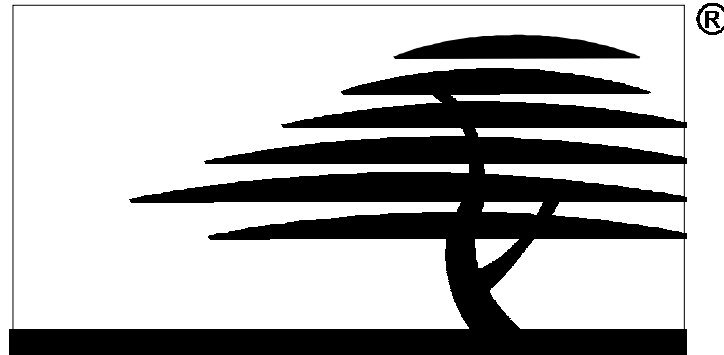




SCO

Introduction to SCO[®] UNIX[®] Systems

Transparency Masters

Course Release: 4.0

For Software: UnixWare[®] 7 and
SCO OpenServer[™] Release 5

© 1987–1998 The Santa Cruz Operation, Inc. All rights reserved.

This publication is protected under copyright laws and international treaties.

Information in this document is subject to change without notice and does not represent a commitment on the part of The Santa Cruz Operation, Inc.

SCO, The Santa Cruz Operation, the SCO logos, VP/ix, SCO OpenServer, UnixWare, dbXtra, Panner, SCO Global Access, SCO MPX, SCO Doctor, SCO Doctor for Networks, SCO Doctor Lite, SCO Premier Motif, SCO Visual Td, SCO VisionFS, SCO ToolWare, SCO TermLite, SCO Vision 97, SCO CIFS Bridge, SCO WebTop, The Internet Way of Computing, and IWoC are trademarks or registered trademarks of The Santa Cruz Operation, Inc. in the USA and other countries. IXI, X.desktop, Deskworks, Wintif, IXI Panorama, Deskterm, and IXI Desktop are trademarks or registered trademarks of IXI Limited, a subsidiary of The Santa Cruz Operation, Inc. Visionware, Devkit.Vision, Esprit, Kodon, Super.Vision, Visionware Super.Vision, Term.Vision, Visionware Direction, the Visionware logo, Visionware SQL-Retriever, and XVision are registered trademarks of Visionware Limited, a subsidiary of The Santa Cruz Operation, Inc. SuperVision, TermVision, X-Visionware, SQL-Retriever, and Vision Builder are trademarks of Visionware Limited. UNIX is a registered trademark of The Open Group in the United States and other countries. Cheyenne and ARCserve are registered trademarks of Cheyenne Software, Inc. Netscape, Netscape Navigator, Netscape Communications Server, Netscape Commerce Server, Netscape Proxy Server, Netscape FastTrack Server, Netscape Enterprise Server, Netscape SuiteSpot, Netscape Catalog Server, Netscape News Server, Netscape Mail Server, and Netscape Navigator Gold are trademarks of Netscape Communications Corporation. NFS was developed by Legent Corporation (formerly Lachman Associates, Inc.) based on LACHMAN SYSTEM V NFS. LACHMAN is a trademark of Legent Corporation. NFS is a trademark of Sun Microsystems, Inc. TCP/IP was developed by Legent Corporation (formerly Lachman Associates, Inc.) based on LACHMAN SYSTEM V STREAMS TCP, a joint development of Lachman Associates and Convergent Technologies. MPX was developed by Corollary, Inc. VP/ix is a product developed and licensed by Phoenix Technologies, Ltd/INTERACTIVE Systems Corporation. XRemote is a registered trademark of Network Computing Devices, Inc. Oracle is a registered trademark of Oracle Corporation, Redwood City, California. Java is a trademark of Sun Microsystems, Inc. All other brand and product names are or may be trademarks of, and are used to identify products or services of, their respective owners.

The SCO software that accompanies this publication is commercial computer software and, together with any related documentation, is subject to the restrictions on US Government use as set forth below. If this procurement is for a DOD agency, the following DFAR Restricted Rights Legend applies:

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the US Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013. Contractor/Manufacturer is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, CA 95060.

If this procurement is for a civilian government agency, this FAR Restricted Rights Legend applies:

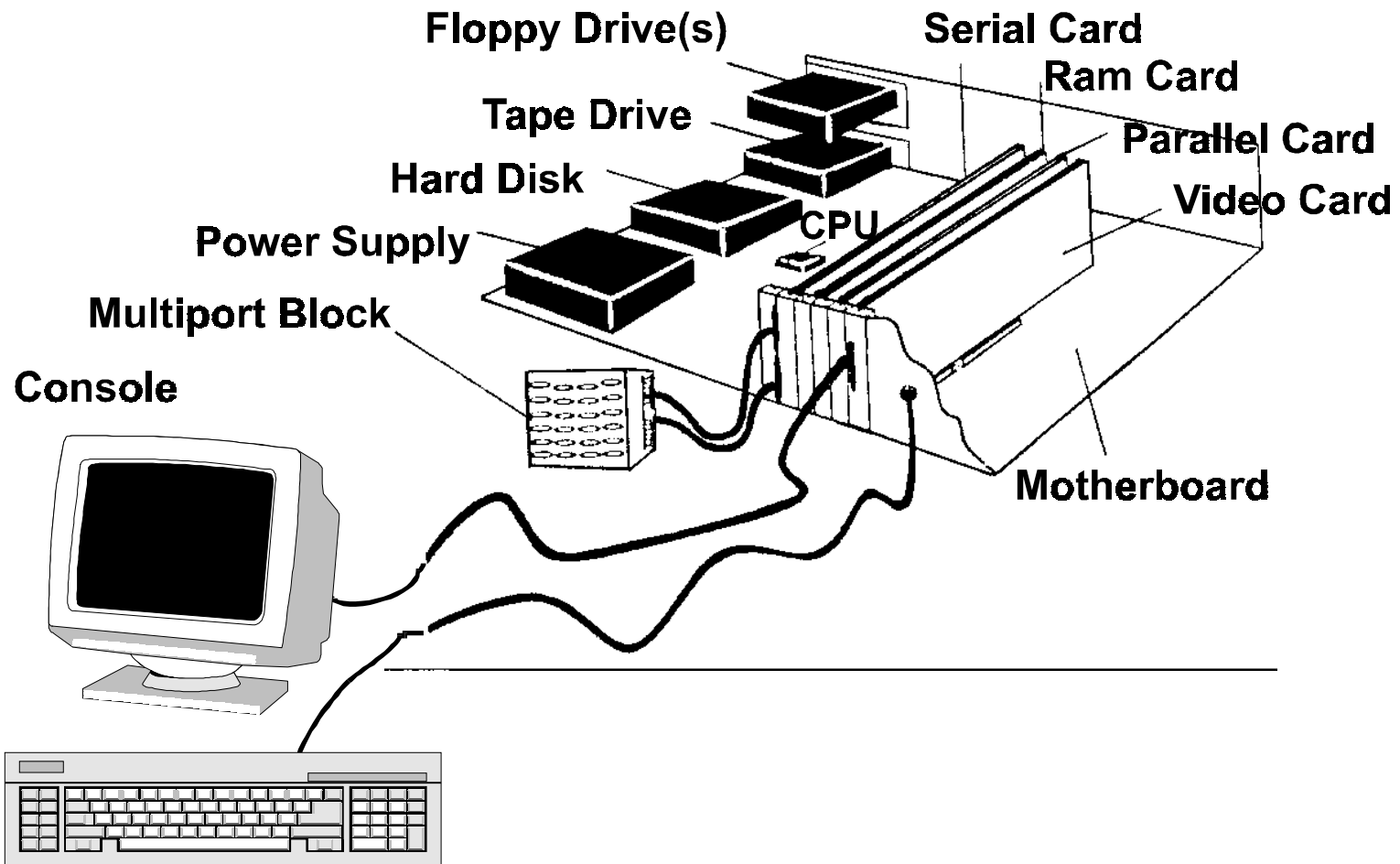
RESTRICTED RIGHTS LEGEND: This computer software is submitted with restricted rights under Government Contract No. _____ (and Subcontract No. _____, if appropriate). It may not be used, reproduced, or disclosed by the Government except as provided in paragraph (g)(3)(i) of FAR Clause 52.227-14 alt III or as otherwise expressly stated in the contract. Contractor/Manufacturer is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, CA 95060.

The copyrighted software that accompanies this publication is licensed to the End User only for use in strict accordance with the End User License Agreement, which should be read carefully before commencing use of the software.

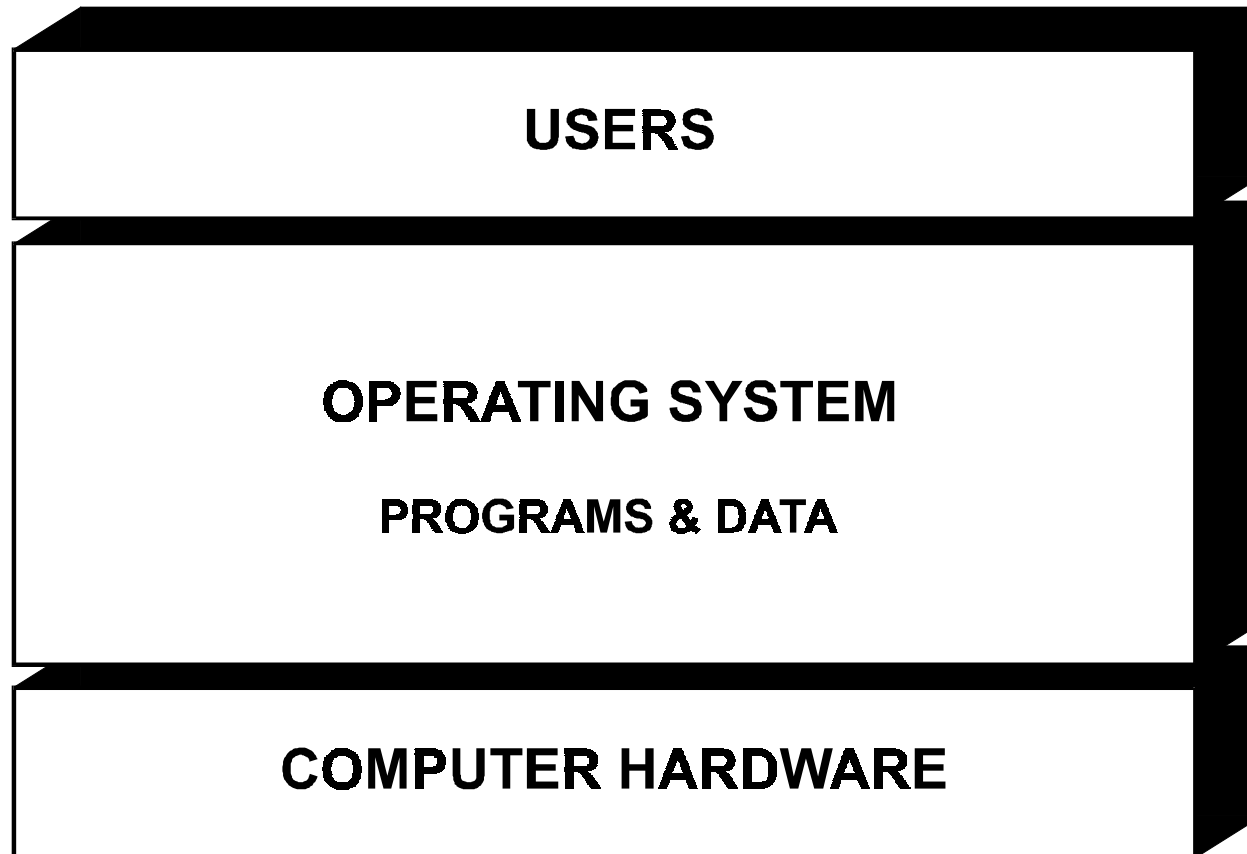
Overview of SCO UNIX Systems

- Describe basic operating system terminology
- List the primary functions of an operating system
- Identify key features of SCO UNIX systems
- List the basic components of SCO UNIX systems

Basic Components: Hardware



Software Interface



Basic Components: Software

OPERATING SYSTEMS

UNIX
XENIX
Windows NT
Macintosh System

SYSTEM SOFTWARE

X Windows
CDE Desktop
Merge
VERITAS

APPLICATIONS

WordPerfect
Lotus Notes
FrameMaker
Microsoft Word

Functions of an Operating System

- Hardware control
- Resource management
- User interfaces
- Input/output processing
- System monitoring
- Communications

Advantages of SCO UNIX

- Multiple users
- Multiple tasks
- Shared peripherals
- Open communications
- Large user base

Command Line Interfaces

Bourne shell (sh)

\$ _

C shell (csh)

% _

Korn shell (ksh)

\$ _

Menuing Interface: scosh

SCO Shell

Application Manager Print Utility Options Quit

Select an application

/u/kathy

Friday May 10, 1994 10:20

Mail -Electronic Mail
Calendar -Event Scheduler
Calculator -Math: + - * / %
System -Run A System Shell

May						
S	M	Tu	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

8:30 - 9:00 Introduction
9:00 - 10:00 Lesson 1: An Overview of SCO System V
10:00 - 10:15 Break
10:15 - 12:00 Lesson 2: Accessing SCO System V
12:00 - 1:00 Lunch
1:00 - 2:30 Lesson 3: Files and Directories

Menuing Interface: scoadmin

SCOadmin

File View Options

Help

* Account Manager

Backup Manager

Hardware/Kernel Manager

License Manager

Software Manager

Video Configuration Manager

> Filesystems

> Mail

> Networks

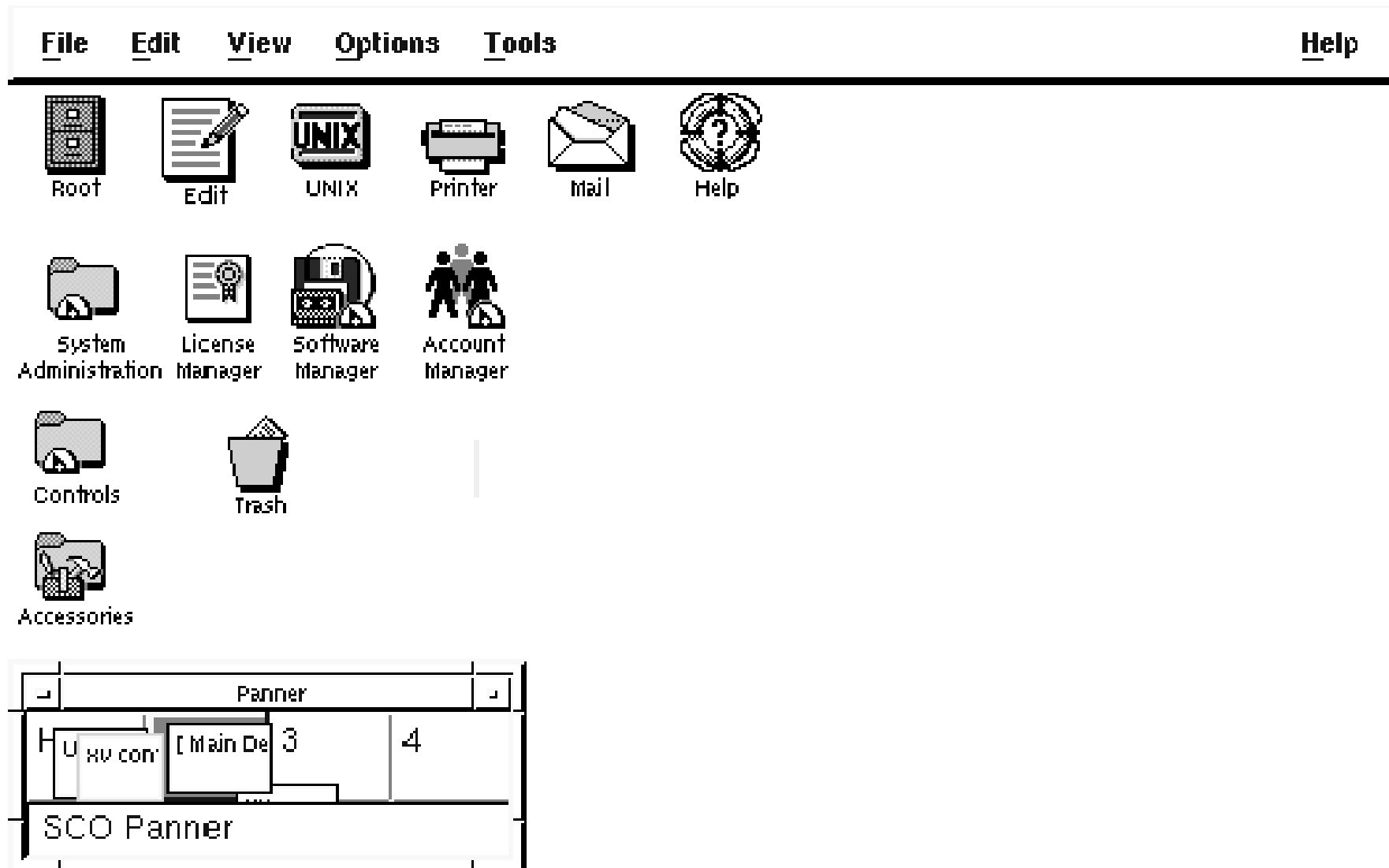
> Printers

> Sysadmsh Legacy

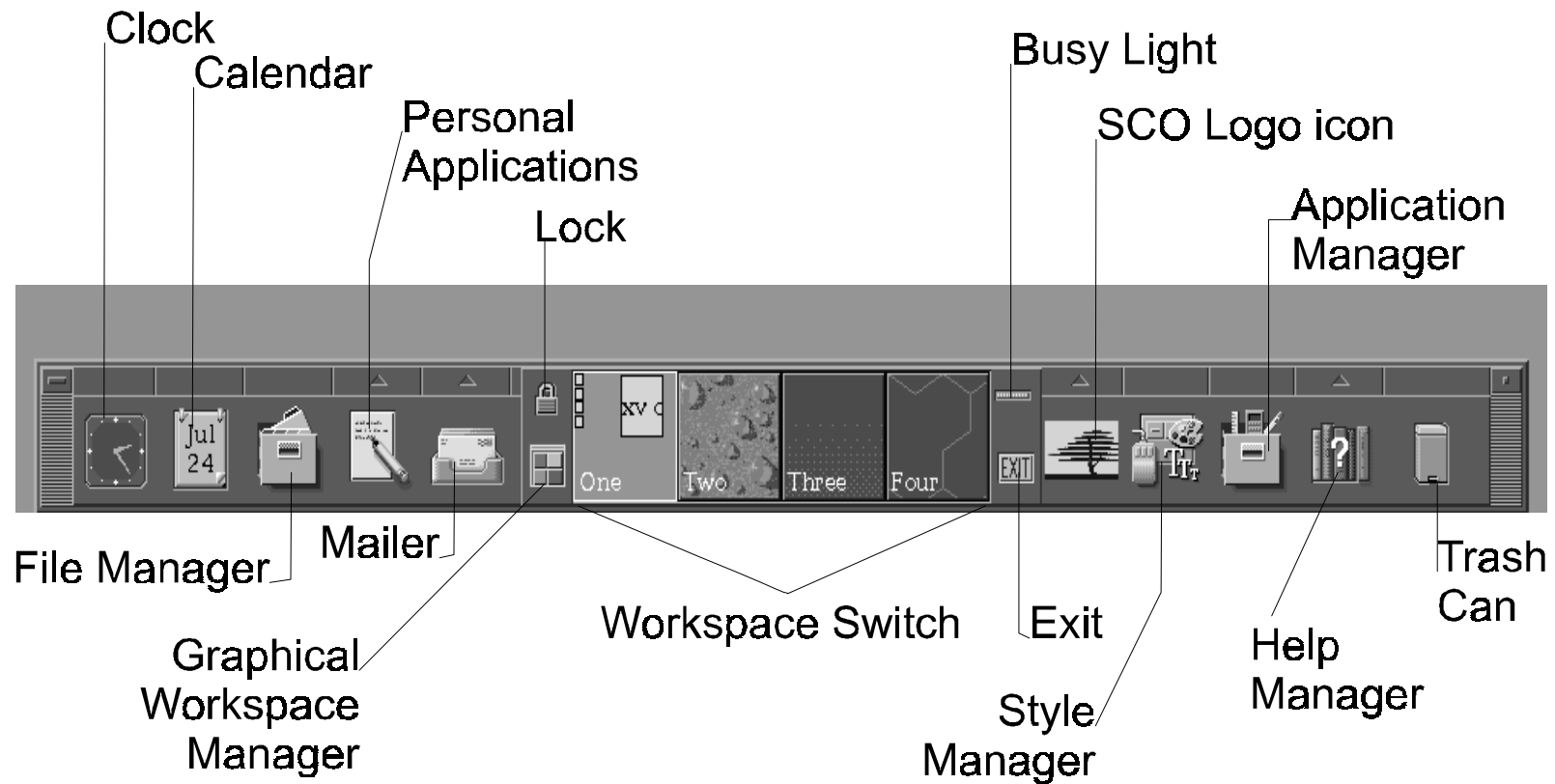
> System

[(++)]

Graphical User Interface: SCO OpenServer



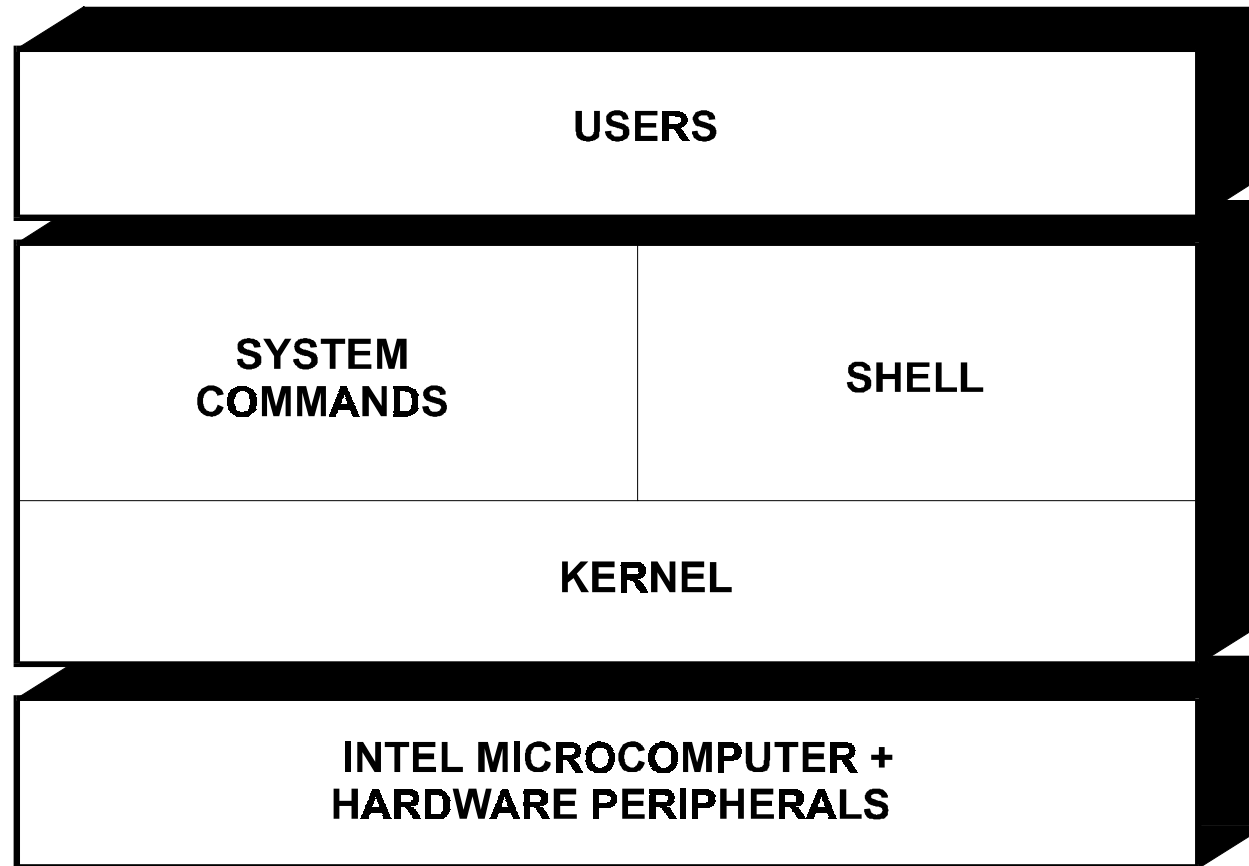
Graphical User Interface: UnixWare 7



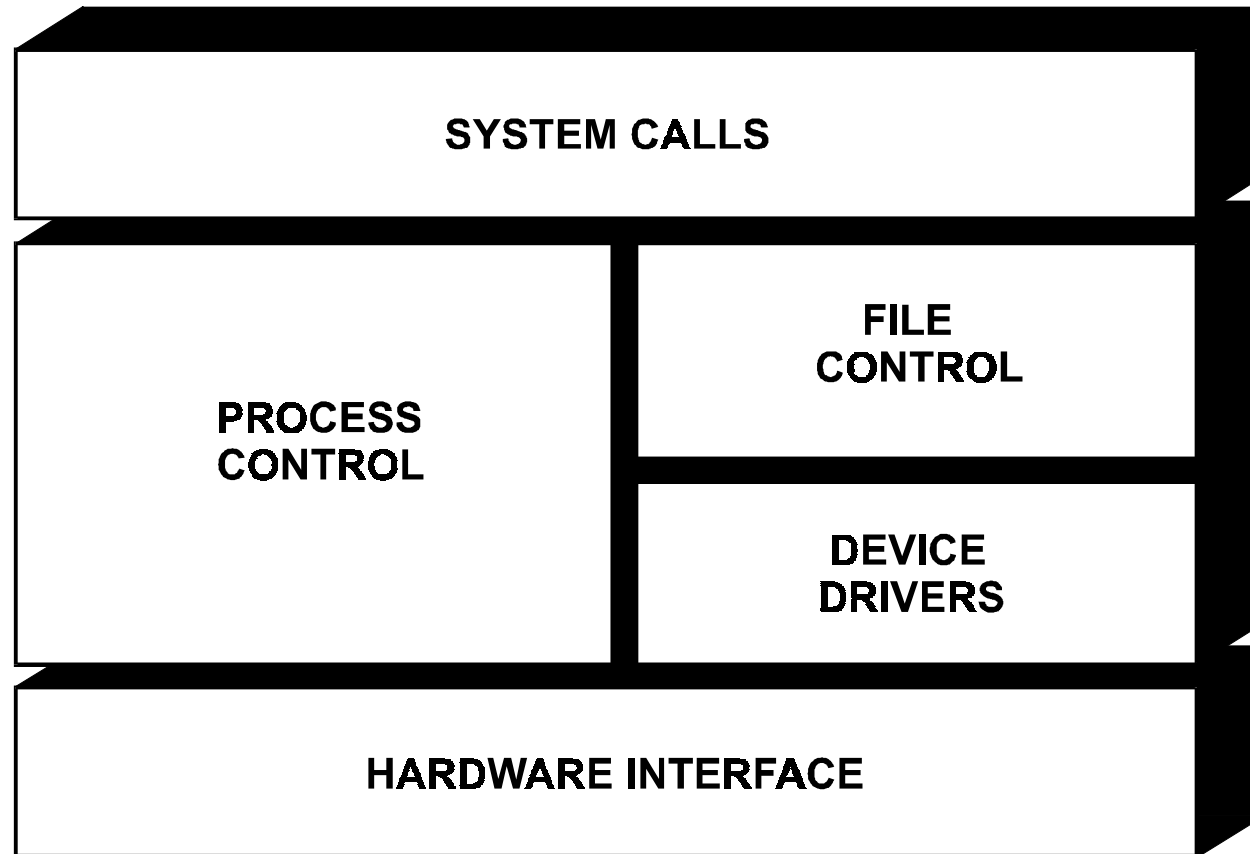
Features of SCO UNIX Systems

- Security
- Scalability
- Open technology
- Interoperability

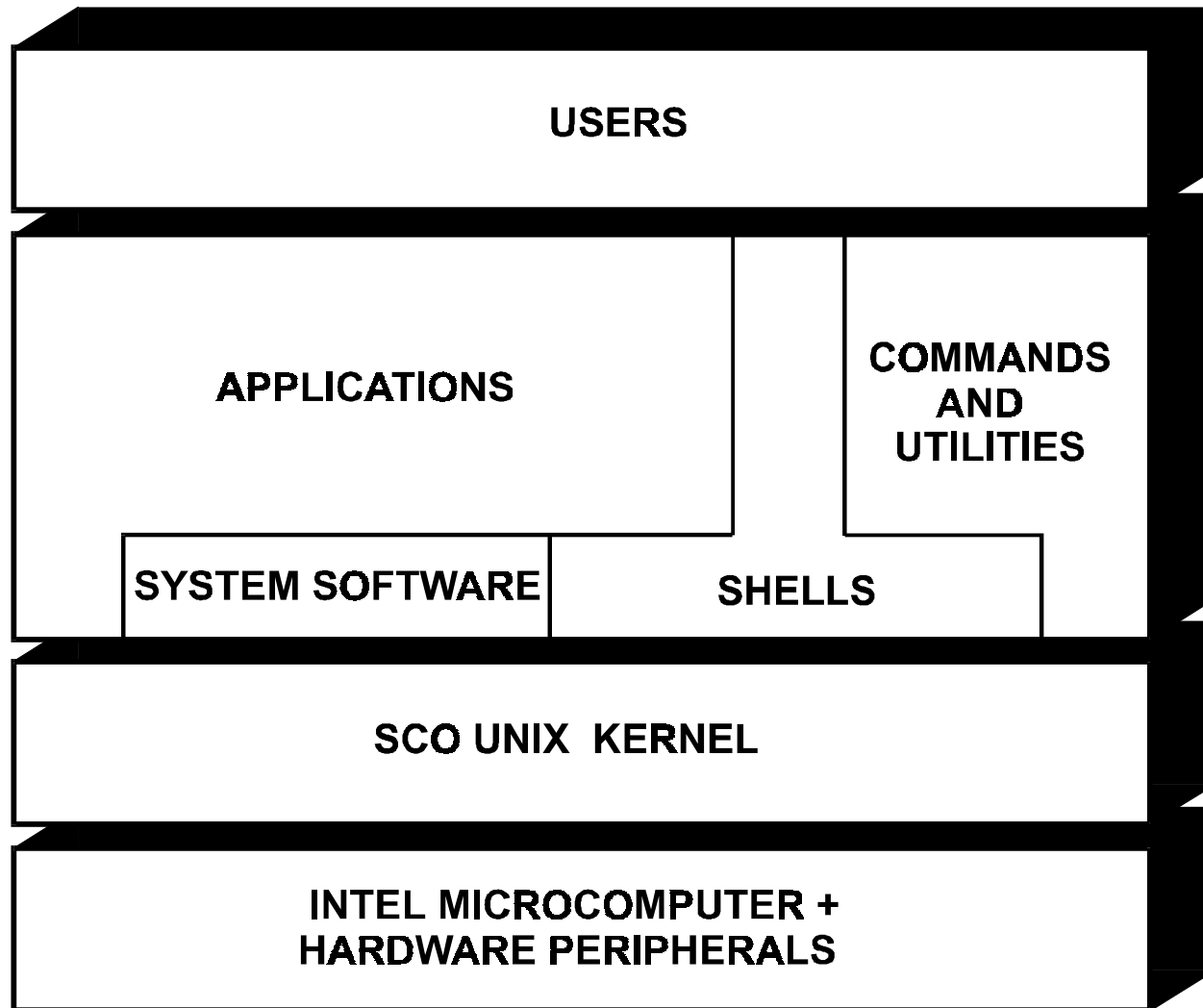
Components of the Operating System



SCO UNIX Kernel



SCO UNIX Architecture



Accessing the SCO UNIX Operating Systems

- Identify secure practices of computer use
- Access the command-line interface
- Perform simple tasks by entering commands
- Identify ways to get help

Logging In: UNIX

ibm486

SCO OpenServer(TM) Release 5

ibm486!login: **larona**

Password:

Changing Your Password: UNIX

```
$ passwd
Setting password for user: joesco
Old password:
Last successful password change for joesco: Mon May 29 14:40:06 1995
Last unsuccessful password change for joesco: Fri May 26 12:21:01 1995
```

Choose Password

You can choose whether you pick a password,
or have the system create one for you.

1. Pick a password
2. Pronouncable password will be generated for you

```
Enter choice (default is 1): 1
Please enter new password (at least 5 characters):
New password:
Re-enter password:
$ _
```

Command Syntax

COMMAND

Tells the shell what to do
(in this case, list users)

ARGUMENTS

Denote which file(s) the command
will process (In this case, the
file */etc/wtmp*)

\$ who -H /etc/wtmp

SHELL PROMPT

OPTIONS

Modify the behavior of the
command (in this case, displays
a header line)

Basic Commands

banner [text]	Displays text in a large format
cal	Displays a calendar
clear	Clears the screen
date	Displays the current time and day
echo [text]	Displays the text you type
finger	Displays information on users
who	Displays the users who are currently on the system

Accessing Online Manual Pages

<code>man</code>	Displays a usage message for man
<code>man <i>command</i></code>	Displays the manual page for the specified command
<code>man -k <i>keyword</i></code>	Locates commands by keyword (same as the command, apropos)

The **banner** Manual Page

`$ man banner`

UNIX System V

BANNER(C)

Name

`banner` - prints large letters

Syntax

`banner strings`

Description

`banner` prints its arguments (each up to 10 characters long) in large letters on the standard output. This is useful for printing names at the front of printouts.

See Also

`echo(C)`

Standards Conference

`banner` is in conformant with:

AT&T SVID Issue 2, Select Code 307-127

and the X/Open Portability Guide II of January 1987

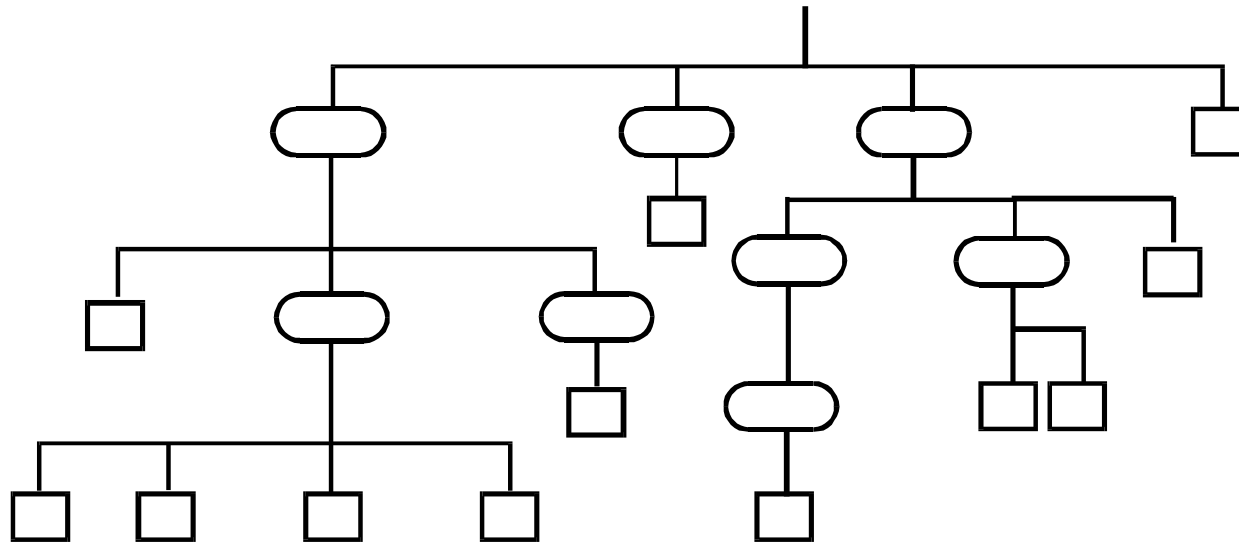
(printed 2/15/90)

BANNER(C)

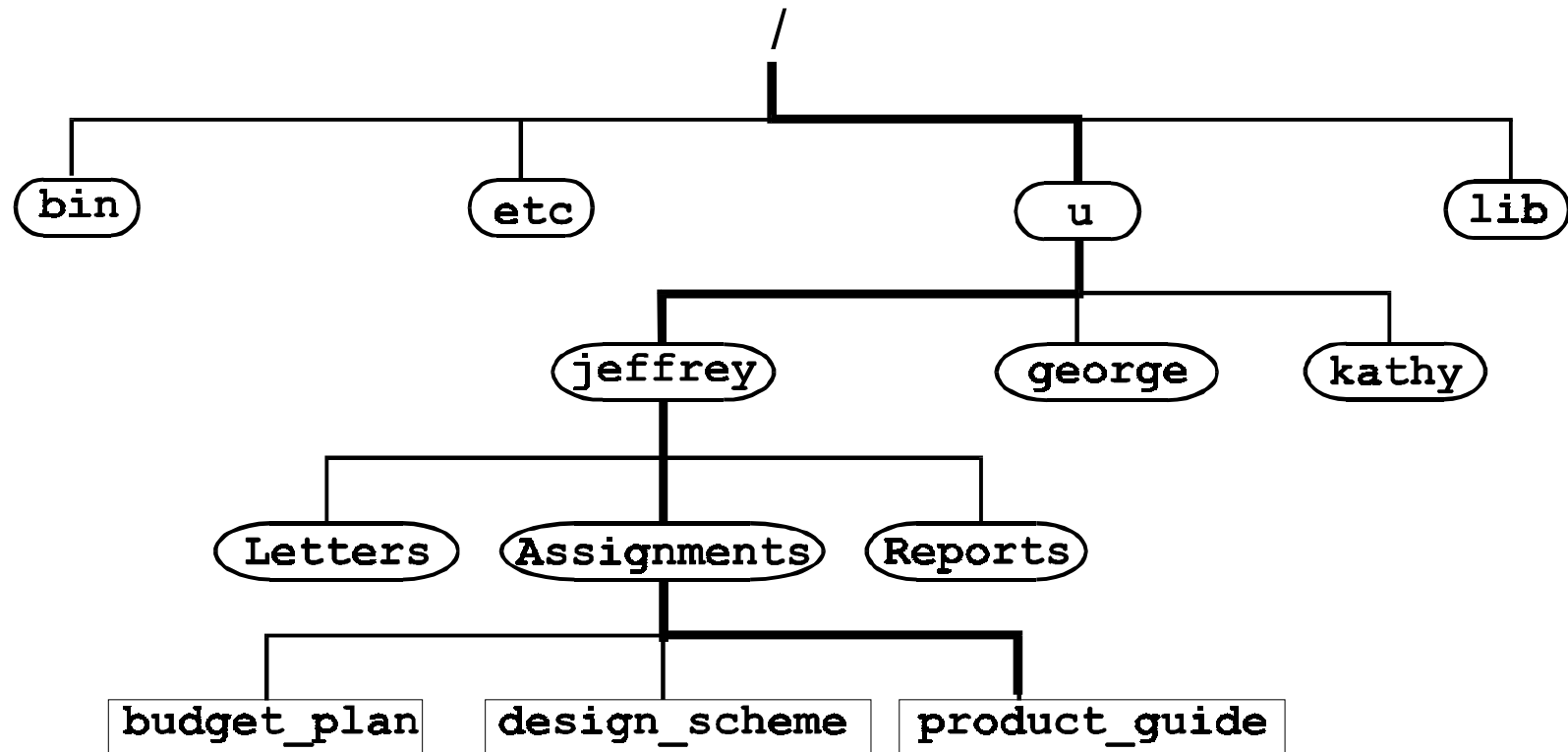
Files and Directories

- Describe the heirarchical directory structure
- Identify the different file types supported by SCO UNIX
- Navigate around the filesystem using `cd`
- Manage your home directory
- Use filename expansion characters to access groups of files
- Use appropriate commands for finding and comparing files

Tree Structure

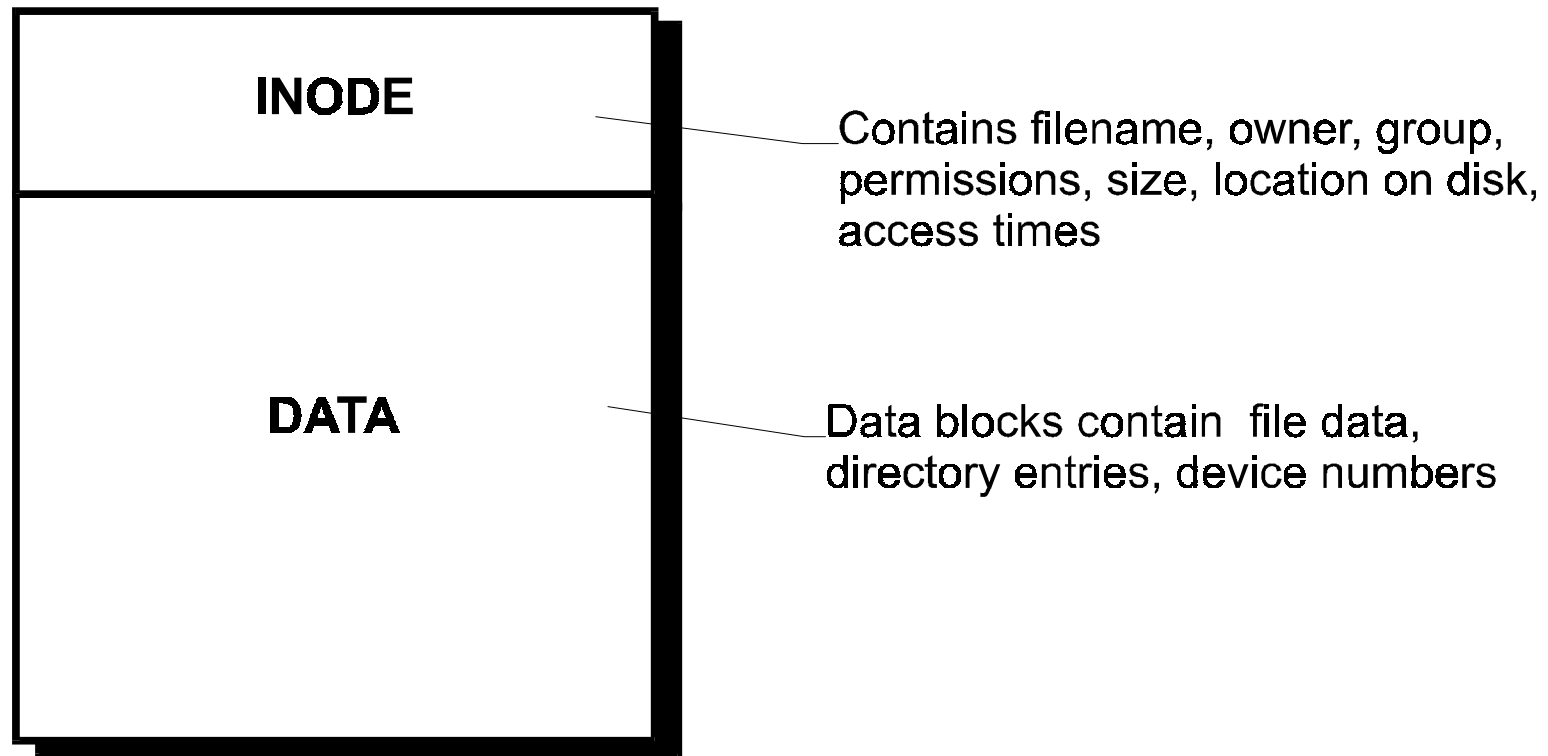


File and Directory Structure



path: /u/jeffrey/Assignments/product_guide

A UNIX File



File Types

Ordinary	- r w - r w - - - -
Directory	d r w x r w x - - -
Symbolic link	l r w x r w x r w x
Device files	c r w - r w - - - -
	b r w - r w - - - -

The If Command

```
$ if
```

```
Letters/  Projects/  
News/    Reports/
```

Directories

```
datecal*  
mbox
```

Executable file

```
mystery@  proposal2  
proposal1 schedule
```

Symbolic link

Text files

The l Command

Disk blocks

\$ l

total 44

drwx-----	2	jon	train	68	Aug 13 10:23	Letters
drwxrwxrwx	2	jon	train	41	Jul 23 14:12	News
drwxrwx---	2	jon	train	113	Aug 02 09:15	Reports
lrwxrwxrwx	1	jon	train	13	Sep 09 08:10	project -> /Rpts/proj
-rw-r--r--	1	jon	train	2095	Aug 26 11:45	schedule
-rwxr-x---	1	jon	train	762	Jun 02 09:53	timecal

Permissions

Owner

Group

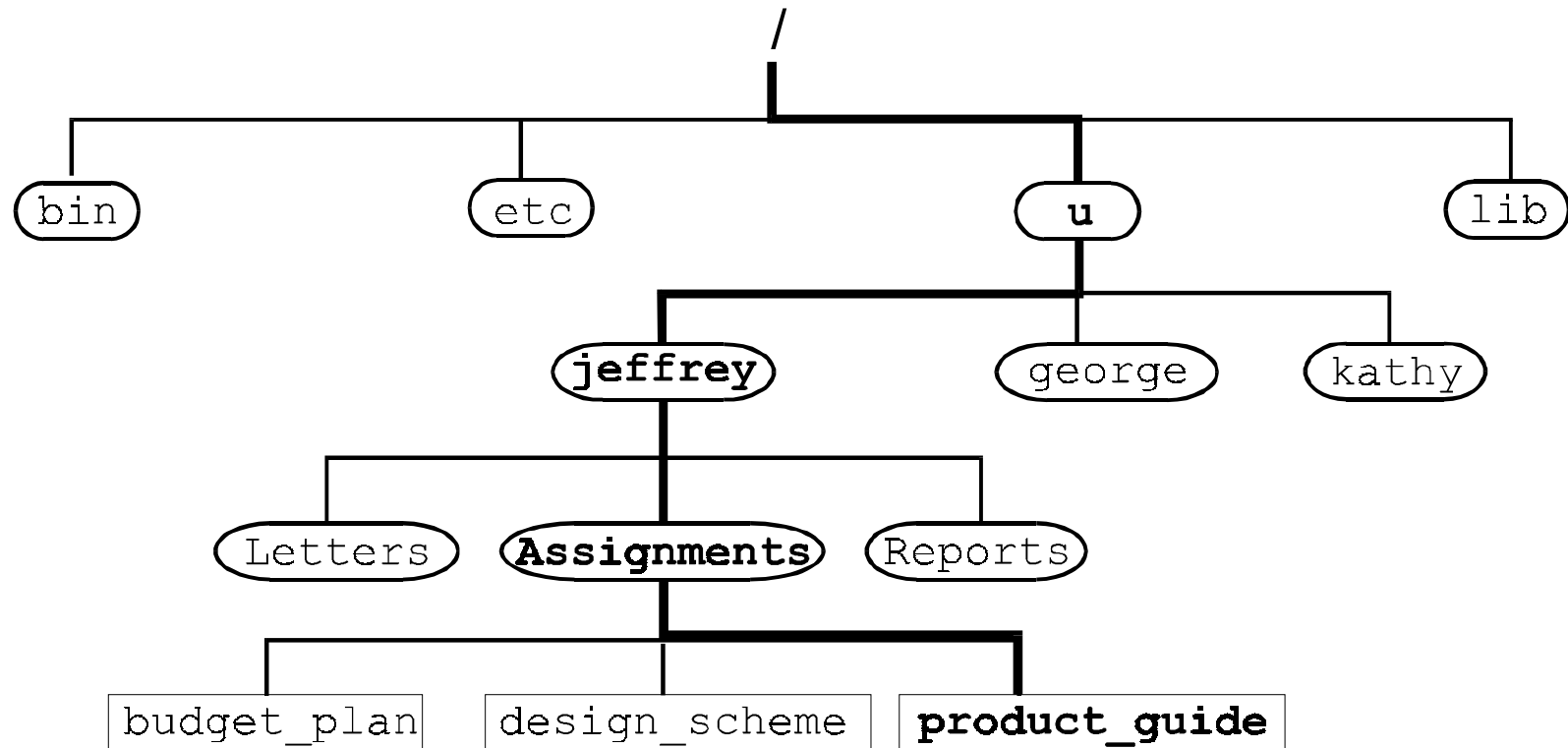
Date modified

Filename

Number of
links

File size

Types of Pathnames



Absolute pathname:

/u/jeffrey/Assignments/product_guide

Relative pathname (when the current directory is */u/jeffrey*)

Assignments/product_guide

Displaying Text Files

cat	Displays entire file to the screen
more	Displays a text file one page at a time
pg	Displays a text file one page at a time
head	Displays the first ten lines of a file
tail	Displays the last ten lines of a file

The **more** Command

\$ more letter

Dear George,

.
. .
. .
. .
. .
. .

The new release will be on the first of the month at the Education Center on Broadway. I hope you can make it.

--More-- (90%)

[SPACE] displays the next screen of text

[ENTER] displays the next line of text

[H] displays the help screen

Examples: **cp** and **mv**

```
$ command    source    destination
```

```
$ cp apples oranges peaches Foods
```

Copies *apples*, *oranges*, and *peaches* to the directory *Foods*.

```
$ cp oranges Foods/citrus
```

Copies the file *oranges* to the directory *Foods* and names it *citrus*.

```
$ mv fruit apples
```

Renames the file *fruit* to *apples*.

Note: The directory *Foods* must exist before you attempt the above examples.

Linking Files

Hard links

```
$ ln greeting blink
```

Symbolic links

```
$ ln -s greeting slink
```

```
$ ls -l greeting ?link
```

```
-rw-r--r-- 2 jim group 111 Apr 6 greeting
-rw-r--r-- 2 jim group 111 Apr 6 blink
lrwxrwxrwx 1 jim group   8 Apr 6 slink -> greeting
```

Filename Expansion

*	Matches any pattern of zero or more characters
?	Matches any single character
[pattern]	Matches any one character in the pattern set
[!pattern]	Matches any one character not in the pattern set

Using Filename Expansion Characters

```
$ lf
```

```
Letters/  Projects/  mysteryA  mysteryC  proposal2  
News/     Reports/   mysteryB  proposal1  timecal*
```

```
$ cat prop*
```

Displays *proposal1* and *proposal2*.

```
$ more mystery?
```

Displays the contents of all three *mystery* files on the screen one page at a time.

```
$ head mystery[AC]
```

Displays the first 10 lines of the files *mysteryA* and *mysteryC*.

Removing Files: `rm`

```
$ rm apples
```

```
$
```

This removes the file `apples` without any further input from the user.

```
$ rm -i apples
```

```
apples: ?
```

The `i` option stands for interactive. The computer asks if you really want to remove the file *apples*.
Type `y`  to remove the file.

Finding Files

```
$ find . -name chapter1 -print
```

Search for files named *chapter1* in the current directory (.) and all directories below it, and send the pathname of any such files to the screen.

```
$ find .. -name 'sonnet*' -print
```

Searches for files whose names begin with 'sonnet' in the parent directory (..) and all directories below it. Quotes must be used around the filename when shell metacharacters are used.

Finding Strings

```
$ grep henry /etc/passwd
```

Search the file */etc/password* for all lines containing the name *henry*.

```
$ grep 'my beloved' Poems/*/*
```

Search for the string *my beloved* in all files that are in subdirectories of the *Poems* directory.

Commands for Comparing Files

- cmp** Compares two files and reports the first character that differs.
- diff** Compares two text files and displays all lines that differ between the two files.
- sum** Computes a unique checksum value for a file.

Comparing Files

```
$ diff greetings /etc/motd
2c2
< Welcome to UNIX
---
> Welcome to SCO UNIX System V
$
$ cmp /usr/bin/more /usr/bin/pg
/usr/bin/more /usr/bin/pg differ: char 25, line 1
$
$ sum /bin/mv
58109 60 /bin/mv
```

File Security

- Identify permissions for ordinary and directory files
- Use **chmod** to change file permissions
- Reassign user and group file ownership
- Define the default permissions for new files

File Ownership

```
$ l letter
```

```
-rw-r--r-- 1 john group 1044 Mar 27 letter
```

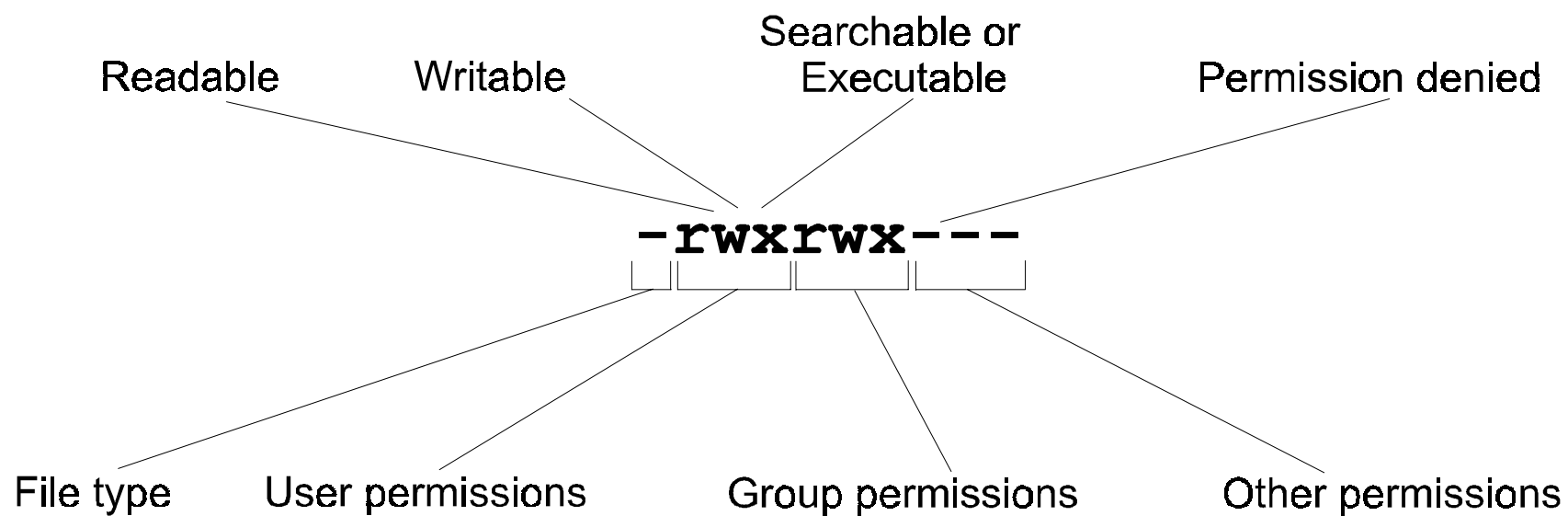
```
$ chgrp sys letter
```

```
$ chown root letter
```

```
$ l letter
```

```
-rw-r--r-- 1 root sys 1044 Mar 27 letter
```

Permissions



File Permission Types

READ

Enables users to open files and read its contents using:
more, cat, grep, sort, view, <

WRITE

Enables users to open a file and change its contents using:
vi, word, >

EXECUTE

Enables users to execute files as commands

Directory Permission Types

READ	Users can view filenames in the directory
WRITE	Users can create or delete files in the directory
SEARCH	Users can search in the directory and change to it using the cd command

Changing Permissions: Symbolic Mode

```
$ 1
```

```
drwx----- 3 helga training 68 Aug 24 14:25 Letters
-rw-r--r-- 1 helga training 2095 Aug 24 14:32 schedule
-rwxrwxr-- 1 helga training 762 Aug 25 19:41 timecal
```

Both the user and the group have read, write, and execute permissions, while others have only read permission.

```
$ chmod g-w, o+x-r timecal
```

```
$ 1
```

```
drwx----- 3 helga training 68 Aug 24 14:25 Letters
-rw-r--r-- 1 helga training 2095 Aug 24 14:32 schedule
-rwxr-x--x 1 helga training 762 Aug 25 19:41 timecal
```

The user retains all permissions, while the group loses write permission, and others lose read permission but gain execute permission.

Changing Permissions: Absolute Mode

Each permission is assigned a number:

r = 4 w = 2 x = 1

Add these numbers for each user category:

-rw-r-----			
421	421	421	
↙	↙	↙	
6	4	0	

Use with **chmod**

\$ chmod 640 file

Changing Permissions: Absolute Mode

\$ 1

drwx-----	3	sandy	training	68	Aug	24	14:25	Letters
-rw-r--r--	1	sandy	training	2095	Aug	24	14:32	schedule
-rwxrwxr-x	1	sandy	training	762	Aug	25	19:41	timecal
7 7 5								

\$ chmod 751 timecal

\$ 1

drwx-----	3	sandy	training	68	Aug	24	14:25	Letters
-rw-r--r--	1	sandy	training	2095	Aug	24	14:32	schedule
-rwxr-x--x	1	sandy	training	762	Aug	25	19:41	timecal
7 5 1								

4 = r: readable
2 = w: writeable
1 = x: executable

Permissions Exercise

Directory1: drwxr-x--x

file1: -rw-rw-r--

owner	☐ modify	☐ delete	☐ read	☐ execute
group	☐ modify	☐ delete	☐ read	☐ execute
other	☐ modify	☐ delete	☐ read	☐ execute

file2: -rwxr-xr-x

owner	☐ modify	☐ delete	☐ read	☐ execute
group	☐ modify	☐ delete	☐ read	☐ execute
other	☐ modify	☐ delete	☐ read	☐ execute

file3: -r-xr-xr--

owner	☐ modify	☐ delete	☐ read	☐ execute
group	☐ modify	☐ delete	☐ read	☐ execute
other	☐ modify	☐ delete	☐ read	☐ execute

Directory2: drwxrwxr-x

file1: -rwxr-xr-x

owner	☐ modify	☐ delete	☐ read	☐ execute
group	☐ modify	☐ delete	☐ read	☐ execute
other	☐ modify	☐ delete	☐ read	☐ execute

Text Editing

- List the primary functions of a text editor
- Add, replace, change, and delete text in a file using the **vi** editor
- Control the editing environment by setting preferences
- Use the system **spell** program to check the spelling in your text files

Invoking vi

This is what vi looks like.

~

~

~

~

~

~

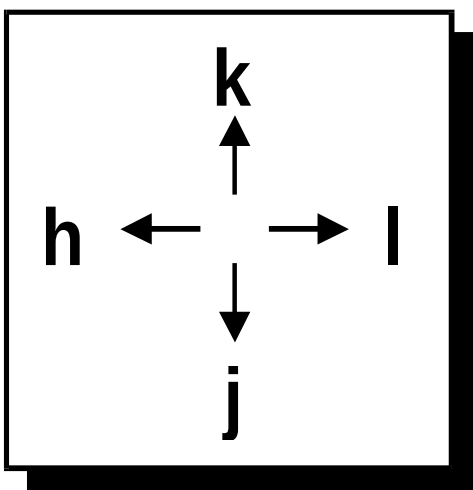
~

"welcome" 1 line, 28 characters

Saving and Quitting in vi

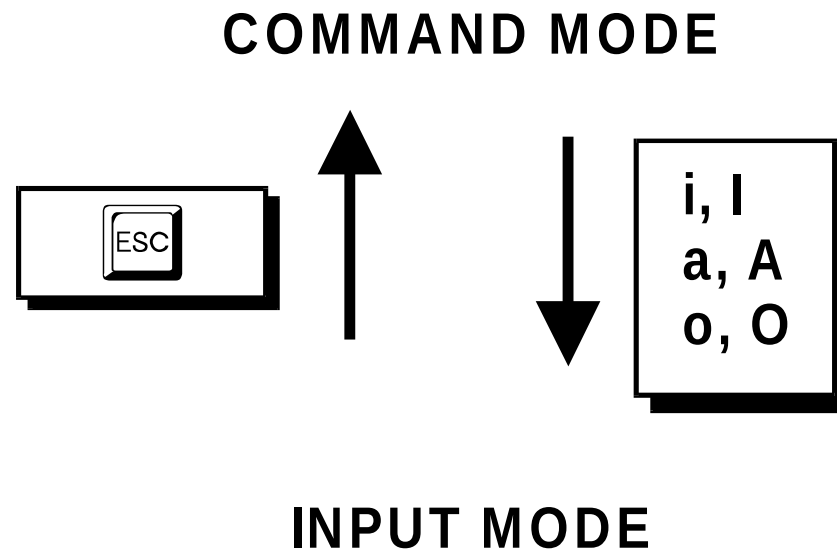
:q	Quit, if no changes were made
:q!	Quit, throw away any changes
:w	Write (save) the file
:wq	Write the file then quit vi
ZZ	Write the file if any changes were made, then quit
:x	Same as ZZ
:w <i>newname</i>	Write the file to a new name

Moving in vi

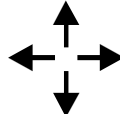


- w** Moves the cursor to the beginning of the next word.
- b** Moves the cursor to the beginning of the previous word.
- \$** Moves the cursor to the end of the current line.
- .** Moves the cursor to the beginning of the current line.
-  u** Moves the cursor up one half screen.
-  d** Moves the cursor down one half screen.

Switching Between vi Modes



Basic vi Commands

Insert a i o	Delete x dw dd	Modifying Text r cw c\$
Saving & Quitting :wq :q!	Cursor Movement  G 1G	Command Mode 

Advanced vi Commands

Buffers p P	Copy y yy	Search / n
Special Commands  g		Command Execution !

Sample .exrc File

```
set wrapmargin=8  
set showmode  
set autoindent  
set number
```

~

~

~

~

~

~

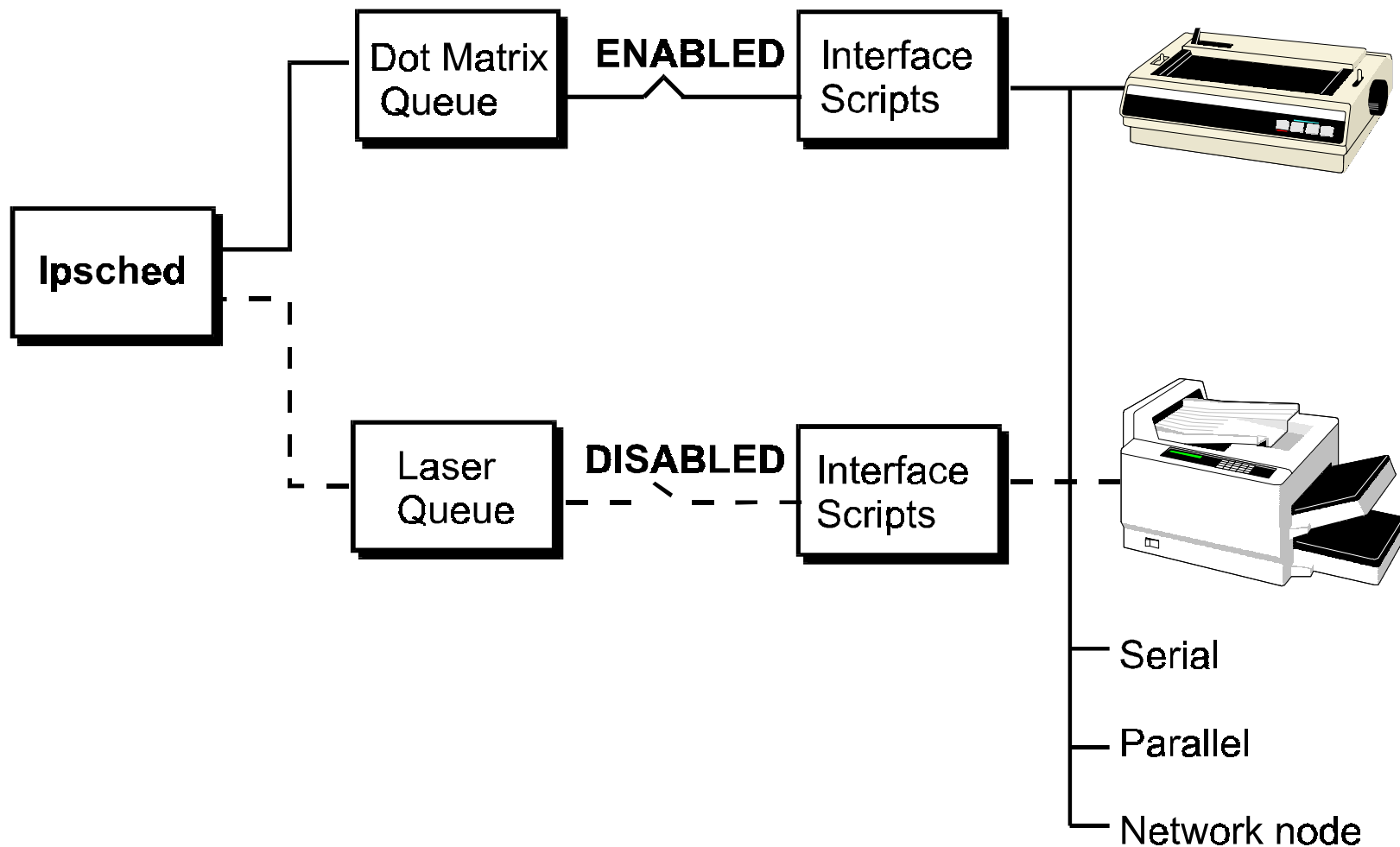
".exrc" [New file] 4 lines, 56 characters

Printing

- Print files to the default printer using the **lp** command
- Direct print requests to other printers
- Monitor the status of print requests and printers
- Cancel print requests
- Describe options to **lp**

Printing a File

```
$ lp file
```



Basic Print Commands

lp filename	Prints a specified file or files
lpstat	Checks the current status of the default printer
cancel request-ID	Cancels a print request

Printer Status

```
5$ lpstat -t
```

```
scheduler is running
```

```
system default destination: laserjet
```

```
device for laserjet: /dev/ttyla
```

```
device for netlp: /dev/null
```

```
device for matrix: /dev/tty2a
```

```
laserjet accepting requests since Thu Jan  8 01:13:29 1998
```

```
netlp accepting requests since Wed Dec 10 02:15:10 1997
```

```
matrix not accepting requests since Thu Jan  8 01:13:28  
1998
```

```
    Disabled/Rejected by root.
```

```
printer laserjet is idle. enabled since Thu Jan  8 01:12:28  
1998. available.
```

```
printer netlp is idle. enabled since Wed Dec 10 02:15:11  
1997. available.
```

```
printer matrix disabled since Thu Jan  8 01:13:28 1998.  
available.
```

```
    Disabled/Rejected by root.
```

```
6$
```

Canceling a Print Request

```
$ lpstat  
laser-201      herbert      5343      Aug 16 10:58  
  
$ cancel laser-201  
request "laser-201" canceled
```

Input and Output

- Identify the three open files an executing program is given when it is started
- Redirect input from files and output to files
- Define the terms pipe, filter, and tee
- Use pipes and tees to combine multiple commands

Standard Flow

Input comes from standard input (*stdin*)

Output goes to standard output (*stdout*)

Error messages go to standard error (*stderr*).



Redirection Syntax

```
command < file
```

```
command > file
```

```
command >> file
```

```
command > file 2>&1
```

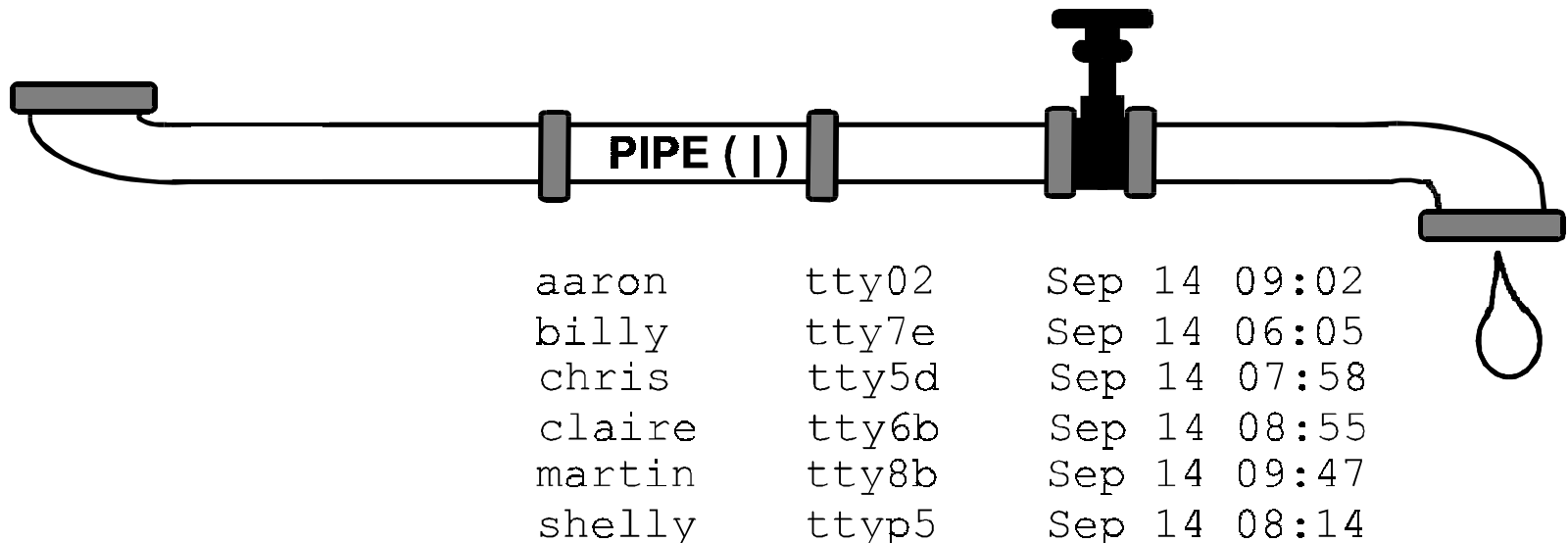
Using Pipes and Filters

\$ who

shelley	ttyp5	Sep 14 08:14
billy	tty7e	Sep 14 06:05
aaron	tty02	Sep 14 09:02
claire	tty6b	Sep 14 08:55
martin	tty8b	Sep 14 09:47
chris	tty5d	Sep 14 07:58

\$ who | sort

FILTER (SORT)



Extended Commands

```
$ grep 'john' employees | sort | pr | lp
```

Search the file *employees* for 'john,' alphabetize the data, format the data for the printer, and print the result.

```
$ grep 'john' employees | sort | tee johns | pr | lp
```

Search the file *employees* for 'john,' alphabetize the data, save the sorted data in the file *johns*, format the data for the printer, and print the result.

Process Control

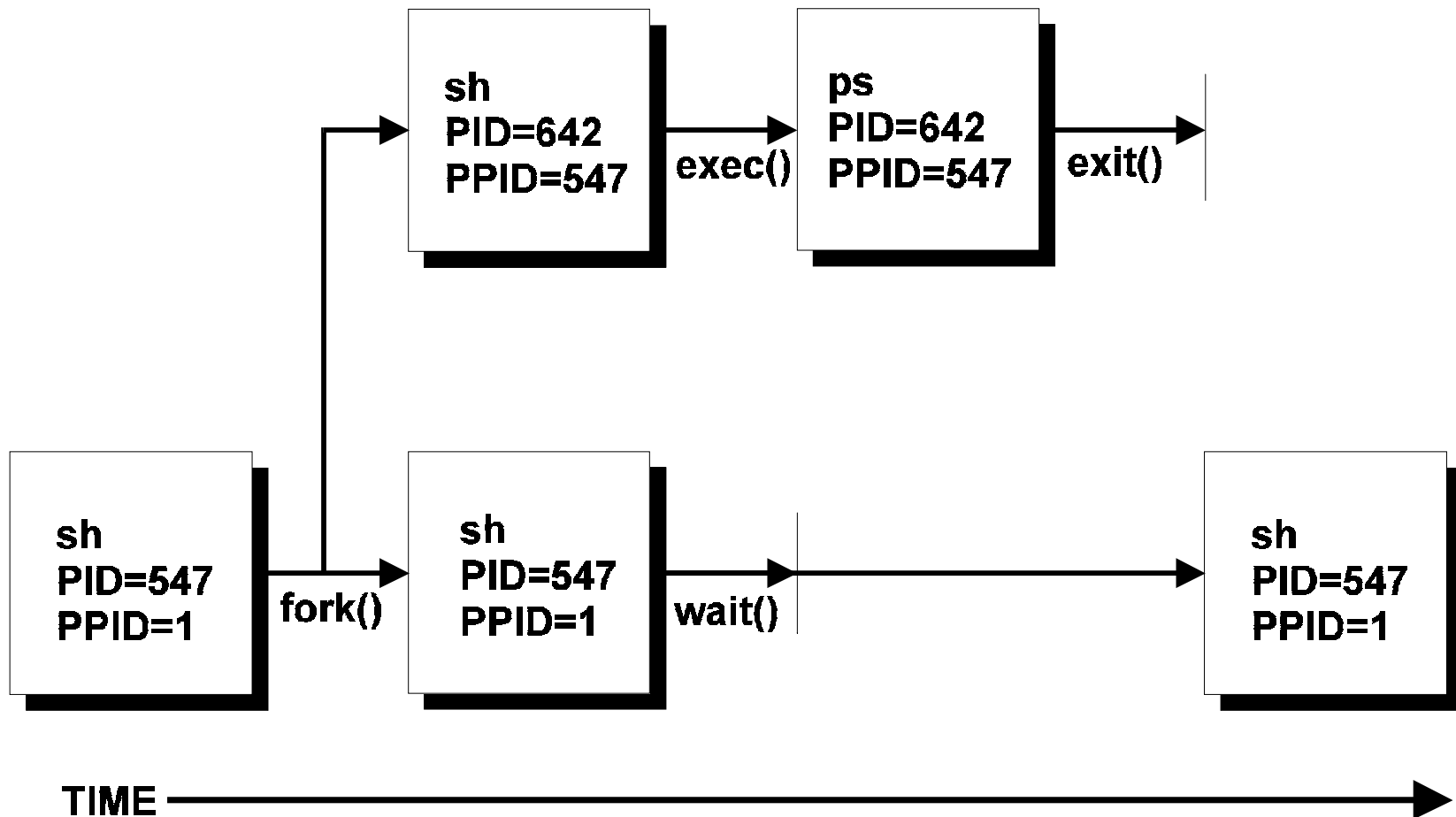
- Define a UNIX process
- Explain how processes are created and how they terminate
- View status information on your processes
- Run processes in the background
- Use signals to terminate processes
- List two ways of distributing process usage

A UNIX Process

PID	PPID
UID	GID
Code	
Data	
File Descriptor Table	
0	STDIN
1	STDOUT
2	STDERR
...	--
n	--

Command Execution

\$ ps

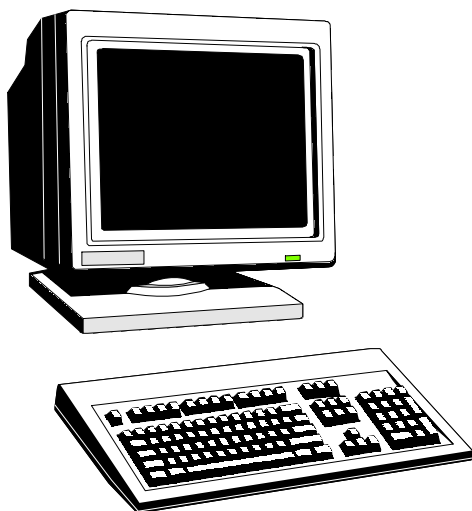


Using **ps**

Process ID number	Controlling terminal for the process	Cumulative execution time for the process	Command name
PID	TTY	TIME	COMMAND
389	5f	0:23	csh
470	5f	0:02	vi
482	5f	0:01	ps

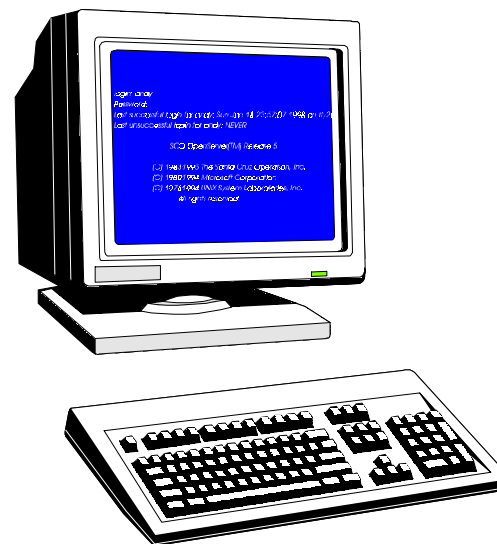
Hung Processes

Dead Terminal



1. 
2. 
3. Cycle power

Live Terminal



1. `ps -t tty`
2. `kill pid`
3. `ps -t tty`
4. `kill -9 pid`

Killing a Process

```
$ ps -u herbert
```

PID	TTY	TIME	COMMAND
389	5f	0:23	cs
470	5f	0:02	vi
482	5f	0:01	ps

```
$ kill 470
```

```
$
```

The Shell Environment

- List the shells offered by SCO UNIX operating systems
- List the features of the Korn shell
- Customize your login environment

Types of Shells

Graphical Interfaces

SCOadmin
SCO Shell
X.desktop
CDE
Panorama

Command-line interfaces

Bourne Shell
C Shell
Korn Shell

Shell Features

- Command editing and execution
- Use of filename expansion characters
- Built-in command library
- Redirection of input and output
- Background processing
- Variables to control the shell environment
- Programming interface

Shell Metacharacters



initiates execution

>

redirects *stdout*

;

separates commands

>>

appends *stdout*

&

executes in background

<

redirects *stdin*

|

pipe

“

quotes a string

?

represents any one character

[]

specifies one from a set of characters

*

represents any string of characters

\$

references a variable

Shell Environment Variables

HOME	Login directory
MAIL	System mailbox
PATH	Command search path
PS1	Primary prompt string
SHELL	Default shell
TERM	Terminal type

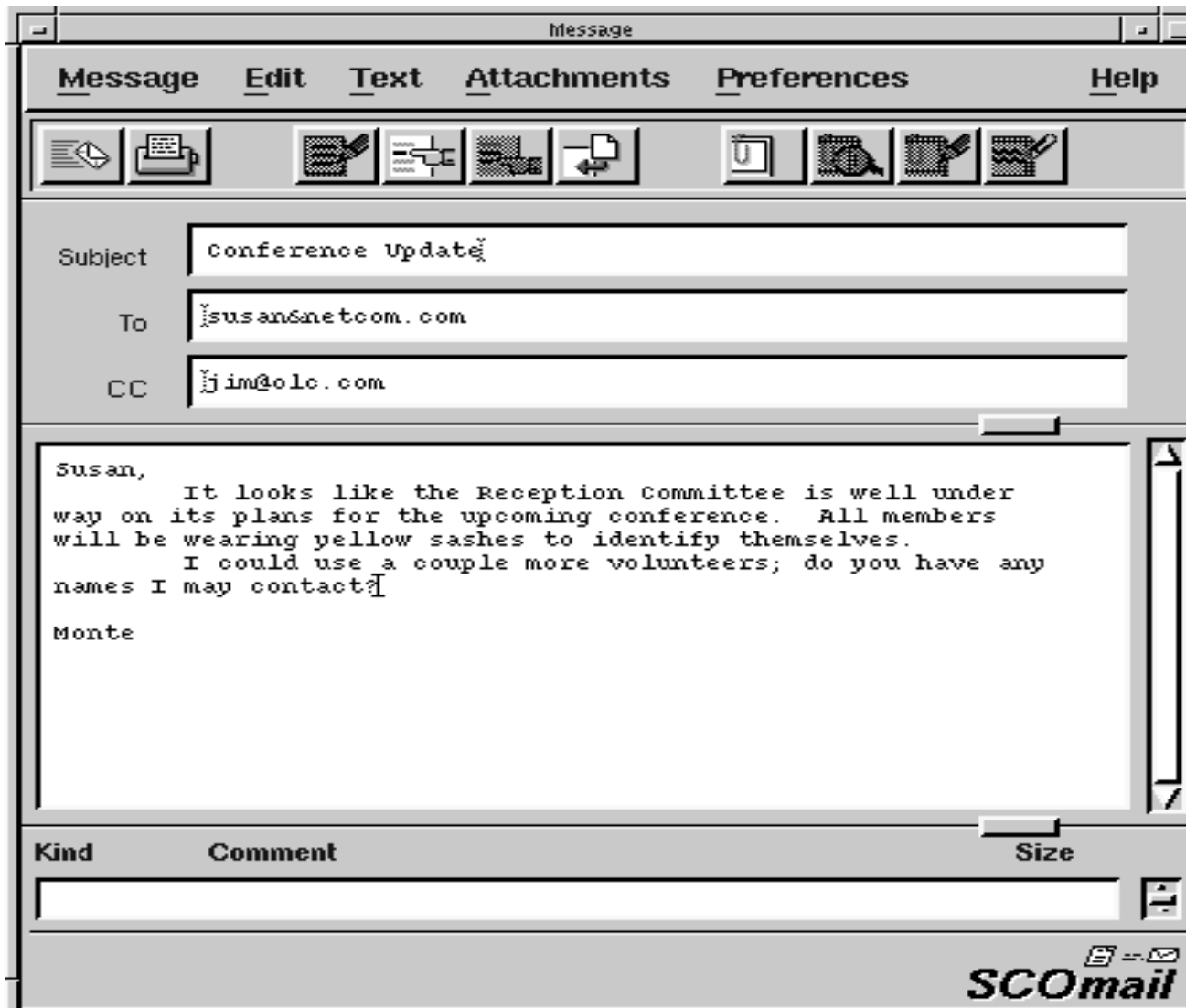
Sample .profile

```
#
# Copyright (C) 19881991 The Santa Cruz Operation, Inc.
# All Rights Reserved.
#
# User $HOME/.profile commands executed at login time
#
# search path:w
PATH=/bin:/usr/bin:/u/bin:/usr/local/bin:/local/bin:${HOME}/bin
:
#
# login name
LOGNAME=charles
#
# mailbox location
MAIL=${HOME}/.mailbox
#
# default permissions for new files
umask 022
#
# terminal setting
eval `tset m ansi:ansi m :\\e?ansi r s Q`
#
export PATH LOGNAME MAIL
```

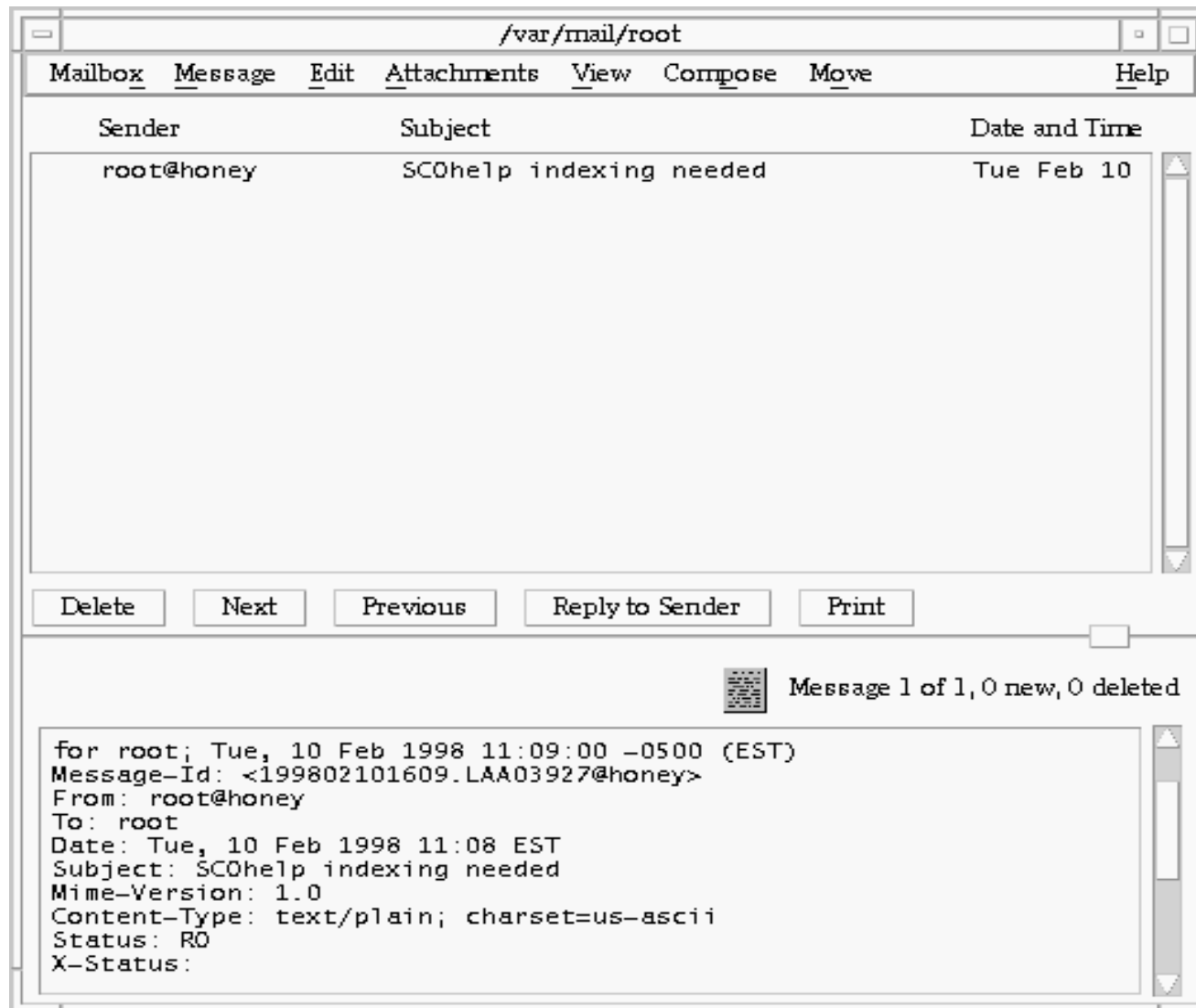
Electronic Mail

- Describe the SCO UNIX mail user agents
- Read mail messages
- Create and send messages
- Respond to mail
- Store messages using folders
- Customize your mail environment

SCO Mail



CDE Mail



Sending a Message

\$ mail susan

Subject: Meeting arrangements

Susan,

**The hospitality suite is coming along fine. Jim and I
will be meeting on Tuesday at lunch for half an hour.
Why don't you join us?**

Chris

Cc: jim

\$

Composing Mail

```
$ mail andy  
Subject: Project files
```

~< !find / -name prop*
executes the **find** command
and reads the result into the
message

The project files you need are in the directory:

```
~< !find / name prop*  
"find / name prop*" 3/77
```

~p

```
Message contains:  
To: andy  
Subject: Project file
```

The **~p** command displays
the message

The project files you need are in the directory:

```
/usr/Projects/development/proposal1  
/usr/Projects/maintenance/proposal4
```

```
(continue)  
Cc: peter
```

Reading Mail

\$ mail

SCO OpenServer Mail Release 5.0 Type ? for help.

"/usr/spool/mail/andy": 4 messages 4 new

N	4	chris	Thu Jan 15 07:31	31/987	Conference
N	3	susan	Thu Jan 15 07:31	25/870	Conference
N	2	rob	Thu Jan 15 07:30	12/308	Pure Poetry
>N	1	susan	Thu Jan 15 07:29	14/463	Project X

&1

Message 1:

From susan Thu Jan 15 07:29:52 1998

From: susan <susan@acme.com>

XMailer: SCO OpenServer Mail Release 5.0

To: andy@acme.com

Subject: Project X

Date: Thu, 15 Jan 98 7:29:52 CST

MessageID: <9801150729.aa15141@mailsrvr.acme.com>

Status: R

For all those who are interested, there will be a presentation this afternoon outlining the important aspects of project X. Please r.s.v.p by mailing the coordinator (Shirley Jones) by noon. See you there!

Susan

&

mail Commands

h [<i>login</i>]	List mail headers
z	Display the next screen of headers
z-	Display the previous screen of headers
d [<i>n</i>]	Delete message number <i>n</i>
u [<i>n</i>]	Undelete message number <i>n</i>
s [<i>n</i>] <i>filename</i>	Save message number <i>n</i> to filename
lp [<i>n</i>]	Send message number <i>n</i> to the default printer
cd <i>directory</i>	Change to the named directory
fo <i>filename</i>	Read the named mail folder
! <i>shell_command</i>	Execute a shell command
sh	Start a shell
x	Exit mail with no changes to the mailbox
q	Quit mail and saves changes to the mailbox
?	Display the mail help screen

Creating Folders

& **h**

>U 3 andy Thu Jan 15 22:43 5291/327578 Support

Budget

U 2 peter Thu Jan 15 22:43 4822/302181 Training

Budget

1 susan Thu Jan 15 07:29 15/474 Project X

& **folder**

"/usr/spool/mail/root": 7 messages 2 new 4 unread 2 saved

& **!mkdir mailfolders**

!

& **set folder=mailfolders**

& **s 2 3 mailfolders/budgets**

"mailfolders/budgets" [New file] 10113/629759

& **s 1 mailfolders/projectX**

"mailfolders/projectX" [New file] 15/474

& **folders**

budgets

projectX

Reading Folders

& **folders**

budgets
projectX

& **fold +budgets**

"/usr/spool/mail/chris" complete
"+budgets": 2 messages 2 new

& **h**

2 andy Thu Jan 15 22:43 5292/327588 Support Budget
> 1 peter Thu Jan 15 22:43 5292/327591 Training Budget

& **fold /usr/spool/mail/chris**

"/usr/chris/mailfolders/budgets" complete
"/usr/spool/mail/chris": 7 messages 4 unread

Encoded Mail

^A^A^A^A

From andy Thu Jan 15 22:43:35 1998

From: andy@ulee.com

XMailer: SCO OpenServer Mail Release 5.0

To: chris@ulee.com

Subject: Support Budget

Date: Thu, 15 Jan 98 22:43:30 CST

MessageID: <9801152243.aa17556@oslep.ulee.com>

Status: R0

Chris, the Support Budget for 4Q1997 is enclosed below. Andy

begin 600 budget.xls

M!@ 3T" ' 30(< %= @ " &T" (!A0(D "5

M @ "0 :T" * 00(H ') @ "P D" + !Y0(H

.

.

.

M %5# !X ' =FET86PY > ! %#3P > !@ &1A=&\$U

M !0 B+'@ 0 !43U0 T,\1X*&Q&N\$

/@ # +

end

^A^A^A^A

Responding to Mail

U	3	rob	Fri Jan 16 09:55	42/593	Pure Poetry
U	2	mark	Thu Jan 15 07:30	13/318	Conference
	1	susan	Wed Jan 14 12:29	15/474	Project X

& R 2

To: mark

Subject: Re: Conference

Cc: shirley

I will be able to help with the conference. Just let me know what you need.

John

& r 1

To: susan

Subject: Re: Project X

The numbers for this project ...

& f 3 poets

Interpolating: 3

Message 3 forwarded to poets.

mail Settings

askcc	Prompts for the carbon copy list
autoprint	Automatically prints messages after d and u commands
chron	Displays messages in chronological order
hold	Keeps all messages in <i>mailbox</i> after reading
header	Prints the header summary when entering mail
record=filename	Records all outgoing messages in <i>filename</i>
save	Saves interrupted messages in the file <i>dead.letter</i> (default)
sign=string	Sets the variable inserted into a message when the ~a (autograph) command is used
outfolder	Records outgoing messages in files located in the directory specified by the folder variable

The *.mailrc* File

```
$ more $HOME/.mailrc
```

```
set askcc  
set asksub  
set folder=$HOME/mailfolders  
set header  
set keep  
set mchron  
set save
```

```
.mailrc: END
```

Transferring Files Between Systems

- Format floppy disks
- Backup files to UNIX-format floppy disks
- Backup files to DOS-format floppy disks
- Determine the type of archive on a floppy disk
- Restore archived files to the hard disk
- List the names of systems on the network
- Transfer files between systems using network commands

Creating tar Backups

```
$ lf
```

```
.kshrc          .profile      ex  
.lastlogin      .sh_history   mail/
```

```
$ tar cv .
```

```
Volume ends at 1439K, blocking factor = 9K
```

```
seek = 1K      a ./profile 2K  
seek = 3K      a ./kshrc 1K  
seek = 4K      a ./lastlogin 0K  
seek = 5K      a ./sh_history 1K  
seek = 6K      a ./ex 320K  
seek = 327K    a ./mail/message1 1K  
seek = 328K    a ./mail/message2 1K  
seek = 329K    a ./mail/message3 1K  
seek = 330K    a ./mail/message4 1K  
seek = 331K    a ./mail/message5 1K  
seek = 332K    a ./mail/message6 1K  
seek = 334K    a ./mail/message7 1K
```

```
$
```

tar Devices

\$ **tar**

Usage: tar {txruc}[09vfbkelmnopwAFLTP] [tapefile] [blocksize]
[tapesize] files...

Key	Device	Block	Size (K)	Tape
0	/dev/rfd048ds9	18	360	No
1	/dev/rfd148ds9	18	360	No
2	/dev/rfd096ds15	10	1200	No
3	/dev/rfd196ds15	10	1200	No
4	/dev/rfd0135ds9	18	720	No
5	/dev/rfd1135ds9	18	720	No
6	/dev/rfd0135ds18	18	1440	No
7	/dev/rfd1135ds18	18	1440	No
8	/dev/rct0	20	0	Yes
9	/dev/rctmini	20	0	Yes
10	/dev/rdisk/fp03d	18	720	No
11	/dev/rdisk/fp03h	18	1440	No

Extracting tar Backups

```
$ mkdir restore
$ cd restore
$ tar xv2
x andybud, 1037 bytes, 2K
x en, 114 bytes, 1K
x fold, 719 bytes, 1K
x list, 440 bytes, 1K
x mail.reply, 462 bytes, 1K
x mailfolders/budgets, 655179 bytes, 640K
x mailhe, 913 bytes, 1K
x moreen, 134 bytes, 1K
x sendm, 204 bytes, 1K
x set, 127 bytes, 1K
x tar, 547 bytes, 1K
x tar.list, 838 bytes, 1K
x test, 515 bytes, 1K
x tilde, 482 bytes, 1K
x what_am_i, 352 bytes, 1K
x what_data_out, 352 bytes, 1K
$
```

Creating **cpio** Backups

```
$ find . depth print | cpio -ocv > /dev/rfd0  
:saved  
root  
lmxadmin  
chris  
peter  
andy  
su  
.  
257 blocks
```

Extracting **cpio** Archives

```
$ cpio -icv < /dev/fd0
```

```
.
```

```
fruit
```

```
file.dos
```

```
better_town
```

```
where_am_i
```

```
5 blocks
```

```
$ lf
```

```
better_town
```

```
file.dos
```

```
fruit
```

```
where_am_i@
```

Connecting with **ftp**

```
$ ftp support
```

```
Connected to support.
```

```
220
```

```
220 support.acme.com FTP server (Version 2.1WU(1)) ready.
```

```
Name (support:andy): andy
```

```
331 Password required for andy.
```

```
Password:
```

```
230 User andy logged in.
```

```
Remote system type is UNIX.
```

```
Using binary mode to transfer files.
```

```
ftp>
```

ftp Commands

get <i>remote-file</i> [<i>local-file</i>]	Retrieves a copy of a file from the remote host
put <i>local-file</i> [<i>remote-file</i>]	Puts a copy of a local file on the remote host
mput <i>local-files</i>	Same as put , but allows use of wildcards
mget <i>remote-files</i>	Same as get , but allows use of wildcards
cd <i>remote-directory</i>	Changes the working directory on the remote host
lcd [<i>local-directory</i>]	Changes the working directory on the local system
ls [<i>remote-directory</i>]	Displays the contents of a remote directory
mkdir <i>remote-directory</i>	Creates a new directory on the remote host
delete <i>remote-file</i>	Deletes a file on the remote host
binary	Sets the transfer type to support binary (data) files
! [<i>command</i>]	Executes a command on the local system
? [<i>command</i>]	Displays a help screen

Using telnet

```
$ telnet ftpsrvr
```

```
Trying 130.128.209.xxx...
```

```
Connected to ftpsrvr
```

```
Escape character is '^]'.
```

```
SCO OpenServer(TM) Release 5 (ftpsrvr.euro.com) (ttyp5)
```

```
login: andy
```

```
Password:
```

```
Last successful login for andy: Thu Mar 15 22:28:59 1998 on  
tty06
```

```
Last unsuccessful login for andy: NEVER
```

```
SCO OpenServer(TM) Release 5
```

```
you have mail
```

```
TERM = (ansi) 
```

```
Terminal type is ansi
```

```
$
```

telnet Command Prompt

```
$ hostname  
ftpsrvr.euro.com
```

```
$  ]  
telnet>!hostname  
mysystem   
$
```

```
telnet>!  
$ uname -a  
SCO_SV mysystem 3.2 2 i386  
$  d 
```

```
$ hostname  
ftpsrvr.euro.com
```

AZ71003P001