang, wobei die drei höchstwertigen Bits 110 enthalten. Anlagen mit einem IP-Netzwerk-Cluster können wahlweise eine einzige Netzwerknúmer für den Cluster benutzen. Dies geschieht mittels Teilnetz-Adressierung. Der Hostnúmernteil der Adresse wird weiter unterteilt in einen Teilnetznummernteil und einen Hostnúmernteil. Innerhalb eines Clusters erscheint jedes Teilnetz wie ein individuelles Netzwerk. Nach außen erscheint die gesamte Gruppe wie ein einziges, einheitliches Netzwerk, das nur einen einzigen Routing-Eintrag benötigt. Die Adressierung von Teilnetzen wird durch die folgenden `ioctl(2)`-Aufrufe ermöglicht und geprüft; sie haben das gleiche Format wie der `SIOCSIFADDR`-Aufruf (siehe `if(3N)`).

SIOCSIFNETMASK	Schnittstelle für Netzwerkmaske setzen. Die Netzwerkmaske gibt den Netzwerkteil der Adresse an. Falls sie mehr enthält als durch den Adreßtyp angezeigt ist, bedeutet dies, daß mit Teilnetzen gearbeitet wird.
SIOCGIFNETMASK	Schnittstelle für Netzwerkmaske einlesen.

ADRESSIERUNG

IP-Adressen sind vier Byte lange Bereiche, die in der Reihenfolge der Netzwerkbytes abgespeichert sind. IP-Adressen sollten unter Verwendung der Routinen zur Umsetzung der Byte-Reihenfolge (siehe `byteorder(3N)`) bearbeitet werden.

Adressen der Internet-Protokollfamilie verwenden folgende Struktur:

```
struct sockaddr_in {
    short sin_family;
    u_short sin_port;
    struct in_addr sin_addr;
    char sin_zero[8];
};
```

Es stehen Bibliotheksroutinen zur Verfügung, um Strukturen dieser Art zu bearbeiten (siehe `inet(3N)`).

Das `sin_addr` Feld der `sockaddr_in` Struktur enthält eine lokale oder ferne IP-Adresse. Jede Netzwerk-Schnittstelle hat ihre eigene eindeutige IP-Adresse. Der Sonderwert `INADDR_ANY` kann in diesem Feld angegeben werden, um Wildcard-Umsetzung anzustoßen. Wird dieser Sonderwert in einem `bind(2)`-Aufruf angegeben, so bleibt die IP-Adresse des Sockets frei, so daß der Socket alle Verbindungen oder Meldungen empfängt, die an irgendeine gültige IP-Adresse des Systems geschickt wurden. Dies kann sich als nützlich erweisen, wenn ein Prozeß die lokale IP-Adresse nicht kennt oder wenn ein Prozeß zum Empfang von Anforderungen alle seine Netzwerk-Schnittstellen ausnutzen möchte. In der `sockaddr_in` Struktur, die im Aufruf `bind(2)` angegeben wird, muß für `in_addr` entweder der Wert `INADDR_ANY` oder eine im System gültige IP-Adresse angegeben werden. Anforderungen, eine andere Adresse einzubinden, ergeben den Fehler `EADDRNOTAVAIL`. Wird ein `connect(2)`-Aufruf für einen Socket abgesetzt, der in einer lokalen Adresse Platzhalter enthält, setzt das System das `sin_addr` Feld des Sockets auf die IP-Adresse der Netzwerk-Schnittstelle, über die die Pakete dieser Verbindung geleitet werden.

Das `sin_port` Feld der `sockaddr_in`-Struktur spezifiziert eine Anschlußnummer, die von TCP oder UDP benutzt wird. Die lokale Anschlußadresse, die im Aufruf `bind(2)` angegeben wird, muß größer sein als `IPPORT_RESERVED` (definiert in `<netinet/in.h>`), es sei denn, der erzeugende Prozeß läuft unter der Kennung `root` ab und stellt einen Bereich für geschützte Anschlußnummern zur Verfügung. Darüberhinaus darf die lokale Anschlußadresse nicht von einem Socket der gleichen Adreßfamilie und des gleichen Typs benutzt werden. Anforderungen, Sockets mit Anschlußnummern zu verbinden, die bereits von anderen Sockets benutzt werden, ergeben den Fehler `EADDRINUSE`. Wenn die lokale Anschlußadresse mit Null angegeben wird, verwendet das System eine eindeutige Anschlußadresse, die größer als `IPPORT_RESERVED` ist. Ebenso wird eine

eindeutige lokale Anschlußadresse vom System ausgewählt, wenn ein nicht eingebundener Socket in einem `connect(2)`- oder `sendto`-Aufruf (siehe `send(2)`) verwendet wird. Auf diese Weise können Programme, die sich nicht darum kümmern, welche lokale Anschlußnummer benutzt wird, TCP-Verbindungen aufbauen, indem sie einfach `socket(2)` und danach `connect(2)` aufrufen. Außerdem können sie UDP-Datagramme absenden, indem sie nacheinander `socket(2)` und `sendto(2)` aufrufen.

Obwohl diese Implementierung nur eindeutige lokale Anschlußnummern für Sockets zuläßt, erlaubt TCP mehrfache simultane Verbindungen für die gleiche lokale Anschlußnummer, solange die fernen IP-Adressen oder -Anschlußnummern für jede Verbindung unterschiedlich sind. Programme können die Socket-Einschränkungen explizit außer Kraft setzen, indem sie die Socket-Option `SO_REUSEADDR` mittels `setsockopt` setzen (siehe `getsockopt(3N)`).

TLI wendet leicht unterschiedliche Semantikregeln bei der Einbindung von lokalen Anschlußnummern an. Diese Semantikregeln treffen dann zu, wenn Protokolle der Internet-Familie über TLI benutzt werden.

SIEHE AUCH

`ioctl(2)`, `send(2)`, `bind(3N)`, `connect(3N)`, `getsockopt(3N)`, `if(3N)`, `byteorder(3N)`, `gethostent(3N)`, `getnetent(3N)`, `getprotoent(3N)`, `getservent(3N)`, `socket(3N)`, `arp(7)`, `icmp(7)`, `ip(7)`, `tcp(7)`, `udp(7)`.

Network Information Center, *DDN Protocol Handbook* (3 vols.), Network Information Center, SRI International, Menlo Park, Calif., 1985.

BEZEICHNUNG

IP – Internet Protokoll

ÜBERBLICK

```
#include <sys/socket.h>
#include <netinet/in.h>

s = socket(AF_INET, SOCK_RAW, proto);
t = t_open ("/dev/rawip", O_RDWR);
d = open ("/dev/ip", O_RDWR);
```

BESCHREIBUNG

IP ist das Netzwerkprotokoll für Datagramme, das im Mittelpunkt der Internet-Protokollfamilie steht. Programme können IP über Protokolle auf höherer Stufe benutzen, wie z.B. das Protokoll der Übertragungssteuerung (TCP) oder das Benutzer-Datagramm-Protokoll (UDP), oder sie können eine direkte Schnittstelle zu IP aufbauen; siehe `tcp(7)` und `udp(7)`. Der Direktzugriff kann über die Socket-Schnittstelle (unter Verwendung eines `raw socket` oder über das Transport-Level-Interface (TLI) erfolgen. Die Protokoll-Optionen, die in der IP-Spezifikation angegeben sind, können in abgehenden Datagrammen gesetzt werden.

Der STREAMS-Treiber `/dev/rawip` ist der TLI-Transport-Provider, der den zeichenweisen Zugriff auf IP zur Verfügung stellt. Das Gerät `/dev/ip` ist der STREAMS-Multiplex-Treiber, der die Protokollverarbeitung des IP bereitstellt. Letzterer baut Verbindungen nach unten zu Verbindungsvorrichtungen (Schnittstellentreibern, siehe `if(3N)`) und nach oben zu Transport-Providern wie TCP und UDP auf.

Zeichenorientierte IP-Sockets sind verbindungslos und werden normalerweise in den Aufrufen `sendto()` und `recvfrom()` verwendet, (siehe `send(2)` und `recv(2)`). Der `connect(2)`-Aufruf kann ebenfalls verwendet werden, um die Zieladresse für zukünftige Datagramme festzulegen (in diesem Fall können auch `read(2)` oder `recv(2)` sowie `write(2)` oder `send(2)` Aufrufe benutzt werden). Falls `proto` gleich Null ist, wird das Vorgabeprotokoll `IPPROTO_RAW` verwendet. Falls `proto` ungleich Null ist, wird der entsprechende Wert als Protokollnummer in abgesendeten Protokollen gesetzt; bei eingehenden Protokollen wird er verwendet, um die richtigen Protokolle auszuwählen. Für jedes abgesendete Datagramm wird ein IP Anfangsetikett generiert und dem Datagramm vorangestellt; empfangene Datagramme werden mit dem unveränderten IP-Anfangsetikett und denselben Optionen zurückgegeben.

Nur die Socketoption `IP_OPTIONS` wird auf der IP-Stufe unterstützt. Diese Socket-Option kann benutzt werden, um in jedes abgesendete Datagramm IP-Optionen einzubinden. IP-Optionen, die abgesendet werden sollen, werden mit `setsockopt()` gesetzt (siehe `getsockopt(2)`). Der Aufruf `getsockopt(2)` gibt die IP-Optionen wieder, die im letzten `setsockopt()`-Aufruf gesetzt wurden. IP-Optionen in eingehenden Datagrammen sind für Benutzerprogramme nur dann sichtbar, wenn zeichenorientierte IP-Sockets verwendet werden. Das Format der in `setsockopt()` angegebenen IP-Optionen entspricht dem in der IP-Spezifikation beschriebenen Format mit einer Ausnahme: die Liste der Adressen der Routing-Optionen muß das nächstgelegene Gateway am Anfang der Liste der Gateways beinhalten. Die nächstgelegene Gateway-Adresse wird aus der Liste der Optionen entnommen und

die Größe vor der Verwendung entsprechend ausgerichtet. IP-Optionen können für jeden Socket-Typ der Internet-Familie verwendet werden.

Auf der Socketebene kann die Socket-Option `SO_DONTROUTE` angewendet werden. Diese Option bewirkt für gesendete Datagramme, daß das Routing während der Ausgabe übergangen wird. Normalerweise wählt IP die Netzwerkschnittstelle, an die das Datagramm gesendet wird, und eventuell ein zwischengeschaltetes Gateway anhand eines Eintrags in der Routing-Tabelle aus; siehe `routing(4)`. Ist `SO_DONTROUTE` gesetzt, so wird beim Absenden des Datagramms die Schnittstelle verwendet, deren Netzwerknummer oder deren volle IP-Adresse der Zieladresse entspricht. Wird keine Entsprechung gefunden, so wird der Fehler `ENETUNRCH` zurückgemeldet.

Zeichenorientierte IP-Datagramme können auch gesendet und empfangen werden, wenn Sie verbindungslose TLI-Grundfunktionen benutzen.

Datagramme bewegen sich zwischen den IP-Ebenen in zwei Richtungen: vom Netzwerk *nach oben* zu den Benutzerprozessen und von den Benutzerprozessen *nach unten* zum Netzwerk. Aufgrund dieser Orientierung ist IP *oberhalb* des Netzwerkschnittstellen-Treibers und *unterhalb* der Transportprotokolle wie UDP und TCP angesiedelt. Das Internet-Protokoll der Steuerungsmeldungen (ICMP) ist ein logischer Bestandteil von IP; siehe `icmp(7)`.

IP sieht eine Prüfsumme für den Kopfteil, nicht aber für den Datenteil des Datagramms vor. Der Wert der Prüfsumme wird beim Senden des Datagramms errechnet und gesetzt und beim Empfang des Datagramms geprüft. Die IP Prüfsummenüberprüfung für Anfangsetiketten kann zum Zweck der Ablaufverfolgung außer Kraft gesetzt werden, indem die Kernvariable `ipcksum` durch einen Patch auf Null gesetzt wird.

IP-Optionen in empfangenen Datagrammen werden auf der IP Ebene entsprechend der Protokoll-Spezifikation verarbeitet. IP-Optionen, die derzeit erkannt werden, sind: security, loose source und record route (LSRR), strict source und record route (SSRR), record route, stream identifier und internet timestamp.

Die IP-Ebene leitet normalerweise empfangene Datagramme weiter, die nicht für sie bestimmt sind. Die Weiterleitung wird von der Kernvariablen `ipforwarding` gesteuert: Ist `ipforwarding` gleich Null, werden IP-Datagramme nicht weitergeleitet; ist `ipforwarding` gleich Eins, werden IP-Datagramme weitergeleitet. `ipforwarding` wird normalerweise nur für Rechner mit mehr als einer Netzwerkschnittstelle (Internetwork-Knoten) auf Eins gesetzt. Diese Kernvariable kann gepatcht werden, um die Weiterleitung ein- oder auszuschalten.

Die IP-Ebene sendet in vielen Fällen eine ICMP-Meldung an den Quellrechner zurück, wenn sie ein Datagramm empfängt, das sie nicht verarbeiten kann. Wenn im Feld "time to live" im IP-Anfangsetikett während der Weiterleitung eines Datagramms der Wert Null erscheint, wird eine "time exceeded" ICMP-Meldung abgeschickt. Falls das Datagramm nicht weitergeleitet werden kann, weil es keinen Weg für seine Zielbestimmung gibt oder falls das Datagramm nicht fragmentiert werden kann, wird eine "destination unreachable" Meldung abgeschickt. Ist das Datagramm an den lokalen Rechner adressiert, aber für ein Protokoll bestimmt, das nicht unterstützt wird oder für einen Anschluß, der nicht existiert, wird ebenfalls die Meldung "destination unreachable" abgeschickt. Die IP-Ebene kann eine ICMP-

Meldung "source quench" abschicken, wenn Datagramme in zu schneller Folge ankommen. ICMP-Meldungen werden nur für das erste Fragment eines fragmentierten Datagramms abgeschickt und werden niemals als Antwort auf Fehler in anderen ICMP-Meldungen zurückgesendet.

Die IP-Ebene unterstützt Fragmentierung und Zusammenfügung. Datagramme werden bei der Ausgabe fragmentiert, wenn sie länger als die größte Übertragungseinheit (MTU) der Netzwerkschnittstelle sind. Fragmente von empfangenen Datagrammen werden aus den Warteschlangen für Zusammenfügung entfernt, wenn das gesamte Datagramm nicht innerhalb kurzer Zeit rekonstruiert wird.

Fehler, die beim Senden auftreten und auf der Ebene des Netzwerkschnittstellentreibers entdeckt werden, werden von IP an den Benutzerprozeß zurückgeschickt.

SIEHE AUCH

`read(2)`, `write(2)`, `connect(3N)`, `getsockopt(3N)`, `recv(3N)`, `send(3N)`, `routing(4)`, `icmp(7)`, `inet(7)` `tcp(7)`, `udp(7)`.

Postel, Jon, *Internet Protocol - DARPA Internet Program Protocol Specification*, RFC 791, Network Information Center, SRI International, Menlo Park, Calif., September 1981.

FEHLERMELDUNGEN

Mißlingt eine Socket-Operation, wird einer der folgenden Fehler zurückgegeben:

EACCESS	Es wurde eine IP-Broadcast-Zieladresse angegeben und der aufrufende Benutzer war nicht privilegiert.
EISCONN	Es wurde versucht, eine Verbindung für einen Socket herzustellen, für den schon eine Verbindung existierte. Oder es wurde versucht, ein Datagramm mit der angegebenen Zieladresse abzuschicken, obwohl der Socket bereits angeschlossen war.
EMSGSIZE	Es wurde versucht, ein Datagramm abzusenden, das zu groß für die betreffende Schnittstelle war, aber nicht fragmentiert werden durfte (wie z.B. broadcasts).
ENETUNREACH	Es wurde versucht, eine Verbindung herzustellen oder ein Datagramm abzuschicken, obwohl kein entsprechender Eintrag in der Routing-Tabelle existiert bzw. obwohl eine ICMP-Meldung "destination unreachable" empfangen wurde.
ENOTCONN	Es wurde ein Datagramm abgeschickt, obwohl keine Zieladresse angegeben und der Socket noch nicht angeschlossen war.
ENOBUFS	Der Speicherplatz des Systems für Fragmentierungspuffer oder andere interne Datenstrukturen reicht nicht aus.
EADDRNOTAVAIL	Es wurde versucht, einen Socket mit einer lokalen Adresse aufzubauen, die keiner Netzwerkschnittstelle entspricht, oder es wurde eine IP-Broadcast-Zieladresse angegeben, obwohl die Netzwerkschnittstelle <code>broadcast</code> nicht unterstützt.

Folgende Fehler können beim Setzen oder Einholen von IP-Optionen auftreten:

EINVAL Es wurde ein unbekannter Name für eine Socket-Option angegeben.

EINVAL Das IP-Optionsfeld hatte ein falsches Format; das Optionsfeld war entweder kürzer als der zulässige Minimalwert oder länger als der vorgesehene Optionspuffer.

HINWEIS

Benutzer von Protokollen höherer Stufe, wie z.B. TCP und UDP, sollten in der Lage sein, empfangene IP-Optionen anzuschauen.

BEZEICHNUNG

keyboard – Tastatur der Systemkonsole

BESCHREIBUNG

Die Systemkonsole besteht aus zwei unterschiedlichen Teilen: der Tastatur und dem Bildschirm (siehe `display` (7)).

Die Tastatur wird benutzt, um Daten einzugeben und um bestimmte Steuerzeichen an den Rechner zu übermitteln. Die SINIX Software führt Terminal-Emulation für den Bildschirm und die Tastatur der Konsole aus und verwendet dabei einige besondere Tasten und Tastenkombinationen. Diese Tasten und Tastenkombinationen haben Sondernamen, die im SINIX System eindeutig sind und möglicherweise der Beschriftung Ihrer Tastatur entsprechen.

Wenn Sie eine Taste drücken, läuft eine der folgenden Aktionen ab:

- Ein ASCII Wert wird eingegeben.
- Die Bedeutung einer anderen Taste bzw. anderer Tasten wird verändert.
- Eine Zeichenfolge wird an den Rechner abgeschickt.
- Eine Funktion wird angestoßen.

Wenn eine Taste gedrückt wird, schickt die Tastatur einen Scan Code an den Rechner. Dieser Scan Code wird vom Tastatur-Treiber interpretiert. Die aktuelle Codefolge, die an die Terminal-Eingaberoutine (siehe `termio` (7)) weitergegeben wird, ist in einer Reihe von internen Tabellen im Treiber definiert. Diese Tabellen können mittels Software-Kommandos modifiziert werden (siehe die Beschreibung der `ioctl` Aufrufe weiter unten). Zusätzlich kann der Treiber veranlaßt werden, die Scan Codes beim Drücken und Loslassen der Tastatur direkt und ohne Übersetzung zu übergeben.

Ändern der Bedeutung

Die Aktion, die durch eine Taste angestoßen wird, kann dadurch verändert werden, daß die Taste in Kombination mit bestimmten anderen Tasten verwendet wird. Die `SHIFT` Taste ändert zum Beispiel die ASCII Werte von alphanumerischen Tasten. Wenn Sie die Taste `CTRL` festhalten, während Sie eine andere Taste drücken, so wird ein Steuercode (wie z.B. `CTRL-D`, `CTRL-S` oder `CTRL-Q`) abgeschickt. Wird die `ALT`-Taste gedrückt, so modifiziert dies ebenfalls den Wert einer Taste. Die `SHIFT`-, `CTRL`- und `ALT`-Tasten können miteinander kombiniert werden.

Bildschirm wechseln

Wenn der aktuelle Bildschirm gewechselt werden soll, geben Sie `ALT-SYSREQ` (wird in einigen Systemen auch mit `ALT-PRINTSCRN` bezeichnet) ein und drücken danach eine Taste, die den gewünschten Bildschirm bezeichnet. Es kann jeder beliebige Bildschirm ausgewählt werden, wenn nach `ALT-SYSREQ` *F_n* gedrückt wird, wobei *F_n* eine der Funktionstasten bezeichnet. *F1* bezieht sich auf den ersten virtuellen Terminal-Bildschirm, *F2* bezeichnet den zweiten virtuellen Terminal-Bildschirm usw.. `ALT-SYSREQ 'h'` bezeichnet den Bildschirm der System-Konsole (`/dev/console`). Der nächste aktive Bildschirm kann mit `ALT-SYSREQ 'n'` und der vorhergehende Bildschirm mit `ALT-SYSREQ 'p'` ausgewählt werden.

Die Voreinstellung für die Sequenz zur Aktivierung des Bildschirmwechsels (ALT-SYSREQ) kann konfiguriert werden. Die Einträge in der `SYSREQ` Tabelle können über Software-Kommandos modifiziert werden (siehe Beschreibung der `ioctl` Aufrufe weiter unten).

Sondertasten

Aus der folgenden Tabelle wird ersichtlich, welche Tasten einer typischen Konsole den SINIX Systemtasten entsprechen. Ein Bindestrich (-) bedeutet in dieser Tabelle, daß Sie die erste Taste festhalten müssen, während Sie die zweite Taste drücken. Die Abbildung der Zeichen, die Signale erzeugen, auf die tatsächlich erzeugten Signale wird mit `stty(1)` gesetzt und kann verändert werden (siehe `stty(1)`).

Name	Tastenbezeichnung	Aktion
INTR	DEL	Stoppt die derzeitige Aktion und springt in die Shell zurück. Diese Taste heißt auch die RUB OUT- oder INTERRUPT- Taste.
BACKSPACE	←	Löscht das erste Zeichen links von der Schreibmarke. Es ist zu beachten, daß die "Schreibmarke links" Taste ebenfalls einen Pfeil nach links (←) als Tastenbezeichnung hat, daß damit aber nicht zurückgesetzt werden kann.
CTRL-D	CTRL-D	Signalisiert das Ende der Eingabe von der Tastatur; bewirkt auch das Verlassen der aktuellen Shell.
CTRL-H	CTRL-H	Löscht das erste Zeichen links von der Schreibmarke. Diese Taste heißt auch die ERASE-Taste.
CTRL-Q	CTRL-Q	Startet den Ausdruck wieder, nachdem er mit CTRL-S angehalten wurde.
CTRL-S	CTRL-S	Hält den Ausdruck auf dem Bildschirm an (stoppt aber nicht das Programm).
CTRL-U	CTRL-U	Löscht alle Zeichen der aktuellen Zeile. Diese Taste heißt auch die KILL-Taste.
CTRL-\	CTRL-\	Bricht das aktuelle Kommando ab und erzeugt eine Datei mit Speicherabzug, wenn dies zulässig ist (Wird nur zur Ablaufverfolgung empfohlen).
ESCAPE	ESC	Sondercode für einige Programme. Wechselt z.B. vom Einfügemodus in den Kommandomodus des vi(1) Text-Editors.
RETURN	(Pfeil nach links unten oder ENTER)	Beendet eine Kommandozeile und stößt eine Aktion der Shell an.
F <i>n</i>	F <i>n</i>	Funktionstaste <i>n</i> . F1-F12 werden ohne SHIFT betätigt, F13-F24 sind F1-F12 zusammen mit SHIFT, F25-F36 sind F1-F12 zusammen mit CTRL, und F37-F48 sind F1-F12 zusammen mit CTRL und SHIFT. Die Tasten F49-F60 befinden sich auf dem Nummernblock (ohne SHIFT): F49 - '7' F50 - '8' F51 - '9' F52 - '-' F53 - '4' F54 - '5' F55 - '6' F56 - '+' F57 - '1' F58 - '2' F59 - '3' F60 - '0'

Tastaturabbildung

Die Abbildungsstruktur für die Tastatur ist in `/usr/include/sys/kd.h` definiert. Jede Taste kann zehn verschiedene Zustände haben. Die ersten acht Zustände sind:

- BASE - CTRL-SHIFT
- SHIFT - ALT-SHIFT
- CTRL - ALT-CTRL
- ALT - ALT-CTRL-SHIFT

Die beiden verbleibenden Zustände bezeichnen zwei Sonderbytes. Das erste Byte beschreibt einen besonderen Zustand. Die Bits darin zeigen an, ob die Taste in einem oder mehreren der ersten acht Zustände als spezielle Taste gekennzeichnet wurde. Das zweite Byte enthält einen der vier möglichen Codes C, N, B oder O, die anzeigen, in welcher Weise die Sperrtasten sich auf die betreffende Taste auswirken.

Die folgende Tabelle beschreibt die Vorgabewerte für die Tastaturabbildung. Alle Werte mit Ausnahme besonderer Schlüsselwörter (die weiter hinten beschrieben werden) sind ASCII Zeichenwerte.

Überschrift	Beschreibung
SCAN CODE	Diese Spalte enthält den Scanner-Code, der von der Tastatur-Hardware beim Drücken einer Taste erzeugt wird. Für das Freigeben einer Taste gibt es keinen Scanner-Code in der Tabelle.
BASE	Diese Spalte enthält den Normalwert einer gedrückten Taste.
SHIFT	Diese Spalte enthält den Wert einer gedrückten Taste, wenn die Taste SHIFT ebenfalls gedrückt wird.
LOCK	Diese Spalte bezeichnet die Sperrtasten, die sich auf die betreffende Taste auswirken: - C bedeutet CAPSLOCK - N bedeutet NUMLOCK - B bedeutet beide - O bedeutet Sperre ausgeschaltet

Die übrigen Spalten enthalten die Werte von Tasten, die gleichzeitig mit einer Kombination der CTRL, ALT und SHIFT Tasten gedrückt werden.

Der Spalteneintrag `SRQTAB` in dieser Tabelle stellt einen einfachen Index von der vorgegebenen Tastenauswahl des virtuellen Terminals zum Scan Code dar, der für diese Tastenauswahl angegeben wurde. Die aktuelle `SRQTAB` Tabelle ist eine in sich abgeschlossene Tabelle, die über den `KDGKBENT` und den `KDSKBENT` `ioctl`-Aufruf gelesen oder geschrieben werden kann.

keyboard (7)

(WX200)

keyboard (7)

SCAN CODE	BASE	SHIFT	CTRL	CTRL SHIFT	ALT	ALT SHIFT	ALT CTRL	ALT CTRL SHIFT	LOCK	SRQTAB
1	esc	esc	esc	esc	esc	esc	esc	esc	O	nop
2	'1'	'!'	'1'	'1'	escn	escn	nop	nop	O	nop
3	'2'	'@'	'2'	nul	escn	escn	nop	nop	O	nop
4	'3'	'#'	'3'	'3'	escn	escn	nop	nop	O	nop
5	'4'	'\$'	'4'	'4'	escn	escn	nop	nop	O	nop
6	'5'	'%'	'5'	'5'	escn	escn	nop	nop	O	nop
7	'6'	'^'	'6'	rs	escn	escn	nop	nop	O	nop
8	'7'	'&'	'7'	'7'	escn	escn	nop	nop	O	nop
9	'8'	'*'	'8'	'8'	escn	escn	nop	nop	O	nop
10	'9'	'('	'9'	'9'	escn	escn	nop	nop	O	nop
11	'0'	')'	'0'	'0'	escn	escn	nop	nop	O	nop
12	'_'	'-'	'_'	ns	escn	escn	nop	nop	O	nop
13	'='	'+'	'='	'='	escn	escn	nop	nop	O	nop
14	bs	bs	bs	bs	bs	bs	bs	bs	O	nop
15	ht	btabs	ht	btabs	ht	btabs	ht	btabs	O	nop
16	'q'	'Q'	dc1	dc1	escn	escn	nop	nop	C	nop
17	'w'	'W'	etb	etb	escn	escn	nop	nop	C	nop
18	'e'	'E'	enq	enq	escn	escn	nop	nop	C	nop
19	'r'	'R'	dc2	dc2	escn	escn	nop	nop	C	nop
20	't'	'T'	dc4	dc4	escn	escn	nop	nop	C	nop
21	'y'	'Y'	em	em	escn	escn	nop	nop	C	nop
22	'u'	'U'	nak	nak	escn	escn	nop	nop	C	nop
23	'i'	'I'	ht	ht	escn	escn	nop	nop	C	nop
24	'o'	'O'	si	si	escn	escn	nop	nop	C	nop
25	'p'	'P'	dle	dle	escn	escn	nop	nop	C	K_PREV
26	'l'	'{'	esc	nop	escn	escn	nop	nop	O	nop
27	'j'	'}'	gs	nop	escn	escn	nop	nop	O	nop
28	cr	cr	cr	cr	cr	cr	cr	cr	O	nop
29	lctrl	lctrl	lctrl	lctrl	lctrl	lctrl	lctrl	lctrl	O	nop
30	'a'	'A'	soh	soh	escn	escn	nop	nop	C	nop
31	's'	'S'	dc3	dc3	escn	escn	nop	nop	C	nop
32	'd'	'D'	eot	eot	escn	escn	nop	nop	C	nop
33	'f'	'F'	ack	ack	escn	escn	nop	nop	C	K_FRCNEXT
34	'g'	'G'	bel	bel	escn	escn	nop	nop	C	nop
35	'h'	'H'	bs	bs	escn	escn	nop	nop	C	K_VTF
36	'j'	'J'	nl	nl	escn	escn	nop	nop	C	nop
37	'k'	'K'	vt	vt	escn	escn	nop	nop	C	nop
38	'l'	'L'	np	np	escn	escn	nop	nop	C	nop
39	'/'	'/'	'/'	'/'	escn	escn	nop	nop	O	nop
40	'''	'''	'''	'''	escn	escn	nop	nop	O	nop
41	'''	'''	'''	'''	escn	escn	nop	nop	O	nop

keyboard (7)

(WX200)

keyboard (7)

SCAN CODE	BASE	SHIFT	CTRL	CTRL		ALT		ALT		ALT	
				SHIFT	ALT	SHIFT	ALT	SHIFT	CTRL	SHIFT	CTRL
42	lshift	lshift	lshift	lshift	lshift	lshift	lshift	lshift	lshift	lshift	lshift
43	\'\'	l'	fs	l'	escn	escn	escn	escn	escn	escn	escn
44	'z'	'Z'	sub	sub	escn	escn	escn	escn	escn	escn	escn
45	'x'	'X'	can	can	escn	escn	escn	escn	escn	escn	escn
46	'c'	'C'	etx	etx	escn	escn	escn	escn	escn	escn	escn
47	'v'	'V'	syn	syn	escn	escn	escn	escn	escn	escn	escn
48	'b'	'B'	stx	stx	escn	escn	escn	escn	escn	escn	escn
49	'n'	'N'	so	so	escn	escn	escn	escn	escn	escn	escn
50	'm'	'M'	cr	cr	escn	escn	escn	escn	escn	escn	escn
51	'<'	'<'	'<'	'<'	escn	escn	escn	escn	escn	escn	escn
52	'>'	'>'	'>'	'>'	escn	escn	escn	escn	escn	escn	escn
53	'/'	'/'	'/'	'/'	escn	escn	escn	escn	escn	escn	escn
54	rshift	rshift	rshift	rshift	rshift	rshift	rshift	rshift	rshift	rshift	rshift
55	'**'	'**'	'**'	'**'	escn	escn	escn	escn	escn	escn	escn
56	halt	halt	halt	halt	halt	halt	halt	halt	halt	halt	halt
57	'\'	'\'	nul	nul	escn	escn	escn	escn	escn	escn	escn
58	clock	clock	clock	clock	clock	clock	clock	clock	clock	clock	clock
59	fkey1	fkey13	fkey25	fkey37	fkey1	fkey13	fkey25	fkey37	fkey1	fkey13	fkey25
60	fkey2	fkey14	fkey26	fkey38	fkey2	fkey14	fkey26	fkey38	fkey2	fkey14	fkey26
61	fkey3	fkey15	fkey27	fkey39	fkey3	fkey15	fkey27	fkey39	fkey3	fkey15	fkey27
62	fkey4	fkey16	fkey28	fkey40	fkey4	fkey16	fkey28	fkey40	fkey4	fkey16	fkey28
63	fkey5	fkey17	fkey29	fkey41	fkey5	fkey17	fkey29	fkey41	fkey5	fkey17	fkey29
64	fkey6	fkey18	fkey30	fkey42	fkey6	fkey18	fkey30	fkey42	fkey6	fkey18	fkey30
65	fkey7	fkey19	fkey31	fkey43	fkey7	fkey19	fkey31	fkey43	fkey7	fkey19	fkey31
66	fkey8	fkey20	fkey32	fkey44	fkey8	fkey20	fkey32	fkey44	fkey8	fkey20	fkey32
67	fkey9	fkey21	fkey33	fkey45	fkey9	fkey21	fkey33	fkey45	fkey9	fkey21	fkey33
68	fkey10	fkey22	fkey34	fkey46	fkey10	fkey22	fkey34	fkey46	fkey10	fkey22	fkey34
69	nlock	nlock	nlock	nlock	nlock	nlock	nlock	nlock	nlock	nlock	nlock
70	stlock	stlock	brk	brk	stlock	stlock	brk	brk	stlock	stlock	brk
71	fkey49	'7'	fkey49	'7'	fkey49	escn	escn	escn	escn	escn	escn
72	fkey50	'8'	fkey50	'8'	fkey50	escn	escn	escn	escn	escn	escn
73	fkey51	'9'	fkey51	'9'	fkey51	escn	escn	escn	escn	escn	escn
74	fkey52	'.'	fkey52	'.'	fkey52	escn	escn	escn	escn	escn	escn
75	fkey53	'4'	fkey53	'4'	fkey53	escn	escn	escn	escn	escn	escn
76	fkey54	'5'	fkey54	'5'	fkey54	escn	escn	escn	escn	escn	escn
77	fkey55	'6'	fkey55	'6'	fkey55	escn	escn	escn	escn	escn	escn
78	fkey56	'+'	fkey56	'+'	fkey56	escn	escn	escn	escn	escn	escn
79	fkey57	'1'	fkey57	'1'	fkey57	escn	escn	escn	escn	escn	escn
80	fkey58	'2'	fkey58	'2'	fkey58	escn	escn	escn	escn	escn	escn
81	fkey59	'3'	fkey59	'3'	fkey59	escn	escn	escn	escn	escn	escn
82	fkey60	'0'	fkey60	'0'	fkey60	escn	escn	escn	escn	escn	escn
83	del	' '	del	' '	del	escn	escn	escn	escn	escn	escn

04/92

7-65

keyboard (7)

(WX200)

keyboard (7)

SCAN CODE	BASE	SHIFT	CTRL	CTRL SHIFT	ALT	ALT SHIFT	ALT CTRL	ALT CTRL SHIFT	LOCK	SRQTAB
84	fkey60	fkey26	fkey60	nop	sysreq	sysreq	sysreq	sysreq	O	
85	fkey58	fkey58	fkey58	fkey58	fkey58	fkey58	fkey58	fkey58	O	
86	fkey53	fkey53	fkey53	fkey53	fkey53	fkey53	fkey53	fkey53	O	
87	fkey11	fkey23	fkey35	fkey47	fkey11	fkey23	fkey35	fkey47	O	K_VTF+11
88	fkey12	fkey24	fkey36	fkey48	fkey12	fkey24	fkey36	fkey48	O	K_VTF+12
89	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
90	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
91	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
92	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
93	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
94	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
95	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
96	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
97	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
98	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
99	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
100	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
101	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
102	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
103	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
104	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
105	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
106	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
107	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
108	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
109	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
110	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
111	fkey51	fkey51	nop	nop	nop	nop	nop	nop	O	K_NOP
112	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
113	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
114	ralt	ralt	ralt	ralt	ralt	ralt	ralt	ralt	O	K_NOP
115	rctrl	rctrl	rctrl	rctrl	rctrl	rctrl	rctrl	rctrl	O	K_NOP
116	cr	cr	cr	cr	cr	cr	cr	cr	O	K_NOP
117	'/'	'/'	nop	nop	escn	escn	nop	nop	O	K_NOP
118	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
119	brk	brk	brk	brk	brk	brk	brk	brk	O	K_NOP
120	fkey50	fkey50	nop	nop	nop	nop	nop	nop	O	K_NOP
121	del	del	del	del	del	del	del	del	O	K_NOP
122	fkey57	fkey57	nop	nop	nop	nop	nop	nop	O	K_NOP
123	fkey60	fkey60	nop	nop	nop	nop	nop	nop	O	K_NOP

SCAN CODE	BASE	SHIFT	CTRL	CTRL		ALT		ALT CTRL		LOCK	SRQTAB
				SHIFT	ALT	SHIFT	CTRL	SHIFT	CTRL		
124	nop	nop	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
125	fkey55	fkey55	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
126	fkey59	fkey59	nop	nop	nop	nop	nop	nop	nop	O	K_NOP
127	fkey49	fkey49	nop	nop	nop	nop	nop	nop	nop	O	K_NOP

In der folgenden Tabelle sind die Werte der speziellen Schlüsselwörter aufgeführt, die in der vorhergehenden Tabelle benutzt wurden. Diese Schlüsselwörter wurden in der vorhergehenden Tabelle nur aus Gründen der Lesbarkeit benutzt, in der tatsächlichen Tastatur- Umsetztabelle wird ein spezielles Schlüsselwort anhand seines Wertes dargestellt und gleichzeitig das entsprechende Bit für besonderen Zustand gesetzt.

Name	Wert	Bedeutung
nop	0	Keine Operation - keine Aktion durch Tastendruck
lshift	2	verschieben nach links
rshift	3	verschieben nach rechts
clock	4	Caps lock (nur Großbuchstaben)
nlock	5	Numeric lock (nur numerische Werte)
slock	6	Scroll lock (Scrollen ein-/aus)
alt	7	ALT-Taste
btabs	8	Rücksetztaste - generiert eine feste Sequenz (ESC[Z)
ctrl	9	Kontrolltaste
lalt	10	linke ALT-Taste
ralt	11	rechte ALT-Taste
lctrl	12	linke Kontrolltaste
rctrl	13	rechte Kontrolltaste
fkey1	27	Funktionstaste #1
.	.	.
.	.	.
.	.	.
fkey96	122	Funktionstaste #96
sysreq	123	Systemanforderung
brk	124	Unterbrechungstaste
escn	125	Generieren einer ESC N x Folge, wobei x der Wert des Scan Codes ohne ALT-Taste ist
esco	126	Generieren einer ESC O x Folge, wobei x der Wert des Scan Codes ohne ALT-Taste ist
escl	127	Generieren einer ESC L x Folge, wobei x der Wert des Scan Codes ohne ALT-Taste ist
rboot	128	System neu laden
debug	129	Ablaufverfolger des Systemkerns aufrufen
NEXT	130	Weiterschalten zum nächsten virtuellen Terminal in der Warteschlange
PREV	131	Zurückschalten zum vorigen virtuellen Terminal in der Warteschlange

Name	Wert	Bedeutung
FNEXT	132	unbedingtes Weiterschalten zum nächsten virtuellen Terminal in der Warteschlange
FPREV	133	unbedingtes Zurückschalten zum vorigen virtuellen Terminal in der Warteschlange
VTF	134	Erstes virtuelles Terminal (VT00)
.	.	.
.	.	.
.	.	.
VTL	148	Letztes virtuelles Terminal (VT14)
MGRF	149	Erste Sequenz für VTM. Ermöglicht es, den Tastenfolgen für besondere Aktionen der Fensterverwaltung für virtuelle Terminals eine besondere Bedeutung zuzuweisen. Wird in der SRQTAB Tabelle benutzt.
.	.	.
.	.	.
.	.	.
MGRL	179	Letzte Sequenz für VTM. Wird in der SRQTAB Tabelle benutzt.

In der folgenden Tabelle sind die Namen und Dezimalwerte für die ASCII Zeichen aus der vorhergehenden Tabelle aufgeführt. Anstelle von numerischen Konstanten werden Namen verwendet, um das Lesen der Scanner-Code-Tabelle zu erleichtern. Im `ioctl` Puffer werden nur die Dezimalwerte eingetragen. Diese Werte werden aus `ascii(5)` entnommen.

Name	Wert	Name	Wert
nul	0	dc1	17
soh	1	dc2	18
stx	2	dc3	19
etx	3	dc4	20
eot	4	nak	21
enq	5	syn	22
ack	6	etb	23
bel	7	can	24
bs	8	em	25
ht	9	sub	26
nl	10	esc	27
vt	11	fs	28
np	12	gs	29
cr	13	rs	30
so	14	ns	31
si	15	del	127
dle	16		

Belegung der Funktionstasten

Die Umsetztabelle für Zeichenfolgen ist ein Bereich von 512 Byte Länge (typedef `strmap_t`). Dieser Bereich enthält Zeichenfolgen mit dem Endekennzeichen Null, die Funktionstasten redefinieren. Die erste mit Null abgeschlossene Zeichenfolge wird der ersten Funktionstaste zugeordnet, die zweite Zeichenfolge der zweiten Funktionstaste usw.

Für die Länge einer einzelnen Zeichenfolge gibt es keine Einschränkungen, solange die gesamte Tabelle einschließlich der Nullen 512 Bytes nicht überschreitet. Um eine Zeichenfolge auf Null zu setzen, werden zusätzliche Null-Zeichen angefügt. Die folgende Tabelle enthält die Vorgabewerte für Funktionstasten.

Vorgabewerte für Funktionstasten				
Funktions- taste #	Funktion	Shift Funktion	Ctrl Funktion	Ctrl Shift Funktion
1	ESC OP	ESC Op	ESC OP	ESC Op
2	ESC OQ	ESC Oq	ESC OQ	ESC Oq
3	ESC OR	ESC Or	ESC OR	ESC Or
4	ESC OS	ESC Os	ESC OS	ESC Os
5	ESC OT	ESC Ot	ESC OT	ESC Ot
6	ESC OU	ESC Ou	ESC OU	ESC Ou
7	ESC OV	ESC Ov	ESC OV	ESC Ov
8	ESC OW	ESC Ow	ESC OW	ESC Ow
9	ESC OX	ESC Ox	ESC OX	ESC Ox
10	ESC OY	ESC Oy	ESC OY	ESC Oy
11	ESC OZ	ESC Oz	ESC OZ	ESC Oz
12	ESC OA	ESC Oa	ESC OA	ESC Oa

ioctl-Aufrufe:

KDGBMODE

Mit diesem Aufruf wird der aktuelle Tastaturmodus festgestellt. Er meldet einen der folgenden Werte zurück, die in `/usr/include/sys/kd.h` definiert sind:

```
#define K_RAW          0x00    /* Scan-Code für Zeile senden */
#define K_XLATE        0x01    /* Umsetzung in ASCII */
```

KDSKBMODE

Dieser Aufruf setzt den aktuellen Tastaturmodus. Das Argument für diesen Aufruf ist entweder `K_RAW` oder `K_XLATE`. Wenn zeichenorientiert gearbeitet wird, kann das Programm die zeichenweisen up/down Scanner-Codes von der Tastatur ablesen. Im Übersetzungsmodus werden die Übersetzungstabellen benutzt, um den entsprechenden Zeichencode zu generieren.

KDKBTYPE

Mit diesem Aufruf wird der Tastaturtyp festgestellt. Er meldet einen der folgenden Werte zurück, die in `/usr/include/sys/kd.h` definiert sind:

```
#defineKB_84          0x00    /* Tastatur mit 84 Tasten */
#defineKB_101         0x01    /* Tastatur mit 101 Tasten */
#defineKB_OTHER       0x03    /* Anderer Tastaturtyp */
```

KDGKBENT

Dieser Aufruf liest einen der Einträge aus der Übersetzungstabelle. Das Argument für den Aufruf ist die Adresse einer der folgenden Strukturen, die in `/usr/include/sys/kd.h` definiert sind. Dabei sind die ersten beiden Felder versorgt.

```
struct kbentry {
    unchar kb_table;    /* zu verwendende Tabelle */
    unchar kb_index;    /* Eintrag in Tabelle */
    ushort kb_value;    /* zu lesender/setzender Wert */
};
```

Zulässige Werte für das Feld `kb_table` sind:

```
#defineK_NORMTAB      0x00    /* BASE */
#defineK_SHIFTTAB     0x01    /* SHIFT */
#defineK_ALTTAB       0x02    /* ALT */
#defineK_ALTSHIFTTAB  0x03    /* SHIFT ALT */
#defineK_SQRTAB       0x04    /* SRTAB */
```

Der `ioctl`-Aufruf holt den angegebenen Eintrag aus der Übersetzungstabelle und gibt ihn im dritten Feld zurück.

Der `K_SQRTAB` Wert für das Feld `kb_table` erlaubt den Zugriff auf die mit dem Scan-Code indizierte Tabelle. Dadurch wird die Zuweisung eines bestimmten virtuellen Terminal-Selektors (`K_VTF`–`K_VTL`) oder die "specialkey" Zuweisung der Fensterverwaltung für virtuelle Terminals (`K_MGRF`–`K_MGRL`) möglich.

Der virtuelle Terminal-Selektor (`K_VTF`) ist normalerweise mit `/dev/tty00` verknüpft, auf dem sich üblicherweise die Login-Shell befindet. Die folgenden Terminal-Selektoren werden ebenfalls zur Auswahl von virtuellen Terminals verwendet:

```
K_VTF+1 für das erste virtuelle Terminal (/dev/vt01)
K_VTF+2 für das zweite virtuelle Terminal (/dev/vt02)
:
:
K_VTF+12 für das 12. virtuelle Terminal (/dev/vt12)
```

KDSKBENT

Dieser Aufruf setzt einen Eintrag in einer der Übersetzungstabellen. Er verwendet die gleiche Struktur wie der `KDGKBENT ioctl`, wobei das dritte Feld mit dem Wert versorgt wird, der in die Übersetzungstabelle eingetragen werden soll. Dies kann dazu benutzt werden, um die Tastatur teilweise oder ganz umzuadressieren.

Der `kd` Treiber sieht die Unterstützung von virtuellen Terminals vor. Das Konsolengerät `/dev/vtmon` sieht Tasten-Anforderungen des virtuellen Terminals vom `kd` Treiber aus an den Prozeß vor, der `/dev/vtmon` geöffnet hat. Zwei `ioctl` Aufrufe sind zur Unterstützung virtueller Terminals vorgesehen:

VT_GETSTATE

Der `VT_GETSTATE` `ioctl`-Aufruf meldet globale Statusinformation über das virtuelle Terminal zurück. Er trägt das aktive virtuelle Terminal im Feld `v_active` und die Nummer des aktiven virtuellen Terminals sowie eine Bit-Maske des globalen Zustands im Feld `vt_state` ein. Dabei bezeichnet das Bit `x` den Status von `vt x` (1 zeigt an, daß das virtuelle Terminal geöffnet ist).

VT_SENDSIG

Der `VT_SENDSIG` `ioctl`-Aufruf bezeichnet ein Signal (in `vt_signal`), das an eine Bit-Maske für virtuelle Terminals (in `vt_state`) geschickt werden soll.

Die von den `VT_GETSTATE` und `VT_SENDSIG` `ioctl`-Aufrufen verwendete Datenstruktur ist:

```
struct vt_stat {
    ushort v_active; /* aktives virt. Terminal */
    ushort v_signal; /* zu sendendes Signal (VT_SENDSIG) */
    ushort v_state; /* Bit-Maske (VT_SENDSIG und VT_GETSTATE) */
};
```

und ist in `/usr/include/sys/vt.h` definiert.

VT_OPENQRY

Der `VT_OPENQRY` `ioctl` wird verwendet, um das nächste zur Verfügung stehende virtuelle Terminal festzustellen. Dieser Wert wird im letzten Argument des `ioctl` (2) Aufrufs abgesetzt.

GIO_KEYMAP

Dieser Aufruf holt die gesamte Abbildungstabelle für die Tastatur vom Systemkern. Die Struktur des Arguments ist in `/usr/include/sys/kd.h` angegeben.

PIO_KEYMAP

Dieser Aufruf setzt die gesamte Abbildungstabelle für die Tastatur. Die Struktur des Arguments ist in `/usr/include/sys/kd.h` angegeben.

GIO_STRMAP

Dieser Aufruf holt die Abbildungstabelle für Zeichenfolgen vom Systemkern. Die Struktur des Arguments ist in `/usr/include/sys/kd.h` angegeben.

PIO_STRMAP

Dieser Aufruf setzt die Abbildungstabelle für Zeichenfolgen. Die Struktur des Arguments ist in `/usr/include/sys/kd.h` angegeben.

TIOCKBOF

Der Modus für erweiterten Zeichensatz wird deaktiviert. Dies ist der vorgegebene Modus.

TIOCKBON

Mit diesem Aufruf werden erweiterte Zeichen für die Übermittlung an das Benutzerprogramm zugelassen. Die erweiterten Zeichen werden in Form eines Null-Bytes, gefolgt von einem zweiten Byte, übertragen, das den erweiterten Code des Zeichens enthält. Wenn ein echtes Null-Byte übertragen werden soll, wird es in Form von zwei aufeinanderfolgenden Null-Bytes abgeschickt.

Wenn die Tastatur komplett aktiviert ist, kann ein 8-Bit Zeichencode dadurch erreicht werden, daß die ALT-Taste gedrückt und gleichzeitig über den Nummernblock der dreistellige Dezimalwert des Zeichens eingegeben wird. Das Zeichen wird übertragen, sobald Sie die ALT-Taste loslassen.

Einige Zeichen der Tastatur haben eine spezielle Bedeutung. Voreingestellt ist, daß durch Drücken der DELETE Taste ein Unterbrechungssignal erzeugt wird, das an alle Prozesse geschickt wird, denen dieses Terminal als kontrollierendes Terminal zugeordnet ist. Wenn die Tastatur komplett aktiviert ist, bewirkt das Herunterdrücken der ALT- Taste und das gleichzeitige Drücken der Taste 8 auf der Home-Tastatur (nicht auf dem Nummernblock), daß ein Nullbyte, gefolgt von 0x7F, zurückgegeben wird. Dies hat die gleiche Wirkung wie die DELETE Taste (0x7F), es sei denn, es wurde ein `stty(1)` Kommando mit der `-isig` Option ausgeführt.

KBENABLED

Wenn die Tastatur komplett aktiviert ist (TIOCKBON), wird ein Wert ungleich Null zurückgegeben. Falls die Tastatur nicht komplett aktiviert ist (TIOCKBOF), wird der Wert Null zurückgemeldet.

GETFKEY

Stellt die aktuelle Definition einer Funktionstaste zur Verfügung. Das Argument für diesen Aufruf ist die Adresse einer der folgenden Strukturen, die in `/usr/include/sys/kd.h` definiert sind:

```
struct fkeyarg {
    unsigned int  keynum;
    char  keydef [MAXFK]; /* mittels comcrt.h aus ioctl.h */
    char  flen;
};
```

Die Funktionstaste muß in *keynum* übergeben werden (siehe *arg* Struktur weiter oben). Nach Ausführung des `ioctl` wird die der Taste derzeit zugeordnete Tastenfolge in *keydef* und die Länge der Folge in *flen* zurückgegeben,

SETFKEY

Ordnet einer gegebenen Tastenfolge eine Funktionstaste zu. Der Aufruf verwendet die gleiche Struktur wie der GETFKEY `ioctl`. Die Nummer der Funktionstaste muß in *keynum*, die Tastenfolge in *keydef* und die Länge der Tastenfolge (Anzahl der Zeichen) in *flen* übergeben werden.

DATEIEN

```
/dev/console
/dev/vt00-n
/usr/include/sys/kd.h
```

SIEHE AUCH

stty(1), console(7), display(7), termio(7).

ioctl(2), ascii(5) im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

lad – SCSI Treiber für Worm-Geräte

BESCHREIBUNG

Der Treiber `lad` unterstützt das optische Plattenlaufwerk LMS 510 Ser. 100. Andere optische Plattenlaufwerke, die der SCSI Spezifikation genügen, sollten mit dem Treiber arbeiten können. Die Kommandos `set/get mode` verhalten sich jedoch nicht auf allen Laufwerken gleich. Optische Platten (Worm-Platten) werden mit einer festen Blockgröße vorformatiert (z.B. 512 Bytes für LMS). Das Laufwerk weist fehlerhaften Blöcken automatisch Reserveblöcke zu. Falls zu wenig Reserveblöcke vorhanden sind, ist der Rest des Datenträgers unbrauchbar.

Schreibvorgänge werden immer überprüft (entweder von der Hardware oder vom Treiber), was eine zusätzliche Umdrehung des Datenträgers erfordert. Daher ist der Durchsatz beim Schreiben nur etwa halb so groß wie beim Lesen.

Wie bei allen vorformatierten Datenträgern kann ein Anwendungsprogramm nur jeweils ganze Blöcke lesen bzw. schreiben. Die Blockgröße hängt vom verwendeten Laufwerk ab. Beim LMS beträgt sie z.B. 512 Bytes. Dienstprogramme wie `tar(1)`, `dd(1)`, `cpio(1)` müssen die gleiche Blockgröße benutzen.

Die Gerätenummer (minor device number) ist so aufgebaut:

- Bits 0-3
sind für eine Partitionsnummer reserviert. Sie werden nicht benutzt.
- Bits 4-6
beschreiben eine logische Einheit (logical unit, LUN) und werden nicht benutzt. Sie sind deshalb bei LMS immer 0.
- Bits 7-9
bezeichnen die SCSI-Targetadresse.
- Bits 10-12
bezeichnen die Host Adapter- (ADP32 Controller- bzw. AHA1542B-) Nummer.

ioctl-Aufrufe

Die verfügbaren `ioctl`-Aufrufe und Datenstrukturen sind in `sys/ladioc1.h` definiert.

DATEIEN

Die Gerätedateien haben folgende Namen:

`/dev/rdisk/ladYtZsO` für die Raw-Schnittstelle,

`/dev/disk/ladYtZsO` für die Block-Schnittstelle.

Dabei steht Y für eine Controller-Nummer, Z für eine SCSI-Targetadresse und O für eine Option (beim LMS immer 0). Für die MX300 und die MX500 gilt: Eine 1 für Y bezeichnet den ersten ADP32 Controller (`adp32.0`), eine 2 den zweiten ADP32 Controller (`adp32.1`) usw. Bei der WX200 bezeichnet eine 1 für Y den ersten AHA1542B Controller (`stos.0`), eine 2 den zweiten AHA1542B Controller (`stos.1`) usw.

Die Gerädateien (Geräteknoten, device nodes) werden beim Hochfahren des Systems erzeugt, nachdem ein neuer Kern generiert wurde (z.B. bei der Installation). Die jeweiligen Geräte müssen angeschaltet und mit dem System verbunden sein, damit die Geräteknoten korrekt erzeugt werden können.

SIEHE AUCH

adp32(7), aha(7)

FEHLERMELDUNGEN

Der Treiber gibt folgende Fehlermeldungen aus:

ladYtZ: *error text*

Dabei bezeichnet Y die Nummer des Controllers, Z die SCSI-Targetnummer und *error text* den Fehler.

BEZEICHNUNG

lo – Softwaremäßige Loopback-Netzwerk-Schnittstelle

ÜBERBLICK

```
d = open ("/dev/loop", O_RDWR);
```

BESCHREIBUNG

Das loopback-Gerät ist eine softwaremäßige Verbindungsvorrichtung (datalink provider), die alle empfangenen Pakete an ihre Quelle zurückschickt, ohne daß irgendein Hardware-Gerät beteiligt wird. Es handelt sich dabei um ein STREAMS-Gerät, das der Schnittstelle für Verbindungsvorrichtungen (DLPI) entspricht. Die allgemeine Beschreibung für Netzwerk-Schnittstellen siehe unter if(7).

Die loopback-Schnittstelle wird dazu verwendet, auf die Internet-Dienste des lokalen Rechners zuzugreifen. Da sie auf allen Rechnern einschließlich solcher ohne hardwaremäßiger Netzwerk-Schnittstelle zur Verfügung steht, garantiert sie allen Programmen den Zugriff auf die lokalen Server. Eine typische Anwendung ist der comsat(1M), der Mitteilungen für Posteingang von einem lokalen Client annimmt. Die loopback-Schnittstelle wird außerdem für die Leistungsanalyse und für Testzwecke verwendet.

Der Konvention entsprechend heißt die loopback-Schnittstelle lo0 und wird mit der Internet-Adresse 127.0.0.1. konfiguriert. Diese Adresse kann mit dem SIOCSIFADDR ioctl() verändert werden.

SIEHE AUCH

comsat(1M), if(7), inet(7).

BEZEICHNUNG

lp – Schnittstelle für parallele Anschlüsse

BESCHREIBUNG

Der Treiber lp für parallele Anschlüsse unterstützt gleichzeitig sowohl den primären (monochromen) als auch den sekundären Adapter für parallele Anschlüsse. Bis zu zwei Drucker werden unterstützt. Falls der Adapter für einen Drucker nicht angeschlossen ist, mißlingt ein Versuch, ihn mit open zu öffnen. Ein close Aufruf durch den Benutzer wartet solange, bis die Ausgabe vollständig beendet ist, bevor er zurückkehrt. Der lp-Treiber erlaubt immer nur einem Prozeß, auf den Adapter zu schreiben. Ist er gerade in Betrieb, gibt ein open-Aufruf einen Fehler zurück. Dagegen läßt der Treiber mehrere open-Operationen für Lesezugriffe gleichzeitig zu.

Die Adapter für parallele Drucker sind zeichenorientierte Geräte. Die Gerätenummer entspricht dem primären oder sekundären Adapter für parallele Drucker. So entspricht die Gerätenummer 0 dem primären und die Gerätenummer 1 dem sekundären Adapter für parallele Drucker.

Die parallelen Anschlüsse verhalten sich wie in termio (7) beschrieben.

DATEIEN

/dev/lp*

SIEHE AUCH

stty(1), termio(7).

ioctl(2) im *Referenzhandbuch für Programmierer*.

BEZEICHNUNG

mbad – MB-I Adapter

BESCHREIBUNG

Der Treiber mbad unterstützt bis zu drei MB-I Adapter (MBAD, Multibus-I Adapter). Jeder MB-I Adapter steuert ein MB-I Gehäuse über ein MB-I Interface (MBIF). Controller werden in die MB-I Gehäuse eingesteckt.

Adressen-Zuweisungen für MX500-90:

Controller	Nr.	Verbindung	I/O-Adresse	Speicher-Adr.	Interr.	Maps
XYL472	0	XT0	300		7	34
Storager	0		3000		2	47
NCR-ADP 32	0	NCR0 S.E.	6000		4	36
NCR-ADP 32	0	NCR3 DIFF	6000		4	36
NCR-ADP 32	1	NCR1 S.E.	6100		4	68
NCR-ADP 32	1	NCR2 DIFF	6100		4	68
NCR-ADP 32	2	NCR2 S.E.	6200		4	68
NCR-ADP 32	2	NCR1 DIFF	6200		4	68
NCR-ADP 32	3	NCR3 S.E.	6300		4	68
NCR-ADP 32	3	NCR0 DIFF	6300		4	68
EXOS	0	CL0	2000		3	67
EXOS	1	CL1	2500		3	67
D419-V1	0	CW0	1800	2 0000	5	
D419-V1	1	CW1	1900	3 0000	4	
D419-V1	2	CW2	2100	4 b000	5	
D419-V1	3	CW3	2200	5 b000	4	
D419-V1	4	CW4	2300	6 b000	6	
D419-V1	5	CW5	2400	7 b000	7	
D364-Vx(16)	0	SR0	4000	8 0000		
D364-Vx(16)	1	SR1	4100	8 4000		
D364-Vx(16)	2	SR2	4200	8 8000		
D364-Vx(16)	3	SR3	4300	8 c000		
D364-Vx(16)	4	SR4	4400	9 0000		
D364-Vx(16)	5	SR5	4500	9 4000		
D364-Vx(16)	6	SR6	4600	9 8000		
D364-Vx(16)	7	SR7	4700	9 c000		
D364-Vx(16)	8	SR8	4800	a 0000		
D364-Vx(16)	9	SR9	4900	a 4000		
D364-Vx(16)	10	SR10	4a00	a 8000		
D364-Vx(16)	11	SR11	4b00	a c000		
D364-Vx(16)	12	SR12	4c00	b 0000		
D364-Vx(16)	13	SR13	4d00	b 4000		
D364-Vx(16)	14	SR14	4e00	b 8000		
D364-Vx(16)	15	SR15	4f00	b c000		

Controller	Nr.	Verbindung	I/O-Adresse	Speicher-Adr.	Interr.	Maps
D501-V4(16)	0	SX0	4000	8 0000		
D501-V4(16)	1	SX1	4100	8 4000		
D501-V4(16)	2	SX2	4200	8 8000		
D501-V4(16)	3	SX3	4300	8 c000		
D501-V4(16)	4	SX4	4400	9 0000		
D501-V4(16)	5	SX5	4500	9 4000		
D501-V4(16)	6	SX6	4600	9 8000		
D501-V4(16)	7	SX7	4700	9 c000		
D501-V4(16)	8	SX8	4800	a 0000		
D501-V4(16)	9	SX9	4900	a 4000		
D501-V4(16)	10	SX10	4a00	a 8000		
D501-V4(16)	11	SX11	4b00	a c000		
D501-V4(16)	12	SX12	4c00	b 0000		
D501-V4(16)	13	SX13	4d00	b 4000		
D501-V4(16)	14	SX14	4e00	b 8000		
D501-V4(16)	15	SX15	4f00	b c000		
D695	0	SI0	5000	8 0000	5	18*
D695	1	SI1	5100	8 4000	5	18*
D695	2	SI2	5200	8 8000	5	18*
D695	3	SI3	5300	8 c000	5	18*
D695	4	SI4	5400	9 0000	5	18*
D695	5	SI5	5500	9 4000	5	18*
D695	6	SI6	5600	9 8000	5	18*
D695	7	SI7	5700	9 c000	5	18*
D695	8	SI8	5800	a 0000	5	18*
D695	9	SI9	5900	a 4000	5	18*
D695	10	SI10	5a00	a 8000	5	18*
D695	11	SI11	5b00	a c000	5	18*
D695	12	SI12	5c00	b 0000	5	18*
D695	13	SI13	5d00	b 4000	5	18*
D695	14	SI14	5e00	b 8000	5	18*
D695	15	SI15	5f00	b c000	5	18*

* Zwei von den 18 sind zum TAK-Laden reserviert.

D364(16) steht für den Terminal-Multiplexer von SIEMENS-NIXDORF. D501(16) steht für den SIEMENS-NIXDORF Terminal Multiplexer mit Modem-Unterstützung. D419 steht für den SIEMENS-NIXDORF Communication Controller. STOR steht für den Interphase Storager III D. NCR steht für den NCR SCSI Host Adapter (ADP32 Controller). EXOS steht für den EXCELAN EXOS 201 Ethernet Controller. D695 oder SIM steht für den SIEMENS-NIXDORF Serial Interface Multiplexer.

In allen Fällen müssen die Möglichkeiten so kombiniert werden, daß an keinem der drei MB-I ein Adressenkonflikt entsteht und nicht mehr als die verfügbaren 256 Map-Register pro MB-I benutzt werden.

mbad (7)

(MX500)

mbad (7)

EBENE

Multiprozessor-Erweiterung

SIEHE AUCH

mdevice(4), sdevice(4), acc(7), adp32(7), exa(7), fd(7), hd(7), lad(7), shd(7), si(7),
sr(7), xmt(7).

BEZEICHNUNG

mem, kmem – Hauptspeicher

ÜBERSICHT

/dev/mem

/dev/kmem

BESCHREIBUNG

Die Datei /dev/mem ist eine Gerätedatei, die ein Abbild des Speichers des Rechners darstellt. Sie kann z.B. verwendet werden, um das System zu überprüfen und eventuell sogar durch Patchen zu ändern.

Byte-Adressen in /dev/mem werden wie Speicheradressen interpretiert. Verweise auf nicht vorhandene Adressen bewirken die Rückmeldung eines Fehlers.

Falls Geräteregister abgefragt und gepatcht werden, führt dies vermutlich zu unerwarteten Ergebnissen, wenn Bits für Lese- oder Schreibschutz vorhanden sind.

Die Datei /dev/kmem und die Datei /dev/mem sind gleich, außer daß im einen Fall auf den virtuellen Speicher des Systemkerns zugegriffen wird, im andern Fall auf den physischen Speicher.

DATEIEN

/dev/mem

/dev/kmem

HINWEIS

Einige /dev/kmem-Dateien können wegen lesegeschützter Adressen oder nicht bestückter Speicheradressen nicht gelesen werden.

null(7)

null(7)

BEZEICHNUNG

null – Null-Datei

ÜBERSICHT

/dev/null

BESCHREIBUNG

Daten, die in die Null-Geräte-datei /dev/null geschrieben werden, werden ignoriert.

Lese-Operationen in einer Null-Geräte-datei geben stets 0 Bytes zurück.

DATEIEN

/dev/null

BEZEICHNUNG

prf – Auslastung des Betriebssystems

BESCHREIBUNG

Die Gerätedatei `/dev/prf` ermöglicht den Zugriff auf Informationen über die Aktivität des Betriebssystems. Das Schreiben der Datei bewirkt, daß die Meßeinrichtung mit den zu überwachenden Text-Adressen geladen wird. Beim Lesen der Datei werden diese Adressen sowie eine Reihe von Zählern zurückgegeben, die die Aktivitäten zwischen nebeneinanderliegenden Adressen anzeigen.

Der Aufzeichnungsmechanismus wird durch den System-Takt gesteuert und testet den Programmzähler bei jeder Zeile. Operationen, die das Betriebssystem betreffen, werden mit den gespeicherten Textadressen verglichen und entsprechende Zähler zur späteren Verarbeitung hochgesetzt.

Die Datei `/dev/prf` ist ein Pseudo-Gerät ohne entsprechende Hardware.

DATEIEN

`/dev/prf`

SIEHE AUCH

`profiler(1M)`.

BEZEICHNUNG

rtc – Schnittstelle für die Echtzeit-Uhr

BESCHREIBUNG

Der rtc-Treiber unterstützt das Chip für die Echtzeit-Uhr. Er ermöglicht das Setzen der korrekten lokalen Zeit und das Ablesen der Zeit von diesem Chip.

ioctl-Aufrufe

RTCRTIME

Dieser Aufruf wird benutzt, um die lokale Zeit von dem Chip für die Echtzeit-Uhr zu lesen. Das Argument für `ioctl` ist die Adresse eines Puffers `RTCNREG` vom Typ "unsigned char" (`RTCNREG` ist in `<sys/rtc.h>` definiert). Der `ioctl` speichert den Inhalt des Chip-Registers in dem Puffer ab. Derzeit hat `RTCNREG` den Wert 14 und die Byte-Register haben folgende Bedeutung:

Register	Contents
0	Sekunden
1	Sekunden-Signal
2	Minuten
3	Minuten-Signal
4	Stunden
5	Stunden-Signal
6	Wochentag
7	Tag des Monats
8	Monat
9	Jahr
A	Zustandsregister A
B	Zustandsregister B
C	Zustandsregister C
D	Zustandsregister D

Weitere Informationen über die Funktionen dieser Register finden Sie in Ihrem Technischen Referenzhandbuch zur Hardware.

RTCSTIME

Dieser Aufruf wird benutzt, um die Zeit in dem Chip für die Echtzeit-Uhr einzusetzen. Das Argument für `ioctl` ist die Adresse eines Puffers mit `RTCNREGP` vom Typ "unsigned char" (`RTCNREGP` ist in `<sys/rtc.h>` definiert). In diesen Bytes sollten die gewünschten Inhalte der Chip-Register stehen. Derzeit hat `RTCNREGP` den Wert 10 und stellt die Register 0–9 wie oben abgebildet dar. Es ist zu beachten, daß nur der Superuser das Gerät für die Echtzeit-Uhr für Schreibzugriffe öffnen darf und daß der `RTCSTIME` `ioctl` für jeden anderen Benutzer zu Fehlern führt.

DATEIEN

/dev/rtc

BEZEICHNUNG

shd – Treiber für SCSI-Festplatten am NCR Host Adapter

BESCHREIBUNG

Der Treiber `shd` unterstützt SCSI-Festplatten wie z.B. Micropolis am NCR Host Adapter (ADP32 Controller). Er unterstützt bis zu 7 Targetgeräte pro Host Adapter und bis zu 8 logische Untereinheiten (LUN, logical unit). Die tatsächliche Anzahl der verwendbaren Festplatten hängt vom `shd`-Eintrag in der Konfigurationsdatei `mdevice` und den verfügbaren Host Adaptern ab. Der Treiber bestimmt die Geometrie der Festplatte dynamisch (siehe unten). Für die einzelnen Slices (Slice 1 bis 15) bietet der Treiber jeweils eine Raw-Schnittstelle und eine Block-Schnittstelle an. Auf die gesamte Partition kann über die Raw-Schnittstelle von Slice 0 zugegriffen werden.

Die Nummer des Geräts (minor device number), auf das zugegriffen wird, legt fest, wie das entsprechende Laufwerk behandelt wird. Die Gerätenummer (minor device number) ist so aufgebaut:

- Bits 0-3 kennzeichnen die Slice (0-15).
- Bits 4-6 sind für logische Untereinheiten (LUN) reserviert und werden nicht benutzt. Sie sind immer 0.
- Bits 7-9 geben die SCSI-Targetnummer an einem SCSI-Bus an.
- Bits 10-12 wählen einen der Host Adapter aus.

Slice 0 steht für die komplette UNIX-Systempartition, wie sie in der `fdisk`-Tabelle definiert ist. Die weiteren Slices werden durch Angaben definiert, die im Inhaltsverzeichnis des Datenträgers (VTOC: volume table of contents) stehen. Wenn auf Slice 0 zugegriffen wird, werden die Grenzen der anderen Slices ignoriert und es findet keine Abbildung defekter Blöcke auf Ersatzblöcke statt. Bei einem solcher Zugriff auf die Festplatte ist daher Vorsicht geboten.

Die ganze Festplatte wird auf zwei Ebenen partitioniert: Erstens sind Plattenbereiche, die von einem anderen Betriebssystem benutzt werden sollen, in der `fdisk`-Tabelle beschrieben. Diese Tabelle liegt im ersten Block der Festplatte. Zweitens werden die UNIX-Bereiche der Festplatte weiter aufgeteilt. Dazu wird die Information aus dem VTOC verwendet, das in einem beliebigen Block enthalten sein kann. Zur Zeit befindet sich das VTOC im Sektor 0 der aktiven UNIX-Partition. Das VTOC enthält auch Informationen über andere Betriebssysteme, die in der `fdisk`-Tabelle beschrieben sind. Wenn die zu der Festplatte gehörige Gerätedatei geöffnet wird, liest der Treiber das VTOC und benutzt es, um seine Tabellen über logische Platten (logische Laufwerke) zu füllen. Für die Zuweisung verwendet der Treiber die Gerätenummer (minor device number).

Jeder Partition wird in der `fdisk`-Tabelle ein Typ zugeordnet (z.B. DOS, UNIX o.a.). Das UNIX-System kann eine Partition (Dateisystem) nur dann nutzen, wenn sie vom richtigen Typ ist. Eine DOS-Partition kann z.B. vom UNIX-System nicht benutzt werden außer im Raw-Modus ohne Dateisystem (raw, non-file system device).

Zylinder 0 ist auf jedem Laufwerk reserviert für Laufwerkscharakteristika und die `fdisk`-Partitionstabelle; beide liegen im physikalischen Festplattensektor 0 mit der Länge 512 Bytes. Weiter ist der Zylinder 0 reserviert für eine Volume- und Dateikennung, die beide im physikalischen Festplattensektor 1 mit der Länge 512

Bytes zu finden sind; und dann liegt im Zylinder 0 noch der Bootbereich für das Hochfahren des Systems. Die Startadresse des Bootbereichs wird durch ein Byte in der Dateikennung angezeigt: `flb.f_boe[4]` enthält den Wert in Kilobytes (normalerweise 4 KB). Zur Zeit ist es nicht möglich, von einer shd SCSI-Festplatte an einem NCR Host Adapter zu booten.

Die `fdisk`-Tabelle beschreibt die Partitionierung der Festplatte wie bei DOS. Es sind höchstens `FD_NUMPARTS` (= 4) Partitionen möglich. Normalerweise existiert nur eine Partition für UNIX - die aktive UNIX Partition -, die die ganze Festplatte mit Ausnahme von Zylinder 0 belegt.

Defekte Sektoren werden vom SCSI-Controller ausgeblendet. Da diese Abbildung auf Hardware-Ebene stattfindet, bleibt die Geometrie der Festplatte unverändert. Für die Software besteht die Festplatte aus einer Folge intakter Blöcke.

Auf der aktiven UNIX Partition befinden sich im Sektor `PDINFO SEC` (= Sektor 0 der Partition) die Datenstrukturen `pdinfo` und `vtoc`. `pdinfo` (physical disk info) beschreibt die physikalischen Parameter der Festplatte, während `vtoc` (volume table of contents) die einzelnen UNIX-Slices innerhalb der gesamten UNIX-Partition beschreibt. Die UNIX-Slice 0 ist die "Backup-Slice", sie enthält die komplette UNIX-Partition.

Um fehlerhafte Sektoren auf Ersatzsektoren abzubilden, steht auf der MX500 `addbad` zur Verfügung.

ioctl-Aufrufe

Die verfügbaren `ioctl`-Aufrufe und Datenstrukturen sind in `sys/shdioc1.h` und `sys/vtoc.h` definiert.

Partitionen

In der `fdisk`-Tabelle können Partitionen an Zylindergrenzen beginnen, im VTOC können Slices jedoch auf Spurgrenzen beginnen. Dies wird für den Boot-Bereich der UNIX Systempartition benutzt, um die erste Spur (mit dem Boot-Code) nicht mit in die Slice aufzunehmen. Die `fdisk`-Tabelle erlaubt höchstens vier Partitionen auf einer Festplatte, mit dem VTOC kann die UNIX-Partition jedoch in bis zu 16 Slices aufgeteilt werden. Jede Slice wird durch die Gerätenummer identifiziert. Einer Slice wird ihre Gerätenummer zugeordnet, wenn zum ersten Mal auf die Platte zugegriffen wird. Die Zuordnung ist in der Datei `/etc/partitions` festgelegt und bleibt so lange unverändert, bis das Dienstprogramm `disksetup` (1M) erneut gestartet wird.

Versuche, ein Dateisystem zu öffnen, schlagen fehl (non-existent device), wenn dafür keine Slice existiert. Ebenso erfolglos sind Versuche, Slices einzuhängen (siehe `mount` (1M)), die kein UNIX-Dateisystem enthalten.

DATEIEN

Die Gerätedateien haben folgende Namen:

```
/dev/dsk/shdYtZsO, ...  
/dev/rdisk/shdYtZsO, ...
```

Dabei steht Y für die Nummer des Controllers und Z für die SCSI-Targetnummer. Eine 1 für Y bezeichnet den ersten ADP32 Controller (adp32.0), eine 2 den zweiten ADP32 Controller (adp32.1) usw. O bezeichnet die Slicenummer. Sie kann bei Geräten, die im Raw-Modus betrieben werden, Werte von 0 bis 15 annehmen. Bei Geräten im Block-Modus sind Slicenummern von 1 bis 15 möglich.

Die Gerätedateien (Geräteknoten, device nodes) werden beim Hochfahren des Systems erzeugt, nachdem ein neuer Kern generiert wurde (z.B. bei der Installation). Die jeweiligen Geräte müssen angeschaltet und mit dem System verbunden sein, damit die Geräteknoten korrekt erzeugt werden können.

SIEHE AUCH

fdisk(1M), disksetup(1M), ioctl(2), fs(4).

FEHLERMELDUNGEN

Je nach Art des Fehlers wiederholt der Treiber fehlgeschlagene Übertragungen bis zu zehnmal. Bei manchen Fehlern findet keine Wiederholung statt. Der Treiber gibt eine Fehlermeldung aus, wenn während der Übertragung ein Fehler auftritt. Vom Treiber gemeldete Fehler haben folgende Form:

shdYtZ: *error text*

Dabei bezeichnet Y die Nummer des Controllers, Z die SCSI-Targetnummer und *error text* den Fehler.

BEZEICHNUNG

shd – Treiber für SCSI-Festplatten am Host Adapter AHA1542B

BESCHREIBUNG

Der Treiber `shd` unterstützt SCSI-Festplatten wie z.B. Micropolis am Host Adapter AHA1542B. Er unterstützt bis zu 7 Targetgeräte pro Host Adapter und bis zu 8 logische Untereinheiten (LUN, logical unit). Die tatsächliche Anzahl der verwendbaren Festplatten hängt vom `shd`-Eintrag in der Konfigurationsdatei `mdevice` und den verfügbaren Host Adapters ab. Der Treiber bestimmt die Geometrie der Festplatte dynamisch (siehe unten). Für die einzelnen Slices (Slice 1 bis 15) bietet der Treiber jeweils eine Raw-Schnittstelle und eine Block-Schnittstelle an. Auf die gesamte Partition kann über die Raw-Schnittstelle von Slice 0 zugegriffen werden.

Die Nummer des Geräts (minor device number), auf das zugegriffen wird, legt fest, wie das entsprechende Laufwerk behandelt wird. Die Gerätenummer (minor device number) ist so aufgebaut:

- Bits 0-3 kennzeichnen die Slice (0-15).
- Bits 4-6 sind für logische Untereinheiten (LUN) reserviert und werden nicht benutzt. Sie sind immer 0.
- Bits 7-9 geben die SCSI-Targetnummer an einem SCSI-Bus an.
- Bits 10-12 wählen einen der Host Adapter aus.

Slice 0 steht für die komplette UNIX-Systempartition, wie sie in der `fdisk`-Tabelle definiert ist. Die weiteren Slices werden durch Angaben definiert, die im Inhaltsverzeichnis des Datenträgers (VTOC: volume table of contents) stehen. Wenn auf Slice 0 zugegriffen wird, werden die Grenzen der anderen Slices ignoriert und es findet keine Abbildung defekter Blöcke auf Ersatzblöcke statt. Bei einem solcher Zugriff auf die Festplatte ist daher Vorsicht geboten.

Die ganze Festplatte wird auf zwei Ebenen partitioniert: Erstens sind Plattenbereiche, die von einem anderen Betriebssystem benutzt werden sollen, in der `fdisk`-Tabelle beschrieben. Diese Tabelle liegt im ersten Block der Festplatte. Zweitens werden die UNIX-Bereiche der Festplatte weiter aufgeteilt. Dazu wird die Information aus dem VTOC verwendet, das in einem beliebigen Block enthalten sein kann. Zur Zeit befindet sich das VTOC im Sektor 0 der aktiven UNIX-Partition. Das VTOC enthält auch Informationen über andere Betriebssysteme, die in der `fdisk`-Tabelle beschrieben sind. Wenn die zu der Festplatte gehörige Gerätedatei geöffnet wird, liest der Treiber das VTOC und benutzt es, um seine Tabellen über logische Platten (logische Laufwerke) zu füllen. Für die Zuweisung verwendet der Treiber die Gerätenummer (minor device number).

Jeder Partition wird in der `fdisk`-Tabelle ein Typ zugeordnet (z.B. DOS, UNIX o.a.). Das UNIX-System kann eine Partition (Dateisystem) nur dann nutzen, wenn sie vom richtigen Typ ist. Eine DOS-Partition kann z.B. vom UNIX-System nicht benutzt werden außer im Raw-Modus ohne Dateisystem (raw, non-file system device).

Zylinder 0 ist auf jedem Laufwerk reserviert für Laufwerkscharakteristika und die `fdisk`-Partitionstabelle; beide liegen im physikalischen Festplattensektor 0 mit der Länge 512 Bytes. Weiter ist der Zylinder 0 reserviert für eine Volume- und Dateikennung, die beide im physikalischen Festplattensektor 1 mit der Länge 512 Bytes zu finden sind; und dann liegt im Zylinder 0 noch der Bootbereich für das

Hochfahren des Systems. Die Startadresse des Bootbereichs wird durch ein Byte in der Dateikennung angezeigt: `flb.f_boe[4]` enthält den Wert in Kilobytes (normalerweise 4 KB).

Die `fdisk`-Tabelle beschreibt die Partitionierung der Festplatte wie bei DOS. Es sind höchstens `FD_NUMPARTS` (= 4) Partitionen möglich. Normalerweise existiert nur eine Partition für UNIX - die aktive UNIX Partition -, die die ganze Festplatte mit Ausnahme von Zylinder 0 belegt.

Defekte Sektoren werden vom SCSI-Controller ausgeblendet. Da diese Abbildung auf Hardware-Ebene stattfindet, bleibt die Geometrie der Festplatte unverändert. Für die Software besteht die Festplatte aus einer Folge intakter Blöcke.

Auf der aktiven UNIX Partition befinden sich im Sektor `PDINFO_SEC` (= Sektor 29 der Partition) die Datenstrukturen `pdinfo` und `vtoc`. `pdinfo` (physical disk info) beschreibt die physikalischen Parameter der Festplatte, während `vtoc` (volume table of contents) die einzelnen UNIX-Slices innerhalb der gesamten UNIX-Partition beschreibt. Die UNIX-Slice 0 ist die "Backup-Slice", sie enthält die komplette UNIX-Partition.

ioctl-Aufrufe

Die verfügbaren `ioctl`-Aufrufe und Datenstrukturen sind in `sys/shdioc1.h` und `sys/vtoc.h` definiert.

Partitionen

In der `fdisk`-Tabelle können Partitionen an Zylindergrenzen beginnen, im VTOC können Slices jedoch auf Spurgrenzen beginnen. Dies wird für den Boot-Bereich der UNIX Systempartition benutzt, um die erste Spur (mit dem Boot-Code) nicht mit in die Slice aufzunehmen. Die `fdisk`-Tabelle erlaubt höchstens vier Partitionen auf einer Festplatte, mit dem VTOC kann die UNIX-Partition jedoch in bis zu 16 Slices aufgeteilt werden. Jede Slice wird durch die Gerätenummer identifiziert. Einer Slice wird ihre Gerätenummer zugeordnet, wenn zum ersten Mal auf die Platte zugegriffen wird. Die Zuordnung ist in der Datei `/etc/partitions` festgelegt und bleibt so lange unverändert, bis das Dienstprogramm `disksetup` (1M) erneut gestartet wird.

Versuche, ein Dateisystem zu öffnen, schlagen fehl (non-existent device), wenn dafür keine Slice existiert. Ebenso erfolglos sind Versuche, Slices einzuhängen (siehe `mount`(1M)), die kein UNIX-Dateisystem enthalten.

DATEIEN

Die Gerätedateien haben folgende Namen:

```
/dev/dsk/cYtZsO, ...  
/dev/rdisk/cYtZsO, ...
```

Dabei steht *Y* für die Nummer des Controllers. Eine 8 bezeichnet den ersten Controller (*stos.0*), eine 9 den zweiten (*stos.1*). *Z* steht für die SCSI-Targetnummer. Mögliche Werte für *Z* sind 0, 1, 2, 3, 4, 5, und 6. *O* bezeichnet die Slicenummer. Sie kann bei Geräten, die im Raw-Modus betrieben werden, Werte von 0 bis 15 annehmen. Bei Geräten im Block-Modus sind Slicenummern von 1 bis 15 möglich.

Die Gerätedateien (Geräteknoten, device nodes) werden beim Hochfahren des Systems erzeugt, nachdem ein neuer Kern generiert wurde (z.B. bei der Installation). Die jeweiligen Geräte müssen angeschaltet und mit dem System verbunden sein, damit die Geräteknoten korrekt erzeugt werden können.

SIEHE AUCH

`fdisk(1M)`, `disksetup(1M)`, `ioctl(2)`, `fs(4)`.

FEHLERMELDUNGEN

Je nach Art des Fehlers wiederholt der Treiber fehlgeschlagene Übertragungen bis zu zehnmal. Bei manchen Fehlern findet keine Wiederholung statt. Der Treiber gibt eine Fehlermeldung aus, wenn während der Übertragung ein Fehler auftritt. Vom Treiber gemeldete Fehler haben folgende Form:

shdYtZ: error text

Dabei bezeichnet *Y* die Nummer des Controllers, *Z* die SCSI-Targetnummer und *error text* den Fehler.

BEZEICHNUNG

si – Multiplexer für die serielle Schnittstelle am MB-I

BESCHREIBUNG

Der Treiber `si` unterstützt bis zu 16 SIM Controller an der MX500 und bis zu 3 an der MX300. Jeder SIM Controller steuert zwei AFP Zweidrahtleitungen. Auf jeder dieser Leitungen können bis zu zwei Terminal-Konzentratoren (TAKs) betrieben werden. An jeden TAK können bis zu vier SS97 Geräte (über die Schnittstellen S0, S1, S2, S3) oder bis zu vier V24 Geräte (über die Schnittstellen V0, V1, V2, V3) angeschlossen werden. Von einem SIM Controller können maximal 16 Geräte gesteuert werden. Die Gerätenummern (minor device numbers) 0,...,15 sprechen SIM0 an, 16,...,31 sprechen SIM1 an, 32,...,47 sprechen SIM2 an usw. .

Zusätzlich zum `si` Treiber gibt es den Administrationstreiber `sv`, der benutzt wird, um die SIM und TAK Controller zu laden, zu konfigurieren und Diagnose-Informationen zu liefern. Die Namen der Gerätedateien sind:

Name der Gerätedatei	minor device number	Verwendung
/dev/sim/smsload<board#>	16*<board#>+0	SIM Ladegerät
/dev/sim/smsdiag<board#>	16*<board#>+1	SIM Diagnosegerät
/dev/sim/smtload<board#>	16*<board#>+2	TAK Ladegerät
/dev/sim/smtdiag<board#>	16*<board#>+3	TAK Diagnosegerät
/dev/sim/smadmin<board#>	16*<baord#>+4	Administrationsgerät

Beim Hochfahren des Systems lädt das Dienstprogramm `simload` Software in den SIM Controller. Der Name des geladenen Programms ist `csimsw`. In der Datei `SIMCONFIG` stehen Informationen über die Konfiguration. Mit dem Dienstprogramm `Tak` wird der TAK geladen, sowie Sende- und Empfangspuffer, Anzahl der HDLC-Wiederholungen usw. ausgegeben. Mit dem Dienstprogramm `simmsg` werden Geräte auf den SIM/TAK Controllern konfiguriert und dekonfiguriert. Alle TAK und SIM Dienstprogramme stehen in den Verzeichnissen `/sbin/sim`. Die Konfigurationsdatei `SIMCONFIG` und das zu ladende Programm `csimsw` stehen in `/etc/sim`.

Auf einer Maschine, bei der die Boards 2 und 8 eingesteckt sind, sieht die Datei `SIM-CONFIG` folgendermaßen aus:

SI2 at /dev/sim

MINORDEV	SIM	LINE	TAK	TAKCHANNEL
032	2	0	1	0
033	2	0	3	0
034	2	1	4	0
035	2	0	1	1
036	2	0	1	2
037	2	0	1	4
038	2	0	3	1
039	2	0	3	4
040	2	0	3	5
041	2	1	4	1
042	2	1	4	4
043	2	1	4	5

SI8 at /dev/sim

MINDEV	SIM	LINE	TAK	TAKCHANNEL
128	8	0	1	1
129	8	0	1	2
130	8	0	1	0
131	8	1	2	0
132	8	1	2	1
133	8	1	2	4
134	8	1	2	5
135	8	0	3	0
136	8	0	3	1
137	8	0	3	4
138	8	0	3	5
139	8	1	4	0
140	8	1	4	1
141	8	1	4	4
142	8	1	4	5

In der Spalte "MINORDEV" steht die Gerätenummer (minor device number), die am SIM Controller "SIM" über die Leitung "LINE" und dort am TAK "TAK" der Schnittstelle "TAKCHANNEL" zugeordnet ist. So bezeichnet 037 die Gerätenummer der Schnittstelle 4 am TAK 1, an der Leitung 0 des SIM Controllers 2. Die Schnittstellen werden von 0 bis 7 angegeben; 0 bezeichnet die erste SS97 Schnittstelle S0, 1 die zweite SS97 Schnittstelle S1, usw.; 4 bezeichnet die erste V24 Schnittstelle V0, 5 die zweite V24 Schnittstelle V1 usw. .

Das Programm `simload` sucht die Datei `SIMCONFIG` und lädt für jeden Eintrag `SI<board#>` at `/dev/sim` den SIM Controller über das Ladegerät `/dev/sim/smsload<board#>`, startet die Software auf dem Board und schickt dem Controller für jede Gerätenummer eine Konfigurationsmeldung.

Beim Hochfahren wird `simload` vom Shellskript `/etc/rc` ohne Parameter aufgerufen. Dadurch werden alle Controller geladen und alle Geräte konfiguriert.

Das Administrationsprogramm ruft `simload` folgendermaßen auf:

```
simload [-s] <board#>
```

Damit wird nur der Controller `<board#>` geladen. Der Parameter `-s` (synchron) schickt die Konfigurationsmeldungen eine nach der anderen und nicht parallel.

ioctl-Aufrufe

Die verfügbaren `ioctl`-Aufrufe und Datenstrukturen sind in `sys/termios.h` und `sys/si.h` definiert.

DATEIEN

```
/dev/term/tty*  
/sbin/sim/simload  
/etc/sim/SIMCONFIG
```

SIEHE AUCH

`signal(2)`, `termios(2)`, `termio(7)`.

BEZEICHNUNG

sr – Serielles Ein/Ausgabeboard für 6 Leitungen am MB-I

BESCHREIBUNG

Der Treiber `sr` unterstützt Controller vom Typ SERAD (D279), SERAG (D312) oder SEAAB (D364Vx). Jeder Controller steuert bis zu sechs V24 oder SS97 Leitungen. Die Controller kommunizieren über "dual ported RAM" mit dem Host. Dieser Speicherbereich heißt auch Mailbox. Die Controller werden im Poll-Modus betrieben, d.h. alle 50 msec wird überprüft, ob auf den am Controller angeschlossenen Leitungen Zeichen eingegangen sind.

Auf die einzelnen Leitungen (Kanäle) wird über

```
/dev/term/tty<nnn> ,
```

zugegriffen. Dabei ist `<nnn>` die Nummer des Kanals. Sie hat den gleichen Wert wie die Gerätenummer (minor device number) und berechnet sich nach der Formel:

$$(\text{Nummer des Boards} \times 6) + (\text{Nummer des Kanals auf dem Board})$$

Mögliche Nummern für ein Board sind 0, 1, ..., `<mm>`; (`<mm>`+1) ist dabei die Maximalzahl der erlaubten Boards. Mögliche Kanalnummern sind 0, 1, ..., 5.

Der `sr` Treiber basiert auf STREAMS. In der Datei

```
/etc/ap/chan.ap
```

gibt es einen Eintrag für `sr`. Er bewirkt, daß bei der Initialisierung des Systems mit den "line discipline"-Modulen `ldterm` und `ttcompat` eine "autopush"-Operation durchgeführt wird.

ioctl-Aufrufe

Siehe *X/Open Portability Guide/XPG3*.

EINSCHRÄNKUNGEN

`sr`, `si` und `sx` Boards benutzen die gleichen Ein-/Ausgabeadressen auf dem MB-I. Solche Controller dürfen nie mit der gleichen Nummer am gleichen MB-I benutzt werden.

Es gibt keine Modem-Unterstützung für RS232-Leitungen. Ebenso wird BREAK nicht unterstützt.

DATEIEN

```
/dev/term/tty*
/etc/ap/chan.ap
```

SIEHE AUCH

`signal(2)`, `termios(2)`, `mbad(7)`, `si(7)`,

BEZEICHNUNG

ssd – Treiber für Festplatten am SCSI-Adapter

BESCHREIBUNG

Der Treiber `ssd` für Festplatten unterstützt bis zu vier SPA Controller mit jeweils zwei SCSI-Kanälen pro Controller. Normalerweise befinden sich zwei Controller auf einem Board. An einem SCSI-Kanal können bis zu sieben SCSI-Geräte (Targets) angeschlossen sein. Der Controller verwaltet keine Untereinheiten (logical subunits). Er kann bis zu 56 Festplattenlaufwerke verwalten. 12 davon befinden sich im Hauptgehäuse und 12 im Erweiterungsgehäuse. Die Eigenschaften des Laufwerks werden beim Hochfahren des Systems mit dem SCSI-Kommando `GETMODES` ermittelt. Der Treiber bestimmt die Geometrie der Festplatte dynamisch (siehe unten). Er bietet Zugriff im Block- und im Raw-Modus auf die einzelnen Partitionen der Festplatte sowie auf die gesamte physikalische Platte.

Die Nummer des Geräts (minor device number), auf das zugegriffen wird, legt fest, wie das entsprechende Laufwerk behandelt wird. Die Gerätenummer (minor device number) ist so aufgebaut:

- Bits 0 - 3
bestimmen die Partition (0 bis 15);
- Bits 4 - 6
sind für logische Untereinheiten (logical subunits) reserviert und werden nicht benutzt;
- Bits 7 - 9
geben die SCSI-Targetgerätnummer am SCSI-Kanal an;
- Bits 10 - 12
wählen einen der SCSI-Kanäle aus.

Partition 0 steht für die komplette UNIX-Systempartition, wie sie in der `fdisk`-Tabelle definiert ist. Die weiteren Partitionen werden durch Informationen im VTOC (volume table of contents) definiert. Wenn auf Partition 0 zugegriffen wird, werden die Grenzen anderer Partitionen ignoriert und defekte Blöcke nicht ersetzt. Bei einem solchen Zugriff ist daher Vorsicht geboten.

Die ganze Festplatte wird auf zwei Ebenen partitioniert: Erstens werden Plattenbereiche, die von diversen Betriebssystemen benutzt werden können, durch die `fdisk`-Tabelle beschrieben, die im ersten Block der Festplatte enthalten ist. Zweitens werden die UNIX-Bereiche der Festplatte weiter partitioniert. Dazu wird die Information aus dem VTOC verwendet, das in einem beliebigen Block enthalten sein kann. Zur Zeit befindet sich das VTOC im Sektor 0 der aktiven UNIX-Partition. Das VTOC enthält auch Informationen über andere Betriebssysteme, die in der `fdisk`-Tabelle beschrieben sind. Wenn die zu der Festplatte gehörige Gerätedatei geöffnet wird, liest der Treiber das VTOC und benutzt es, um seine Tabellen über logische Platten (logische Laufwerke) zu füllen. Für die Zuweisung verwendet der Treiber die Gerätenummer (minor device number).

Jeder Partition wird in der `fdisk`-Tabelle ein Typ zugeordnet (z.B. DOS, UNIX o.a.). Das UNIX-System kann eine Partition (Dateisystem) nur dann nutzen, wenn sie den richtigen Typ hat. Eine DOS-Partition kann z.B. vom UNIX-System nur über die `ioctl`-Aufrufe `RD_ABS` und `WR_ABS` benutzt werden.

Zylinder 0 ist auf jedem Laufwerk reserviert für Laufwerkscharakteristika und die `fdisk`-Partitionstabelle; beide liegen im physikalischen Festplattensektor 0 mit der Länge 512 Bytes. Weiter ist der Zylinder 0 reserviert für eine Volume- und Dateikennung, die beide im physikalischen Festplattensektor 1 mit der Länge 512 Bytes zu finden sind; außerdem liegt im Zylinder 0 der Bootbereich für das Hochfahren des Systems. Die Startadresse des Bootbereichs wird durch ein Byte in der Dateikennung angezeigt: `flb.f_boe[4]` enthält den Wert in Kilobytes (normalerweise 4 KB).

Die `fdisk`-Tabelle beschreibt die Partitionierung der Festplatte wie bei DOS. Es sind höchstens `FD_NUMPARTS` (= 4) Partitionen möglich. Normalerweise existiert nur eine Partition für UNIX - die aktive UNIX Partition -, die die ganze Festplatte beschreibt mit Ausnahme von Zylinder 0 und einigen Zylindern, die für Diagnosezwecke reserviert sind.

Defekte Sektoren werden vom SCSI-Controller ausgeblendet. Da diese Abbildung auf Hardware-Ebene stattfindet, bleibt die Geometrie der Festplatte unverändert. Für die Software besteht die Festplatte aus einer Folge intakter Blöcke.

Auf der aktiven UNIX Partition befinden sich im Sektor `PDINFO_SEC` (= Sektor 0 der UNIX-Partition) die Datenstrukturen `pdinfo` und `vtoc`. `pdinfo` (physical disk info) beschreibt die physikalischen Parameter der Festplatte, während `vtoc` (volume table of contents) die einzelnen UNIX-Partitionen innerhalb des gesamten UNIX-Bereichs beschreibt. Die UNIX-Partition 0 ist die "Backup-Partition", sie enthält die komplette UNIX-Partition. Ein Benutzer mit Systemverwalterrechten kann mit den `ioctl`-Aufrufen `V_RDABS/V_WRABS` auf die gesamte physikalische Platte zugreifen, indem er die UNIX-Partition 0 im Raw-Modus öffnet. Diese Funktionen sollten nur von dem Dienstprogramm `disksetup(1M)` benutzt werden, um neue Festplatten zu installieren.

Versuche, ein Dateisystem zu öffnen, für das keine Partition existiert, schlagen fehl (non-existent device). Ebenso Versuche, mit dem `mount`-Kommando (siehe `mount(1M)`) Partitionen einzuhängen, die kein UNIX-Dateisystem enthalten.

ioctl-Aufrufe

Die verfügbaren `ioctl`-Aufrufe und Datenstrukturen sind in `sys/ssdioctl.h` und `sys/vtoc.h` definiert.

DATEIEN

`/dev/dsk/c0d0s0`, ...
`/dev/rdisk/c0d0s0`,

SIEHE AUCH

`fdisk(1M)`, `disksetup(1M)`, `ioctl(2)`, `fs(4)`.

FEHLERMELDUNGEN

Je nach Art des Fehlers wiederholt der Treiber fehlgeschlagene Übertragungen bis zu zehnmal. Bei manchen Fehlern findet keine Wiederholung statt. Der Treiber gibt eine Fehlermeldung aus, wenn während der Übertragung ein Fehler auftritt. Fehler des Festplattencontrollers werden wie folgt ausgegeben:

`ssd(<channel>, <target>): <SCSI command> error - status <status>, typ <typ>.`

BEZEICHNUNG

sx – Serielles Modem-Ein/Ausgabeboard für 6 Leitungen am MB-I

BESCHREIBUNG

Der Treiber sx unterstützt Controller vom Typ SEAAC-V4 (D501V4). Jeder Controller steuert bis zu sechs RS232 Leitungen mit optionaler Unterstützung eines Modems. Die Controller werden im Poll-Modus betrieben, d.h. alle 50 msec wird überprüft, ob auf den am Controller angeschlossenen Leitungen Zeichen eingegangen sind.

Auf die einzelnen Leitungen (Kanäle) wird über

```
/dev/term/tty5<nn> ,
```

zugegriffen. Dabei ist <nn> die Nummer des Kanals. Bei direkten Verbindungen (ohne Modem) ist die Gerätenummer gleich der Kanalnummer. Bei Modemverbindungen ist die Gerätenummer gleich der Kanalnummer + 100. Die Kanalnummer wird genauso berechnet wie beim Treiber sr.

Der Treiber basiert auf STREAMS. Es gibt in der Datei

```
/etc/ap/chan.ap
```

einen Eintrag für sx. Er bewirkt, daß bei der Initialisierung des Systems mit den "line discipline"-Modulen ldterm und ttcompat eine "autopush"-Operation durchgeführt wird.

ioctl-Aufrufe

Siehe *X/Open Portability Guide/XPG3*.

EINSCHRÄNKUNGEN

sr Boards mit 16 KB Mailbox, si und sx Boards benutzen die gleichen Ein-/Ausgabeadressen am MB-I. Solche Controller dürfen nie mit der gleichen Nummer am gleichen MB-I benutzt werden.

Die Signalleitungen "Calling Indicator" (M3) und "Received Line Signal Detector" (M4) werden nicht ausgewertet.

DATEIEN

```
/dev/term/tty*  
/etc/ap/chan.ap
```

SIEHE AUCH

signal(2), termios(2), sr(7), mbad(7), si(7),

BEZEICHNUNG

sxt – Treiber für virtuelle Geräte

BESCHREIBUNG

Die Gerätedatei `/dev/sxt` ist ein Treiber für virtuelle Geräte, der zwischen standardmäßigen `tty`-Leistungsprotokollen und dem echten Gerätetreiber ein Protokoll einschiebt. Die Standard-Protokolle behandeln virtuelle `tty`-Strukturen (Kanäle), die vom `/dev/sxt`-Treiber angegeben werden. `/dev/sxt` verhält sich dabei wie ein Protokoll zur Behandlung von echten `tty`-Strukturen, die von einem echten Geräte-Treiber angegeben wurden. Derzeit wird der `/dev/sxt`-Treiber nur vom `shl(1)`-Kommando benutzt.

Virtuelle `tty`s werden durch I-Nodes im Unterverzeichnis `/dev/sxt` abgebildet und werden in Achtergruppen reserviert. Um eine solche Gruppe zuzuweisen, muß ein Programm eine Datei exklusiv eröffnen, deren Namen im Format `/dev/sxt/??0` (Kanal 0) angegeben ist; danach muß das Programm einen `SXTIOCLINK ioctl`-Aufruf absetzen, um die multiplexe Verarbeitung einzuleiten.

Nur jeweils ein Kanal, nämlich der kontrollierende Kanal, kann eine Eingabe von der Tastatur empfangen; die Leseversuche anderer Kanäle werden abgeblockt.

Es gibt zwei Gruppen von `ioctl(2)`-Aufrufen, die von `sxt` unterstützt werden. Die erste Gruppe enthält neben den in `termio(7)` beschriebenen Standard-`ioctl`-Kommandos folgende Aufrufe:

<code>TIOCEXCL</code>	Modus für exklusive Benutzung setzen: es werden keine weiteren Open-Operationen zugelassen, bis die Datei geschlossen wird.
<code>TIOCNXCL</code>	Modus für exklusive Benutzung zurücksetzen: weitere Open-Operationen sind wieder zugelassen.

Die zweite Gruppe sind `sxt`-Kommandos. Einige davon können nur über Kanal 0 ausgeführt werden.

`SXTIOCLINK` Kanalgruppe zuweisen und die virtuellen `tty`s auf echte `tty` im Multiplexerverfahren koppeln. Das Argument für diesen Aufruf ist die Nummer der zuzuweisenden Kanäle. Dieses Kommando kann nur über Kanal 0 ausgeführt werden. Fehler, die auftreten können:

<code>EINVAL</code>	Das Argument liegt außerhalb der Grenzwerte.
<code>ENOTTY</code>	Das Kommando wurde nicht von einem echten <code>tty</code> abgesetzt.
<code>ENXIO</code>	<code>linesw</code> ist für <code>sxt</code> nicht konfiguriert.
<code>EBUSY</code>	Für dieses echte <code>tty</code> wurde schon ein <code>SXTIOCLINK</code> -Kommando abgesetzt.
<code>ENOMEM</code>	Es steht kein Systemspeicher für die Zuweisung der virtuellen <code>tty</code> -Strukturen zur Verfügung.
<code>EBADF</code>	Kanal 0 ist nicht vor diesem Aufruf geöffnet worden.

<code>SXTIOCSWTCH</code>	Kontrollierenden Kanal setzen. Fehler, die auftreten können: <code>EINVAL</code> Es wurde eine ungültige Kanal-Nummer angegeben. <code>EPERM</code> Das Kommando wurde nicht von Kanal 0 abgesetzt.
<code>SXTIOCWF</code>	Ein Kanal muß warten, bis er der kontrollierende Kanal ist. Dieses Kommando meldet den Fehler <code>EINVAL</code> zurück, wenn eine ungültige Kanal-Nummer angegeben wurde.
<code>SXTIOCUBLK</code>	<code>loblk</code> -Kontrollmarke im virtuellen tty für den angezeigten Kanal ausschalten. Es wird der Fehler <code>EINVAL</code> zurückgemeldet, wenn eine ungültige Kanal-Nummer oder Kanal 0 angegeben wurde.
<code>SXTIOCSTAT</code>	Zustand eines jeden Kanals feststellen (geblockt bei Ein- oder Ausgabe) und in der <code>sxtblock</code> -Struktur abspeichern, die im Argument angegeben ist. Falls die Struktur nicht geschrieben werden kann, wird der Fehler <code>EFAULT</code> zurückgemeldet.
<code>SXTIOCTRACE</code>	Überwachung einschalten. Die Überwachungs-Information wird auf die Konsole geschrieben. Dieses Kommando hat keine Wirkung, wenn keine Überwachung konfiguriert ist.
<code>SXTIOCNOTRACE</code>	Überwachung ausschalten. Dieses Kommando hat keine Wirkung, wenn keine Überwachung konfiguriert ist.

DATEIEN

`/dev/sxt/??[0-7]` Virtuelle tty-Geräte

SIEHE AUCH

`sh1(1)`, `stty(1)` `ioctl(2)`, `open(2)`, `termio(7)`

BEZEICHNUNG

TCP – Internet-Protokoll für die Übertragungssteuerung

ÜBERBLICK

```
#include <sys/socket.h>
#include <netinet/in.h>

s = socket(AF_INET, SOCK_STREAM, 0);
t = t_open("/dev/tcp", O_RDWR);
```

BESCHREIBUNG

TCP ist das virtuelle Leitungsprotokoll der Internet-Protokollfamilie. Es sorgt für die zuverlässige, flußgesteuerte, geordnete, zweiseitige Übertragung von Daten. Es ist ein Byte-Stream-Protokoll, das über dem Internet-Protokoll (IP) angesiedelt ist, dem Netzwerkprotokoll für Datagramme der Internet-Protokollfamilie.

Programme können auf TCP zugreifen, indem sie die Socket-Schnittstelle als `SOCK_STREAM`-Sockettyp verwenden oder indem sie die Schnittstelle Transport-Level-Interface (TLI) dort benutzen, wo sie die verbindungsorientierten (`T_COTS_ORD`) Service-Typen unterstützt.

TCP verwendet die Adressierung der IP Rechner-Ebene und fügt seine jeweils eigene Sammlung von Port-Adressen an. Die Endpunkte einer TCP Verbindung werden durch eine Kombination von IP Adressen und einer TCP Anschlußnummer bestimmt. Obwohl andere Protokolle wie z.B. das Benutzer-Datagramm-Protokoll (UDP) das gleiche Adreßformat für Rechner und Anschluß benutzen können, ist der Anschlußraum für diese Protokolle anders. Einzelheiten über allgemeine Adressierungsaspekte der Internet-Protokollfamilie finden Sie unter `inet(7)`.

Sockets, die TCP benutzen, sind entweder aktiv oder passiv. Aktive Sockets erzeugen Verbindungen zu passiven Sockets. Für beide Arten von Sockets müssen die zugehörigen IP-Adressen und die TCP-Anschlußnummern mit dem Systemaufruf `bind(2)` gebunden werden, nachdem der Socket erzeugt wurde. Voreingestellt ist, daß TCP-Sockets aktiv sind. Ein passiver Socket wird mit dem Systemaufruf `listen(2)` erzeugt, nachdem der Socket mit `bind(2)` gebunden wurde. Dies legt einen Warteschlangen-Parameter für den passiven Socket fest. Danach können Verbindungen zu dem passiven Socket mit dem Systemaufruf `accept(2)` aufgebaut werden. Aktive Sockets verwenden nach dem Binden den Aufruf `connect(2)`, um Verbindungen einzuleiten.

Durch die Verwendung des speziellen Werts `INADDR_ANY` kann die lokale IP-Adresse im `bind(2)` Aufruf für aktive oder passive TCP-Sockets offengelassen werden. Diese Funktion wird üblicherweise dann benutzt, wenn die lokale Adresse entweder unbekannt oder irrelevant ist. Wenn die Adresse nicht angegeben ist, wird die lokale IP-Adresse zum Zeitpunkt des Verbindungsaufbaus an die Adresse der Netzwerk-Schnittstelle gebunden, die in dieser Verbindung verwendet wird.

Wenn eine Verbindung einmal aufgebaut ist, können Daten mit den Systemaufrufen `read(2)` und `write(2)` übertragen werden.

TCP unterstützt eine Socket-Option, die mit `setsockopt()` gesetzt und mit `getsockopt(2)` getestet wird. In den meisten Fällen schickt TCP Daten zum Zeitpunkt ihres Auftretens ab. Wenn anstehende Daten noch nicht bestätigt worden sind, faßt TCP kleine Mengen von Ausgabedaten zusammen, die in einem einzigen Paket

gesendet werden, sobald eine Bestätigung eingetroffen ist. Für eine kleine Anzahl von Clienten wie z.B. Fenster-Systeme, die einen Datenstrom von Maus-Ereignissen senden, für die keine Antwort erfolgt, kann diese Paketbildung erhebliche Verzögerungen bewirken. Daher sieht TCP die boolische Option `TCP_NODELAY` vor (definiert in `/usr/include/netinet/tcp.h`), um diesen Algorithmus aufzuheben. Die Options-Ebene für den `setsockopt()`-Aufruf ist die Protokollnummer für TCP, die mit `getprotobyname()` zur Verfügung gestellt wird (siehe `getprotoent(3N)`).

Optionen auf der IP-Ebene können von TCP verwendet werden; siehe `ip(7)`.

TCP stellt einen Mechanismus für eilige Daten bereit, der unter Verwendung der out-of-band Vorrichtung von `send(2)` aufgerufen werden kann. Der aufrufende Prozeß kann ein Byte mit dem `MSG_OOB`-Schalter als eilig für den `send(2)`-Aufruf kennzeichnen. Dies setzt einen sogenannten eiligen Zeiger, der auf dieses Byte im TCP-Datenstrom verweist. Der Empfänger auf der anderen Seite des Datenstroms wird über ein `SIGURG`-Signal von den eiligen Daten in Kenntnis gesetzt. Die `SIOCATMARK` `ioctl()`-Anforderung meldet einen Wert zurück, der anzeigt, ob der Datenstrom die Eil-Marke erreicht hat. Da das System niemals Daten über die Eil-Marke hinweg mit einem einzigen `read(2)`-Aufruf zurückgibt, ist es möglich, auf die eiligen Daten in einer einfachen Schleife vorzusetzen. Diese Schleife liest Daten ein und testet den Socket mit der `SIOCATMARK` `ioctl()`-Anforderung, bis die Marke erreicht ist.

Ankommende Verbindungsanforderungen, die eine IP-Routing Option beinhalten, werden notiert und auf dem gleichen Weg an die Quelladresse zurückgesandt.

Eine Prüfsumme für alle Daten unterstützt die Zuverlässigkeit von TCP. TCP verwendet einen fensterorientierten Mechanismus für die Flußsteuerung, der positive Rückmeldungen, Folgenummern und eine Rückübertragungs-Strategie benutzt. So kann normalerweise eine Wiederherstellung durchgeführt werden, wenn Datagramme zerstört sind, verzögert oder doppelt auftreten oder wenn Datagramme vom zugrundeliegenden Kommunikationsmedium außer der Reihe geliefert werden.

Wenn das lokale TCP über längere Zeit keine Bestätigung von seinem Peer erhält, wie das z.B. beim Zusammenbruch des fernen Rechners der Fall sein kann, wird die Verbindung abgebrochen und dem Benutzer ein Fehler gemeldet. Wenn ein ferner Rechner neu lädt oder auf andere Weise Status-Informationen über eine TCP-Verbindung verliert, wird die Verbindung abgebrochen und dem Benutzer ein Fehler gemeldet.

SIEHE AUCH

`read(2)`, `write(2)`, `accept(3N)`, `bind(3N)`, `connect(3N)`, `getprotoent(3N)`, `getsockopt(3N)`, `listen(3N)`, `send(3N)`, `inet(7)`, `ip(7)`.

Postel, Jon, *Transmission Control Protocol - DARPA Internet Program Protocol Specification*, RFC 793, Network Information Center, SRI International, Menlo Park, Calif., September 1981.

FEHLERMELDUNGEN

Eine Socket-Operation kann zu folgenden Fehlern führen:

EISCONN	Es wurde versucht, eine <code>connect()</code> -Operation für einen Socket auszuführen, für den bereits eine <code>connect()</code> -Operation durchgeführt war.
ETIMEDOUT	Eine Verbindung wurde wegen exzessiver Rückübertragung aufgegeben.
ECONNRESET	Der ferne Peer erzwang einen Abbruch der Verbindung (üblicherweise, weil aufgrund eines Systemzusammenbruchs Statusinformation über die Verbindung verloren gegangen ist).
ECONNREFUSED	Der ferne Peer ergriff selbst Maßnahmen zur Verhinderung des Verbindungsaufbaus (üblicherweise, weil kein Prozeß am Anschluß horcht).
EADDRINUSE	Es wurde versucht, eine <code>bind()</code> -Operation für einen Socket mit einem Netzwerk-Adreß/Anschluß-Paar durchzuführen, das schon an einen anderen Socket gebunden ist.
EADDRNOTAVAIL	Es wurde versucht, eine <code>bind()</code> -Operation für einen Socket mit einer Netzwerk-Adresse auszuführen, für die keine Netzwerk-Schnittstelle existiert.
EACCES	Es wurde versucht, eine <code>bind()</code> -Operation mit einer reservierten Anschlußnummer durchzuführen, die effektive Benutzernummer des Prozesses bezeichnete aber keinen privilegierten Benutzer.
ENOBUFS	Der Speicherplatz des Systems für interne Datenstrukturen reicht nicht aus.

BEZEICHNUNG

termio – Allgemeines Terminal-Interface

ÜBERSICHT

```
#include <termio.h>

ioctl(int fildes, int request, struct termio *arg);
ioctl(int fildes, int request, int arg);

#include <termios.h>

ioctl(int fildes, int request, struct termios *arg);
```

BESCHREIBUNG

System V unterstützt eine allgemeine, hardware-unabhängige Schnittstelle zu asynchronen Kommunikationskanälen. Diese Schnittstelle besteht vor allem aus Funktionsaufrufen, die in `termios(2)` beschrieben sind, und aus `ioctl` Kommandos, die hier beschrieben sind. Hier werden ebenfalls die Eigenschaften des Terminalsystems besprochen, die beiden Schnittstellen gemeinsam sind.

Beim Öffnen einer Terminaldatei wird der Prozeß angehalten, bis die Verbindung aufgebaut ist. In der Praxis öffnen Benutzerprogramme nur selten Terminals. Meistens werden sie vom System geöffnet und dem Benutzer als Standardeingabe-, -ausgabe- und -fehlerdatei zur Verfügung gestellt. Das als erstes geöffnete Terminal einer Sitzung, das keiner anderen zugeordnet ist, wird deren Kontrollterminal. Auf seine spezielle Rolle bei der Behandlung von Quit- und Interrupt-Signalen wird später eingegangen. Dieses Kontrollterminal wird während eines `fork(2)` an den Kindprozeß vererbt. Ein Prozeß kann diese Zuordnung beenden, indem er seine Sitzung durch `setsid(2)` ändert.

Ein diesen Dateien zugeordnetes Terminal arbeitet normalerweise im Voll-Duplex-Modus. Zeichen können jederzeit, auch während einer Ausgabe, eingegeben werden und gehen nur in den seltensten Fällen verloren. Möglich ist ein Verlust von Zeichen, wenn die Eingabepuffer des Systems überlaufen; dazu kommt es z.B. wenn die Zahl der Zeichen im Zeilenpuffer der Eingabeprozedur größer wird als `MAX_CANON` und `IMAXBEL` (siehe weiter unten) nicht gesetzt ist. Oder Zeichen gehen verloren, wenn der Benutzer `MAX_INPUT` Eingabezeichen angesammelt hat, die noch von keinem Programm gelesen wurden. Nach dem Erreichen der Eingabegrenzen werden alle bislang eingetroffenen Zeichen ohne weiteren Hinweis weggeworfen.

Sitzungsmanagement (Auftragssteuerung)

Ein Kontrollterminal betrachtet eine der Prozessgruppen, die zur Sitzung gehören, als Vordergrund-Prozeßgruppe. Alle weiteren Prozeßgruppen der Sitzung werden als Hintergrund-Prozeßgruppen behandelt. Die Vordergrund-Prozeßgruppe spielt eine spezielle Rolle bei der Behandlung von Eingabezeichen, die Signale auslösen (siehe unten). Standardmäßig wird beim Allokieren eines Kontrollterminals die Prozeßgruppe des Kontrollprozesses als Vordergrund-Prozeßgruppe eingesetzt.

Hintergrund-Prozeßgruppen, die zur Sitzung des Kontrollprozesses gehören und die versuchen auf ihr Kontrollterminal zuzugreifen, müssen sich an die Übertragungsprozedur (line discipline) der Auftragssteuerung halten. Ist für eine Prozeßgruppe nichts anderes vorgegeben, kann sie durch Signale unterbrochen werden. Nur Mitglieder von verwaisten Prozeßgruppen werden anders behandelt.

Da kein Mitglied einer solchen Prozeßgruppe einen Vater in einer anderen Prozeßgruppe derselben Sitzung hat, teilen es auch mit niemanden dasselbe Kontrollterminal. Greift ein Mitglied einer verwaisten Prozeßgruppe auf das Kontrollterminal zu, könnte nach einem Abbruch niemand mit der Bearbeitung fortfahren, deshalb wird hier ein Fehler zurückgegeben.

Wenn ein Mitglied einer Hintergrund-Prozeßgruppe versucht, von seinem Kontrollterminal zu lesen, wird seiner Prozeßgruppe ein SIGTTIN-Signal gesendet. Normalerweise werden dadurch die Mitglieder dieser Prozeßgruppe angehalten. Falls jedoch ein Prozeß SIGTTIN ignoriert, dieses Signal selbst behandelt oder zu einer verwaisten Prozeßgruppe gehört, mißlingt das Lesen, `errno` wird auf `EIO` gesetzt und kein Signal wird gesendet.

Wenn ein Mitglied einer Hintergrund-Prozeßgruppe versucht, auf sein Kontrollterminal zu schreiben, obwohl das `TOSTOP`-Bit im `c_lflag`-Feld gesetzt ist, wird seiner Prozeßgruppe ein SIGTTOU-Signal gesendet. Normalerweise werden dadurch die Mitglieder dieser Prozeßgruppe angehalten. Falls ein Prozeß SIGTTOU ignoriert oder selbst behandelt, wird das Schreiben durchgeführt. Bei einem verwaisten Prozeß, der SIGTTOU weder ignoriert noch selbst behandelt, wird nicht geschrieben, sondern `errno` wird auf `EIO` gesetzt und kein Signal wird gesendet.

Versucht ein Mitglied einer Hintergrund-Prozeßgruppe bei gesetztem `TOSTOP` auf sein Kontrollterminal über `ioctl` zuzugreifen und damit die Kontrollparameter zu modifizieren (z.B. mit `TCSETA`, `TCSETAW`, `TCSETAF`, oder `TIOCSPPGRP`), wird seiner Prozeßgruppe ein SIGTTOU-Signal geschickt. Normalerweise werden dadurch die Mitglieder dieser Prozeßgruppe angehalten. Falls ein Prozeß SIGTTOU ignoriert oder selbst behandelt, wird der Vorgang durchgeführt. Bei einem verwaisten Prozeß, der SIGTTOU weder ignoriert noch selbst behandelt, wird nichts ausgeführt, `errno` auf `EIO` gesetzt und kein Signal gesendet.

Eingabeverarbeitung im kanonischen Modus

Normalerweise wird die Terminaleingabe zeilenweise verarbeitet. Eine Zeile wird begrenzt durch ein Newline-Zeichen (ASCII LF), ein Dateiende-Zeichen (ASCII EOT) oder ein Zeilenende-Zeichen. Das bedeutet, daß ein Programm, das zu lesen versucht, warten muß, bis eine ganze Zeile eingegeben wurde. Ebenso wird maximal eine einzige Zeile ans Programm übergeben, unabhängig davon, wieviele Zeichen der Lesevorgang angefordert hat. Es ist jedoch nicht nötig, jedesmal eine ganze Zeile auf einmal zu lesen. Vielmehr kann jede beliebige Anzahl Zeichen angefordert werden, auch ein einzelnes, ohne daß Information verloren geht.

Zeichen und Zeilen werden wie üblich gelöscht. Das ERASE-Zeichen (per Voreinstellung das Zeichen #) löscht das zuletzt eingegebene Zeichen. Das WERASE-Zeichen (CTRL-W) löscht das zuletzt eingegebene Wort der aktuellen Zeile, jedoch nicht vorausgehende Leerzeichen und Tabulatoren. Ein "Wort" ist definiert als Folge nicht leerer Zeichen, wobei Tabulatoren als Leerzeichen gerechnet werden. Weder ERASE noch WERASE löschen über den Anfang der aktuellen Zeile hinaus. Das KILL-Zeichen (per Voreinstellung das Zeichen @) löscht die gesamte Eingabezeile und gibt optional ein Newline-Zeichen aus. All diese Zeichen arbeiten gemäß der tatsächlichen Tastatureingabe, ohne Rückwärtsschritte oder Tabulatoren weiter zu berücksichtigen. Das REPRINT-Zeichen (CTRL-R) gibt ein Newline-Zeichen, gefolgt von allen noch nicht gelesenen Zeichen, aus. Dies geschieht auch automatisch, wenn Zeichen, die normalerweise wieder vom Schirm gelöscht

werden, durch Programmausgaben versehentlich überschrieben worden sind. Diese Zeichen werden wiedergegeben, als würden sie geecho't. Ist ECHO also nicht gesetzt, werden sie nicht nochmals ausgegeben.

Die Zeichen ERASE und KILL können direkt durch ein Voranstellen des Escape-Zeichens (`\`) eingegeben werden. Das Escape-Zeichen wird dabei überlesen. Die Zeichen für Erase wie auch für Kill können geändert werden.

Eingabeverarbeitung im nicht-kanonischen Modus

Bei der Eingabeverarbeitung im nicht-kanonischen Modus werden Eingabezeichen nicht zeilenweise aufgesammelt. Auch gibt es keine Löschfunktionen. Die Werte `MIN` und `TIME` bestimmen die Verarbeitung der empfangenen Zeichen.

`MIN` steht für die minimale Anzahl Zeichen, die empfangen werden sollen, damit das Lesen als erfolgreich gilt (d.h. die Zeichen an den Benutzer zurückgegeben werden). `TIME` definiert einen Timer, der bis auf 0.10 Sekunden genau, schnell eintreffende Daten kurzer Länge kontrollieren soll.

Vier Einstellungen von `MIN` und `TIME` sind möglich:

Fall A: `MIN > 0`, `TIME > 0`

Hier definiert `TIME` einen Timer zwischen den einzelnen Zeichen. Er wird jeweils nach Eintreffen eines Zeichen aufgezogen, erstmals also nachdem das erste Zeichen angekommen ist. Treffen dann `MIN` Zeichen ein, ohne daß der Timer abläuft, war das Lesen erfolgreich. Man beachte dabei, daß der Timer nach jedem einzelnen Zeichen neu aufgezogen wird. Läuft der Timer ab, bevor `MIN` Zeichen eingetroffen sind, werden nur die bis dahin eingetroffenen Zeichen an den Anwender zurückgegeben. Es muß also mindestens ein Zeichen eingetroffen sein, weil der Timer erst nach dem ersten Zeichen erstmals aufgezogen wird. Ist `MIN > 0` und `TIME > 0`, schläft der Lesevorgang, bis der `MIN` und `TIME` Mechanismus durch ein erstes eintreffendes Zeichen aktiviert wird. Wenn die Anzahl der gewünschten Zeichen geringer ist als die Anzahl der bereits verfügbaren, wird der Timer nicht aufgezogen und der Lesevorgang sofort durchgeführt.

Fall B: `MIN > 0`, `TIME = 0`

Da der Wert von `TIME` Null ist, spielt der Timer keine weitere Rolle, nur `MIN` ist von Bedeutung. Hier wird ein Lesevorgang erst abgeschlossen, wenn `MIN` Zeichen eingetroffen sind. Ein Programm, das Blöcke fester Größe vom Terminal liest und Fall B gewählt hat, kann beim Lesen unendlich lange blockiert sein.

Fall C: `MIN = 0`, `TIME > 0`

Hier fungiert `TIME`, da `MIN = 0` ist, nicht als Taktgeber zwischen den Zeichen. Der Timer wird vielmehr sofort nach dem Absetzen von `read` aufgezogen. `read` wird dann entweder durch den Ablauf des Timers (und dann ohne ein Zeichen zurückzugeben) oder durch das erste eintreffende Zeichen beendet. Der Lesevorgang kann also nicht beliebig lange blockieren. Wenn kein Zeichen eintrifft, wird stattdessen nach `TIME*.10` Sekunden die Bearbeitung abgebrochen und nichts zurückgegeben.

Fall D: `MIN = 0`, `TIME = 0`

Es wird nicht auf das Eintreffen neuer Zeichen gewartet. Sind Zeichen vorhanden, wird die gewünschte Anzahl von Zeichen zurückgegeben, höchstens aber so viele, wie gerade verfügbar sind.

Vergleich der verschiedenen Fälle von MIN, TIME

Einige Anmerkungen zu MIN und TIME:

1. MIN und TIME werden nicht symmetrisch verwendet. Ist $\text{MIN} > 0$ und $\text{TIME} = 0$, dann hat TIME keine weitere Wirkung. Ist aber $\text{MIN} = 0$ und $\text{TIME} > 0$, dann spielen sowohl MIN als auch TIME eine Rolle, weil MIN durch ein eintreffendes Zeichen erfüllt werden kann.
2. Im Fall A ($\text{MIN} > 0$, $\text{TIME} > 0$) dient TIME als Timer zwischen den einzelnen Zeichen, im Fall C ($\text{TIME} = 0$, $\text{TIME} > 0$) aber als Timer für den gesamten Lesevorgang.

Beide Punkte zeigen nochmals die unterschiedlichen Möglichkeiten von MIN/TIME. Die Fälle A und B ($\text{MIN} > 0$) dienen dazu, Transaktionen zu unterstützen, bei denen Daten rasch nacheinander eintreffen (z.B. Datentransferprogramme) und ein Programm mindestens MIN Zeichen pro Zeiteinheit bearbeiten will. Im Fall A wird aus Sicherheitsgründen ein Timer aktiviert, im Fall B nicht.

Die Fälle C und D dienen dem Transfer einzelner Zeichen in einem gewissen zeitlichen Rahmen. Diese Fälle treten oft bei Applikationen auf, die beispielsweise Zeichen einlesen und anschließend auf dem Bildschirm anzeigen sollen. Der Fall C wird von einem Timer überwacht, der Fall D nicht.

Wichtig ist, daß MIN immer nur ein Minimum und keine Blockgröße bezeichnet. Wenn ein Programm etwa einen Lesevorgang von 20 Bytes startet, dabei MIN gleich 10 ist und 25 Zeichen vorhanden sind, werden 20 Zeichen an den Aufrufer zurückgegeben.

Zeichenausgabe

Ein oder mehrere geschriebene Zeichen werden an das Terminal übermittelt, sobald die Ausgabe vorher geschriebener Zeichen beendet ist. Bei eingeschaltetem Echo werden eingegebene Zeichen so wiedergegeben, wie sie geschrieben wurden. Ein Prozeß, der Zeichen schneller erzeugt, als sie ausgegeben werden können, wird angehalten, sobald die Größe seiner Ausgabeschlange einen bestimmten Wert übersteigt. Ist die Ausgabeschlange wieder auf einen normalen unteren Wert geschrumpft, wird das Program wieder aufgenommen.

Sonderzeichen

Einige Zeichen haben für die Eingabe eine besondere Funktion. Im folgenden werden diese Funktionen mit den dafür voreingestellten Zeichen aufgelistet:

INTR	(Rubout oder ASCII DEL) erzeugt ein SIGINT-Signal. SIGINT wird an alle dem Kontrollterminal zugeordneten Prozesse geschickt. Ein betroffener Prozeß beendet sich normalerweise dadurch, wenn er keine Vorkehrungen getroffen hat, das Signal zu ignorieren oder selbst zu behandeln (siehe auch <code>signal(5)</code>).
QUIT	(CTRL- oder ASCII FS) erzeugt ein SIGQUIT-Signal. Die Behandlung ist identisch mit der des Interrupt-Signals (INTR). Jedoch wird, soweit der empfangende Prozeß keine anderweitigen Vorkehrungen getroffen hat, zusätzlich ein Speicherabzug (<code>core</code>) im aktuellen Arbeitsverzeichnis erzeugt.
ERASE	(#) löscht das vorangegangene Zeichen. Das Löschen geht nicht über den durch die Zeichen NL, EOF, EOL, oder EOL2 festgelegten Zeilenanfang hinaus.
WERASE	(CTRL-W or ASCII ETX) löscht das vorangegangene "Wort". Das Löschen geht nicht über den durch die Zeichen NL, EOF, EOL, oder EOL2 festgelegten Zeilenanfang hinaus.
KILL	(@) löscht die gesamte, durch NL, EOF, EOL oder EOL2 festgelegte Zeile.
REPRINT	(CTRL-R oder ASCII DC2) gibt nach einem Zeilenvorschub alle noch nicht gelesenen Zeichen aus.
EOF	(CTRL-D oder ASCII EOT) erzeugt ein Dateiende-Zeichen vom Terminal aus. Beim Empfang werden alle ungelesenen Zeichen an das Programm übergeben, ohne auf das Newline-Zeichen zu warten. Das EOF Zeichen wird verworfen. Somit werden keine Zeichen übergeben, falls keine Zeichen zum Lesen bereit stehen (d.h. EOF am Zeilenanfang auftritt). Dies ist die standardmäßige Kennzeichnung des Dateiendes. Das EOF Zeichen wird nur geechoet, wenn ein Fluchtzeichen davor steht oder ECHOCTL gesetzt ist. Weil EOT gemäß Voreinstellung das EOF Zeichen ist, wird die Verbindung bei Terminals, die auf EOT antworten, nicht unterbrochen.
NL	(ASCII LF) ist das standardmäßig vorgegebene Zeilenbegrenzungszeichen. Es kann nicht geändert oder entwertet werden.
EOL	(ASCII NULL) ist ein zusätzliches Zeichen zur Zeilenbegrenzung. Es wird normalerweise nicht benutzt.
EOL2	ist ein weiteres Zeilenbegrenzungszeichen.
SWTCH	(CTRL-Z oder ASCII EM) wird nur benutzt, falls <code>sh1</code> aufgerufen ist.
SUSP	(CTRL-Z oder ASCII SUB) erzeugt ein SIGTSTP-Signal. SIGTSTP hält alle Prozesse in der Vordergrund-Prozeßgruppe des Terminals an.
DSUSP	(CTRL-Y oder ASCII EM) generiert ein SIGTSTP Signal (wie auch SUSP), allerdings nur dann, wenn ein Prozeß in der Vordergrund-Prozeßgruppe versucht, das DSUSP Zeichen zu lesen und nicht, wenn es eingegeben wird.

- STOP** (CTRL-S oder ASCII DC3) kann verwendet werden, um Ausgaben temporär zu unterdrücken. Bei CRT Terminals kann man damit verhindern, daß die Ausgaben verschwinden, bevor sie gelesen werden konnten. Während die Ausgabe unterdrückt ist, werden STOP- Zeichen ignoriert und nicht gelesen.
- START** (CTRL-Q oder ASCII DC1) kann verwendet werden, um vorher durch ein STOP-Zeichen unterdrückte Ausgaben wieder aufzunehmen. Ist die Ausgabe nicht unterdrückt, werden START-Zeichen ignoriert und nicht gelesen.
- DISCARD** (CTRL-O oder ASCII SI) unterdrückt solange alle nachfolgenden Ausgaben, bis ein weiteres DISCARD Zeichen eingegeben wird, zusätzliche Eingaben erfolgen oder die Bedingung von einem Programm aus abgeschaltet wird.
- LNEXT** (CTRL-V oder ASCII SYN) unterdrückt die Sonderbedeutung des nächsten Zeichens. Dies ist für alle oben angeführten Zeichen möglich. Damit können Zeichen eingegeben werden, die andernfalls durch das System interpretiert werden würden (z.B. KILL, QUIT).

Die Zeichen für INTR, QUIT, ERASE, WERASE, KILL, REPRINT, EOF, EOL, EOL2, SWTCH, SUSP, DSUSP, STOP, START, DISCARD und LNEXT können nach individuellen Bedürfnissen geändert werden. Ist der Wert eines Sonderzeichens `_POSIX_VDISABLE` (0), wird die Bedeutung dieses Sonderzeichens deaktiviert. Die Zeichen ERASE, KILL und EOF können durch ein vorangestelltes `\` Zeichen entwertet werden. Dann wird keine Sonderfunktionalität ausgeführt. Jedem der Sonderzeichen kann ein LNEXT Zeichen vorausgehen. Dadurch wird dessen besondere Bedeutung unterdrückt.

Abbruch der Modem-Verbindung

Ist eine Modem-Verbindung unterbrochen und wird dies erkannt, wird dem Kontrollprozeß des Terminals das Signal `SIGHUP` gesendet. Standardmäßig wird dadurch der Prozeß beendet. Wird `SIGHUP` ignoriert oder abgefangen, liefert jedes nachfolgende Lesen bis zum Schliessen des Terminals den Wert Dateieinde (end-of-file).

Gehört der Kontrollprozeß nicht zur Gruppe der Vordergrund-Prozesse des Terminals, wird das Signal `SIGTSTP` an die Vordergrund-Prozeßgruppe des Terminals gesendet. Sind keine anderweitigen Vorkehrungen getroffen, verursacht das Signal die Beendigung der Prozesse.

Wenn Prozesse einer Hintergrund-Prozeßgruppe auf ein Kontrollterminal zugreifen, nachdem die Modem-Verbindung unterbrochen wurde, dieses Terminal aber noch zur Sitzung gehört, erhalten diese Prozesse `SIGTTTOU` resp. `SIGTTIN` Signale. Sind keine anderweitigen Vorkehrungen getroffen, verursacht das Signal die Beendigung der Prozesse.

Das kontrollierende Terminal verbleibt in seinem Zustand, bis es vom kontrollierenden Prozeß entweder durch ein erfolgreiches Öffnen reinitialisiert oder aber von ihm freigegeben wird.

Terminal-Parameter

Die Parameter der `termios` Schnittstelle, die das Verhalten von Geräten und Modulen steuern, sind in der `termios` Struktur festgelegt. Diese Struktur wird in `<termios.h>` definiert und von verschiedenen `ioctl(2)` Systemaufrufen verwendet. Die Struktur ist so aufgebaut:

```

tcflag_t  c_iflag;           /* Eingabemodi */
tcflag_t  c_oflag;           /* Ausgabemodi */
tcflag_t  c_cflag;           /* Steuermodi */
tcflag_t  c_lflag;           /* lokale Modi */
cc_t      c_cc[NCCS];        /* Steuerzeichen */

```

Die speziellen Steuerzeichen sind in dem Feld `c_cc` definiert. Der symbolische Name `NCCS` bezeichnet die Größe dieses Feldes, er ist ebenfalls in `<termios.h>` definiert. Die relativen Positionen, Bezeichnungen und typischen Initialisierungen für die einzelnen Funktionen sind folgende:

0	VINTR	DEL
1	VQUIT	FS
2	VERSE	#
3	VKILL	@
4	VEOF	EOT
5	VEOL	NUL
6	VEOL2	NUL
7	VSWTCH	NUL
8	VSTRT	DC1
9	VSTOP	DC3
10	VSUSP	SUB
11	VDSUSP	EM
12	VREPRINT	DC2
13	VDISCRD	SI
14	VWERSE	ETB
15	VLNEXT	SYN
16-19	reserviert	

Eingabemodi

Das Feld `c_iflag` beschreibt die grundlegende Eingabesteuerung für Terminals:

IGNBRK	break-Bedingung ignorieren.
BRKINT	Signalunterbrechung bei break.
IGNPAR	Zeichen mit Paritätsfehlern ignorieren.
PARMRK	Paritätsfehler markieren.
INPCK	Paritätstest der Eingabe aktivieren.
ISTRIP	Zeichen maskieren.
INLCR	NL in der Eingabe in CR übersetzen.
IGNCR	CR ignorieren.
ICRNL	CR in der Eingabe in NL übersetzen.
IUCLC	Großbuchstaben in der Eingabe in Kleinbuchstaben übersetzen.
IXON	Start-/Stop-Ausgabesteuerung aktivieren.
IXANY	Jedes Zeichen zum Neustart der Ausgabe zulassen.
IXOFF	Start-/Stop-Eingabesteuerung aktivieren.
IMAXBEL	BEL bei zu langer Eingabezeile.

Ist `IGNBRK` gesetzt, wird die break-Bedingung (ein Zeichenrahmenfehler, bei dem alle Datenbits auf Null gesetzt sind) in der Eingabe ignoriert. Sie wird also nicht in die Eingabeschlange gestellt und damit auch von keinem Prozeß gelesen. Ist `BRKINT` gesetzt, nicht aber `IGNBRK`, dann muß bei der break-Bedingung die Eingabe- und Ausgabeschlange geleert werden. Falls das Terminal das Kontrollterminal einer Vordergrund-Prozeßgruppe ist, wird zudem ein `SIGINT`-Signal an diese Vordergrund-Prozeßgruppe geschickt. Ist weder `IGNBRK` noch `BRKINT` gesetzt, wird die break-Bedingung entweder als einzelnes ASCII NULL Zeichen (`\0`) gelesen, oder, wenn `PARMRK` gesetzt ist, als `\377`, `\0`, `\0`.

Ist `IGNPAR` gesetzt, wird ein Byte mit Rahmen- oder Paritätsfehler (außer break) ignoriert.

Ist `PARMRK` gesetzt, nicht aber `IGNPAR`, wird ein Byte mit Rahmen- oder Paritätsfehler (außer break) als Sequenz von drei Zeichen an die Anwendung weitergegeben: `\377`, `\0`, `X`. `X` entspricht dabei dem fehlerhaft empfangenen Byte. Um Zweideutigkeiten zu vermeiden, wird bei nicht gesetztem `ISTRIP` ein gültiges Zeichen `\377` in der Form `\377`, `\377` an die Anwendung weitergegeben. Ist weder `IGNPAR` noch `PARMRK` gesetzt, wird ein Rahmen- oder Paritätsfehler (außer break) als einzelnes ASCII NULL Zeichen (`\0`) an die Anwendung weitergegeben.

Ist `INPCK` gesetzt, wird der Paritätstest der Eingabe aktiviert. Ist `INPCK` nicht gesetzt, ist der Paritätstest der Eingabe deaktiviert. Dies ermöglicht, auch bei Paritätsfehlern der Eingabe, die Paritätserzeugung für die Ausgabe. Dabei ist zu beachten, daß das Aktivieren und Deaktivieren der Paritätstests unabhängig vom Aktivieren und Deaktivieren der Paritätserkennung ist. Ist die Paritätserkennung, nicht aber der Paritätstest aktiviert, erkennt die mit dem Terminal verbundene Hardware zwar das Paritätsbit, der Terminalgerätetreiber überprüft aber nicht, ob es korrekt gesetzt ist.

Ist `ISTRIP` gesetzt, werden gültige Eingabezeichen auf 7 Bit reduziert. Andernfalls werden alle 8 Bits verarbeitet.

Ist `INLCR` gesetzt, wird ein empfangenes NL Zeichen in ein CR Zeichen übersetzt. Ist `IGNCR` gesetzt, wird ein empfangenes CR Zeichen ignoriert (nicht gelesen). Sonst wird, falls `ICRNL` gesetzt ist, ein empfangenes CR Zeichen in ein NL Zeichen übersetzt.

Ist `IUCLC` gesetzt, wird ein empfangener Großbuchstabe in den entsprechenden Kleinbuchstaben übersetzt.

Ist `IXON` gesetzt, wird die Start-/Stop-Eingabesteuerung aktiviert. Ein empfangenes STOP-Zeichen hält die Ausgabe an, ein empfangenes START Zeichen nimmt sie wieder auf. Die Zeichen STOP und START werden nicht gelesen, sondern übernehmen nur Aufgaben der Datenflußsteuerung. Ist `IXANY` gesetzt, nimmt jedes eingegebene Zeichen die angehaltene Ausgabe wieder auf.

Ist `IXOFF` gesetzt, überträgt das System ein STOP Zeichen, wenn die Eingabeschlange nahezu voll ist, und ein START Zeichen, wenn genügend Eingaben gelesen wurden und die Eingabeschlange wieder nahezu leer ist.

Ist `IMAXBEL` gesetzt, wird beim Überlaufen des Eingabestroms ein ASCII BEL Zeichen ausgegeben. Weitere Eingaben werden nicht gespeichert, die bislang eingetroffenen Zeichen aber bleiben erhalten. Ist `IMAXBEL` nicht gesetzt, wird bei einem Überlauf des Eingabestroms kein BEL Zeichen ausgegeben und alle bisher eingetroffenen Zeichen gehen verloren.

Die Voreinstellung für die Eingabesteuerung ist `BRKINT`, `ICRNLC`, `IXON`, `ISTRIP`.

Ausgabemodi

Das Feld `c_oflag` beschreibt die Behandlung der Ausgabe durch das System:

<code>OPOST</code>	Ausgabe nachbearbeiten.
<code>OLCUC</code>	Kleinbuchstaben in der Ausgabe in Großbuchstaben übersetzen.
<code>ONLCR</code>	NL in der Ausgabe in CR-NL übersetzen.
<code>OCRNLC</code>	CR in der Ausgabe in NL übersetzen.
<code>ONOCR</code>	Kein CR in Spalte 0 ausgeben.
<code>ONLRET</code>	NL übernimmt CR Funktionalität.
<code>OFILL</code>	Füllzeichen zur Verzögerung verwenden.
<code>OFDEL</code>	DEL ist Füllzeichen, ansonsten NULL.
<code>NLDLY</code>	NL-Verzögerung (newline) auswählen:
<code>NL0</code>	
<code>NL1</code>	
<code>CRDLY</code>	CR-Verzögerung (carriage-return) auswählen:
<code>CR0</code>	
<code>CR1</code>	
<code>CR2</code>	
<code>CR3</code>	
<code>TABDLY</code>	Horizontale Tabulator-Verzögerung oder -Umwandlung auswählen:
<code>TAB0</code>	
<code>TAB1</code>	
<code>TAB2</code>	
<code>TAB3</code>	Tabulatoren zu Leerzeichen expandieren.
<code>XTAB3</code>	Tabulatoren zu Leerzeichen expandieren.
<code>BSDLY</code>	BS-Verzögerung (backspace) auswählen:
<code>BS0</code>	
<code>BS1</code>	
<code>VTDLY</code>	Vertikale Tabulator-Verzögerung auswählen:
<code>VT0</code>	
<code>VT1</code>	
<code>FFDLY</code>	Seitenvorschub-Verzögerung (form feed) auswählen:
<code>FF0</code>	
<code>FF1</code>	

Ist `OPOST` gesetzt, werden Ausgabezeichen entsprechend der übrigen Einstellungen nachbearbeitet. Andernfalls werden die Zeichen ohne Änderungen übertragen.

Ist `OLCUC` gesetzt, wird ein Kleinbuchstabe als Großbuchstabe übermittelt. Diese Funktionalität wird oft in Verbindung mit `IUCLC` benutzt.

Ist `ONLCR` gesetzt, wird ein NL Zeichen als ein CR-NL Zeichenpaar übermittelt. Ist `OCRNLC` gesetzt, wird ein CR Zeichen als ein NL Zeichen übermittelt. Ist `ONOCR` gesetzt, wird in Spalte 0 (Zeilenanfang) kein CR Zeichen übermittelt. Ist `ONRET` gesetzt, übernimmt das NL Zeichen die Funktionalität von Wagenrücklauf

(carriage-return). Der Spaltenzeiger wird auf 0 gesetzt und die festgelegten Verzögerungswerte für CR angewandt. Andernfalls übernimmt das NL Zeichen nur die Funktionalität von Zeilenvorschub (line-feed), der Spaltenzeiger bleibt unverändert. Der Spaltenzeiger wird auf 0 gesetzt, falls das CR Zeichen tatsächlich übermittelt wird.

Die Verzögerungsbits legen fest, wie lange beim Senden gewisser Zeichen die Übertragung ausgesetzt wird, um mechanische oder sonstige Bewegungen beim Empfänger zu ermöglichen. In allen Fällen steht der Wert 0 für keine Verzögerung. Ist `OFILL` gesetzt, werden anstelle einer zeitlichen Verzögerung Füllzeichen zur Verzögerung verwendet. Dies ist geeignet für Terminals mit hoher Übertragungsrate und geringer Verzögerung. Bei gesetztem `OFDEL` wird DEL das Füllzeichen, andernfalls ist es `NULL`.

Ist eine Seitenvorschub- oder horizontale Tabulator-Verzögerung angegeben, dauert diese ungefähr 2 Sekunden.

Eine NL-Verzögerung (newline) dauert ungefähr 0.10 Sekunden. Ist `ONLRET` gesetzt, werden die CR-Verzögerungen (carriage-return) anstelle der NL-Verzögerungen (newline) verwendet. Bei gesetztem `OFILL` werden zwei Füllzeichen übermittelt.

Die CR-Verzögerung (carriage-return) vom Typ 1 ist abhängig von der aktuellen Spaltenposition, Typ 2 verzögert um etwa 0.10 Sekunden, Typ 3 um ungefähr 0.15 Sekunden. Ist `OFILL` gesetzt, werden bei Typ 1 zwei Füllzeichen und bei Typ 2 vier Füllzeichen übermittelt.

Die horizontale Tabulator-Verzögerung vom Typ 1 ist abhängig von der aktuellen Spaltenposition, bei Typ 2 dauert sie ungefähr 0.10 Sekunden, Typ 3 expandiert Tabulatoren zu Leerzeichen. Ist `OFILL` gesetzt, werden für jeden Typ zwei Füllzeichen zur Verzögerung übermittelt.

Die BS-Verzögerung (backspace) dauert ungefähr 0.05 Sekunden. Ist `OFILL` gesetzt, wird ein Füllzeichen übermittelt.

Die aktuellen Verzögerungswerte hängen von der Leitungsgeschwindigkeit und der Systemlast ab.

Die Voreinstellung für die Ausgabesteuerung ist `OPOST`, `ONLCR`, `TAB3`.

Steuermodi

Das Feld `c_cflag` beschreibt die Hardwaresteuerung des Terminals:

CBAUD	Baudraten:
B0	Verbindung unterbrochen (hang up)
B50	50 Baud
B75	75 Baud
B110	110 Baud
B134	134 Baud
B150	150 Baud
B200	200 Baud
B300	300 Baud
B600	600 Baud
B1200	1200 Baud
B1800	1800 Baud
B2400	2400 Baud

B4800	4800 Baud
B9600	9600 Baud
B19200	19200 Baud
EXTA	Extern A
B38400	38400 Baud
EXTB	Extern B
CSIZE	Größe eines Zeichens:
CS5	5 Bits
CS6	6 Bits
CS7	7 Bits
CS8	8 Bits
CSTOPB	Zwei Stopbits senden, ansonsten eines.
CREAD	Empfänger aktivieren.
PARENB	Parität aktivieren.
PARODD	Ungerade Parität, ansonsten gerade.
HUPCL	Verbindung beim letzten Schließen (close) unterbrechen.
CLOCAL	Standleitung, ansonsten Wählleitung.
CIBAUD	Baudrate bei Eingabe, falls unterschiedlich von Ausgabe.
PAREXT	Erweiterte Parität für Mark und Space Parität.

Die CBAUD-Bits legen die Baudrate fest. B0 wird zur Unterbrechung der Verbindung verwendet. In diesem Fall liegt das Signal DTR (data terminal ready) nicht an. Dadurch wird normalerweise die Leitung unterbrochen. Sind die CIBAUD Bits nicht Null, spezifizieren sie die Baudrate der Eingabe. In diesem Fall legt CBAUD die Baudrate der Ausgabe fest. Sonst legt CBAUD die Baudrate sowohl für die Eingabe, als auch für die Ausgabe fest. Die Werte für CIBAUD entsprechen denen für CBAUD, die Bits werden lediglich um IBSHIFT nach links geschoben. Unausführbare Änderungen der Übertragungsgeschwindigkeit werden von der Hardware ignoriert.

Die Bits in CSIZE geben die Zeichenbreite in Bits für Senden und Empfangen an. Ein eventuell vorhandenes Paritätsbit ist nicht in dieser Zeichenbreite enthalten. Ist CSTOPB gesetzt, werden zwei Stopbits übertragen, sonst nur eines. Bei 110 Baud sind beispielsweise zwei Stopbits nötig.

Bei gesetztem PARENB wird das Paritätsbit gesetzt und überprüft. Jedem Zeichen wird dazu ein Paritätsbit angefügt. Ist PARODD gesetzt, wird eine ungerade, sonst eine gerade Parität verwendet.

Der Empfänger wird durch CREAD eingeschaltet, andernfalls werden keine Zeichen empfangen.

Ist HUPCL gesetzt, dann wird die Leitung unterbrochen, wenn der letzte Prozeß, der sie verwendet, die Leitung schließt oder sich beendet.

Wenn CLOCAL gesetzt ist, wird angenommen, daß eine lokale Leitung ohne Modemsteuerung betrieben wird. Andernfalls wird die Modemsteuerung verwendet.

Die Bits der Hardwaresteuerung sind beim Öffnen einer Verbindung auf B300, CS8, CREAD und HUPCL gesetzt.

Lokale Modi

Das `c_lflag` der `termios` Struktur wird von der Übertragungsprozedur zur Steuerung der Terminal-Funktionen verwendet. Die Basisprozedur für diese Übertragung (basic line discipline) unterstützt dabei folgende Werte:

ISIG	Signalbehandlung aktivieren.
ICANON	Kanonische Eingabe (Behandlung von erase und kill).
XCASE	Kanonische Übergabe bei Groß-/Kleinschreibung.
ECHO	Echo aktivieren.
ECHOE	erase-Zeichen als BS-SP-BS echoen.
ECHOK	NL nach einem kill-Zeichen echoen.
ECHONL	NL echoen.
NOFLSH	Nach einem Interrupt oder Quit die Zeichen nicht wegwerfen.
TOSTOP	SIGTTOU bei Hintergrund-Ausgabe schicken.
ECHOCTL	Kontrollzeichen als <code>^char</code> , delete als <code>??</code> echoen.
ECHOPRT	erase als gelöschttes Zeichen echoen.
ECHOKE	BS-SP-BS löscht beim Zeilenlöschen die ganze Zeile.
FLUSHO	Ausgabepuffer leeren.
PENDIN	Anstehende Eingaben beim nächsten Lesen oder Eingeben wiederholen.
IEXTEN	Zusätzliche (implementationabhängige) Funktionen einschalten.

Wenn ISIG gesetzt ist, wird jedes Zeichen mit den speziellen Steuerzeichen INTR, QUIT, SWTCH, SUSP, STATUS und DSUSP verglichen. Entspricht ein Eingabezeichen einem der Steuerzeichen, wird die dazugehörige Funktion aufgerufen. Diese Überprüfung findet nur statt, wenn ISIG gesetzt ist.

ICANON aktiviert die kanonische Behandlung. Dadurch werden die erase- und kill-Funktionen aktiviert und eingegebene Zeichen in Zeilen verarbeitet, die durch NL, EOF, EOL, und EOL2 begrenzt sind. Ist ICANON nicht gesetzt, werden Leseanfragen direkt aus der Eingabeschlange befriedigt, sobald mindestens MIN Zeichen angekommen sind oder die durch TIME begrenzte Zeit zwischen zwei Zeichen abgelaufen ist. So können sowohl rasch nacheinander eintreffende Zeichenströme wie auch einzelne Zeichen gelesen werden. TIME wird dabei in Zehntel-Sekunden angegeben.

Sind XCASE und ICANON gleichzeitig gesetzt, müssen Großbuchstaben bei der Eingabe durch das Fluchtsymbol `\` geschützt werden. Sowohl `\` als auch der Großbuchstabe werden dann unverändert an die Ausgabe weitergegeben. In diesem Modus werden die folgenden Escape-Sequenzen ausgegeben und bei der Eingabe erwartet:

für:	verwende:
<code>^</code>	<code>\^</code>
<code> </code>	<code>\ </code>
<code>~</code>	<code>\~</code>
<code>{</code>	<code>\{</code>
<code>}</code>	<code>\}</code>
<code>\</code>	<code>\\</code>

Einige Beispiele: A wird als \a eingegeben, \n als \\n, sowie \N als \\N.

Ist ECHO gesetzt, werden die Zeichen ohne Änderung geechot.

Wenn ICANON gesetzt ist, sind folgende Echofunktionen möglich:

1. ECHO und ECHOE sind gesetzt, ECHOPRT ist nicht gesetzt: ERASE und WERASE Zeichen werden als ein oder mehrere ASCII BS SP BS geechot. Dadurch werden bei einem CRT-Bildschirm das oder die letzte(n) Zeichen gelöscht.
2. ECHO und ECHOPRT sind gesetzt: Das erste ERASE und WERASE-Zeichen einer Sequenz wird als Backslash (\) geechot, gefolgt von dem oder den zu löschenden Zeichen. Aufeinanderfolgende ERASE und WERASE echoen die zu löschenden Zeichen in umgekehrter Reihenfolge. Dem nächsten von einem erase-Zeichen verschiedene Zeichen wird ein (/) vorangestellt. Für Drucker sollte ECHOPRT verwendet werden.
3. ECHOKE ist gesetzt: Das kill-Zeichen wird geechot. Dadurch werden alle Zeichen der Zeile gelöscht. Der dabei verwendete Mechanismus wird durch ECHOE und ECHOPRT bestimmt.
4. ECHOK ist gesetzt, ECHOKE ist nicht gesetzt: Nach einem kill-Zeichen wird ein NL geechot. Dadurch soll deutlich gemacht werden, daß diese Zeile gelöscht wurde. Das Fluchtzeichen (\) bzw. das LNEXT Zeichen heben die Spezialbedeutung eines anschließenden kill- bzw. erase-Zeichens auf.
5. ECHONL ist gesetzt: Das NL Zeichen wird geechot, auch wenn ECHO nicht gesetzt ist. Dies ist bei Terminals mit lokalem Echo (half-duplex) empfehlenswert.

Wenn ECHOCTL gesetzt ist, werden Kontrollzeichen (zwischen 0 und 37 oktal) als ^X ausgegeben. X ist dabei das Kontrollzeichen plus 100 oktal (aus 1 wird also ^A, aus ASCII DEL mit dem Code 177 oktal ^?). Ausgenommen davon sind ASCII TAB, ASCII NL, das START Zeichen, das STOP Zeichen, ASCII CR und ASCII BS.

Wenn NOFLSH gesetzt ist, wird das standardmäßige Leeren der Ein- und Ausgabepuffer nach INTR, QUIT oder SUSP nicht vorgenommen. Dieses Bit sollte gesetzt sein, wenn ein Systemaufruf wiederholt wird, der von einem Terminal liest oder auf ein Terminal schreibt (siehe `sigaction(2)`).

Wenn TOSTOP gesetzt ist, wird das Signal SIGTTOU an denjenigen Prozeß geschickt, der versucht, auf sein Kontrollterminal zu schreiben, obwohl er nicht in der Vordergrund-Prozeßgruppe für dieses Terminal ist. Normalerweise wird dieser Prozeß dadurch beendet. Andernfalls wird die Ausgabe dieses Prozesses auf dem aktuellen Ausgabestrom vorgenommen. Prozesse, die SIGTTOU ignorieren oder selbst behandeln, dürfen durchaus Ausgabe produzieren.

Wenn FLUSHO gesetzt ist, wird die bisherige Eingabe verworfen. Dieses Bit wird durch ein FLUSH-Zeichen gesetzt und kann durch ein FLUSH Zeichen wieder aufgehoben werden.

Wenn PENDIN gesetzt ist, werden alle Eingaben, die bis dahin noch nicht gelesen wurden, beim nächsten ankommenden Zeichen nochmals wiedergegeben.

Wenn `IEXTEN` gesetzt ist, kann auf folgende implementationsabhngige Funktionen zugegriffen werden: Spezielle Steuerzeichen (`WERASE`, `REPRINT`, `DISCARD` und `LNEXT`), lokale Flags (`TOSTOP`, `ECHOCTL`, `ECHOPRT`, `ECHOKE`, `FLUSHO` und `PENDIN`).

Grundeinstellung bei der Zeilenbehandlung ist `ISIG`, `ICANON`, `ECHO`, `ECHOK`.

Minimum und Timeout

Die Werte `MIN` und `TIME` wurden weiter oben im Abschnitt *Eingabeverarbeitung im nicht-kanonischen Modus* behandelt. Der voreingestellte Wert `fr MIN` ist 1, `fr TIME` 0.

Terminalgre

Die Anzahl der Zeilen und Spalten auf einer Terminalanzeige ist in der `winsize` Struktur festgelegt. Sie ist in `<sys/termios.h>` definiert und enthlt folgende Angaben:

```
unsigned short  ws_row;    /* Zeilen in Zeichen */
unsigned short  ws_col;    /* Spalten in Zeichen */
unsigned short  ws_xpixel; /* Breite in Pixel */
unsigned short  ws_ypixel; /* Hhe in Pixel */
```

Die Struktur termio

Die System V `termio` Struktur wird von einigen `ioctl`s Aufrufen verwendet. Sie ist in `<sys/termio.h>` definiert und enthlt folgende Felder:

```
unsigned short  c_iflag;    /* Eingabemodi */
unsigned short  c_oflag;    /* Ausgabemodi */
unsigned short  c_cflag;    /* Steuernodi */
unsigned short  c_lflag;    /* lokale Modi */
char            c_line;     /* Leitungsprozedur */
unsigned char   c_cc[NCC];  /* Steuerzeichen */
```

Die speziellen Steuerzeichen sind in dem Feld `c_cc` definiert. Der symbolische Name `NCC` bezeichnet die Gre des Feldes `fr` die Steuerzeichen und ist ebenfalls in `<termio.h>` definiert. Die relativen Positionen, Bezeichnungen und typischen Initialisierungen `fr` die einzelnen Funktionen sind wie folgt:

```
0  VINTR    DEL
1  VQUIT    FS
2  VERASE    #
3  VKILL     @
4  VEOF      EOT
5  VEOL      NUL
6  VEOL2     NUL
7  reserviert
```

Alle Aufrufe, die die `termio` Struktur verwenden, knnen nur die Flags und Kontrollzeichen dieser Struktur verndern. Alle weiteren Flags und Kontrollzeichen sind nicht betroffen.

Modem-Leitungen

Bei Gertedateien fr serielle Schnittstellen knnen die von der Hardware untersttzten Kontroll-Leitungen des Modems gelesen bzw. dessen Status-Leitungen gendert werden. Nicht alle Gerte untersttzen die folgenden Kontroll- und Status-Leitungen. Sie sind in `<sys/termios.h>` definiert.

TIOCM_LE	Leitung aktivieren
TIOCM_DTR	Daten-Terminal bereit
TIOCM_RTS	Anfrage schicken
TIOCM_ST	Sekundrbermittlung
TIOCM_SR	Sekundrempfang
TIOCM_CTS	Fertig zum Senden
TIOCM_CAR	Trger gefunden
TIOCM_RNG	Klingel
TIOCM_DSR	Datenset fertig

TIOCM_CD ist ein Synonym fr TIOCM_CAR und TIOCM_RI ein Synonym fr TIOCM_RNG.

IOCTLS

Die von Gerten und STREAMS Modulen untersttzten `ioctl`-Aufrufe der `termios` Schnittstelle werden unten aufgefhr. Nicht alle Aufrufe werden von allen Gerten oder Modulen unterstzt. Dieselbe Funktionalitt bietet die in `termios(2)` definierte Schnittstelle.

TCGETS	Das Argument ist ein Zeiger auf eine <code>termios</code> -Struktur. Die aktuellen Terminal-Parameter werden ermittelt und in dieser Struktur gespeichert.
TCSETS	Das Argument ist ein Zeiger auf eine <code>termios</code> -Struktur. Die Terminal-Parameter werden mit den Werten der bergebenen Struktur sofort initialisiert.
TCSETSW	Das Argument ist ein Zeiger auf eine <code>termios</code> -Struktur. Die Terminal-Parameter werden mit den Werten der bergebenen Struktur initialisiert. Die nderung wirkt sich erst dann aus, wenn alle Zeichen der Ausgabewarteschlange ermittelt wurden. Dieser Aufruf sollte immer dann verwendet werden, wenn die nderung von Parametern auch die Terminal-Ausgabe betrifft.
TCSETSF	Das Argument ist ein Zeiger auf eine <code>termios</code> -Struktur. Die Terminal-Parameter werden mit den Werten der bergebenen Struktur initialisiert. Die nderung wirkt sich erst dann aus, wenn alle Zeichen der Ausgabewarteschlange ermittelt wurden. Alle Zeichen der Eingabewarteschlange werden verworfen.
TCGETA	Das Argument ist ein Zeiger auf eine <code>termio</code> -Struktur. Die aktuellen Terminal-Parameter werden ermittelt und, soweit mglich, in dieser Struktur gespeichert.
TCSETA	Das Argument ist ein Zeiger auf eine <code>termio</code> -Struktur. Die Terminal-Parameter werden mit den Werten der bergebenen Struktur initialisiert. Dies hat sofortige Wirkung.

TCSETAW	Das Argument ist ein Zeiger auf eine <code>termio</code> -Struktur. Die Terminal-Parameter werden mit den Werten der gegebenen Struktur initialisiert. Die Änderung wirkt sich erst dann aus, wenn alle Zeichen der Ausgabewarteschlange ermittelt wurden. Dieser Aufruf sollte immer dann verwendet werden, wenn die Änderung von Parametern auch die Terminal-Ausgabe betrifft.
TCSETAF	Das Argument ist ein Zeiger auf eine <code>termio</code> -Struktur. Die Terminal-Parameter werden mit den Werten der gegebenen Struktur initialisiert. Die Änderung wirkt sich erst dann aus, wenn alle Zeichen der Ausgabewarteschlange ermittelt wurden. Alle Zeichen der Eingabewarteschlange werden verworfen.
TCSBRK	Das Argument ist ein <code>int</code> -Wert und unterbricht die Weiterleitung der Ausgabe. Wird 0 gegeben, dann wird ein <code>break</code> gesendet (mit 0 besetzte Bits für 0.25 Sekunden).
TCXONC	Start/Stop-Kontrolle. Das Argument ist ein <code>int</code> -Wert. Der Wert 0 unterdrückt die Ausgabe, der Wert 1 startet die unterdrückte Ausgabe wieder. Der Wert 2 unterdrückt die Eingabe, der Wert 3 startet die unterdrückte Eingabe wieder.
TCFLSH	Das Argument ist ein <code>int</code> -Wert. Der Wert 0 leert den Eingabepuffer, der Wert 1 leert den Ausgabepuffer, der Wert 2 leert beide.
TIOCGPGRP	Das Argument ist ein Zeiger auf <code>pid_t</code> . <code>pid_t</code> wird mit der ID der Vordergrund-Prozessgruppe initialisiert. Siehe <code>termios(2)</code> für eine Beschreibung von <code>TCGETPGRP</code> .
TIOCSGRP	Das Argument ist ein Zeiger auf <code>pid_t</code> . Die durch <code>pid_t</code> beschriebene Prozessgruppe wird als Vordergrund-Prozessgruppe des Terminals eingesetzt. Der Wert der neuen Prozessgruppe muss im Bereich der erlaubten Werte für ID-Werte liegen. Andernfalls wird ein <code>EPERM</code> Fehler zurückgegeben. Siehe <code>termios(2)</code> für eine Beschreibung von <code>TCSETPGRP</code> .
TIOCGSID	Das Argument ist ein Zeiger auf <code>pid_t</code> . <code>pid_t</code> wird mit der Sitzungs-ID des Terminals initialisiert.
TIOCGWINSZ	Das Argument ist ein Zeiger auf eine <code>winsize</code> -Struktur. Die Größe des Terminals aus der Sicht des Terminaltreibers wird ermittelt und in die <code>winsize</code> -Struktur gestellt.
TIOCSWINSZ	Das Argument ist ein Zeiger auf eine <code>winsize</code> -Struktur. Die Größe des Terminals aus der Sicht des Terminaltreibers wird mit dem gegebenen Wert initialisiert. Unterscheidet sich die neue Größe von der alten, wird ein <code>SIGWINCH</code> -Signal an die Prozessgruppe des Terminals geschickt.
TIOCMBS	Das Argument ist ein Zeiger auf einen <code>int</code> -Wert. Dieser Wert enthält eine Maske mit allen Modem-Kontroll-Leitungen, die aktiviert werden sollen. Gesetzte Bits aktivieren dabei die jeweilige Leitung. Andere Kontroll-Leitungen werden nicht beeinflusst.

TIOCMBIC	Das Argument ist ein Zeiger auf einen <code>int</code> -Wert. Dieser Wert enthält eine Maske mit allen Modem-Kontroll-Leitungen, die deaktiviert werden sollen. Gesetzte Bits deaktivieren dabei die jeweilige Leitung. Andere Kontroll-Leitungen werden nicht beeinflusst.
TIOCMGET	Das Argument ist ein Zeiger auf einen <code>int</code> -Wert. Der aktuelle Zustand der Modem-Status-Leitungen wird ermittelt und in diesem Wert gespeichert.
TIOCMSET	Das Argument ist ein Zeiger auf einen <code>int</code> -Wert. Die Modem-Kontroll-Leitungen werden mit diesem Wert initialisiert. Ein entsprechend gesetztes Bit aktiviert, ein nicht gesetztes Bit deaktiviert eine Leitung.

DATEIEN

Dateien in oder unter `/dev`

SIEHE AUCH

`fork(2)`, `ioctl(2)`, `setsid(2)`, `signal(2)`, `termios(2)`, `streamio(7)`.

BEZEICHNUNG

termiox – Allgemeine Terminal-Schnittstelle mit Erweiterungen

BESCHREIBUNG

Die allgemeine Terminal-Schnittstelle mit Erweiterungen ergänzt die `termio(7)` Terminal-Schnittstelle durch die Unterstützung von hardwareseitiger, asynchroner Flußsteuerung, isochroner Flußsteuerung mit entsprechenden Takteinstellungen und lokal implementierten zusätzlichen asynchronen Funktionen. Einige Systeme können auf Grund von Hardware- oder Softwarebeschränkungen nicht alle diese Möglichkeiten ausnutzen. Weiter können bestimmte Funktionen in einigen Systemen nicht gesperrt werden; in diesen Fällen werden die entsprechenden Bits ignoriert. Die Datei `<termiox.h>` gibt an, welche Funktionen im einzelnen verfügbar sind.

Hardwareseitige Flußsteuerung

Die hardwareseitige Flußsteuerung ergänzt die in `termio(7)` definierte zeichenorientierte Flußsteuerung `IXON`, `IXOFF` und `IXANY`. Im Falle einer zeichenorientierten Flußsteuerung kontrolliert ein Gerät die Datenübertragung eines anderen Geräts durch das Einfügen von Steuerzeichen in den übertragenen Datenstrom. Im Falle einer hardwareseitigen Flußsteuerung kontrolliert ein Gerät die Datenübertragung eines anderen Geräts durch die Ausnutzung von elektronischen Steuersignalen in der asynchronen Schnittstelle. Von einer isochronen Flußsteuerung spricht man, wenn ein Gerät die Datenübertragung des anderen Geräts durch das Hinzufügen oder Wegnehmen der Sende-Zeittakte dieses Geräts kontrolliert. Es ist möglich, zeichenorientierte und hardwareseitige Flußsteuerung gleichzeitig zu verwenden.

Für die hardwareseitige Flußsteuerung bei asynchronen Voll-Duplex Verbindungen werden meistens die im Standard EIA-232-D der Electronic Industries Association festgelegten Leitungen für Request To Send (RTS) und Clear To Send (CTS) verwendet. Andere Standards für die hardwareseitige Flußsteuerung sind ebenfalls in der Schnittstelle enthalten.

Im EIA-232-D Standard ist lediglich die hardwareseitige Flußsteuerung für einseitige Verbindungen definiert - die Datenübertragungseinrichtung (DCE) zeigt dem Datenendgerät (DTE) die Unterbrechung der Datenübertragung an. Demgegenüber erlaubt die `termiox(7)`-Schnittstelle sowohl einseitige als auch beidseitige Flußsteuerung; wenn die beidseitige Flußsteuerung eingeschaltet ist, kann sowohl DCE als auch DTE gegenseitig die Unterbrechung der Datenübertragung über die Schnittstelle signalisieren. Hinweis: Es wird vorausgesetzt, daß der asynchrone Port als DTE konfiguriert ist. Falls das angeschlossene Gerät ebenfalls ein DTE und kein DCE ist, kann ein Null-Modem verwendet werden, um DTE mit DTE zu verbinden (z.B.: Bildschirm oder Druckeranschluß).

Möglichkeiten der Takterzeugung

Isochrone Datenübertragung ist eine Variation der asynchronen Verbindung, wobei zwei Datenübertragungsgeräte dem jeweils anderen Gerät den Sende- und/oder Empfangstakt zur Verfügung stellen können. Auf der Empfängerseite können die Taktsignale direkt vom Baudraten-Generator auf dem isochronen Adapter genommen werden, vom CCITT V24 Leitung 114: "Transmitter Signal Element Timing" - DCE Quelle (EIA-232-D Pin 15), oder aber vom CCITT V24 Leitung 115: "Receiver Signal Element Timing" - DCE Quelle (EIA-232-D Pin 17). Auf der Senderseite können

Taktsignale auf dem CCITT V24 Leitung 113: Transmitter Signal Element Timing - DTE Quelle (EIA-232-D Pin 24), auf dem CCITT V24 Leitung 128, Receiver Signal Element Timing - DTE Quelle (kein EIA-232-D Pin) gesendet werden, oder überhaupt nicht.

Was die Möglichkeiten der Takterzeugung anbelangt, wird die übliche asynchrone Datenübertragung einfach dadurch implementiert, daß der eigene Baudraten-Generator als Quelle für den ankommenden Sende- und Empfangstakt verwendet wird; es werden keine Taktsignale gesendet.

Parameter für Datenübertragungsgeräte

Alle Parameter, die zur Kontrolle von Datenübertragungsgeräten mit der `termiox`-Schnittstelle dienen, sind in der `termiox`-Struktur festgelegt, die wiederum in der Header-Datei `<sys/termiox.h>` definiert wird. Einige `ioctl(2)`-Systemaufrufe, die diese Parameter verwenden oder ändern, greifen auf diese Struktur zu:

```
#define NFF          5
struct termiox {
    unsigned short    x_hflag;    /* Eingestellte Hardware
                                   Flußsteuerung */
    unsigned short    x_cflag;    /* Takterzeugungsart */
    unsigned short    x_rflag[NFF]; /* Reserviert */
    unsigned short    x_sflag;    /* Frei für eigene
                                   Benutzung */
};
```

Der `x_hflag` Eintrag beschreibt die hardwareseitige Flußsteuerung:

RTSXOFF	0000001	Erlaube RTS Flußsteuerung beim Empfangen.
CTSXON	0000002	Erlaube CTS Flußsteuerung beim Senden.
DTRXOFF	0000004	Erlaube DTR Flußsteuerung beim Empfangen.
CDXON	0000010	Erlaube CD Flußsteuerung beim Senden.
ISXOFF	0000020	Erlaube isochrone Flußsteuerung beim Empfangen.

Die EIA-232-D DTR und CD-Leitungen werden für den Verbindungsaufbau zwischen zwei Systemen verwendet. Darüberhinaus wird die RTS-Leitung für den Verbindungsaufbau mit einem Modem verwendet. Das bedeutet, daß DTR und RTS aktiviert werden, sobald ein asynchroner Port geöffnet wird. Falls DTR für hardwareseitige Flußsteuerung herangezogen wird, muß RTS für den Verbindungsaufbau benutzt werden. Falls CD für hardwareseitige Flußsteuerung herangezogen wird, muß CTS für den Verbindungsaufbau benutzt werden. Damit können RTS und DTR (bzw. CTS und CD) nicht gleichzeitig für die hardwareseitige Flußsteuerung genutzt werden. Weitere gegenseitige Ausschließungen können gelten, wie zum Beispiel das gleichzeitige Setzen von `termio(7)` HUPCL sowie den `termiox(7)` DTRXOFF Bits, die beide die "DTE Ready"-Leitung für verschiedene Zwecke verwenden.

Verschiedene Methoden der hardwareseitigen Flußsteuerung können durch Setzen der entsprechenden Bits ausgewählt werden. So kann beispielsweise die beidseitige RTS/CTS-Flußsteuerung durch Setzen der beiden Bits `RTSXOFF` und `CTSXON`

ausgewählt werden, die beidseitige DTR/CTS Flußsteuerung durch das Setzen von `DTRXOFF` und `CTSXON`. Für Modemkontrolle bzw. einseitige CTS-Flußsteuerung muß lediglich das Bit `CTSXON` gesetzt werden.

Wie bereits oben erwähnt, wird vorausgesetzt, daß der lokale asynchrone Adapter (z.B. der Rechner) als DTE konfiguriert ist. Falls das angeschlossene Gerät (z.B. ein Drucker) ebenfalls ein DTE ist, sollte es mit dem asynchronen Adapter über ein Null-Modem angeschlossen sein, das die Steuersignale (meistens RTS und CTS) vertauscht. In diesem Fall steuert das DTE auf der Sendeseite das RTS-Signal, und das Null-Modem vertauscht RTS und CTS, so daß das lokale DTE auf der Empfängerseite RTS als CTS empfängt. Falls das `CTSXON` Bit für die hardwareseitige Flußsteuerung gesetzt wird, würde ein Rücksetzen des RTS-Signals durch den Drucker zu einem Rücksetzen des CTS-Signals auf der Rechnerseite führen. Die Druckerausgabe wird bis zum Heraufsetzen des RTS Signals unterbrochen, was auf der Rechnerseite zum Heraufsetzen des CTS Signals führen würde.

Falls das Bit `RTSXOFF` gesetzt ist, wird die "Request To Send" (RTS)-Leitung heraufgesetzt und erst dann vom asynchronen Port heruntergesetzt, wenn die Eingabe unterbrochen werden soll. Ist die RTS-Leitung heruntergesetzt, sollte das angeschlossene Gerät seine Ausgabe bis zum erneuten Heraufsetzen von RTS ebenfalls unterbrechen.

Falls das Bit `CTSXON` gesetzt ist, werden nur dann Daten gesendet, wenn die "Clear To Send" (CTS)-Leitung vom angeschlossenen Gerät heraufgesetzt wird. Falls die CTS-Leitung vom angeschlossenen Gerät heruntergesetzt wird, wird die Ausgabe bis zum erneuten Heraufsetzen von CTS unterbrochen.

Falls das Bit `DTRXOFF` gesetzt ist, wird die "DTE Ready" (DTR)-Leitung heraufgesetzt; sie wird vom asynchronen Port wieder heruntergesetzt, wenn die Eingabe unterbrochen werden soll. Ist die DTR-Leitung heruntergesetzt, sollte das angeschlossene Gerät seine Ausgabe bis zum erneuten Heraufsetzen von DTR ebenfalls unterbrechen.

Falls das Bit `CDXON` gesetzt ist, werden nur dann Daten gesendet, wenn die "Received Line Signal Detector" (CD)-Leitung vom angeschlossenen Gerät heraufgesetzt wird. Falls die CD-Leitung vom angeschlossenen Gerät heruntergesetzt wird, wird die Ausgabe bis zum erneuten Heraufsetzen von CD unterbrochen.

Falls das Bit `ISXOFF` gesetzt ist und der isochrone Port die Eingabe unterbrechen möchte, wird das ausgehende Taktsignal angehalten. Das angeschlossene Gerät sollte dieses Taktsignal für die Erzeugung der eigenen Sendedaten verwenden. Die Taktgeber für Senden und Empfangen werden über den `x_cflag`-Eintrag programmiert. Falls der Port nicht für die externe Takterzeugung programmiert ist, wird das `ISXOFF` ignoriert. Die isochrone Flußsteuerung der Ausgangsdaten wird durch geeignete Programmierung des Taktgebers im `x_cflag`-Eintrag erreicht und dadurch für das angeschlossene Gerät zugelassen.

Das `x_cflag`-Feld bestimmt die Behandlung der verschiedenen Taktgenerierungarten.

XMTCLK	0000007	Quelle für Sendetakt.
XCIBRG	0000000	Erzeuge Sendetakt von internem Baudraten-Generator.
XCTSET	0000001	Erzeuge Sendetakt vom "transmitter signal element timing" Signal (DCE Quelle), CCITT V24 Leitung 114, EIA-232-D Pin 15.
XCRSET	0000002	Erzeuge Sendetakt vom "receiver signal element timing" Signal (DCE Quelle), CCITT V24 Leitung 115, EIA-232-D pin 17.
RCVCLK	0000070	Quelle für Empfangstakt.
RCIBRG	0000000	Erzeuge Empfangstakt von internem Baudraten-Generator.
RCTSET	0000010	Erzeuge Empfangstakt vom "transmitter signal element timing" Signal (DCE Quelle), CCITT V24 Leitung 114, EIA-232-D Pin 15.
RCRSET	0000020	Erzeuge Empfangstakt vom "receiver signal element timing" Signal (DCE Quelle), CCITT V24 Leitung 115, EIA-232-D pin 17.
TSETCLK	0000700	"Transmitter signal element timing" (DTE Quelle) CCITT V24 Leitung 113, EIA-232-D Pin 24, mit folgender Takterzeugung:
TSETCOFF	0000000	Keine TSET Takterzeugung.
TSETCRBRG	0000100	Lege Baudraten-Generator für Empfangsdaten auf Leitung 113.
TSETCTBRG	0000200	Lege Baudraten-Generator für Sendedata auf Leitung 113.
TSETCTSET	0000300	Lege Taktgeber für "transmitter signal element timing" (DCE Quelle) auf Leitung 113.
TSETCRSET	0000400	Lege Taktgeber für "receiver signal element timing" (DCE Quelle) auf Leitung 113.
RSETCLK	0007000	Receiver signal element timing" (DTE Quelle) CCITT V24 Leitung 128, kein EIA-232-D Pin, mit folgender Takterzeugung:
RSETCOFF	0000000	Keine RSET Takterzeugung.
RSETCRBRG	0001000	Lege Baudraten-Generator für Empfangsdaten auf Leitung 128.
RSETCTBRG	0002000	Lege Baudraten-Generator für Sendedata auf Leitung 128.
RSETCTSET	0003000	Lege Taktgeber für "transmitter signal element timing" (DCE Quelle) auf Leitung 128.
RSETCRSET	0004000	Lege Taktgeber für "receiver signal element timing" (DCE) auf Leitung 128.

Falls das XMTCLK Feld den Wert XCIBRG hat, wird der Sendetakt vom eingebauten Baudraten-Generator abgenommen, so wie dies für die asynchrone Datenübertragung üblich ist. Falls XMTCLK = XCTSET ist, wird der Sendetakt vom Taktgeber für "Transmitter Signal Element Timing" (DCE Quelle) abgenommen. Falls XMTCLK = XCRSET ist, wird der Sendetakt vom Taktgeber für "Receiver Signal Element Timing" (DCE Quelle) abgenommen.

Falls das `RCVCLK` Feld den Wert `RCIBRG` hat, wird der Empfangstakt vom eingebauten Baudraten-Generator abgenommen, so wie dies für die asynchrone Datenübertragung üblich ist. Falls `RCVCLK = RCTSET` ist, wird der Empfangstakt vom Taktgeber für "Transmitter Signal Element Timing" (DCE Quelle) abgenommen. Falls `RCVCLK = RCRSET` ist, wird der Empfangstakt vom Taktgeber für "Receiver Signal Element Timing" (DCE Quelle) abgenommen.

Falls das `TSETCLK` Feld den Wert `TSETCOFF` hat, wird kein Takt für "Transmitter Signal Element Timing" (DTE Quelle) erzeugt. Falls `TSETCLK = TSETCRBRG` ist, wird der Takt für "Transmitter Signal Element Timing" (DTE Quelle) durch den Baudraten-Generator für Empfangsdaten erzeugt. Falls `TSETCLK = TSETCTBRG` ist, wird der Takt für "Transmitter Signal Element Timing" (DTE Quelle) durch den Baudraten-Generator für Sendedaten erzeugt. Falls `TSETCLK = TSETCTSET` ist, wird der Takt für "Transmitter Signal Element Timing" (DTE Quelle) vom Taktgeber für "Transmitter Signal Element Timing (DCE Quelle) erzeugt. Falls `TSETCLK = TSETCRBRG` ist, wird der Takt für "Transmitter Signal Element Timing" (DTE Quelle) vom Taktgeber für "Receiver Signal Element Timing (DCE Quelle) erzeugt.

Falls das `RSETCLK` Feld den Wert `RSETCOFF` hat, wird kein Takt für "Receiver Signal Element Timing" (DTE Quelle) erzeugt. Falls `RSETCLK = RSETCRBRG` ist, wird der Takt für "Receiver Signal Element Timing" (DTE Quelle) durch den Baudraten-Generator für Empfangsdaten erzeugt. Falls `RSETCLK = RSETCTBRG` ist, wird der Takt für "Receiver Signal Element Timing" (DTE Quelle) durch den Baudraten-Generator für Sendedaten erzeugt. Falls `RSETCLK = RSETCTSET` ist, wird der Takt für "Receiver Signal Element Timing" (DTE Quelle) vom Taktgeber für "Transmitter Signal Element Timing" (DCE Quelle) erzeugt. Falls `RSETCLK = RSETCRBRG` ist, wird der Takt für "Receiver Signal Element Timing" (DTE Quelle) vom Taktgeber für "Receiver Signal Element Timing (DCE Quelle) erzeugt.

Das `x_rflag` Feld ist für zukünftige Erweiterungen reserviert und sollte daher nicht verwendet werden. Das `x_sflag` Feld steht für eigene Anpassungen der Terminal-Schnittstelle zur Verfügung; hierzu können die `termiox(7)` `ioctl`-Systemaufrufe verwendet werden.

IOCTL

Die `ioctl(2)` Systemaufrufe haben die Form:

```
ioctl(dateideskriptor, kommando, arg)
struct termiox *arg;
```

Die untenstehenden Kommandos verwenden diese Struktur:

TCGETX

Das Argument ist ein Zeiger auf eine `termiox`-Struktur. Die aktuellen Terminal-Parameter werden ermittelt und in dieser Struktur zurückgegeben.

TCSETX

Das Argument ist ein Zeiger auf eine `termiox`-Struktur. Die neuen Terminal-Parameter werden mit den in dieser Struktur enthaltenen Werten gesetzt und haben sofortige Wirkung.

TCSETXW

Das Argument ist ein Zeiger auf eine `termiox`-Struktur. Die neuen Terminal-Parameter werden mit den in dieser Struktur enthaltenen Werten gesetzt; die Änderungen werden erst wirksam, nachdem alle Zeichen in der Ausgabewarteschlange übertragen wurden. Dieser Aufruf sollte immer dann benutzt werden, wenn die Änderung von Parametern auch die Terminal-Ausgabe betrifft.

TCSETXF

Das Argument ist ein Zeiger auf eine `termiox`-Struktur. Die neuen Terminal-Parameter werden mit den in dieser Struktur enthaltenen Werten gesetzt. Die Änderungen werden erst wirksam, nachdem alle Zeichen in der Ausgabewarteschlange übertragen wurden; eventuell vorliegende Zeichen in der Eingabewarteschlange werden ignoriert.

DATEIEN

`/dev/*`

SIEHE AUCH

`stty(1)`, `ioctl(2)`, `termio(7)`.

BEZEICHNUNG

ttcompat – Module kompatibel zu V7, 4BSD und XENIX STREAMS

ÜBERSICHT

```
#include <sys/stream.h>
#include <sys/stropt.h>
#include <sys/ttcompat.h>
#include <sys/ttold.h>

ioctl(fd, I_PUSH, "ttcompat");
```

BESCHREIBUNG

ttcompat ist ein STREAMS-Modul, der den ioctl-Aufruf älterer Terminaltreiber in den Aufruf von ioctl überführt, wie er von der neuen termio-Schnittstelle unterstützt wird (siehe termio(7)). Mit "älteren Treibern" sind hier frühere Terminaltreiber aus Version 7, 4BSD und XENIX gemeint. Nicht betroffene Teile in den Nachrichten werden unverändert durchgereicht. Das Verhalten der Aufrufe read und write bleibt identisch zu früher. Das gleiche gilt für alle Aufrufvarianten von ioctl, die von ttcompat nicht unterstützt werden.

Es ist nicht nötig, das ttcompat-Modul explizit auf den Stream-Stack zu pushen. Dies kann mit dem autopush-Mechanismus automatisch erledigt werden. Die Verwendung dieses Moduls setzt voraus, daß alle anderen Moduln die termios-Schnittstelle benutzen und daß das Anwenderprogramm in der Lage ist, den Treiber auf den Stream-Stack zu pushen. Außerdem müssen sie alle ioctl-Aufrufe unterstützen, in denen TCGETS, TCSETS und TCSETSF angegeben ist. Ist das nicht der Fall und werden diese Aufrufe dazu verwendet, irgendwelche Werte zu setzen oder Informationen auszulesen, kann das dazu führen, daß bestimmte Funktionen der V7/4BSD/XENIX Versionen nicht korrekt ablaufen. Es folgen einige Beispiele hierfür. Sind in dem Feld c_cflag die Bits CBAUD nicht versorgt, dann werden die Funktionen nicht unterstützt, die durch die Felder sg_ispeed und sg_ospeed in der Struktur sgttyb zur Verfügung gestellt werden (siehe weiter unten). Eine fehlende Unterstützung des Aufrufs TCFLSH ioctl hat zur Folge, daß der Aufruf TIOCFUSH ioctl nicht unterstützt wird. Eine fehlende Unterstützung des Aufrufs TCXONC ioctl hat die Konsequenz, daß die durch die Aufrufe TIOCSTOP ioctl und TIOCSTART ioctl ausgelösten Funktionen nicht unterstützt werden. Eine fehlende Unterstützung der Aufrufe TIOCMBS ioctl und TIOCMBS ioctl führt dazu, daß die durch die Aufrufe TIOCSDBR ioctl und TIOCCDBR ioctl ausgelösten Funktionen nicht unterstützt werden.

Die Grundfunktionen der ioctl-Aufrufe verwenden die Struktur sgttyb, die folgendermaßen definiert ist:

```
struct sgttyb {
    char    sg_ispeed;
    char    sg_ospeed;
    char    sg_erase;
    char    sg_kill;
    int     sg_flags;
};
```

Die beiden Felder `sg_ispeed` und `sg_ospeed` beschreiben die Gerätegeschwindigkeit bei der Ein- und Ausgabe. Sie entsprechen dem Inhalt des Felds `c_cflag` in der Struktur `termios`. Die beiden Felder `sg_erase` und `sg_kill` der obigen Struktur legen das Löschr- bzw. das Abbruchzeichen fest und entsprechen den Elementen `VERASE` und `VKILL` des Felds `c_cc` in der `termios` Struktur.

Das Feld `sg_flags` der obigen Struktur enthält diverse Schalter, die die Behandlung des Terminals durch das System bestimmen. Sie werden auf entsprechende Schalter von Feldern der `termios` Struktur abgebildet.

Der Verzögerungstyp 0 wird stets auf sein Äquivalent im Feld `c_oflag` der `termios` Struktur abgebildet. Die Abbildung anderer Verzögerungen wird wie folgt durchgeführt:

<code>sg_flags</code>	<code>c_oflag</code>
BS1	BS1
FF1	VT1
CR1	CR2
CR2	CR3
CR3	nicht unterstützt
TAB1	TAB1
TAB2	TAB2
XTABS	TAB3
NL1	ONLRET CR1
NL2	NL1

Unter der Voraussetzung, daß der `raw`-Modus nicht gesetzt ist und vorausgehende `ioctl`-Aufrufe, in denen `TIOCLSET` oder `TIOCLBIS` angegeben war, weder den Modus `LITOUT` noch `PASS8` eingestellt haben, gilt folgendes. Im Feld `c_iflag` der `termios`-Struktur wird der Schalter `ISTRIP` gesetzt. Außerdem steuern die beiden Schalter `EVENP` und `ODDP` die Zeichenparität der Zeichen vom und zum Terminal folgendermaßen:

Keine Paritätsprüfung bei der Eingabe und keine Erzeugung von Paritätsbits bei der Ausgabe;

die Zeichengröße wird auf `CS8` eingestellt und im Feld `c_cflag` der `termios`-Struktur wird der Schalter rückgesetzt.

Für Ein- und Ausgabe soll gerade Parität gelten;

die Zeichengröße wird auf `CS7` eingestellt und in den Feldern `c_iflag` und `c_cflag` der `termios`-Struktur wird der Schalter gesetzt.

Für Ein- und Ausgabe soll ungerade Parität gelten;

die Zeichengröße wird auf `CS7` eingestellt und in den Feldern `c_iflag` und `c_cflag` der `termios`-Struktur wird der Schalter gesetzt.

Für die Ausgabe soll gerade Parität gelten und bei der Eingabe soll beliebige Parität zugelassen sein;

die Zeichengröße wird auf `CS7` eingestellt und in den Feldern der `termios`-Struktur wird der Schalter in `c_iflag` gelöscht und in `c_cflag` gesetzt.

Ist der Schalter für `raw` gesetzt, findet keinerlei Bearbeitung von Ein- oder Ausgabe statt. Bezüglich der Ausgabe werden durch ihn die Schalter `OPOST` im Feld `c_oflag` und `XCASE` im Feld `c_lflag` der `termios`-Struktur rückgesetzt. Bezüglich der Eingabe werden im Feld `c_iflag` der `termios` Struktur alle Schalter außer `IXOFF` und `IXANY` rückgesetzt. Für Ein- und Ausgabe gilt eine Zeichenlänge von 8 Bit; sie ist demzufolge auf `CS8` gesetzt. Die beiden Schalter `PARENB` und `PARODD` im `c_cflag` Feld der `termios`-Struktur werden rückgesetzt. Die Steuerzeichen für Signalerzeugung und Zeilenorientierung werden außer Kraft gesetzt, indem die beiden Schalter `ISIG` und `ICANON` im `c_lflag` Feld der `termios`-Struktur rückgesetzt werden.

Ist der Schalter `CRMOD` gesetzt, werden bei der Eingabe Wagenrücklaufzeichen in Neuezeile-Zeichen umgesetzt und bei der Ausgabe wird vor Neuezeile-Zeichen ein Wagenrücklaufzeichen eingefügt. Außerdem werden in der `termios`-Struktur im Feld `c_iflag` der Schalter `ICRNL` und im Feld `c_oflag` die beiden Schalter `OPOST` und `ONLCR` gesetzt.

Ist der Schalter `LCASE` gesetzt, werden bei der Eingabe alle Großbuchstaben des ASCII-Zeichensatzes durch ihre Kleinbuchstaben und bei der Ausgabe alle Kleinbuchstaben des ASCII-Zeichensatzes durch ihre Großbuchstaben ersetzt. Bezüglich der Eingabe wird hierzu im Feld `c_iflag` der Schalter `IUCUC` und bezüglich der Ausgabe im Feld `c_oflag` der Schalter `OLCUC` gesetzt. Um auch bestimmte ASCII-Zeichen zuzulassen, die von älteren Terminals nicht unterstützt werden, werden für die Eingabe Escape-Sequenzen zugelassen und bei der Ausgabe entsprechend erzeugt. In diesem Fall ist der Schalter `XCASE` im Feld `c_lflag` gesetzt.

Die übrigen Schalter werden direkt auf die entsprechenden Schalter in der `termios`-Struktur wie folgt abgebildet:

<code>sg_flags</code>	Schalter in der <code>termios</code> -Struktur
<code>CBREAK</code>	Komplement von <code>ICANON</code> im <code>c_lflag</code> Feld
<code>ECHO</code>	<code>ECHO</code> im <code>c_lflag</code> Feld
<code>TANDEM</code>	<code>IXOFF</code> im <code>c_iflag</code> Feld

Eine weitere jedem Terminal zugeordnete Struktur beschreibt Zeichen besonderer Bedeutung, die sowohl für Terminalschnittstellen der älteren Version 7 als auch der neueren 4BSD Version gelten. Diese Struktur ist folgendermaßen definiert:

```
struct tchars {
    char  t_intrc;      /* Unterbrechung */
    char  t_quite;      /* Abbruch */
    char  t_startc;     /* Ausgabestart */
    char  t_stopc;      /* Ausgabestop */
    char  t_eofc;       /* Dateiende-Zeichen */
    char  t_brkc;       /* Eingabeabschluß (z.B. Neue-Zeile) */
};
```

Unter XENIX hat die Struktur `tchar` den Namen `tc`. Deren Elemente vom Typ `char` werden wie folgt auf die Elemente des Feldes `c_cc` der `termios`-Struktur abgebildet:

<code>tchars</code>	<code>c_cc</code> index
<code>t_intrc</code>	<code>VINTR</code>
<code>t_quitc</code>	<code>VQUIT</code>
<code>t_startc</code>	<code>VSTART</code>
<code>t_stopc</code>	<code>VSTOP</code>
<code>t_eofc</code>	<code>VEOF</code>
<code>t_brkc</code>	<code>VEOL</code>

Weiter gibt es für jedes Terminal noch lokale Schalter. Sie sind in einem Element vom Typ `word` abgelegt und unterstützen Schalter der neuen 4BSD-Terminalschnittstelle. Die meisten davon werden direkt wie folgt auf die Schalter der `termios`-Struktur abgebildet:

lokale Schalter	Schalter in der <code>termios</code> -Struktur
<code>LCRTBS</code>	nicht unterstützt
<code>LPRTERA</code>	<code>ECHOPRT</code> im <code>c_lflag</code> -Feld
<code>LCRTERA</code>	<code>ECHOE</code> im <code>c_lflag</code> -Feld
<code>LTILDE</code>	nicht unterstützt
<code>LTOSTOP</code>	<code>TOSTOP</code> im <code>c_lflag</code> -Feld
<code>LFLUSHO</code>	<code>FLUSHO</code> im <code>c_lflag</code> -Feld
<code>LNOHANG</code>	<code>CLOCAL</code> im <code>c_cflag</code> -Feld
<code>LCRTKIL</code>	<code>ECHOKE</code> im <code>c_lflag</code> -Feld
<code>LCTLECH</code>	<code>CTLECH</code> im <code>c_lflag</code> -Feld
<code>LPENDIN</code>	<code>PENDIN</code> im <code>c_lflag</code> -Feld
<code>LDECCTQ</code>	Komplement von <code>IXANY</code> im <code>c_iflag</code> -Feld
<code>LNOFLSH</code>	<code>NOFLSH</code> im <code>c_lflag</code> -Feld

Eine weitere, Terminals zugeordnete Struktur ist die `ltchars`-Struktur, in der die Steuerzeichen der neuen 4BSD-Terminalschnittstelle definiert sind. Sie ist folgendermaßen aufgebaut:

```
struct ltchars {
    char    t_suspc;        /* Prozeßhaltesignal */
    char    t_dsuspc;       /* verzögertes Prozeßhaltesignal */
    char    t_rprntc;       /* Zeile wiederholen */
    char    t_flushc;       /* Ausgabe leeren (Flipflop) */
    char    t_werasc;       /* Wort löschen */
    char    t_lnextc;       /* nächstes Zeichen unverändert */
};
```


Die Elemente vom Typ `char` werden wie folgt auf Elemente des Feldes `c_cc` der `termios`-Struktur abgebildet:

<code>ltchars</code>	<code>c_cc index</code>
<code>t_suspc</code>	<code>VSUSP</code>
<code>t_dsuspc</code>	<code>VDSUSP</code>
<code>t_rprntc</code>	<code>VREPRINT</code>
<code>t_flushc</code>	<code>VDISCARD</code>
<code>t_werasc</code>	<code>VWERASE</code>
<code>t_lnextc</code>	<code>VLNEXT</code>

IOCTL

Im folgenden sind die `ioctl`-Aufrufe aufgeführt, auf die `ttcompat` reagiert. Alle anderen Aufrufe werden an den darunterliegenden Modul durchgereicht.

<code>TIOCGTP</code>	Lesen des aktuellen Terminalstatus. Als Argument wird ein Zeiger auf eine <code>sgttyb</code> -Struktur übergeben, in der die einzelnen Zeichen sowie Werte wie Geschwindigkeit für Ein- und Ausgabe abgespeichert und die aktuellen Schalter des Terminals in geeigneter Form in dem Feld <code>sg_flags</code> untergebracht werden.
<code>TIOCEXCL</code>	Einschalten des Exklusivmodus. Solange die Datei nicht geschlossen wird, werden keine weiteren Open-Aufrufe zugelassen.
<code>TIOCNXCL</code>	Ausschalten des Exklusivmodus.
<code>TIOCSPT</code>	Setzen des aktuellen Terminalstatus. Als Argument wird ein Zeiger auf eine <code>sgttyb</code> -Struktur übergeben, deren Inhalt der Neudefinierung des Terminalzustands dient. Dazu werden die einzelnen Zeichen und die Geschwindigkeit für Ein- und Ausgabe aus der Struktur in die Statusbeschreibung des Terminals übertragen und die Schalter werden entsprechend denen im <code>sg_flags</code> gesetzt. Das Ändern des Zustands wird mittels des <code>TCSETSF</code> <code>ioctl</code> -Aufrufs erreicht. Dabei wird solange gewartet, bis keine Ausgabe mehr ansteht und bevor der Terminalzustand geändert wird, werden alle bis dahin noch nicht gelesenen Zeichen verworfen.
<code>TIOCSETN</code>	Ändert den Terminalzustand ähnlich wie der <code>TIOCSPT</code> <code>ioctl</code> -Aufruf. Allerdings wird hier weder gewartet, noch werden Eingabezeichen verworfen. Als Argument wird ein Zeiger auf eine <code>sgttyb</code> -Struktur übergeben.
<code>TIOCHPCL</code>	Setzt im Terminal in <code>c_cflag</code> den Schalter <code>HUPCL</code> . Ein eventuelles Argument bleibt unberücksichtigt.
<code>TIOCFUSH</code>	Als Argument wird ein Zeiger auf eine Variable vom Typ <code>int</code> übergeben. Ist deren Inhalt Null, werden alle Zeichen in Warteschlangen für Ein- oder Ausgabe gelöscht. Andernfalls wird ihr Inhalt zu einer logischen ODER -Verknüpfung mit den Schaltern <code>FREAD</code> und <code>FWRITE</code> verwendet, die in <code><sys/file.h></code> definiert sind. Ist im Ergebnis das Bit <code>FREAD</code> gesetzt, werden alle Zeichen in der Eingabe gelöscht. Ist im Ergebnis das Bit <code>FWRITE</code> gesetzt, werden alle Zeichen in der Ausgabe gelöscht.

TIOCBRK	Setzt das Break-Bit für dieses Gerät. Ein eventuelles Argument bleibt unberücksichtigt.
TIOCCBRK	Setzt das Break-Bit für dieses Gerät zurück. Ein eventuelles Argument bleibt unberücksichtigt.
TIOCSDR	Setzt das DTR-Bit (Data Terminal Ready) für dieses Gerät. Ein eventuelles Argument bleibt unberücksichtigt.
TIOCCDR	Setzt das DTR-Bit (Data Terminal Ready) für dieses Gerät zurück. Ein eventuelles Argument bleibt unberücksichtigt.
TIOCSTOP	Die Ausgabe wird angehalten, als ob das STOP Zeichen eingegeben worden wäre. Ein eventuelles Argument bleibt unberücksichtigt.
TIOCSTART	Die Ausgabe wird weitergeführt, als ob das START Zeichen eingegeben worden wäre. Ein eventuelles Argument bleibt unberücksichtigt.
TIOCGTC	Lesen des aktuellen Terminalzustands. Als Argument wird ein Zeiger auf eine tchars-Struktur übergeben, die die gelesene Information aufnimmt.
TIOCSETC	Versetzt das Terminal in einen neuen Zustand. Als Argument wird ein Zeiger auf eine tchars-Struktur übergeben, deren Inhalt den neuen Zustand des Terminals beschreibt.
TIOCLGET	Lesen des aktuellen Terminalzustands. Als Argument wird ein Zeiger auf eine Variable vom Typ int übergeben, der als Zieladresse für die aus den ausgelesenen Schaltern gebildeten lokalen Schalter dient.
TIOCLBIS	Lesen und Neusetzen des aktuellen Terminalzustands. Als Argument wird ein Zeiger auf eine Variable vom Typ int übergeben, deren Inhalt eine Maske der einzustellenden lokalen Schalter darstellt. Aus den Schaltern des Terminals werden die Werte der lokalen Schalter bestimmt und gemäß den in der Maske angegebenen Werten neu gesetzt und anschließend im Terminal gesetzt.
TIOCLBIC	Lesen und Neusetzen des aktuellen Terminalzustands. Als Argument wird ein Zeiger auf eine Variable vom Typ int übergeben, deren Inhalt eine Maske der rückzusetzenden lokalen Schalter darstellt. Aus den Schaltern des Terminals werden die Werte der lokalen Schalter bestimmt und gemäß den in der Maske angegebenen Werten rückgesetzt und anschließend im Terminal rückgesetzt.
TIOCLSET	Als Argument wird ein Zeiger auf eine Variable vom Typ int übergeben, deren Inhalt eine Reihe neuer lokaler Schalter darstellt. Dementsprechend werden die Schalter im Terminal gesetzt.
TIOCLTC	Als Argument wird ein Zeiger auf eine ltchars-Struktur übergeben, in der die Werte des Terminalzustands geeignet abgespeichert werden.
TIOCSLTC	Als Argument wird ein Zeiger auf eine ltchars-Struktur übergeben, deren Werte zum Setzen des neuen Terminalzustands gelesen werden.

FIORDCHK	FIORDCHK gibt die Anzahl der sofort für das Lesen verfügbaren Zeichen zurück. Ein eventuelles Argument bleibt unberücksichtigt.
FIONREAD	Als Argument wird ein Zeiger auf eine Variable vom Typ <code>int</code> übergeben. FIONREAD gibt die Anzahl der sofort für das Lesen verfügbaren Zeichen in dieser Variablen zurück.
LDSMAP	Ruft die Funktion <code>emsetmap</code> (<i>tp</i> , <i>mp</i>) auf, vorausgesetzt, sie ist im Systemkern konfiguriert.
LDGMAP	Ruft die Funktion <code>emgetmap</code> (<i>tp</i> , <i>mp</i>) auf, vorausgesetzt, sie ist im Systemkern konfiguriert.
LDNMAP	Ruft die Funktion <code>emunmap</code> (<i>tp</i> , <i>mp</i>) auf, vorausgesetzt, sie ist im Systemkern konfiguriert.

Aus Kompatibilitätsgründen werden die folgenden Aufrufe von `ioctl`s als fehlerfrei durchgeführt rückgemeldet, obwohl sie offensichtlich nichts ausgerichtet haben (d.h. der Terminalzustand ist überhaupt nicht verändert).

TIOCSETD	LDOPEN
TIOCGETD	LDCLOSE
DIOCSETP	LDCHG
DIOCSETP	LDSETT
DIIOGETP	LDGETT

SIEHE AUCH

`ioctl(2)`, `termios(2)`, `termio(7)`, `ldterm(7)`.

HINWEIS

TIOCBRK und TIOCCBRK sollten vom Treiber behandelt werden. FIONREAD und FIORDCHK sollten vom Stream-Kopf behandelt werden.

BEZEICHNUNG

tty – Schnittstelle des steuernden Terminals

ÜBERSICHT

/dev/tty

BESCHREIBUNG

Für jeden Prozeß einer Prozeßgruppe, die (wenn überhaupt) ein steuerndes Terminal besitzt, ist der Dateiname `/dev/tty` eine synonyme Bezeichnung für dieses Terminal. Programme oder Shell-Prozeduren, die sichergehen wollen, daß auszugebende Meldungen auch auf dem Terminal landen, ganz gleich, wie ihre Ausgabe auch umgelenkt sein mag, können diese Bezeichnung wählen. Das gleiche gilt auch, wenn Programme für ihre Ausgabe einen Dateinamen brauchen, es aber zu mühsam ist, das zur Zeit aktive Terminal ausfindig zu machen.

DATEIEN

/dev/tty

/dev/tty*

SIEHE AUCH

`console(7)`, `ports(7)`.

BEZEICHNUNG

UDP – Protokoll für Internet-Datagramme

ÜBERSICHT

```
#include <sys/socket.h>
#include <netinet/in.h>

s = socket(AF_INET, SOCK_DGRAM, 0);
t = t_open("/dev/udp", O_RDWR);
```

BESCHREIBUNG

UDP ist ein einfaches Datagramm-Protokoll, das als Schicht unmittelbar über dem Internet-Protokoll angesiedelt ist (IP). Programme können auf zweierlei Art auf UDP zugreifen. Zum einen über die Socket-Schnittstelle, wo es den Socket-Typ `SOCK_DGRAM` unterstützt. Zum andern über die TLI-Schnittstelle (Transport Level Interface), wo es den verbindungsfreien Servicetyp (`T_CLTS`) unterstützt.

Bei Verwendung der Socket-Schnittstelle wird UDP normalerweise in Verbindung mit den Aufrufen `sendto()`, `sendmsg()`, `recvfrom()` und `recvmsg()` benutzt (siehe `send(2)` und `recv(2)`). Ist der Bestimmungsort für weitere Pakete mit Hilfe des Aufrufs `connect(2)` festgelegt, können auch die Aufrufe `recv(2)` oder `read(2)` und `send(2)` oder `write(2)` verwendet werden.

Das Format der UDP-Adressen ist identisch zu dem des Transmission Control Protocol (TCP). UDP benutzt zur Identifizierung des Kommunikationsendpunkts, wie TCP auch, eine Schnittstellenummer und eine Adresse. Die beiden Mengen der Schnittstellenummern für UDP und TCP sind disjunkt (d.h., daß ein UDP-Anschluß nicht mit einem TCP-Anschluß verbunden werden kann). Zur Einstellung der lokalen Adressen und der Schnittstellenummer eines UDP-Socket kann der `bind(2)`-Aufruf verwendet werden. Wird in dem `bind`-Aufruf der spezielle Wert `INADDR_ANY` verwendet, braucht man die lokale IP-Adresse nicht anzugeben. Wird `bind()` nicht aufgerufen, so erfolgt die Zuordnung von lokaler IP-Adresse und Schnittstellenummer an den Kommunikationsendpunkt zum Zeitpunkt der ersten Paketübermittlung. Unter der Voraussetzung, daß das zugrundeliegende Netzwerk `broadcast` unterstützt wird, können solche Pakete unter Verwendung einer reservierten `broadcast`-Adresse verschickt werden. Diese Adresse hängt von der Netzwerk-Schnittstelle ab. Rundsprüche dürfen nur von privilegierten Benutzern versandt werden.

Optionen, die auf der Ebene von IP gültig sind, dürfen auch im Zusammenhang mit UDP verwendet werden (siehe `ip(7)`).

UDP-Pakete können auf die unterschiedlichste Art und Weise verloren gehen oder fehlerhaft übertragen werden. Dazu zählen auch Fehler im zugrundeliegenden Kommunikationsmechanismus. UDP bildet über den Datenanteil im Paket eine Prüfsumme. Ist in einem empfangenen Paket diese Prüfsumme nicht in Ordnung, wird dieses Paket ohne jeden Hinweis an den Benutzer verworfen. Für jeden UDP-Socket wird eine eigene Warteschlange eingerichtet, deren Kapazität begrenzt ist. Wird durch eintreffende Pakete die Höchstgrenze der Aufnahmekapazität überschritten, werden diese Pakete ohne jeden Hinweis verworfen.

UDP verarbeitet Fehlermeldungen des ICMP, die als Antwort auf versandte UDP-Pakete eintreffen. Diese Meldungen gehorchen dem Format des Internet-Protokolls für Steuermeldungen (Internet Control Message Protocol – ICMP). Siehe auch `icmp(7)`. ICMP-Meldungen bezüglich gelöschter Nachrichten seitens der Quelle bleiben unberücksichtigt. ICMP-Meldungen mit der Aussage "unerreichbares Bestimmungsziel", "Zeitüberlauf" oder "Parameterproblem" führen dazu, daß die Verbindung der Socket-Schnittstelle zu ihrem Peer gelöst wird. Nachfolgende Versuche, über diese Socket-Schnittstelle Pakete zu versenden, werden mit einem Fehler beendet. UDP stellt nicht sicher, daß die Pakete auch in derselben Reihenfolge ausgeliefert werden, in der sie verschickt wurden. Außerdem kann es vorkommen, daß Paketduplikate auftauchen.

SIEHE AUCH

`read(2)`, `write(2)`, `bind(3N)`, `connect(3N)`, `recv(3N)`, `send(3N)`, `icmp(7)`, `inet(7)`, `ip(7)`, `tcp(7)`.

Postel, Jon, *User Datagram Protocol*, RFC 768, Network Information Center, SRI International, Menlo Park, Calif., August 1980.

FEHLERMELDUNGEN

Im folgenden sind mögliche Fehlerfälle einer Socket-Operation aufgezählt:

EISCONN	Es wurde der Versuch unternommen, eine <code>connect()</code> -Operation auf eine Socket-Schnittstelle anzuwenden, für die diese Operation bereits erfolgt war. Es war jedoch nicht möglich, die alte Verbindung erfolgreich aufzulösen, um die neue herzustellen.
EISCONN	Es wurde der Versuch unternommen, mittels <code>sendto()</code> oder <code>sendmsg()</code> an die darin angegebene Adresse eine Nachricht zu schicken. Diese Adresse bezog sich jedoch auf eine Socket-Schnittstelle, für die bereits eine <code>connect()</code> -Operation erfolgt war.
ENOTCONN	Es wurde der Versuch unternommen, entweder mittels <code>send()</code> oder <code>write()</code> oder aber mittels <code>sendto()</code> oder <code>sendmsg()</code> eine Nachricht zu verschicken, ohne daß eine Zieladresse spezifiziert war. Für die betroffene Socket-Schnittstelle war jedoch noch keine <code>connect()</code> -Operation erfolgt.
EADDRINUSE	Mittels <code>bind()</code> wurde eine Verbindungsherstellung versucht. Das angegebene Paar Adresse/Anschluß bezieht sich jedoch auf eine bereits verbundene Socket-Schnittstelle.
EADDRNOTAVAIL	Es wurde der Versuch unternommen, eine <code>bind()</code> -Operation auf eine Socket-Schnittstelle anzuwenden, für deren Adresse im Netzwerk keine Schnittstelle existiert.
EINVAL	Es wurde der Versuch unternommen, eine <code>sendmsg()</code> -Operation durchzuführen, wobei für <code>msg_accrights</code> ein Wert ungleich NULL angegeben war.

UDP(7)

UDP(7)

EACCES

Es wurde durch einen nicht privilegierten Benutzer der Versuch unternommen, eine `bind()`-Operation mit einer reservierten Schnittstellennummer durchzuführen.

ENOBUFS

Der Speicherbedarf für interne Datenstrukturen konnte nicht befriedigt werden.

BEZEICHNUNG

xmt – Xylogics 472 MB-I Schnittstelle für Magnetband

BESCHREIBUNG

Xylogics 472 und ein "Cipher formatted interface transport" nach Industrie-Standard ergeben zusammen eine standardisierte Schnittstelle für Bandlaufwerke. Geschwindigkeit und Aufzeichnungsdichte wählen Sie entsprechend der Transportart und des Formatierers über bestimmte Bits der Gerätenummer (minor device number) aus.

Die Bits haben im einzelnen folgende Bedeutung:

Die beiden niederwertigsten Bits wählen die logische Einheit (logical unit) aus. Es werden bis zu vier Transporteinheiten bei jeder Kombination von Controller und Formatierer unterstützt. Das dritte Bit gibt an, daß das Band nicht zurückgespult wird, wenn die zugehörige Gerätedatei geschlossen wird.

Die Anzahl der MULTIBUS-Maps, die bei idconfig(1M) angegeben ist, sollte mindestens 10 betragen. Der Controller für das Bandlaufwerk muß auf 24-Bit-Adressierung eingestellt sein.

AUSWAHL VON SCHREIBDICHTEN UND GESCHWINDIGKEIT

Bit 010 und Bit 0100 der Gerätenummer legen die Aufzeichnungsdichte des Bandes fest, Bit 040 die Geschwindigkeit.

Bit 010	Bit 0100	Bit 040	Knotenname	Dichte	Geschwindigkeit
0	0	0	rmt0, rmt4	1600 bpi	50 ips
0	0	1	rmt32, rmt36	1600 bpi	100 ips
1	0	0	rmt8, rmt12	3200 bpi	50 ips
1	0	1		3200 bpi	100 ips *)
0	1	0	rmt64, rmt68	800 bpi	50 ips
0	1	1	rmt96, rmt100	800 bpi	100 ips
1	1	0	rmt72, rmt76	6250 bpi	50 ips
1	1	1	rmt104, rmt108	6250 bpi	100 ips

*) Nicht unterstützt bei FS1000/FS2000

BINÄRE KONFIGURATION

Soll ein neuer Kernel mit MULTIBUS Magnetband erzeugt werden, muß die binäre Konfigurationsdatei /etc/conf/pack.d/xmt/space.c auf dem aktuellsten Stand sein. Sie enthält u.a. folgende Konstanten und Variablen:

XMTCTLR Die maximale Anzahl der vom Treiber unterstützten xmt-Controller.

xmtdensel Wird auf eins gesetzt, falls bei dem Bandlaufwerk die Schreibdichte ausgewählt werden kann. Manche Cipher-Bandlaufwerke unterstützen nur 1600-bpi-Operationen.

xmtunits Dieses Feld beschreibt die Controller- und Transport-Zuweisungen, die das System unterstützt. Mit Wildcards kann der Treiber gezwungen werden, jeden Controller auf die angegebenen Bandlaufwerke hin zu überprüfen. Das Feld wird mit der

logischen Einheitennummer indiziert, die in der Gerätenummer (minor device number) enthalten ist. Das folgende Beispiel beschreibt zwei mögliche Controller mit jeweils zwei Bandlaufwerken:

```
struct xmt_unit xmtunits[] = {
/*  ctrl  drive  */
  { ANY,   ANY }, /* xmt0: any controller, any
                    drive */
  { ANY,   1 }, /* xmt1: any controller, drive 1 */
  { ANY,   0 }, /* xmt2: any controller, drive 0 */
  { ANY,   1 }, /* xmt3: any controller, drive 1 */
};
```

DATEIEN

/dev/mtxxx Gerätedateien für den Zugriff auf Magnetband im Raw-Modus

/etc/conf/pack.d/xmt/space.ci

/etc/conf/sdevice.d/xmt Binäre Konfigurationsdatei

SIEHE AUCH

cpio(1), tar(1)

FEHLERMELDUNGEN

xmt%d: drive %d ctrl %d conflict. Ein Laufwerk wurde in der binären Konfiguration nicht mit Wildcards angegeben, aber das gewünschte Laufwerk war bereits in Gebrauch und einer vorher überprüften Gerätedatei zugeordnet.

xmt%d: no write ring. Beim Versuch, auf dem Bandlaufwerk zu schreiben, fehlte der Schreibring.

xmt%d: not online: drive status = %x. Das Band, auf das zugegriffen werden sollte, war im Zustand offline.

xmt%d: can't change density in mid-tape. Es wurde versucht, mit einer anderen Dichte auf ein Band zu schreiben als dort angegeben war.

xmt%d: hard error bn%d er=0x%x. Beim Block *bn* trat ein Bandfehler auf. Der Fehlercode wird im Xylogics-Handbuch beschrieben. Falls nicht im Raw-Modus gearbeitet wird, führt jeder Fehler zum Abbruch. Bevor der Fehler gemeldet wird, haben Treiber oder Controller bereits mehrfach versucht, die fehlgeschlagene Operation zu wiederholen.

xmt%d: recovered error bn%d. Der Controller konnte einen Fehler beim Lesen oder Schreiben korrigieren.

FEHLER

Wird nicht im Raw-Modus gearbeitet und ein Fehler erkannt, der nicht mit den Daten zusammenhängt, so tut der Treiber so lange nichts mehr, bis die Gerätedatei geschlossen wird.

Wegen Hardware-Einschränkungen erkennt der Treiber `xmt` nicht den Unterschied zwischen einem fehlenden Bandlaufwerk und einem Bandlaufwerk, in dem kein Band eingelegt ist. Daher scheinen alle konfigurierten Bandlaufwerke an jedem Controller im System vorhanden zu sein, wenn die Laufwerke überprüft werden. Bei der binären Konfiguration sind Wildcards mit Vorsicht zu verwenden.

Wegen Hardware-Einschränkungen ist es nicht möglich, mehr als ein Bandlaufwerk an einem Controller seriell zu verketten.

BEZEICHNUNG

xt – Multiplexer STREAMS-Treiber für fensterfähige Terminals von AT&T

BESCHREIBUNG

Der xt-Treiber stellt virtuelle TTY-Kanäle (siehe `tty(7)`) zur Verfügung, die im Multiplexbetrieb auf STREAMS-Gerätetreibern aufsetzen. Der Pseudogerätetreiber xt ist STREAMS-orientiert und als höherer Multiplexer zwischen Stream-Kopf und Hardware-Gerätetreiber angesiedelt.

Virtuelle TTY-Kanäle werden durch zeichenorientierte Gerätedateien angesprochen, die folgenden Namenskonventionen gehorchen. Sie sind von der Form `/dev/xt/???`, wobei die letzten drei Zeichen Ziffern sind, von denen die ersten beiden die Kanalnummer bezeichnen. Die dritte Ziffer ist aus dem Bereich von 0-7 und steht für die virtuelle TTY-Nummer der Kanalgruppe (vgl. `tty(7)`). Gibt man beim Öffnen einer solchen Gerätedatei den Schalter `O_EXCL` an und wählt den Namen der Gerätedatei so, daß das letzte Zeichen 0 ist, wird eine neue Kanalgruppe dynamisch erzeugt. Nachdem das Öffnen erfolgreich verlaufen ist, sollte die TTY-Datei für die Multiplex-Kanäle an xt durchgereicht werden. Dazu wird in der `streamio(7)` Anforderung `I_LINK` gesetzt. Ab da verhalten sich alle TTY-Kanäle der Gruppe wie normale TTY-Dateien und die Daten werden in Paketen über die echte TTY-Leitung transportiert.

Das implementierte xt-Treiberprotokoll ist in `xtproto(5)` und in `layers(5)` beschrieben. Die Pakete sind von der Form, wie sie in `xtproto(5)` beschrieben sind, während die Inhalte der Pakete der Beschreibung in `layers(5)` genügen.

xt kennt vier Gruppen von `ioctl(2)` Anforderungen. Die erste enthält die normalen `ioctl(2)` TTY-Anforderungen, wie sie in `termio(7)` beschrieben sind. Die folgende gilt zusätzlich:

<code>TIOCGWINSZ</code>	Diese Angabe erfordert als Argument die Adresse einer Struktur vom Typ <code>winsize</code> . In diese Struktur wird die Größe des Shell-Fensters kopiert, das über den Dateideskriptor angesprochen wird, der ebenfalls Argument des <code>ioctl(2)</code> Aufrufs ist.
-------------------------	--

Die zweite Gruppe betrifft `ioctl(2)` Anforderungen zur Steuerung des fensterfähigen Terminals. In `layers(5)` sind Anforderungen, die auf Kommunikationsaustausch mit dem Terminal hinauslaufen, genauer beschrieben. Die Anforderungskonstanten sind in der Include-Datei `<sys/jioctl.h>` wie folgt definiert:

<code>JTYPE, JMPX</code>	In beiden Fällen wird <code>JMPX</code> zurückgegeben. Sie werden dazu gebraucht, ein Terminalgerät als xt Kanal zu identifizieren.
<code>JBOOT, JTERM</code>	Beide erzeugen ein geeignetes Kommandopakete, welches an das fensterfähige Terminal geschickt wird. Es beeinflusst das dortige Shell-Fenster, das über den Dateideskriptor angesprochen wird, der ebenfalls Argument des <code>ioctl(2)</code> Aufrufs ist. In beiden Fällen gilt, daß bei einem Fehler bei der Zuweisung von STREAMS-Puffern der Fehlercode <code>EAGAIN</code> zurückgegeben wird.
<code>JTIMOM</code>	Legt die Zeitschranke in Millisekunden fest. Dies gilt nur für Kanal 0. Bei einem Fehler bei der Zuweisung von STREAMS-Puffern wird der Fehlercode <code>EAGAIN</code> zurückgegeben.

JWINSIZE	Diese Angabe erfordert als Argument die Adresse einer Struktur vom Typ <code>jwinsize</code> . In diese Struktur wird die Größe des Shell-Fensters kopiert, das über den Dateideskriptor angesprochen wird, der ebenfalls Argument des <code>ioctl(2)</code> Aufrufs ist.
JTRUN	Diese Angabe erfordert als Argument die Adresse einer Zeichenkette der Form "Kanal, UNIX Systemkommando", welches in dem angegebenen Shell-Fenster ablaufen soll. Bei einem Fehler bei der Zuweisung von STREAMS-Puffern wird der Fehlercode <code>EAGAIN</code> zurückgegeben.
JZOMBBOOT	Erzeugt ein Kommandopakete, welches das fensterfähige Terminal in den Zustand versetzt, in dem das Hinunterladen möglich ist. Dabei wird der Dateideskriptor verwendet, der auch Argument des <code>ioctl(2)</code> Aufrufs ist. Dies ähnelt dem Aufruf mit <code>JBOOT</code> , allerdings wird hier nach dem Hinunterladen das Shell-Fenster in den Zombie-Zustand versetzt, um die Fehlersuche vorzubereiten. Bei einem Fehler bei der Zuweisung von STREAMS-Puffern wird der Fehlercode <code>EAGAIN</code> zurückgegeben.
JAGENT	Die zur Verfügung gestellten Daten werden als Kommandopakete an das fensterfähige Terminal gesandt, um dort eine Behandlungsroutine aufzurufen. Die Antwort des Terminals wird dem aufrufenden Prozeß zugestellt. Dies ist nur auf dem Dateideskriptor für Kanal 0 erlaubt (vgl. <code>jagent(5)</code>). Bei einem Fehler bei der Zuweisung von STREAMS-Puffern wird der Fehlercode <code>EAGAIN</code> zurückgegeben.
JXTPROTO	Legt das <code>xt</code> Protokoll fest (siehe <code>xtproto(5)</code>). Bei einem Fehler bei der Zuweisung von STREAMS-Puffern wird der Fehlercode <code>EAGAIN</code> zurückgegeben.

Die dritte Gruppe betrifft `ioctl(2)` Anforderungen zur Konfiguration von `xt`. Die Konfigurationskonstanten sind in der Include-Datei `<sys/nxt.h>` wie folgt definiert:

XTIOCTYPE	Gibt den Wert <code>XTIOCTYPE</code> zurück und erfüllt den gleichen Zweck wie <code>JMPX</code> .
XTIOCHEX	Legt fest, daß der ENCODING MODE eingeschaltet wird.
XTIOCTRACE	Diese Angabe erfordert als Argument die Adresse einer Struktur vom Typ <code>Tbuf</code> . Diese Struktur wird mit dem Inhalt des Überwachungspuffers versorgt. Die Überwachung wird eingeschaltet (siehe auch <code>xet(1)</code>).
XTIOCIOTRACE	Abschalten der Überwachung.
XTIOGSTATS	Diese Angabe erfordert als Argument die Adresse eines Feldes (array) der Größe <code>S_NSTATS</code> , dessen Elemente vom Typ <code>Stats_t</code> sind. Das Feld wird mit der statistischen Information des Treibers versorgt (siehe auch <code>xls(1)</code>).

Die vierte Gruppe betrifft `ioctl(2)` Anforderungen zur Konfiguration des `streamio(7)` Multiplexers mit den folgenden Konstantendefinitionen:

`I_LINK` Bindet den Hardware-Treiber unter `xt` ein. Die Argumente für den diesbezüglichen `ioctl` Aufruf sind in `streamio(7)` beschrieben.

`I_UNLINK` Hebt die Einbindung des Hardware-Treibers unter `xt` auf. Die Argumente für den diesbezüglichen `ioctl` Aufruf sind in `streamio(7)` beschrieben.

DATEIEN

`/dev/xt/??[0-7]` Gerätedateien für Multiplex-Betrieb

`/usr/include/sys/jioctl.h` Typ der Kommandopakete

`/usr/include/sys/nxtproto.h` Definitionen der Protokolle für Multiplexkanäle

`/usr/include/sys/nxt.h` Definitionen für STREAM-orientierte Treiber

SIEHE AUCH

`layers(1)`, `xts(1M)`, `xtt(1M)`
`ioctl(2)`, `open(2)`
`jagent(5)`, `layers(5)`, `xtproto(5)`
`streamio(7)`, `termio(7)`, `tty(7)`.
CES, Leitfaden und Werkzeuge für die Programmierung mit C.

BEZEICHNUNG

`zero` – Quelldatei für Nullen

ÜBERSICHT

`/dev/zero`

BESCHREIBUNG

Mit der Gerätedatei `zero` bezeichnet man einen imaginären Speicherbereich, der mit Nullen gefüllt ist.

Wird diese Gerätedatei als Quelldatei zum Lesen benutzt, ist der Puffer anschließend stets mit Nullen gefüllt. Die Größe des Puffers spielt hierbei keine Rolle.

Es ist auch jederzeit möglich, in diese Gerätedatei zu schreiben, allerdings werden die Daten verworfen.

Wird `zero` als Abbildungsdatei verwendet, dann wird ein Speicherobjekt ohne Namen erzeugt, das mit Nullen aufgefüllt ist. Das Objekt hat die kleinstmögliche Seitengröße, auf die die Größe der Abbildung aufgerundet werden kann. Diese Seitengröße wird mit `sysconf` ermittelt. Unter der Voraussetzung, daß das Objekt `MAP_SHARED` auf einen gemeinsamen Anker abgebildet wurde, kann eine solche Gerätedatei auch gemeinsam von mehreren Prozessen genutzt werden.

DATEIEN

`/dev/zero`

SIEHE AUCH

`fork(2)`, `sysconf(3C)`, `mmap(2)`



Index

Index

I-1

Index

A

Abbildarchiv eines Dateisystems,
 erstellen (fimage) 1: 186
Abrechnung
 der Verbindungszeit (acctcon) 1: 8
 Kommandos 1: 4
 prozeßspezifische (acct) 4: 2
 Shellprozeduren 1: 13
Abrechnungsdaten, beanspruchter
 Speicherplatz (diskusg) 1: 156
Accounting (acct) 1: 4
Accounting (acctcms) 1: 6
Accounting (acctcon) 1: 8
Accounting (acctmerg) 1: 10
Accounting (acctprc) 1: 11
Accounting (acctsh) 1: 13
ADP32 Controller (adp32) 7: 3
Adreßauflösung
 anzeigen (arp) 1: 18
 Protokoll (ARP) 7: 5
 steuern (arp) 1: 18
Adreßdatenbank für Hostnamen (eth-
 ers) 4: 21
AHA1542B (aha) 7: 4
aktive Prozesse, abbrechen (kill)
 1: 274
aktive UNIX Partition 7: 15
 aufteilen (disksetup) 1: 153
 auswählen (fdisk) 1: 171-173
Aktivität der Prozessoren, darstellen
 (monitor) 1: 339
Akzeptanz binärer Meldungen, prüfen
 (ckbinarsys) 1: 81
Aliasdatei für FACE (pathalias) 4: 89
Archiv
 Dateiformate (ar) 4: 7
 verschieben (migration) 1: 317

ASCII-Zeichensatz (ascii) 5: 2
Aufteilung der aktiven UNIX Partition
 (disksetup) 1: 153
Aufträge eines fernen Systems,
 ausführen (uuxqt) 1: 588
Auftragsabrechnung 1: 216
Auslastung des Betriebssystems (prf)
 7: 84
automatisches Einhängen von
 Dateisystemen (automount) 1: 19

B

Bandlaufwerke, Schnittstelle für
 (xmt) 7: 138
Benutzer
 einrichten (useradd) 1: 574
 wechseln (su) 1: 503
Benutzeraktivitäten, aufzeigen
 (whodo) 1: 593
Benutzeranmeldungen, auflisten
 (logins) 1: 283
Benutzerkennung
 ändern (usermod) 1: 578
 einrichten (useradd) 1: 574
 löschen (userdel) 1: 577
Benutzernamen
 auflisten (dispuid) 1: 161
 ausgeben (id) 1: 231
Benutzernummer, überprüfen (ckuid)
 1: 105
Benutzerquoten, editieren (edquota)
 1: 164
Benutzerumgebung (environ) 5: 4
Betriebsbedingungen
 überprüfen (bp) 1: 69
 überprüfen (bpchk) 1: 70

Betriebsstatus des Systems 1: 268
 Betriebsstufe des Systems 1: 268
 Betriebssystem
 Auslastung des (prf) 7: 84
 beenden (rc6) 1: 427
 Mehrbenutzerumgebung aktivieren
 (rc2) 1: 425
 neuladen (rc6) 1: 427
 Systemdienste beenden (rc0) 1: 423
 Bibliothek, Dateiformate (ar) 4: 7
 biff-Server (comsat) 1: 113
 binäre Meldungen
 Akzeptanz prüfen (ckbinarsys) 1: 81
 Infodatei (binarsys) 4: 13
 Blöcke
 fehlerhafte 1: 16
 freie (df) 1: 139
 Größe analysieren (s5) (fsba) 1: 195
 Boot-Code, auf Platte schreiben 1: 153
 Bootflags
 ändern (bootflags) 1: 67
 anzeigen (bootflags) 1: 67
 Boot-Optionen 1: 51
 Bootparameter 1: 52
 Bootstring 1: 52
 ändern (bootflags) 1: 67
 anzeigen (bootflags) 1: 67

C

CD-ROM 7: 8
 CD-ROM-Laufwerk 7: 8
 C-Programme, übersetzen 1: 217
 cron-Optionen setzen (cron) 4: 14

D

DARPA INTERNET, Netzwerknamen
 (networks) 4: 86
 DARPA Namens-Server 1: 552
 DARPA Reverse Address Resolution
 Protocol 1: 422
 DARPA TELNET Protokoll Server
 1: 547
 DARPA TFTP Protokoll Server 1: 549
 Datei
 aus Abbildarchiv wiederherstellen
 (fimage) 1: 186
 konvertieren (dd) 1: 130
 kopieren (dd) 1: 130
 Dateibehandlung, Optionen (fcntl)
 5: 10
 Dateideskriptoren (fd) 4: 22
 Dateiformat, Systemparameter
 (stune) 4: 138
 Dateiformate 4: 1
 Dateikopf, COFF Dateien (filehdr)
 4: 23
 Dateinamen eines Dateisystems,
 auflisten (ff) 1: 178
 Dateisystem
 aktive Prozesse (proc) 4: 91
 anlegen (makefsys) 1: 316
 Archiv anlegen (fdp) 1: 175
 Archiv erstellen (ffile) 1: 182
 Archiv wiederherstellen (fdp) 1: 175
 Archivdateien wiederherstellen
 (ffile) 1: 182
 Aufbau (filesystem) 7: 41
 aushängen (mount) 1: 346
 aushängen (umountfsys) 1: 358
 automatisches Einhängen von (auto-
 mount) 1: 19

- Blockanzahl ermitteln (quot) 1: 418
- Blockgröße analysieren (s5) (fsba) 1: 195
- Dateinamen auflisten (ff) 1: 178
- Eigentümerstatus auflisten (quot) 1: 418
- einhängen (mount) 1: 346
- einhängen (mountfsys) 1: 358
- einhängen (NFS) (mountd) 1: 357
- erstellen (makefsys) 1: 316
- erstellen (mkfs) 1: 321
- freier Plattenplatz (df) 1: 139
- gemeinsam benutztes (sharetab) 4: 130
- I-Node-Format (bfs) (inode) 4: 46
- Kennsätze definieren (labelit) 1: 275
- kopieren (dcopy) 1: 128
- korrigieren (fsck) 1: 196
- optimieren (tunefs) 1: 561
- Organisation (filesystem) 7: 41
- Pfadnamen und I-Nodes auflisten (ncheck) 1: 363
- physische Kopie erzeugen (volcopy) 1: 589
- prüfen (fsck) 1: 196
- Quoten anzeigen (repquota) 1: 436
- sichern (ufsdump) 1: 564
- Sicherungsplan prüfen (ckbupscd) 1: 82
- Standardwertetabelle (vfstab) 4: 208
- Superblock-Format (bfs) (fs) 4: 25
- Teilarchiv anlegen (incfile) 1: 259
- Teilarchiv wiederherstellen (incfile) 1: 259
- teilweise sichern (ufsdump) 1: 564
- teilweise wiederherstellen (ufsrestore) 1: 567
- Typ bestimmen (fstyp) 1: 209
- Typ konfigurieren (mfsys) 4: 76
- Typinformation (sfsys) 4: 128
- überprüfen (bcheckrc) 1: 71
- überprüfen (checkfsys) 1: 73
- verteiltes (fstypes) 4: 34
- wiederherstellen (bcheckrc) 1: 71
- wiederherstellen (ufsrestore) 1: 567
- zurückspeichern (restore) 1: 437
- Dateisysteme, eingehängte (mnttab) 4: 77
- Dateisystemquoten
 - aktivieren (quotaon) 1: 421
 - deaktivieren (quotaoff) 1: 421
 - Konsistenz prüfen (quotacheck) 1: 420
- Dateiübertragungsprotokolle, Server (ftpd) 1: 210
- Dateiverzeichnis
 - Blockanzahl ermitteln (du) 1: 162
 - s5-Format (dir) 4: 16
 - UFS-Format (dir) 4: 17
 - verschieben (mvdir) 1: 359
- Datenbank
 - der Terminal-Eigenschaften (terminfo) 4: 145
 - für Host-Namen (hosts) 4: 38
 - für Internet-Server (inetd.conf) 4: 40
 - für Netzmasken (netmasks) 4: 83
 - für Netzwerk-Namen (networks) 4: 86
 - für Protokollnamen (protocols) 4: 106
 - Netzwerk-Konfiguration (netconfig) 4: 79
- Datenpartitionen, zurückspeichern (restore) 1: 437
- Datensicherung, Dialog (bkoper) 1: 36
- Datentypen, Sprachen (nl_types) 5: 30

Datenübertragung, prüfen (ckbinarsys) 1: 81

Datum

Eingabe überprüfen (ckdate) 1: 84

einstellen (rdate) 1: 428

regional (strftime) 4: 136

DFS-Protokolldatei (fstypes) 4: 34

Diskette, Treiber (fd) 7: 38

Driver Software Package, Daten ausgeben (idcheck) 1: 234

Drucken

Aufträge abweisen (reject) 1: 3

Aufträge annehmen (accept) 1: 3

Druckerwarteschlange, Prioritäten

festlegen (lpusers) 1: 313

Durchsatzmessung (prf) 7: 84

E

Echtzeit

Dispatcher (rt_dptbl) 4: 110

Zuteilungsroutine (rt_dptbl) 4: 110

Ein/Ausgabeboard, seriell (sr) 7: 95

Einbenutzer-Betrieb 1: 505

Eingabe, überprüfen (ckint) 1: 89

Eingabeformate (utmp) 4: 204

Eingabeformate (utmpx) 4: 206

Eingabeformate (wtmp) 4: 204

Eingabeformate (wtmpx) 4: 206

eingehängte Dateisysteme (mnttab) 4: 77

Einhängetabelle, anlegen (setmnt) 1: 477

einstellbare Parameter

Standardwerte (iddefaults) 1: 238-242

umdefinieren (idtune) 1: 255

einstellbare Systemparameter

(mtune) 4: 78

einstellbare Systemparameter (stune)

4: 138

Ersatzblock 1: 16

Erweiterungs-Befehle, mail (mailsurr)

4: 59

Ethernet

Adreßdatenbank (ethers) 4: 21

Adreßumsetzung 7: 5

exklusiv benutzte Geräte, freigeben

(devfree) 1: 135

F

FACE

Aliase 4: 89

Benutzerpräferenzen (environ) 4: 19

Objektarchitektur-Information (ott) 4: 87

fehlerhafte Blöcke

auflisten (addbad) 1: 16

behandeln (addbad) 1: 16

ersetzen (addbad) 1: 16

Fenster, ändern (relogin) 1: 429

fensterfähige Terminals, steuern (jagent) 5: 24

Fensterprotokoll (layers) 5: 27

Fenstertechnik

Terminalsteuerung (jagent) 5: 24

xt-Protokoll (xtproto) 5: 48

ferne Systeme, Information (binarsys) 4: 13

fernen Rechner, ansprechen (uutry) 1: 587

Festplatte

in Partitionen aufteilen (fdisk)

1: 171-173

konfigurieren (diskadd) 1: 147-151
 über Aufteilung informieren (fdisk)
 1: 171-173
 verwalten (mkpart) 1: 330,334
 Festplatte (disk) 7: 15
 Festplatte (hd) 7: 45-47
 FIFO-Geräte-datei, erstellen (mkfifo)
 1: 320
 Format
 s5-Verzeichnis (dir) 4: 16
 UFS-Verzeichnis (dir) 4: 17
 Formate, regional 4: 136
 freien Speicherplatz, überprüfen
 (idspace) 1: 253
 Front-End, zu cc legen (genc) 1: 217
 ftp Fern-Login Daten (netrc) 4: 84

G

ganze Zahlen, überprüfen (ckrange)
 1: 99
 Gerät
 Attribute auflisten (devattr) 1: 134
 auflisten (getdev) 1: 218
 exklusiv benutztes freigeben (dev-
 free) 1: 135
 Include-Datei (archives) 4: 10
 Kennsatz überprüfen (getvol) 1: 225
 Zugriffsmöglichkeit überprüfen (get-
 vol) 1: 225
 zur exklusiven Nutzung reservieren
 (devreserv) 1: 137
 Geräte-datei 7: 1
 anlegen (mknod) 1: 329
 konvertieren (UFS) (fconv) 1: 170
 suchen (devnm) 1: 136
 Gerätegruppen

auflisten (getgrp) 1: 220
 Mitglieder auflisten (listdgrp) 1: 280
 Tabelle editieren (putdgrp) 1: 414
 Geräteknotten, entfernen (idmknod)
 1: 249
 Gerätetabelle, editieren (putdev)
 1: 410
 Gerätetreiber
 Konfigurationsdaten abrufen (idin-
 stall) 1: 244
 Konfigurationsdaten modifizieren
 (idinstall) 1: 244
 Gesamtabrechnung, addieren
 (acctmerg) 1: 10
 Gesamtabrechnung 4: 2
 getty-Steuerdatei (gettydefs) 4: 35
 Gruppe
 ändern (newgrp) 1: 369
 anlegen (groupadd) 1: 227
 definieren (groupadd) 1: 227
 Definition ändern (groupmod)
 1: 229
 löschen (groupdel) 1: 228
 Gruppendatei
 /etc/group 4: 37
 prüfen (grpck) 1: 416
 Gruppennamen
 auflisten (dispgid) 1: 160
 ausgeben (id) 1: 231
 Gruppennummer, überprüfen (ckgid)
 1: 87

H

Hauptspeicher (kmem) 7: 82
 Hauptspeicher (mem) 7: 82
 Host-Datenbank, /etc/hosts 4: 38

Hostnamen, Adreßdatenbank 4: 21
 Hosts
 vertrauenswürdige
 (/etc/hosts.equiv) 4: 39
 vertrauenswürdige (.rhosts) 4: 39
 Host-Tabelle
 im DoD-Internet-Format abrufen
 (gettable) 1: 222
 umwandeln (htable) 1: 230

I

ICMP ECHO_REQUEST1-Pakete, an
 Netzwerk-Hosts senden 1: 389
 Implementierung, Betriebssystem
 4: 51
 Include-Datei, für Geräte (archives)
 4: 10
 Inhaltsverzeichnis eines Datenträgers
 ändern (edvtoc) 1: 168-169
 init, Steuerdatei 4: 42
 Initialisierung
 mail (mailcnfg) 4: 56
 rmail (mailcnfg) 4: 56
 inittab, erstellen (idmkinit) 1: 247
 I-Node, bfs-Dateisystem (inode) 4: 46
 I-Nodes, freie (df) 1: 139
 Installation
 Antworten speichern (pkgask)
 1: 392
 erstellen (custom) 1: 125
 prüfen (pkgchk) 1: 393
 Software-Paket (pkgadd) 1: 390
 von Kommandos (install) 1: 272
 Voreinstellung (admin) 4: 4
 Internet-Aliase (services) 4: 127
 Internet-Datagramme, Protokoll
 (UDP) 7: 135

Internet-Dienste (inetd) 1: 263
 Internet-Dienste (services) 4: 127
 Internet-Name Server 1: 360
 Internet-Protokoll, Über-
 tragungssteuerung (TCP) 7: 101
 Internet-Protokoll (IP) 7: 56
 Internet-Protokollfamilie (inet) 7: 53
 Internet-Server, Datenbank 4: 40
 IP-Adresse ermitteln (rarpd) 1: 422

J

Jukebox, Treiber (acc) 7: 2

K

Kennsätze für Dateisysteme,
 definieren (labelit) 1: 275
 Kennwortdatei (passwd) 4: 88
 Kernel
 Konfiguration erstellen (idconfig)
 1: 236
 neu aufbauen (idmkunix) 1: 252
 Knotennamen, ändern (setuname)
 1: 478
 Kommando
 Ausnutzung ermitteln (acctcms) 1: 6
 installieren (install) 1: 272
 zu bestimmten Zeiten ausführen
 (cron) 1: 123
 Konfigurations-Report, ausgeben
 (rep) 1: 430
 Konfigurations-Skript
 ersetzen pro Port Monitor (sacad-
 min) 1: 461
 erstellen pro Port Monitor (sacad-
 min) 1: 461

installieren pro Port Monitor (sacad-
min) 1: 461
Konfigurieren
 von Dateisystemtypen (mfsys) 4: 76
 von Festplatten (diskadd) 1: 147-151
Konsole, Typ feststellen (consoletype)
 1: 114
Konstanten
 implementierungsspezifische (lim-
 its) 4: 51
 symbolische (unistd) 4: 201
Konvertierung, UFS-Gerätedateien
 (fconv) 1: 170

L

Lademarken 1: 67
Ladeoptionen
 ändern (bootflags) 1: 67
 anzeigen (bootflags) 1: 67
Ladeprogramm des Systems (boot)
 1: 50,55
Lochkarte, Dateiformat für (pnch)
 4: 90
Login-Bildschirm (issue) 4: 50
Login-Protokolldatei (loginlog) 4: 55
Login-Shellvariablen (login) 4: 53
Login-Sitzung, vorbereiten (getty)
 1: 223
Login-Standardwerte (login) 4: 53
Login-Versuche, fehlgeschlagene
 (loginlog) 4: 55
lokale Ressourcen
 auflisten (DFS) (dfmounts) 1: 143
 auflisten (NFS) (dfmounts) 1: 144
 sperren (DFS) (unshare) 1: 527
 zur Verfügung stellen (DFS) (share)
 1: 480

Loopback-Netzwerk-Schnittstelle (lo)
 7: 77
LP-Anforderungen, umleiten (lpfilter)
 1: 309
LP-Druckdienst
 beenden (lpsched) 1: 309
 ferne Systeme eintragen (lpssystem)
 1: 310
 konfigurieren (lpadmin) 1: 285
 starten (lpsched) 1: 309
LP-Filter, verwalten (lpfilter) 1: 297
LP-Formulare, verwalten (lpforms)
 1: 302

M

Magnetband, Schnittstelle für (xmt)
 7: 138
Master-Datei für Geräte-
 Spezifikationen (mdevice)
 4: 67,72
MB-I Adapter (mbad) 7: 79
MB-I Ein/ Ausgabeboard (sr) 7: 95
Mehrbenutzer-Betrieb 1: 505
Mehrbenutzer-Umgebung, aktivieren
 (rc2) 1: 425
Meldungen
 an alle Benutzer schicken (wall)
 1: 592
 binäre (binarsys) 4: 13
 binäre (ckbinarsys) 1: 81
Menü
 generieren (ckitem) 1: 91
 Option auswählen (ckitem) 1: 91
Modem-Ein/ Ausgabeboard, seriell
 (sx) 7: 98
mount-Tabelle, anlegen (setmnt)
 1: 477

Multiplexer, serielle Schnittstelle (si)
 7: 92
 multi-user-mode 1: 505

N

Nachrichten
 an SMTP schicken (tosmtp) 1: 553
 empfangen (smtpd) 1: 491
 in Warteschlangen bearbeiten
 (smtpsched) 1: 495
 in Warteschlangen einreihen
 (smtpquer) 1: 493
 versenden (smtp) 1: 490
 Name/Finger-Protokoll, Server
 (fingerd) 1: 190
 Name-Server, abfragen (nslookup)
 1: 376
 Name-Server-Routinen,
 Konfigurationsdatei (resolv.conf)
 4: 107
 NCR SCSI Host Adapter (adp32) 7: 3
 Netzmasken für Teilnetze (netmasks)
 4: 83
 Netzwerk
 Anforderungen erkennen (listen)
 1: 281
 Datenbank für Namen (networks)
 4: 86
 Datenbank (netconfig) 4: 79
 ftp Fern-Login Daten (netrc) 4: 84
 Konfiguration (netconfig) 4: 79
 Paket-Routing (routing) 4: 108
 Schnittstelle konfigurieren (ifconfig)
 1: 256
 Status anzeigen (netstat) 1: 366
 Netzwerk-Routen-Dämon (routed)
 1: 448

Netzwerk-Routen-Tabellen, manuell
 bearbeiten (route) 1: 446
 Netzwerk-Service-Steuerung 1: 458
 Netzwerk-Service-Verwaltung 1: 371
 neuen Benutzer, einrichten (useradd)
 1: 574
 NFS-Dämon, starten (biode) 1: 29
 NFS-Dämon (nfsd) 1: 370
 NFS-Dateisystem
 automatisches Einhängen von (auto-
 mount) 1: 19
 einhängen (mountd) 1: 357
 NFS-Ressourcen, auflisten (dfmounts)
 1: 1
 Null-Datei (null) 7: 83
 Nullen, Quelldatei (zero) 7: 144
 NVRAM
 Daten ändern (nvram) 1: 381
 Daten anzeigen (nvram) 1: 381

O

Objektarchitektur-Information, FACE
 (ott) 4: 87

P

Paket-Routing, Verwaltung (routing)
 4: 108
 Parameter, umdefinieren (id tune)
 1: 255
 Paßwort, verschlüsseltes (shadow)
 4: 129
 Paßwortdatei
 prüfen (pwck) 1: 416
 verwalten (passmgmt) 1: 387
 Paßwortdatei (passwd) 4: 88

- periodisches Überprüfen der
 Betriebsbedingungen (bp) 1: 69
- periodisches Überprüfen der
 Betriebsbedingungen (bpchk)
 1: 70
- Pfadnamen, überprüfen (ckpath) 1: 96
- physische Kopie des Dateisystems,
 erzeugen (volcopy) 1: 589
- Platte
 installieren 1: 153
 optische 7: 2,75
 zurückspeichern (restore) 1: 437
- Plattenabrechnungsdaten, erstellen
 (diskusg) 1: 156
- Plattenauslastung, zeigen (du) 1: 162
- Plattenbelegung, anzeigen (quota)
 1: 419
- Plattenblöcke, freie (df) 1: 139
- Plattengeometrie, ausgeben (prtvtoc)
 1: 407
- Plattenkonfiguration, Report ausgeben
 (rep) 1: 430
- Plattenquote, anzeigen (quota) 1: 419
- Plattenwechsler, SCSI-Treiber (acc)
 7: 2
- Port Monitor
 aktivieren (sacadmin) 1: 461
 beenden (sacadmin) 1: 461
 deaktivieren (sacadmin) 1: 461
 entfernen (sacadmin) 1: 461
 hinzufügen (sacadmin) 1: 461
 Informationen ausgeben (sacadmin)
 1: 461
 starten (sacadmin) 1: 461
 verwalten (pmdadmin) 1: 401
 verwalten (sacadmin) 1: 461
- Post
 abhören (comsat) 1: 113
- empfangen (mail_pipe) 1: 314
- Post-Transport, Erweiterungsbefehle
 4: 59
- privilegierter Benutzer 1: 503
- Protokoll zwischen Host und
 fensterfähigem Terminal (layers)
 5: 27
- Protokoll-Familie 4: 79
- Protokollnamen, Datenbank für 4: 106
- Prozeß-Abrechnung (acctprc) 1: 11
- Prozeß-Dateisystem (proc) 4: 91
- Prozesse, identifizieren (fuser) 1: 214
- Prozessoren
 Aktivität darstellen (monitor) 1: 339
 Zustand ändern (offline) 1: 385
 Zustand ändern (online) 1: 386
 Zustand anzeigen (offline) 1: 385
 Zustand anzeigen (online) 1: 386
- Prozeß-Scheduler
 Parameter 4: 110
 verwalten (dispadmin) 1: 158
- Prozeß-Scheduler (ts_dptbl) 4: 195
- Prozeß-Steuerung
 initialisieren (init) 1: 268
 initialisieren (telinit) 1: 268
- ## Q
- Quelldatei, für Nullen (zero) 7: 144
- ## R
- reguläre Ausdrücke (regexp) 5: 31
- Resolver-Routinen (resolv.conf) 4: 107
- Ressourcen
 auflisten (DFS) (dfmounts) 1: 143
 auflisten (DFS) (dfshares) 1: 145

- gemeinsame Nutzung (dfstab) 4: 15
- nicht mehr zur Verfügung stellen
 - (lokal) (unshareall) 1: 483
- sperrern (DFS) (unshare) 1: 572
- zur Verfügung stellen (DFS) (share)
 - 1: 480
- zur Verfügung stellen (lokal)
 - (shareall) 1: 483
- rexec-Server (rexecd) 1: 440
- RFC822-Post, empfangen (fromsmtp)
 - 1: 194
- rhost-Datei (/etc/hosts.equiv) 4: 39
- rlogin-Server (rlogind) 1: 442
- Root-Verzeichnis, ändern (chroot)
 - 1: 75
- Routing 4: 59
- rsh-Server (rshd) 1: 451
- ruptimuptime-Server 1: 456
- rwho-Server 1: 456

S

- SAC 1: 458
- SCCS
 - Delta 4: 115
 - Flags 4: 115
 - Prüfsumme 4: 115
- SCCS-Datei (scsfile) 4: 115
- Schlüsselwort, überprüfen (ckkeywd)
 - 1: 94
- Schnittstelle
 - für Bandlaufwerke (xmt) 7: 138
 - für Magnetband (xmt) 7: 138
 - Parameter konfigurieren (ifconfig)
 - 1: 256
- SCSI-Adapter, Treiber für Festplatten
 - 7: 96

- SCSI-Band
 - am ADP32 Controller 7: 33
 - am Host Adapter AHA1542B 7: 35
 - am NCR Host Adapter 7: 33
- SCSI-Controller für Bandlaufwerke
 - (ct) 7: 10
- SCSI-Festplatte
 - Treiber (shd) 7: 86,89
- SCSI-Geräte
 - am Host Adapter 1: 474
 - am NCR Host Adapter 1: 472
 - anschließbare 7: 3-4
- SCSI-Geräte-dateien
 - erzeugen (scsigen) 1: 472-474
- SCSI-Host-Adapter (aha) 7: 4
- SCSI-Node-Files
 - erzeugen (scsigen) 1: 472-474
- SCSI-Treiber
 - für CD-ROM (cdr) 7: 8
 - für Worm-Geräte (lad) 7: 75
 - für Worm-Plattenwechsler (acc) 7: 2
- Server
 - Name/Finger-Protokoll (fingerd)
 - 1: 190
- Service
 - aktivieren (pmadmin) 1: 401
 - Anforderungen erkennen (listen)
 - 1: 281
 - deaktivieren (pmadmin) 1: 401
 - entfernen (pmadmin) 1: 401
 - hinzufügen (pmadmin) 1: 401
 - Informationen ausgeben (pmadmin)
 - 1: 401
 - Konfigurations-Skript ersetzen
 - (pmadmin) 1: 401
 - Konfigurations-Skript erstellen
 - (pmadmin) 1: 401
- Service Access Controller (SAC)
 - 1: 458,461

-
- Service Access Controller (SAC)
 - 1: 458,461
 - shadow-Datei
 - aktualisieren (pwconv) 1: 417
 - installieren (pwconv) 1: 417
 - Shell-Umgebung, einstellen (profile)
 - 4: 104
 - Shell-Variablen 5: 4
 - Sicherung, Ausnahmeliste (bkexcept)
 - 1: 30
 - Sicherungsoperationen, Status
 - anzeigen (bkstatus) 1: 47
 - Sicherungsplan, für Dateisystem
 - prüfen (ckbupscd) 1: 82
 - Sicherungsregister
 - Inhalt ändern (bkreg) 1: 38
 - Inhalt anzeigen (bkreg) 1: 38
 - Signale 5: 36
 - Signale (signal) 5: 38
 - Signalerzeugung, Information zur (siginfo) 5: 36
 - SIM Controller 7: 92
 - Simple Mail Transfer Protocol
 - Nachrichten bearbeiten (smtpsched)
 - 1: 495
 - Nachrichten einreihen (smtpquer)
 - 1: 493
 - Nachrichten empfangen (smtpd)
 - 1: 491
 - Nachrichten versenden (smtp)
 - 1: 490
 - Single-user-mode 1: 505
 - SINIX-Systemkern, neu aufbauen
 - (idmkunix) 1: 252
 - Slices
 - anlegen (disksetup) 1: 153
 - sichern (sysssave) 1: 541
 - Slices 7: 15
 - SMTP
 - Nachrichten bearbeiten (smtpsched)
 - 1: 495
 - Nachrichten empfangen (smtpd)
 - 1: 491
 - Nachrichten in Warteschlange einreihen (smtpquer) 1: 493
 - Nachrichten versenden (smtp)
 - 1: 490
 - Socket-Schnittstelle 4: 40
 - Software-Pakete
 - Antworten speichern (pkgask)
 - 1: 392
 - Eigenschaften auflisten (pkginfo)
 - 1: 395
 - entfernen (pkgrm) 1: 398
 - Formate umwandeln (pkgtrans)
 - 1: 399
 - installieren (pkgadd) 1: 390
 - Parameter auflisten (pkgparam)
 - 1: 397
 - vom System entfernen (pkgrm)
 - 1: 398
 - Software-Pakete für Gerätetreiber, Daten ausgeben (idcheck) 1: 234
 - Sortierdatenbank, anlegen (colltbl)
 - 1: 109
 - Speicherabzug
 - erlauben (dumping) 1: 163
 - organisieren (savecore) 1: 468,471
 - Speichermedium für Direktzugriff
 - (disk) 7: 15
 - Sperren, lokale Ressourcen (DFS)
 - (unshare) 1: 572
 - Sprachen, Datentypen (nl_types) 5: 30
 - Sprachkonstanten (langinfo) 5: 25
 - Standardwerte für einstellbare Parameter

- vorgeben (iddefaults) 1: 238-242
- stat-Dateien (stat) 5: 42
- stat-Systemaufruf (stat) 5: 42
- Steuerdatei für init (inittab) 4: 42
- Steuerungsmeldungen, Internet Protokoll (ICMP) 7: 51
- STREAMS
 - Binder (slink) 1: 486
 - Fehlerprotokoll-Dämon (strerr) 1: 499
 - Fehlerprotokolle löschen (strclean) 1: 498
 - kompatible Module (ttcompat) 7: 127
 - Konfigurationsdatei für STREAMS TCP/IP (strcf) 4: 131
 - Konfigurationsoperationen (strcf) 4: 131
 - Module konfigurieren (autopush) 1: 24
 - Überwachungsmeldungen ausgeben (strace) 1: 496
- su, Optionen (su) 4: 139
- Superblock, aktualisieren (sync) 1: 508
- Superblock-Format, bfs-Dateisystem (fs) 4: 25
- swap-Bereiche
 - löschen (swap) 1: 506
 - vergrößern (swap) 1: 506
 - verkleinern (swap) 1: 506
 - verwalten (swap) 1: 506
- symbolische Konstanten (unistd) 4: 201
- sysadm-Menü, löschen (delsysadm) 1: 132
- sysadm-Schnittstelle, editieren (edsysadm) 1: 165
- System
 - Aktivität darstellen (monitor) 1: 339
 - beenden (rc6) 1: 427
 - booten (boot) 1: 50,55
 - deaktivieren (rc0) 1: 423
 - herunterfahren (shutdown) 1: 484
 - initialisieren (bcheckrc) 1: 71
 - initialisieren (brc) 1: 71
 - laden (boot) 1: 50,55
 - Mehrbenutzerumgebung aktivieren (rc2) 1: 425
 - neuladen (rc6) 1: 427
 - sichern (backup) 1: 25
 - sichern (syssave) 1: 541
 - wiederherstellen 1: 538
- Systemabbilder, prüfen (crash) 1: 115
- Systemaktivität
 - abfragen (sar) 1: 466
 - analysieren (profile) 1: 406
 - erfassen (sar) 1: 466
- Systemanmeldungen, auflisten (logins) 1: 283
- Systemauslastung (prf) 7: 84
- Systemdatei
 - für Geräte-Spezifikationen (sdevice) 4: 118,125
- Systemdump, erlauben (dumping) 1: 163
- Systemeigentümer, einrichten (setup) 1: 479
- Systemkern, neu aufbauen (idbuild) 1: 232
- Systemkonfiguration
 - Daten ausgeben (idcheck) 1: 234
 - Report ausgeben (rep) 1: 430
- Systemnamen, ändern (setuname) 1: 478
- Systemparameter
 - einstellbare (mtune) 4: 78

einstellbare (stune) 4: 138
 Systemsicherung
 protokollieren (bkhistory) 1: 34
 starten 1: 25
 steuern 1: 25
 Systemstatus, ändern (shutdown)
 1: 484
 Systemstatus-Server (rwhod) 1: 456
 Systemverwaltung
 menügeführt (sysadm) 1: 509,519,528
 Systemzeit, setzen (setclk) 1: 476
 Systemzeit (timezone) 4: 194

T

Tabulatorpositionen 4: 32
 Tagesabrechnung, durchführen
 (runacct) 1: 454
 Talk Server 1: 546
 Teilsicherungen
 ändern (bkexcept) 1: 30
 anzeigen (bkexcept) 1: 30
 termcap-Beschreibung, konvertieren in
 terminfo-Beschreibung (cap-
 toinfo) 1: 72
 Terminal
 allgemeines Interface (termio) 7: 104
 Beschreibung ausgeben (infocmp)
 1: 264
 Beschreibung erstellen (terminfo)
 4: 145
 Bezeichnungen (term) 5: 44
 Eigenschaften (terminfo) 4: 145
 fensterfähiges steuern (jagent) 5: 24
 Geschwindigkeit definieren (getty)
 1: 223
 Information (term) 4: 140
 initialisieren (getty) 1: 223
 Modi definieren (getty) 1: 223
 Protokoll zu fensterfähigem (layers)
 5: 27
 Schnittstelle mit Erweiterungen (ter-
 miox) 7: 121
 Schnittstelle (tty) 7: 134
 steuerndes (tty) 7: 134
 Typ definieren (getty) 1: 223
 Übertragungsprozedur definieren
 (getty) 1: 223
 Terminaldateien 4: 199
 terminfo, Beschreibung (term) 4: 140
 terminfo-Beschreibungen
 ausgeben (infocmp) 1: 264
 vergleichen (infocmp) 1: 264
 terminfo-Compiler 1: 550
 terminfo-Übersetzer 1: 550
 Textdateien, Format (fspec) 4: 32
 Time-Sharing CPU, Parameter
 (ts_dptbl) 4: 195
 Transport von Post 4: 59
 Treiber
 für Diskettenlaufwerke (fd) 7: 38
 für Festplatten am SCSI-Adapter
 zB d) 7: 96
 für Festplatten (hd) 7: 45-47
 für SCSI-Festplatten (shd) 7: 86,89
 für Video-Bandlaufwerk (exa) 7: 33-
 35
 für virtuelle Geräte (sxt) 7: 99
 ttyname, Such-Verzeichnisse (ttysrch)
 4: 199
 tty-Strukturen, virtuelle 7: 99

U

überprüfen

- Akzeptanz binärer Meldungen
(ckbinarsys) 1: 81
- Benutzernummer (ckuid) 1: 105
- Datumseingabe (ckdate) 1: 84
- Eingabe (ckint) 1: 89
- ganze Zahlen (ckrange) 1: 99
- Gruppennummer (ckgid) 1: 87
- Pfadnamen (ckpath) 1: 96
- Schlüsselwort (ckkeywd) 1: 94
- Sicherungsplan für Dateisystem
(ckbupscd) 1: 82
- Uhrzeit (cktime) 1: 103
- yes/no (ckyorn) 1: 107
- Zeichenkette (ckstr) 1: 101
- Uhrzeit, Angabe überprüfen (cktime)
1: 103
- umount-Tabelle, anlegen (setmnt)
1: 477
- UUCP-Daten, übertragen (uucico)
1: 582
- UUCP-Datenübertragung, Auf-
tragsverwaltung (uusched) 1: 586
- UUCP-Spool-Verzeichnis, aufräumen
(uucleanup) 1: 584
- UUCP-Verzeichnisse, prüfen
(uucheck) 1: 581
- UUCP-Zugriffsrechte, prüfen
(uucheck) 1: 581

V

Verbindungen

- Abrechnungssätze bearbeiten
(fwtmp) 1: 216
- Abrechnungssätze bearbeiten
(wtmpfix) 1: 216

- Zeit abrechnen (acctcon) 1: 8
- verfügbare Ressourcen, auflisten (DFS)
(dfshares) 1: 145

Verwaltung

- kontrollieren (uadmin) 1: 563
- steuern (uadmin) 1: 563

Verweise

- einrichten (link) 1: 279
- entfernen (link) 1: 279

Verzeichnis

- s5-Format (dir) 4: 16
- UFS-Format (dir) 4: 17

Verzeichniseintrag,

- dateisystemunabhängig (dirent)
4: 18

virtuelle Geräte, Treiber für (sxt) 7: 99

Voreinstellung, Installation (admin) 4: 4

VTOC einer Platte

- ändern (edvtoc) 1: 168-169
- ausgeben (prtvto) 1: 407

W

Währungsdatenbank, erstellen

- (montbl) 1: 343

Worm-Gerät 7: 75

Worm-Platte 7: 75

Worm-Plattenwechsler, SCSI-Treiber

- (acc) 7: 2

X

XENIX, installieren (xinstall) 1: 598

XENIX-Dateiberechtigung, korrigieren (fixperm) 1: 191

XENIX-Dateisystem

prüfen (xfscck) 1: 595
reparieren (xfscck) 1: 595
wiederherstellen (xrestor) 1: 599
wiederherstellen (xrestore) 1: 599
XENIX-Eigentümerrechte, korrigieren
(fixperm) 1: 191
xt-Treiber-Protokoll (xtproto) 5: 48

Y

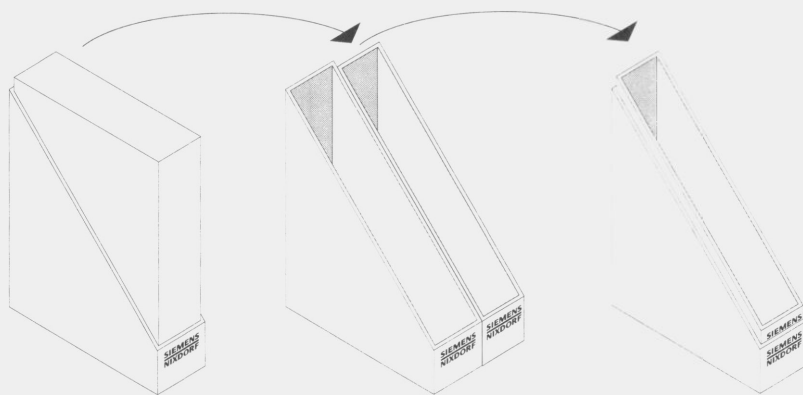
yes/no, überprüfen (ckyorn) 1: 107

Z

Zeichenkette, überprüfen (ckstr)
1: 101
Zeichenklassifikationstabellen, generi-
eren (chrtbl) 1: 76
Zeichensätze, konvertieren (iconv)
5: 12
Zeichenumsetzungstabellen, generi-
eren (chrtbl) 1: 76
Zeit, regional (strftime) 4: 136
Zeitkonvertierung, festlegen (zic)
1: 604
Zeitregel, festlegen (zic) 1: 604
Zeitzone
ausgeben (zdump) 1: 603
Verweis 1: 604
Zeitkonvertierung festlegen (zic)
1: 604
Zeitzone (timezone) 4: 194

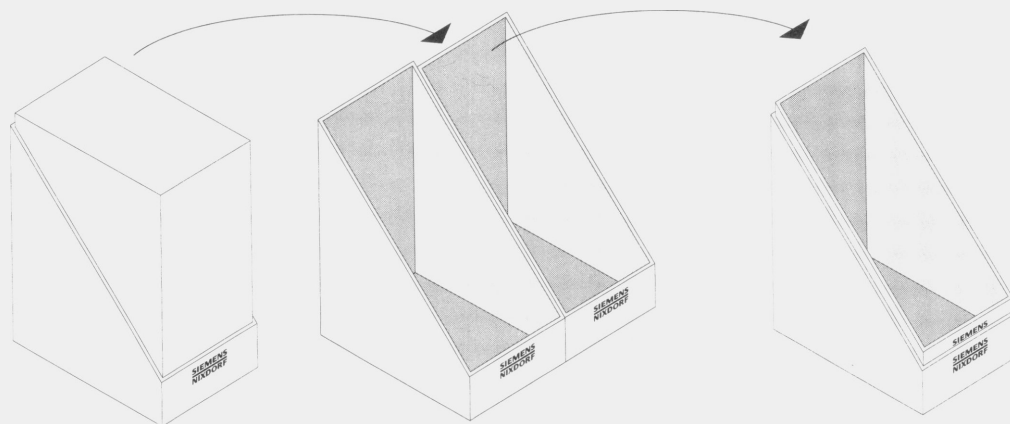
Sammelboxen

Für Handbücher des vorliegenden Formates bieten wir zweiteilige Sammelboxen in zweierlei Größen an. Der Bestellvorgang entspricht dem für Handbücher.



Breite: ca. 5 cm

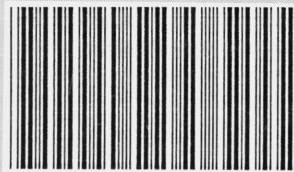
Bestellnummer: U3775-J-Z18-1



Breite: ca. 10 cm

Bestellnummer: U3776-J-Z18-1

970323 P/MCD



9Y503372

Herausgegeben von / Published by
Siemens Nixdorf Informationssysteme AG
Postfach 21 60, W-4790 Paderborn
Postfach 83 09 51, W-8000 München 83

Bestell-Nr. / Order No. **U8888-J-Z145-1**
Printed in the Federal Republic of Germany
17960 AG 2932. (22450)