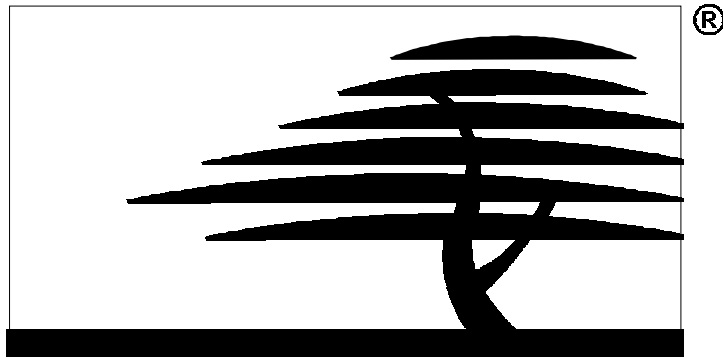




SCO

**UnixWare[®] 7 System
Administration II:
Installation,
Configuration, and
Maintenance**

Student Guide

Course Release: 2.0

For Software: UnixWare 7

© 1987–1999 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 Tcl, 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 SCO[®] Authorized Education class you are about to take may form part of the recommended preparation for those seeking the SCO ACE[™] certification.

The SCO ACE certification program uses a series of exams to measure your ability to support and administer SCO installations with the SCO recommended skills that maximize system performance. As a certified SCO ACE graduate, you earn all of the respect that the industry accords to each certified member of the extended SCO support team. When professionals in the computer industry see your SCO ACE certification, they know that you have passed demanding SCO testing requirements. This certification is the foundation for mastering real-world UNIX[®] systems, including those from other vendors.

Important!

When registering for any SCO ACE exam at a Sylvan Prometric[™] testing center, please quote the course release number on the front of the ***Student Guide*** that the exam covers. This number corresponds to the version number of the SCO ACE exam for that course.

Ask your SCO Authorized Instructor, or call SCO for more information about SCO Certification.

Table of Contents

Lesson 1: Course Introduction	1-1
Introduction	1-3
Course goal and purpose	1-3
Target audience	1-4
Student Materials	1-5
The <i>Student Guide</i>	1-5
Typographical Conventions.....	1-6
Sample Exam Items	1-8
 Lesson 2: System Installation	 2-1
Identifying UnixWare Hardware Requirements	2-4
Disk space requirements	2-5
Verifying supported hardware	2-6
Preparing the System for UnixWare 7 Installation	2-8
Running DOS hardware configuration programs	2-8
Identifying UnixWare 7 Products	2-9
UnixWare 7 versions	2-9
UnixWare 7 media	2-9
Printed documentation	2-11
Additional documentation and support resources.....	2-12
UnixWare 7 Editions	2-13
Licensed software	2-15
Unlicensed software.....	2-16
Optional Services	2-16
Completing Installation Checklists	2-21
General Information Checklist.....	2-22
Installation language	2-22
Zone, locale, and keyboard	2-22
License number, code, and data	2-23
Host Bus Adapter (HBA) diskette(s)	2-24
Manual hardware configuration	2-24
System node name	2-24
Installation type.....	2-25
Use whole disk for installation?.....	2-25
Configuring filesystems on active partition	2-27
Describing filesystems	2-30
Filesystem types.....	2-31
Choose default disk options or customize	2-32
System profiles.....	2-32
System services	2-33
System packages	2-33

Date, time, and timezone	2-33
Security profiles	2-34
System owner name	2-34
System root password	2-34
Save installation responses.....	2-35
Mouse type.....	2-35
Hardware Checklist.....	2-36
Entering the DCU	2-39
Hardware device configuration	2-41
Software device drivers configuration	2-45
Networking Checklist	2-48
Network adapter.....	2-48
Configuring TCP/IP networking.....	2-48
Configuring IPX networking	2-49
NIS configuration.....	2-49
Partition and Filesystems Checklist.....	2-51
Installing other partitions	2-51
Configure slices and filesystems.....	2-52
Installing UnixWare 7	2-55
Responding to prompts	2-55
Lesson Summary.....	2-56
Lab: Installation	2-58
Relevant Commands	2-64
Answers to Review Questions	2-65

Lesson 3: Installation Troubleshooting	3-1
---	------------

Troubleshooting Problems During Installation.....	3-4
System does not initially boot.....	3-5
System hangs or resets during boot	3-5
System cannot find the hard disk	3-5
System cannot find the CD-ROM.....	3-6
Resolving other installation problems	3-6
Troubleshooting Problems After Installation.....	3-7
Mouse not responding.....	3-7
Configuring your video.....	3-9
Lesson Summary	3-16
Relevant Commands	3-17
Answers to Review Questions	3-18

Lesson 4: Post-installation Administration	4-1
---	------------

Creating Emergency Recovery Media	4-4
Requirements for creating emergency recovery diskettes.....	4-4
Creating emergency recovery diskettes	4-5
Creating emergency recovery tapes	4-6
Managing Software Applications	4-9

Software management terminology	4-11
Package instances.....	4-11
The installation software database	4-12
Using the scoadmin (1M) Application Installer.....	4-15
Viewing properties and contents	4-16
Installing packages.....	4-17
Spooling packages	4-18
Package formats	4-18
Removing packages	4-19
Licensing and Registering UnixWare Products	4-20
Licensing the software	4-20
Registering the software	4-23
The registration process	4-23
Copy Protection Policy	4-25
Copy Protection Policy daemons	4-26
Linux Compatibility	4-27
Linux versions.....	4-27
Installing Linux applications	4-27
Running Linux applications.....	4-28
Creating the Documentation Search Indexes	4-29
Lesson Summary.....	4-30
Lab: Installing UnixWare 7 Software Packages.....	4-31
Relevant Commands and Files.....	4-32
Answers to Review Questions	4-33

Lesson 5: Managing System Startup and Shutdown

5-1

Describing the Default Boot Process	5-4
Standard boot procedure	5-4
Initiating the Boot Command Processor (BCP).....	5-6
Entering single-user state.....	5-6
Loading a kernel other than the default kernel	5-9
Booting other operating systems.....	5-9
Describing boot parameters	5-10
Describing Early Initialization.....	5-14
Run-levels	5-14
The <i>/etc/inittab</i> file.....	5-17
Entering the default run-level	5-19
Changing system run-levels	5-20
Changing the <i>/etc/inittab</i> file	5-23
Initializing the Multiuser and Networking States	5-24
Multiuser state (run-level 2).....	5-24
The networking state (run-level 3).....	5-27
Customizing Multiuser Initialization	5-29
How scripts are executed from <i>/etc/rc2.d</i>	5-29
Executing commands at system startup	5-29

Defining other run-level initialization scripts	5-30
Deactivating rc scripts.....	5-30
Troubleshooting System Startup.....	5-31
Shutting Down the System.....	5-33
Analyzing shutdown	5-33
Lesson Summary.....	5-34
Lab: Creating an <i>/etc/rc2.d</i> Script.....	5-35
Relevant Commands and Files.....	5-36
Answers to Review Questions	5-37

Lesson 6: Filesystem Configuration and Disk Management	6-1
---	------------

Describing the UNIX Filesystem Structure	6-5
Data blocks	6-5
Inodes.....	6-5
Superblock	6-6
Describing the Directory Hierarchy.....	6-9
Key directories in the <i>/usr</i> directory structure	6-13
Key directories in the <i>/etc</i> directory structure	6-13
Managing the <i>/dev</i> Directory	6-17
Block device special files.....	6-17
Character device special files.....	6-18
Raw interfaces.....	6-18
Major and minor device numbers	6-21
Creating special device files	6-22
The Device Database	6-23
Alternate names for special device files	6-24
Troubleshooting the <i>/dev</i> directory	6-24
Describing Important vxfs Filesystem Features	6-25
Large file support	6-25
Extent-based allocation.....	6-25
Logical block size	6-26
Quotas	6-26
Enhanced data integrity.....	6-27
Fast filesystem recovery.....	6-27
Describing VTOC	6-29
The fdisk (1M) command.....	6-29
The disksetup (1M) command.....	6-29
Preparing the Filesystem for Access.....	6-31
The <i>/etc/vfstab</i> file.....	6-33
Adding Large file Support	6-35
Checking and Repairing Filesystems.....	6-36
Specifying a filesystem type	6-36
Full vs. fast checking	6-36
Running fsck (1M).....	6-38
Error messages	6-39

Running ncheck (1M)	6-40
Configuring Additional Disks and Filesystems	6-41
Possible configurations for secondary hard disks	6-41
The Storage Device Interface (SDI).....	6-41
Hard disk naming conventions.....	6-43
Multi-Path Input/Output (MPIO).....	6-43
Configuring IDE disks and disk controllers.....	6-44
Using diskadd (M) to configure disks.....	6-45
Adding a hard disk	6-45
Managing the Volume Table of Contents (VTOC)	6-48
Troubleshooting Hard Disks	6-50
Mapping of bad blocks	6-50
Lesson Summary	6-51
Lab: Configuring a Second Hard Disk.....	6-53
Relevant Commands and Files.....	6-55
Answers to Review Questions	6-56

Lesson 7: Online Data Manager	7-1
--------------------------------------	------------

UnixWare 7 Online Data Manager (ODM).....	7-4
VERITAS Filesystem (VxFS).....	7-4
Volume Manager (VxVM).....	7-5
Visual Administrator (VxVA)	7-17
Initializing the VxVM.....	7-18
Checking VxVM processes	7-26
Starting up VxVA.....	7-27
Performing VxVA basic operations	7-27
Creating a concatenated volume	7-27
Lesson Summary	7-32
Relevant Commands and Files.....	7-33
Answers to Review Questions	7-34

Lesson 8: Installing and Managing Serial Ports	8-1
---	------------

Identifying Standard Serial Ports	8-5
Serial port configuration	8-7
Flow control	8-7
Asynchronous data communications	8-7
UARTS.....	8-9
Serial device naming conventions.....	8-9
Installing a Serial Card	8-13
Serial adapter cards	8-13
Installing the hardware.....	8-14
Using Multiscreens	8-15
Describing the Service Access Facility (SAF).....	8-16
The Service Access Controller (SAC)	8-17
The ttymon (1M) port monitor.....	8-19

Configuring Serial Ports	8-21
Installing Serial Terminals	8-25
About serial terminal hardware.....	8-25
Ports and cables	8-25
Terminal settings.....	8-29
Managing Logins	8-31
Logging in.....	8-31
Setting the terminal type	8-31
Disabling or enabling logins	8-32
Troubleshooting Terminals	8-33
Troubleshooting non-functional terminals.....	8-35
Troubleshooting hung or scrambled terminals.....	8-35
Lesson Summary	8-36
Relevant Commands and Files.....	8-37
Answers to Review Questions	8-38

Lesson 9: Configuring Modems	9-1
-------------------------------------	------------

Adding a Modem	9-4
Choosing a serial port	9-4
Modem support	9-4
Connecting the modem	9-4
Configuring the modem	9-6
Testing the modem connection	9-8
Lesson Summary	9-9
Relevant Commands and Files.....	9-10
Answers to Review Questions	9-11

Lesson 10: Installing and Managing Printers	10-1
--	-------------

Installing a Parallel Card.....	10-4
Installing Parallel and Serial Printers.....	10-6
Verifying the hardware installation of a parallel or serial printer	10-6
Setting up RTS/CTS protocol serial printers	10-9
Troubleshooting Printer Installations.....	10-11
Adding Printers	10-12
Adding a local printer	10-12
Adding a TCP/IP printer	10-13
Adding a NetWare printer	10-14
Describing the UNIX Print Service	10-15
Who can administer the print service?.....	10-15
Print scheduler functions	10-15
Print request process	10-17
Viewing the print scheduler	10-19
Managing the Print Request Directories	10-20
Setting Print Request Priorities.....	10-25
Default priority level and limits	10-25

Changing the default priority for all users	10-25
Setting a personal priority limit for a user	10-26
Restoring the default priority limit	10-26
Submitting a print job with a different priority	10-26
Changing the priority of a print job residing in the queue	10-27
Bypassing all other awaiting print requests	10-27
Managing Printer Interface Scripts	10-28
Assigning an interface script.....	10-28
Customizing the standard interface script.....	10-29
Troubleshooting Common Printer Problems	10-31
Printer does not print.....	10-31
Restarting a printer.....	10-31
Laser printers and form feeds	10-32
Testing printers in PostScript mode	10-32
Determining if the print scheduler is active	10-32
Redirecting output.....	10-33
Printer output spacing is wrong	10-34
Lesson Summary.....	10-35
Review questions	10-35
Lab: Printer Troubleshooting.....	10-37
Relevant Commands and Files.....	10-38
Answers to Review Questions	10-40

Lesson 11: Installing Tape Drives and CD-ROM Drives	11-1
--	-------------

Describing Supported Tape Devices	11-4
Adding a Tape Drive	11-5
Adding a SCSI tape drive.....	11-6
Adding a non-SCSI tape drive	11-6
Adding a CD-ROM Drive	11-7
Lesson Summary	11-9
Relevant Commands and Files.....	11-10
Answers to Review Questions	11-11

Lesson 12: Controlling User and System Processes	12-1
---	-------------

Creating Processes	12-5
Distinguishing Types of Processes.....	12-6
Light Weight Processes (LWP)	12-6
System processes	12-9
User Processes	12-10
Checking Process Activity	12-11
The ps (1) command.....	12-11
The sar (1M) command.....	12-17
The timex (1M) command.....	12-20
Managing Process Scheduling.....	12-21
Scheduling classes	12-21

Controlling the nice value.....	12-22
Managing Process Accounting	12-23
Enabling process accounting	12-23
Generating Reports	12-24
Disabling process accounting	12-25
Lesson Summary.....	12-27
Relevant Commands and Files.....	12-28
Relevant commands	12-28
Relevant files.....	12-28
Answers to Review Questions	12-29

Lesson 13: Managing System Security

13-1

Describing Trusted System Concepts	13-4
Describing a Security Profile	13-7
The high security profile	13-7
The improved security profile.....	13-7
The traditional security profile.....	13-9
The low security profile	13-9
Viewing and Changing your System Security Profile.....	13-10
Determining your security profile.....	13-10
Selecting a security profile.....	13-10
Changing a parameter in your security profile	13-12
Using the secdefs(1M) command	13-12
Reviewing the Account Database Files	13-14
System V files	13-14
Trusted facility database files.....	13-15
Delegating System Administration Responsibility	13-16
The system owner account.....	13-16
Limited system owner authorizations	13-16
Allowing users to execute superuser commands	13-19
Adding an administrative role.....	13-19
Adding commands to the role	13-20
Assigning the user the administrative role.....	13-20
Executing the tfadmin(1M) command	13-20
Updating the file privilege database.....	13-21
Managing Access Control Lists (ACLs).....	13-25
ACL entries for individual files	13-27
Default ACLs	13-27
Deleting ACLs	13-27
Advanced Protection Bits	13-29
Examining SUID/SGID bits	13-31
Setting the sticky bit.....	13-31
Restricting root logins.....	13-32
Monitoring the use of the su(1M) command	13-32
Lesson Summary	13-34

Relevant Commands and Files.....	13-36
Answers to Review Questions	13-38
Lesson 14: Performance Tuning	14-1
Influencing System Performance.....	14-4
Hardware factors	14-4
Software factors	14-5
System organization.....	14-5
Setting Performance Goals	14-6
Defining your goals.....	14-6
Listing your goals	14-6
Understanding your goals	14-6
Collecting Performance Data.....	14-7
Identifying CPU saturation	14-8
Tuning CPU resources	14-9
Identifying I/O saturation.....	14-10
Describing the buffer cache	14-10
Tuning I/O performance.....	14-13
Identifying memory saturation.....	14-14
Tuning memory saturation	14-16
Running the System Monitor.....	14-19
Using the System Monitor	14-21
Configuring the System Monitor	14-22
Managing log data	14-22
Setting alarms	14-23
Setting options	14-23
Autotuning Parameters.....	14-25
Interpreting Kernel Error Messages.....	14-26
Changing Kernel Parameters	14-27
The System Tuner	14-27
The id tune (1M) command	14-28
Lesson Summary.....	14-30
Lab: Reconfiguring the kernel	14-32
Relevant Commands and Files.....	14-33
Answers to Review Questions	14-34
Lesson 15: Disaster Recovery	15-1
Preparing for Recovery	15-4
Keys to recovering your system.....	15-4
Recovery components.....	15-4
Running fsck (1M).....	15-5
Recreating a non-root filesystem	15-5
Restoring a non-root filesystem	15-6
Recovering an Unusable System.....	15-8
Recovery components	15-8

Performing the recovery.....	15-8
Repairing the file privilege database.....	15-9
Lesson Summary.....	15-10
Lab: Recovering an unusable system.....	15-11
Relevant Commands	15-12
Answers to Review Questions	15-13

Appendix A: Basic Hardware Configuration	A-1
---	------------

Hardware Overview	A-5
Microprocessors and Memory	A-5
Microprocessors.....	A-9
The Pentium Pro	A-10
Memory.....	A-15
Bus Basics.....	A-19
Bus architectures.....	A-19
Peripheral Component Interconnect (PCI)	A-22
Accelerated Graphics Port (AGP).....	A-23
Adapter cards and controllers	A-25
Supported Hard Disk Configurations.....	A-31
Small Computer System Interface (SCSI) Host Adapters	A-31
SCSI addresses.....	A-35
General compatibility issues	A-35
General configuration parameters.....	A-36
Plug and Play (PnP)	A-37
Clustering.....	A-39
Maintaining a Sample System Administration Log Book	A-40
Lesson Summary.....	A-42
Relevant Commands	A-44
Answers to Review Questions	A-45

1

Course Introduction

Lesson 1

Course Introduction

1.1 Introduction

The ***UnixWare 7® Administration II: System Installation, Configuration, and Maintenance*** course is designed for technically-oriented individuals who already have a solid understanding of UNIX® system administration basics and are required to install and configure UnixWare 7 systems, as well as maintain and troubleshoot these systems on a daily basis.

1.1.1 Course goal and purpose

At the conclusion of this course, students should be able to perform the tasks and duties associated with installing, configuring, and maintaining the system hardware and software. By the end of this course, students will be able to:

- Describe the basic hardware configuration of a system
- Describe files and filesystems
- Plan for a successful operating system installation
- Install the UnixWare 7 operating system and troubleshoot the installation
- Perform post-installation tasks that include licensing their software and creating emergency recovery media
- Manage system startup and shutdown
- Add hard disks and configure filesystems
- Install and manage serial and parallel ports
- Install and manage printers and print services
- Install tape drives and CD-ROM drives

- Control user and system processes
- Manage system security
- Tune the system for optimum performance
- Perform disaster recovery

1.1.2 Target audience

The target audience includes support technicians, developers, engineers, system administrators, and other technical users who require an advanced level of understanding for installing, configuring, and maintaining a UnixWare 7 system. Students are likely to have a wide variety of backgrounds in XENIX® or UNIX, but must have prerequisite skills equal to or exceeding those provided in the ***UnixWare 7 Administration I: User Services*** course.

1.2 Student Materials

Your student guide is yours to keep and write notes in. Your instructor may have other reference materials in the classroom for your use while attending this class. Please do not take or write in these materials as they will be needed again for later classes.

1.2.1 The *Student Guide*

The *Student Guide* contains:

- Outline of presentation
- Note-taking space

In addition, each student should have access to copies of the following hard copy documents:

- *UnixWare 7 Getting Started Guide and Release Notes*
- *UnixWare 7 Handbook*
- *Sample System Administration Log Book*

1.3 Typographical Conventions

This publication presents commands, filenames, keystrokes, and other special elements as shown here:

lp or lp(1)	Commands, device drivers, programs, and utilities (names, icons, or windows); the letter in parentheses indicates the reference manual section in which the command, driver, program, or utility is documented
<i>/new/client.list</i>	Files, directories, and desktops (names, icons, or windows)
root	System or user names
<i>arg</i> or <i>arg</i>	Place holders (replace with appropriate name or value)
[<i>arg</i>]	Surrounds an <i>option</i> or <i>cmdarg</i> that is not mandatory
<i>arg</i> ...	Indicates multiple occurrences of the <i>option</i> or <i>cmdarg</i>
\$	Command line prompt for commands executable from the Bourne shell by any user
#	Command line prompt for commands only executable by the superuser from the root account
	Keyboard keys
 	Press  and  keys simultaneously
  	Press all three keys,  ,  , and  , simultaneously
 d	Press  key. With  still depressed, also press  .
 0240	Press  key. With  still depressed, also press and release in sequence  ,  ,  , and  .
 Q + i	Press  key. With  still depressed, also press  . Release both keys and press  .
 Q + I	Press  key. With  still depressed, also press  . Release both keys and press   .
Exit program?	System output (prompts, messages)

tar tv	User input, such as on a command line
“Description”	Field names or column headings (on screen or in database)
Cancel	Button names
Edit	Menu names
Copy	Menu items
File ⇨ Find ⇨ Text	Sequences of menus and menu items
open or open(S)	Library routines, system calls, kernel functions, C keywords; the letter in parentheses indicates the reference manual section in which the file is documented
\$HOME	Environment or shell variables
SIGHUP	Named constants, signals, and kernel parameters
“adm3a”	Data values
<i>employees</i>	Database names and tables
buf	C program structures
b_b.errno	Structure members

1.4 Sample Exam Items

These questions are representative exam items on the SCO ACE certification exam for this course. Being able to correctly answer these questions does not guarantee that you will be able to pass the actual exam.

1. What is the maximum vxfs filesystem size?

- A. 2GB
- B. 256GB
- C. 512GB
- D. 1 Terabyte

Answer: D

2. Which security profile meets only the minimum requirements for the C2 level of trust?

- A. High
- B. Improved
- C. Traditional
- D. Low

Answer: B

3. How much free space must be remaining in the filesystem used to create the emergency recovery media?

- A. 5 MB
- B. 30 MB
- C. 50 MB
- D. 80MB

Answer B.

2

System Installation

AV 2-1

The slide is titled "Installation Planning" in a blue-bordered box at the top. Below the title, there is a bulleted list of four items. At the bottom of the slide, there is a blue footer bar containing the text "sa2v-0201" on the left and "© The Santa Cruz Operation, Inc." on the right.

Installation Planning

- Identify UnixWare 7 hardware requirements
- Identify UnixWare 7 products
- Complete installation checklists
- Install UnixWare 7

sa2v-0201 © The Santa Cruz Operation, Inc.

Lesson 2

System Installation

Introduction

The successful installation of system hardware and software is a critically important aspect of your job as a system administrator. The first step to a successful installation is a pre-installation analysis.

When you complete a pre-installation analysis, you have an overall picture that describes the amount and kind of work that the system will be expected to perform. In addition, you will have a worksheet or a checklist that will make the actual installation faster and easier. The questions you will need to answer during installation will be decided during the analysis phase, so you do not have to interrupt the installation when you are prompted for that information during the actual installation process.

During the installation, you will use the checklist you completed to respond to the installation prompts.

Objectives

This lesson describes the things you need to do before you install your operating system software. By the end of this lesson, you will be able to:

- Identify UnixWare 7 hardware requirements
- Identify UnixWare 7 products
- Complete installation checklists
- Install UnixWare 7

2.1 Identifying UnixWare Hardware Requirements

SCO recommends that your system meets or exceeds these requirements:

Processor	An Intel Pentium, Intel Pentium II or Intel Pentium Pro microprocessor or a 100% compatible is recommended. The Intel 486DX processor is the minimum. Intel 486SX systems are not supported.
Architecture	Conformance to the Industry Standard Architecture (ISA), (also called an AT bus), Extended Industry Standard Architecture (EISA), Microchannel Architecture (MCA), Peripheral Component Interface (PCI), or Intelligent Input/Output (I ² O) standards.
Memory	A minimum of 32MB of Random Access Memory (RAM) is required. For improved CDE desktop performance, use of ARCserve[®] /IT[™] v6.6 from Computer Associates, Java Workshop, Java Studio, and the Webtop a minimum of 64MB RAM is required. Performance is enhanced as more RAM is added. RAM requirements vary depending on the type of applications you are running, number of users, and the expected system load. The figures above are minimum recommendations. Most production systems will run optimally with RAM configured above the minimum requirements. Small Footprint installations can run on systems with only 16MB of RAM installed. All other system profiles require a minimum of 32MB of RAM. UnixWare 7 supports up to 8GB of RAM for general purpose usage and up to 64GB for use by the dynamic shared memory (DSHM) and fine-grained affinity shared memory (FGA-SHM) subsystems, which are licensed separately.
Mouse	A serial, bus, or PS/2 compatible mouse is recommended to access all graphical desktop functionality.
Video	A Super VGA monitor with a minimum resolution of 800x600 is required to run the graphical desktop.
Media devices	A 3.5-inch diskette drive for booting UnixWare 7 and installing additional software is required.

The system must also have one of the following to install UnixWare 7:

- CD-ROM drive, or
- Supported network adapter, and a preconfigured network installation server

2.1.1 Disk space requirements

Disk space requirements vary depending on the software you choose to install. If you are installing any of the development kits or other applications that requires root filesystem space, you must increase the root filesystem size by the required amount, plus any resulting increases in swap space.

The swap area is a specific section of the hard disk used by the kernel for memory management. It is not a filesystem, but simply formatted space. The special device file for the swap area is */dev/swap*.

The size of the swap area depends on how much memory is installed in your system.

The minimum disk space requirements are:

Disk size	A hard disk of 1GB or larger is recommended. Small footprint installations can run on systems with 300MB or larger UNIX partitions. Other installation types can run on a UNIX partition of 500MB or larger. The UNIX partition must begin below cylinder 1024 of the hard disk. In order to run large Java programs, like Java Workshop or Java Studio (on the UnixWare and OpenServer Development Kit CD-ROM), you need 350MB of swap space on disk. Please change the default size of the swap partition during installation to that value.
-----------	--

UNIX partition	Must have a UNIX partition of at least 500MB to install the Small Footprint Server profile. A 1.2GB UNIX partition is recommended for all other system profiles.
----------------	--

The active UNIX partition must start below cylinder 1024 of the hard disk.

2.1.2 Verifying supported hardware

Your UnixWare 7 installation media includes Host Bus Adapters (HBAs) drivers for supported hardware. See the **UnixWare 7 Getting Started Guide** and **Release Notes**, the `/info/hardware/` directories on the UnixWare 7 Release 7.1.0 Operating System Updates CD-ROM, and the SCO Compatible Hardware Web Pages at: <http://www.sco.com/chwp> for help on determining if your hardware is supported.

The Host Bus Adapter (HBA) drivers:

- Are contained on the installation diskettes provided with UnixWare 7
- Can be provided on floppy diskettes provided by your hardware or software vendor
- Are loaded during the installation

Note that any vendor-supplied disk must be created specifically for UnixWare 7 systems; otherwise, the driver will not successfully load.

The SCO Compatible Hardware Web Pages define hardware that:

- Has been tested by SCO, or
- Is reported by the vendor to work; not tested by SCO

It also defines software drivers that:

- Are included with a product
- Are not part of a product's base distribution, but are provided by SCO
- Are not part of a product's base distribution, but are provided by a vendor

You can also view brands of computers and peripherals that are supported by SCO, including:

- Computer systems
- Host adapters
- Network adapters
- Video adapters
- Tape drives

- CD-ROM/Optical drives
- Additional storage technologies

2.2 Preparing the System for UnixWare 7 Installation

After determining that your hardware is compatible with UnixWare 7, you may need to perform the following:

- Review the manuals provided with your computer and hard disk
- Run any system setup programs provided by the computer manufacturer
- Configure your hardware controllers verifying that they are set to the correct:
 - I/O base address
 - DMA channel
 - Interrupt level
- Run the system self-test described in your computer's system hardware manual to detect any hardware problems

When performing these hardware tests, it may be necessary to boot the vendor-supplied DOS operating system from a diskette.

2.2.1 Running DOS hardware configuration programs

Many platform configuration utilities and/or hardware initialization programs require the DOS operating system. If so, use your own DOS diskettes (or ones provided by your hardware vendor) to reboot your system and configure the hardware.

2.3 Identifying UnixWare 7 Products

The next sections describe the UnixWare 7 versions, media, editions, and licensing.

2.3.1 UnixWare 7 versions

There are three major release versions of the UnixWare 7 operating system:

- UnixWare7 Release 7.0.0 first customer ship (FCS) March 1998
- UnixWare 7 Release 7.0.1 FCS September 1998
- UnixWare 7 Release 7.1 FCS March 1999

Reference **scohelp** ⇒ **UnixWare 7 Features** for a detailed description of the new features for each version of the system.

Once your system is installed, you can check the version of your operating system by using the **uname(1)** command. Use the following command to check the version of your UnixWare 7 operating system:

uname -v

SCO engineering is focused on continuous improvements and new releases for UnixWare 7. Check the UnixWare 7 web page for the latest release and improvement information at <http://www.sco.com/unixware/>.

2.3.1.1 UnixWare 7.1 upgrade licensing

If you are upgrading from a Release 7.0 or Release 7.0.1 system, a new Release 7.1 Upgrade License is required. The Upgrade License must be purchased before you can upgrade from any previous release to Release 7.1.

Licenses for the Business Edition and the Data Center Edition can only be used with the Release 7.1 Media Kit.

2.3.2 UnixWare 7 media

The UnixWare 7 media kit includes both the UnixWare 7 operating system and many supporting products that are used with UnixWare 7.

The UnixWare 7 Media Kit contains the following CD-ROMs:

- UnixWare 7 Release 7.1.0 Installation
- UnixWare 7 Release 7.1.0 Operating System Updates
- UnixWare 7 Release 7.1.0 Optional Services
- UnixWare and OpenServer Development Kit
- SCO Vision2K™
- SCO Skunkware™ 7

For a list of the products and services on each CD-ROM refer to the UnixWare 7, Release 7.1.0 Installation Roadmap and the */info/readme.htm* or the */info/readme.txt* files on the Installation and Operating System Updates CD-ROMs.

With UnixWare 7.1 release, SCO® Tarantella™ is no longer included on the Optional Services CD-ROM. The latest release of SCO Tarantella for UnixWare 7 is available from SCO's Tarantella Division. Visit <http://tarantella.com.com/> for more information on SCO's Tarantella Technology.

With UnixWare 7.1, because Java WorkShop and Java Studio are designated as Open Source products by Sun Microsystems and are freely available on the Web directly from Sun Microsystems, they are no longer included on the Optional Services CD-ROM.

For a brief description of each of the products included on the UnixWare CD-ROMs see Section 2.3.8, "Optional Services," on page 2-16.

Your media kit should also include the following diskettes:

- UnixWare 7 Host Bus Adapter Diskette
- UnixWare 7 Installation Diskettes (2)

2.3.2.1 Diskette images

The UnixWare 7 Release 7.1.0 Installation CD-ROM contains images of the diskettes used to install and troubleshoot UnixWare 7. These images are located in */info/images*. They include the following images:

boot.image.C.1	UnixWare 7.1.0 Installation Diskette 1 of 2
boot.image.C.2	UnixWare 7.1.0 Installation Diskette 2 of 2

hba.image	UnixWare 7.1.0 Host Bus Adapter Diskette
magic.image	Magic Diskette— This diskette is used for installation troubleshooting and is generally not required.
netinstall.image.1	Network Installation Diskette 1 of 2
netinstall.image.2	Network Installation Diskette 2 of 2

Refer to `/info/images/readme.txt` for instructions on how to create the diskettes from the images.

Check the SCO UnixWare 7.1 web site for information regarding future support for bootable CD-ROMs.

2.3.3 Printed documentation

The following printed documentation is included in the UnixWare 7 media kit:

- ***UnixWare 7, Release 7.1.0 Installation Roadmap***

This document provides a brief overview of the UnixWare 7 installation process, contents, and licensing of the software in your UnixWare 7 media kit.

- ***UnixWare 7 Getting Started Guide***

This document provides the information you need to get your UnixWare 7 system up and running. It includes installation and configuration information instructions, as well as chapters with the system ***Release Notes***. The ***Release Notes*** contain important late-breaking information about installation, hardware requirements, and known limitations.

You should read the ***UnixWare 7 Getting Started Guide*** and ***Release Notes*** thoroughly before beginning any installation.

- ***The UnixWare 7 System Handbook***

Describes common system administration and troubleshooting tasks. Use this handbook after your initial software installation. The contents of the handbook are also available in the online documentation.

2.3.4 Additional documentation and support resources

The UnixWare 7 installation CD-ROMs contain a */info* directory with:

- Lists of the products on each CD-ROM
- Installation notes
- Licensing information
- Product information

For a complete list of the contents of each */info/* directory view the */info/readme.htm* or */info/readme.txt* files on each CD-ROM.

If you are connected to the World Wide Web, you can access the:

- Latest Updates—Describes where to find documentation that may be more current than the documentation installed on your system. The URL is:

<http://uw7doc.sco.com/>

Select Late News

- Differences between UnixWare 7 and other SCO UNIX systems—Describes many of the differences between UnixWare 7 and other versions of SCO UnixWare and SCO OpenServer

<http://uw7doc.sco.com/>

Select **UnixWare7 Features** ⇔ **Differences for SCO OpenServer and SCO UnixWare 1 and 2 users**

- SCO Support Library—Contains a searchable list of solutions to common troubleshooting issues

<http://www.sco.com/support>

Select **Access to Technical Articles**

- SCO Home Page—Provides useful information on product offerings and updates

<http://www.sco.com>

2.3.5 UnixWare 7 Editions

The Edition determines which additional software products are fully licensed during the installation of UnixWare 7. It also determines the number of users, CPUs, and the amount of main memory that is licensed.

The UnixWare 7 Editions include:

Base Edition	The Base Edition is a lightweight, single-user UnixWare 7 configuration for workstation environments and for building dedicated or specialized servers. The Base Edition is ideal for smaller footprint systems and embedded systems and is licensed for a single user and 1GB of memory.
Business Edition	The Business Edition is designed to run applications that automate business processes and reliably share business critical information with any client. The Business Edition provides all the fundamental services used by businesses today. In addition to the base features of UnixWare 7 listed above, the Business Edition is licensed with: ● 5-User configuration (additional Network User Licenses available) ● Licensed for up to 4GB of general purpose memory ● Network backup and system restore with ARCserve /IT v6.6 from Computer Associates ● Windows file and print sharing across a UNIX server network using SCO VisionFS[®].
Departmental Edition	Designed specifically for medium-sized businesses and departments in larger organizations. The Departmental Edition is designed for larger user counts and more demanding application usage, including database applications. The Departmental Edition is configured similarly to the Business Edition with the following differences: ● 25-User configuration perfect for workgroups and departments (additional Network User Licenses available)

	● Licensed for 2 CPUs (additional CPU Licenses available)
Messaging Edition	Supports 1 user, 2 CPUs, and up to 4GB of general purpose memory. It also includes full networking functionality, ARCserve /IT v6.6 from Computer Associates, and the Netscape Mail Server™.
Enterprise Edition	A reliable, scalable, high performance, network-ready and secure operating environment to run enterprise-wide business applications and databases that manage, distribute and warehouse business critical information. The Enterprise Edition is configured similar to the Business Edition and adds enterprise class features such as: ● 50-User Configuration (additional Network User Licenses available) ● Licensed for 4 processors (additional CPU Licenses available) ● Licensed for up to 16GB of general purpose memory ● Volume management, software RAID, and disk performance analysis for enterprise-level data management with UnixWare 7 Online Data Manager Release 3.2.
Data Center Edition	The Data Center Edition is designed specifically as a high performance, robust system. In addition to features of the Business Edition, the Data Center Edition includes: ● 150-User Configuration for large organizations (additional Network User Licenses available) ● Licensed for 8 processors ● Licensed for up to 32GB of memory ● Volume management, software RAID and disk performance analysis for enterprise-level data management with UnixWare 7 Online Data Manager Release 3.2 ● New support for Event Logging to a reliable SQL database to manage and log operating system events, resulting in a more robust and manageable system

For a listing of current UnixWare 7 Editions and the software products included with each Edition, contact your SCO software provider or access the following URL:

http://www.sco.com/unixware/product_family

2.3.6 Licensed software

UnixWare 7 Edition License Packs (Base, Business, Departmental, Enterprise, Data Center, Messaging) can be used for fresh (new) installations of older Release 7.0 and Release 7.0.1 systems as well as for fresh installations of Release 7.1.

2.3.6.1 Network User Licenses

Network User Licenses did not change with Release 7.1. Licenses available prior to Release 7.1 can be used on Release 7.1 systems. There are no User License upgrades required and systems that are upgraded from 7.0 or 7.0.1 will automatically recognize existing User Licenses installed on the system. To check the number of users licensed on your system, use the following **uname(1)** command:

```
uname -l
```

2.3.6.2 CPU Licenses

Each UnixWare 7 Edition includes support for a number of CPUs that is designed to optimize the hardware platforms with which Edition is typically delivered. Additional CPUs can be licensed by purchasing the Processor Upgrade and can be used with any UnixWare 7 Edition. The Processor Upgrade License is for a single CPU and can be combined with the CPUs in the Edition License for a cumulative number of CPUs.

For example the Business Edition includes support for two CPUs. If a Processor Upgrade is added, the total number of supported CPUs is three. Users who need support for four CPUs should consider the purchase of a UnixWare 7 Enterprise Edition, which comes pre-configured with support for four CPUs.

Users may also purchase the Business or Departmental Editions and add two Processor Upgrade Licenses. There is no requirement to purchase CPU upgrades when upgrading from Release 7.0 or Release 7.0.1 to Release 7.1. Upgraded systems will automatically recognize existing CPU licenses installed on the system.

2.3.6.3 Memory Upgrade Licensing

Memory Upgrade licenses are designed to match the level of memory provided in a hardware platform. Memory Licenses are available for systems using 4, 16, 32 or 64GB of memory. Any UnixWare 7 Edition may upgrade memory to any level that is higher than that included in the Edition License configuration.

For example, a Base Edition includes support for 1GB of memory. The Base Edition may upgrade to 4, 16, 32 or 64GB of memory. Users with the Business Edition or Departmental Edition may wish to upgrade their memory to a 16, 32 or 64GB of memory. Enterprise Edition users may wish to upgrade to either 32GB or 64GB. Data Center Edition users may wish to upgrade to a 64GB memory license.

2.3.7 Unlicensed software

If you install the UnixWare 7 operating system without a license it will default to the Business System Edition for a 60-day evaluation. Evaluation versions are available for most other products in the Media Kit. To evaluate other UnixWare 7 products, simply install the product without entering a license number and license code. This will enable a 60-day evaluation license period. You can evaluate products that are not included with your license by selecting additional services and packages during the installation, or by installing them later using the **SCOadmin Application Installer**.

To continue using the software after the 60-day trial, you must purchase a license and license the software products using the **SCOadmin License Manager**.

2.3.8 Optional Services

Check the **UnixWare 7 Release 7.1.0 Installation Roadmap** to confirm which services are included with the license of your UnixWare 7 edition.

The optional services expand the capabilities of your UnixWare 7 system in the areas of reliability-availability-serviceability (RAS), interoperability, and application development. For more information on each of these products, reference the */info* directories on your UnixWare 7 CD-ROMs.

RAS options include:

- SCO® Online Data Manager

This is a cost-effective, enterprise-class storage management solution for high availability and online volume management. It provides software RAID Levels 0, 1, 5, 10 (striping, mirroring, striping with distributed parity and striped mirroring) as well as disk-spanning capabilities. It includes a graphical Visual Administrator for exceptional ease-of-use and supports online filesystem resizing, migration, and RAID-level modifications. The full product is included with the Enterprise and Data Center Editions. All other UnixWare 7 editions include the “base” functions of the volume manager.

- Disk Mirroring

UnixWare 7 Disk Mirroring provides increased data availability by providing fault tolerance against disk failures and faster access using software RAID Level 1 (simple disk mirroring). By adding a second disk to the system and enabling Disk Mirroring, administrators will automatically have access to their business-critical data should one disk fail. Designed to work with off-the-shelf disk storage subsystems, UnixWare 7 Disk Mirroring provides a very cost-effective high-availability data storage solution. Disk Mirroring is included with the SCO Online Data Manager license.

- Enhanced Event Logging System (Enhanced query-capable RDBMS licensed separately)

The Enhanced Event Logging System (EELS) provides an infrastructure to centralize the logging, management, and reporting of standard UNIX logging systems, such as syslog and the auditing sub-system. It does this by channeling the various log sources through a driver and filtering mechanism into a searchable flat filesystem or an SQL queryable Relational Database Management System (RDBMS). The Data Center editions includes the advanced RDBMS capabilities. All other editions include the base EELS system and can license the advanced capabilities separately.

- SCO Reliant® HA 1.1.0

The Reliant HA software for UnixWare 7 provides high-availability clustering solutions for enterprises that require high levels of application, system, and data availability. Using UnixWare 7 ReliantHA you can create highly available 2-to-4 nodes loosely coupled clusters ideal for transaction processing, database, Internet/intranet, replicated site and NFS services. Unlike expensive proprietary solutions, UnixWare 7 ReliantHA supports industry standard Intel processor servers with off-the-shelf RAID and networking solutions. Unlike some solutions that use redundant systems and peripherals, all the nodes in a cluster can be used.

Interoperability options include:

- IBM® Network Station™ Manager

The IBM Network Station is IBM's network computer. It is designed to be easy to install, use, and manage. It operates without any internal disk storage, using supporting software and applications from associated servers. The IBM Network Station Manager is developed to provide a common support environment for the IBM Network Station across all IBM and Intel PC-based server platforms.

- Novell® NetWare® Services

This optional component provides Novell NetWare 4.10a file, print, and directory services for both UnixWare 7 and SCO UnixWare 2.1. With NetWare Services a UnixWare application server can be easily and transparently accessed by NetWare clients, enabling seamless integration into existing Novell environments. Business critical environments can also leverage the reliability, performance, and scalability of the UnixWare 7 system to build a single, highly reliable primary network server for NetWare clients with LAN security, multiple client support, and robust application services. Also available in License Packs enabling 10-user, 25-user, 50-user, 100-user, and 250-user licenses. This service is optional to all UnixWare 7 editions except for the free and evaluation editions.

- Advanced File and Print Server 4.0.1

The SCO Advanced File and Print Server enables enterprise-wide, scalable file and printer sharing with PCs running Microsoft Windows 95, Windows NT, Windows 3.x, OS/2, and MS-DOS. This newest version for UnixWare 7 delivers a highly-integrated environment that allows PCs to seamlessly access files and printers in the native Windows format while also accessing mission-critical business applications running on the server. With peer-to-peer compatibility with Microsoft NT 4.0, it appears to desktop clients exactly as if it were an NT server and is ideal for environments which need easy UNIX-to-Windows and UNIX-to-Windows NT integration.

- Netscape® Directory Server™

This product is a fast and flexible LDAP directory service. Netscape Directory Server provides the foundation for centralized administration across intranets and extranets.

- Netscape Messaging Server™

The Netscape Messaging Server quickly delivers mail with embedded sound, graphics, video files, HTML forms, Java applets, and desktop applications. It outperforms other messaging systems in the speed of message processing, handling queues and power of directory lookups. Organizations that use the Netscape Messaging Server on UnixWare 7 will find that they can communicate with virtually all mail systems and gateways.

- RealNetworks RealSystem® 5.0 Servers

RealNetworks software (including servers, encoders, and players) provides a complete solution for streaming media and allows Internet or intranet users to easily access multimedia content in real time.

- SCO Merge™ 4.1.1

This optional service provides users the capability of running Microsoft Windows 95, Windows 3, and DOS sessions on UnixWare® 7.

- SCO TermLite™ 1.0.7

SCO TermLite is a simple, lightweight Windows terminal emulator that allows you to run UNIX programs in a window on your PC alongside local Microsoft Windows applications. It provides fast emulation of SCO ANSI and VT320 terminals over a Telnet (TCP/IP) or NVT (NetBIOS) connection. SCO TermLite supports all versions of Microsoft Windows. SCO TermLite's single executable design makes it ideal for locating centrally on a server, providing a zero-footprint on the client. Alternatively, SCO TermLite can be installed locally on a Windows client, taking up less than 1MB of disk space.

- SCO VisionFS™ 3

SCO VisionFS for UnixWare 7 and SCO OpenServer provides high-performance robust SMB file and printer sharing from UNIX Systems to PC clients running Windows, and provides basic access to server applications. Unlike PC-based NFS products which need to be installed on each individual PC, SCO VisionFS is installed purely on the server, saving the organization both time and money during initial installation, configuration, and ongoing maintenance.

- SCO SCO Vision2K™

The SCO SCO Vision2K™ CD-ROM that is included with UnixWare 7 includes a full suite of windows connectivity and interoperability products.

- SCO® Tarantella™

Tarantella software allows the delivery of any application immediately to any client - without any client software installation. UnixWare 7 includes a single-user Tarantella license for a system administration webtop. This webtop can be accessed from any supported browser, enabling the webtop to travel with the administrator. The result is complete UnixWare system administration from anywhere in the world. Also included is a non-administration single-user license for the Tarantella Application Broker that allows a user to create and use an additional user-oriented webtop. Tarantella software provides a zero disruption route to Network Computing. With UnixWare 7.1, SCO Tarantella™ is no longer included on the Optional Services CD-ROM. The latest release of SCO Tarantella is available for UnixWare 7 from SCO's Tarantella Division. Visit <http://tarantella.com.com/> for more information on SCO's Tarantella Technology.

Development tools are included with the optional UnixWare and OpenServer Development Kit 7.0

There are many useful tools and utilities included free for all editions of the UnixWare 7 system on the Skunkware™ 7 CD-ROM.

For information regarding third-party applications developed for use on UnixWare 7, check the online SCO Applications Directory at:

http://wdb1.sco.com/sdir_web/owa/scodir_seh

2.4 Completing Installation Checklists

An installation checklist is designed to ensure that you correctly and quickly install your UnixWare 7 system. This checklist also provides a written log of your responses should you need to troubleshoot or reinstall your system. These checklists can be found in the ***UnixWare 7 Getting Started Guide***.

Information relevant to all types of installations is contained in the *General information* checklists. These additional checklists might also apply:

Hardware	Complete this checklist to indicate hardware parameters for non-detected hardware.
Network	Complete this checklist if you are configuring TCP/IP or IPX/SPX. SCO strongly recommends configuring the network during installation. Several operating system components and optional services, such as the CDE Desktop, SCOhelp , Netscape FastTrack, and Network Station Manager depend on the presence of the network either during Initial Software Load (ISL) or immediately after the system boots for the first time. If the network is not configured, package installation might fail or the system might fail to boot correctly.
Partitions	Complete this checklist if you want to customize the partition table. If you do not, the installation uses the existing partition table on your system, or creates a single partition using the entire disk for UnixWare 7.
Filesystems	Complete this checklist if you want to alter the default filesystem layout.

Complete the installation analysis for your system by filling out the installation checklists in the ***UnixWare 7 Getting Started Guide***.

The following sections describe the information on each checklist.

2.5 General Information Checklist

The following sections review the information you will need to analyze to complete the general information checklist.

2.5.1 Installation language

Choose the language for use during the system installation from the following options:

- English
- French
- German
- Spanish
- Japanese

2.5.2 Zone, locale, and keyboard

These three interrelated choices determine how the system will display currency, punctuation, and other special characters. Choose from one of these *Zones*:

- Americas (Latin-1)
- Eastern Europe (Latin-5)
- Central Europe (Latin-2)
- Western Europe (Latin-1)
- Northeast Asia
- Other
- All Locales

If you do not know the correct zone, choose All locales. The Locales and Keyboard types displayed differ for the selected Zone. The Locale and Keyboard types displayed below are the results of choosing the Americas (Latin-1) Zone.

Choose from one of these *Locales*:

- C English
- POSIX English
- English for Canada
- English for USA
- Spanish for Argentina
- Spanish for Chile
- Spanish for Mexico
- Spanish for Venezuela
- French for Canada
- Portuguese for Brazil

Setting the zone and locale limit keyboard selection to those that are able to process the desired characters. Choose from one of these *Keyboard* types:

- United States
- Japanese A01
- Japanese AX

2.5.3 License number, code, and data

Obtain all licensing information from the Certificate of License and Authenticity that came with your UnixWare 7 product, or from your vendor's licensing web page. Not all products require License Data.

If you choose to defer licensing, you are issued a sixty (60) day evaluation license.

The license agreement stipulates that the license numbers must be unique. Systems on a network cannot be installed using the same license number.

2.5.4 Host Bus Adapter (HBA) diskette(s)

The installation order of the HBA diskettes is critical. Most system installations will use the following order for loading HBA diskettes:

1. Any vendor-supplied HBA diskette containing the host bus adapter driver for your system. You may or may not have such a diskette.
2. Other vendor-supplied diskettes containing other drivers. These diskettes are also optional.
3. The SCO-supplied HBA diskette. This diskette is always required.

There may be cases where the loading order of the HBA diskettes should differ from the order above. The first driver loaded for any device is the version that will be used. Therefore, the HBA diskette containing the newest or desired driver should be inserted first. The drivers for the boot hard disk controller and the CD-ROM driver are required during installation whereas other drivers can be installed and updated with **pkgadd(1M)** after system installation is complete.

2.5.5 Manual hardware configuration

During installation, SCO recommends that you only use the DCU to configure devices that are needed to install your UnixWare7 system, such as your hard disk or CD-ROM. The UnixWare 7 installation autodetects most common hardware devices. If it is not detected, or two devices share the same interrupt, you must configure the hardware driver using the Device Configuration Utility (DCU). Use the Hardware checklist in Section 2.6 on page 2-36 to review the information required by the DCU.

2.5.6 System node name

The system node name is the name of your system. Networking and mail programs on the system use the system node name to identify the correct data destination. The fully qualified domain name is not needed at this point. Only the node name, i.e., not *money.accounting.yourcompany.com*, just *money*.

The node name must:

- Be unique on the network
- Begin with a letter

- Contain only lowercase letters and numbers and can be up to a maximum of sixty-three (63) characters. For backwards compatibility, keep your machine names at seven (7) characters.

If you are responsible for administering many systems, you can construct a common naming convention. Some sample valid system names include:

- *srv100e*—Describes the building and location where the system resides
- *sales1*—Identifies the set of applications on the system, or the users who will use them
- *rome15*—Describes the general geographic location of the system

2.5.7 Installation type

You can choose to install from:

- CD-ROM
- SPX network server
- TCP network server

If you choose to install from a network server, you will need to complete the appropriate network installation checklists in Section 2.7 on page 2-48.

2.5.8 Use whole disk for installation?

You can use the whole disk to install UnixWare 7 or partition the disk. A disk comprises:

Logical arrays of blocks	On a UnixWare 7 system, the Host Bus Adapter (HBA) controller presents its associated hard disk as an arrays of blocks to the operating system. Usually each block is 512-bytes.
Partitions	You can divide the hard disk into one to four contiguous data areas for the installation of operating systems such as UNIX, DOS™, NT™, and OS/2™. Some vendors' hardware reserve a partition for system information, i.e., Compaq.

You can define up to four (4) partitions. If a second hard disk is detected, you can also create partitions on that disk. During installation, if you choose to customize the partitions, you will need to complete the disk partitioning checklist in Section 2.8 on page 2-51.

AV 2-2

Default Filesystem Types

Filesystems and Raw Disk Slices

Filesystem	Description	Type	Size	Disk
=====	=====	====	====	=====
/	Root File System	vxfs	1868	1
/stand	Boot File System	bfs	20	1
/dev/swap	Swap Slice	slice	118	1
/dev/dump	Dump Slice	off		1
/home	User File System	off		1
/home2	2nd User File System	off		1
/var	Add-ons File System	off		1
/tmp	Temporary File System	memfs	10	1
/var/tmp	Temporary File System	memfs	401	
/dev/volprivate	Private Volume	slice	11	
ALTS TABLE	Alt/Sector Slice	slice	401	
ALTS TABLE	Alt/Sector Slice	off		2

(memfs sizes are included in /dev/swap size)

	Disk 1	Disk 2
Available (MB)	2047	2078
Allocated (MB)	2047	0

<F1>=Help <F2>=Choices <F9>=Back <F10>=NextArrow keys, <Tab>=Move

sa2-v0202
© The Santa Cruz Operation, Inc.

2.5.9 Configuring filesystems on active partition

You can configure multiple filesystems on the active partition. You can either use the default filesystem configuration or complete the filesystem checklist to customize your filesystems. See Section 2.8 on page 2-51 for information on the filesystem checklist.

UnixWare 7 allows you to configure disk slices. A slice is a contiguous section of a partition. Slices provide a method to better manage the space inside the partition. Each slice has a:

- Starting sector
- Length
- Slice type
- Permissions (whether it is mountable or unmountable)

AV 2-3

Swap Space

Size of System	Size of Swap Space
16MB	32MB
64MB	75MB
256MB	200MB
1026MB	430MB
4096MB	1330MB

sa2v-0203© The Santa Cruz Operation, Inc.

A slice can contain a filesystem, swap space for paging, or can be left to be organized by an application such as a database.

If you choose to “Use the default filesystem size and types,” the following slices and filesystems are created:

<i>root (/)</i>	A vxfs filesystem type. Contains the bulk of the system, including files, commands, log files, and other data.
<i>/stand</i>	A bfs filesystem type. Contains all the stand-alone programs and text files necessary to boot UnixWare 7.
<i>/dev/swap</i>	Is not a filesystem. It is an area of the disk that is used to swap processes into and out of memory. Swap space should be twice the size of memory on small systems. Refer to the table on the opposite page for recommended sizes of swap based on the amount of memory installed.
<i>/dev/dump</i>	Is not a filesystem. This slice is not enabled by default. The dump slice is used to store a core image of your system if it should crash. If you choose to create a dump slice, it should be as large as your RAM.
<i>/home, /home2</i>	These filesystems are not enabled by default. If you do not enable these filesystems, they are created as subdirectories in the root filesystem.
<i>/var</i>	This filesystem is not enabled by default. It is created as a subdirectory in the root filesystem and contains installation data and administrative files. If you choose to make it a filesystem, choose the same type as your root filesystem.
<i>/tmp</i>	A memfs filesystem type. It is a temporary filesystem. All files and directories are automatically flushed (removed) during each reboot. Note that if you want to use the UnixWare 7 emergency recovery utilities, you must define <i>/tmp</i> as a memfs filesystem type.
<i>/var/tmp</i>	A memfs filesystem type. It is also a temporary filesystem and acts the same as <i>/tmp</i> during reboot. It is used by add-on software packages and must be configured as a memfs filesystem type.
<i>/dev/volprivate</i>	It is a slice used by the Online Data Manager to ensure recoverability.

ALTS TABLE

These slices provide a mapping of bad blocks to good blocks for use by the disk driver. Some devices, such as SCSI hard disks, provide their own mapping scheme. If this is the case, set this to a small size.

2.5.10 Describing filesystems

The operating system's file subsystem provides access to various types of filesystems and their files and directories. Filesystems can reside:

- On hard disks
- On CD-ROMS
- On floppy disks
- In memory

Each UnixWare 7 system has at least two (2) filesystems on the primary hard disk:

- The main filesystem is */dev/root* which is mounted to the *(/)* directory. Most of the key files and directories previously discussed are contained in the root filesystem.
- The other filesystem is */dev/stand* which is mounted to the */stand* directory. This filesystem contains all of the programs and files needed to boot the UnixWare 7 operating system.

All filesystem types can be mounted with the **scoadmin(1M) Filesystem Manager**. However, not all filesystem types can be configured or unmounted with the **Filesystem Manager**. For example, a *vxfs* filesystem can be mounted, unmounted, and configured with the **Filesystem Manager** but a *ufs* filesystem can only be mounted with the filesystem manager. To unmount a *ufs* filesystem you must use the command line and for reconfiguration of a *ufs* filesystem you must edit the */etc/vfstab* file directly.

For those filesystems that cannot be configured or unmounted with the **Filesystem Manager**, use the command-line interfaces.

2.5.11 Filesystem types

Filesystem types that can be configured on your UnixWare 7 system include:

vxfs (VERITAS)

This filesystem is the default filesystem type for UnixWare 7 systems. It is a high-integrity filesystem that provides enhanced performance and reliability.

A *vxfs* filesystem:

- Supports a maximum file and filesystem size of 1 terabyte (TB). The default is 2GB.
- Uses intent logging. The filesystem transactions are recorded in a log and then committed to disk. This increases system recovery speed with a very small performance penalty.
- Supports extent-based allocation. An extent is defined as one or more adjacent blocks of data within the filesystem. When storage is added to a file on the *vxfs* filesystem, it is grouped in extents rather than being allocated one block at a time.
- Provides enhanced data integrity. If the system crashes or is shut down improperly, only files that are currently being written to media can lose data.
- Supports a default logical data block size of 1KB. Can support block sizes of 2048, 4096, and 8192 bytes.
- Supports filesystem concatenation.
- Is not a bootable filesystem.

bfs

The *bfs* filesystem type is a special-purpose filesystem. It contains all of the stand-alone programs (such as */stand/unix*) and all of the text files necessary for booting your system. It is not a user filesystem.

cdfs

The *cdfs* CD-ROM filesystem type supports the Rockridge protocol and the High Sierra/ISO-9660 specifications.

<i>dosfs</i>	The <i>dosfs</i> filesystem type lets users access DOS-formatted floppies and hard disk partitions using UNIX system commands and calls.
<i>nucfs</i> (NetWare® for UNIX Client Filesystem)	The <i>nucfs</i> filesystem is a UnixWare filesystem that provides NetWare file services to a UnixWare platform.
<i>nfs</i> (Network File System)	The <i>nfs</i> filesystem is a service that lets machines of different architectures running different operating systems to share resources across a network.
<i>s5</i>	The <i>s5</i> filesystem type is the only SCO OpenServer filesystem that UnixWare 7 supports.

2.5.12 Choose default disk options or customize

If you choose to “Customize disk options,” you can:

- Install a new boot sector. The boot code from another operating system is removed.
- Scan disk for bad blocks. The surface of the hard disk is scanned for bad sectors ensuring that the boot data is not installed on a bad block.
- Reset disk geometry (destructive). Recalculates the disk cylinder, head, and sector data for the hard disk. Any existing data is destroyed.

If you choose “Use default disk options”:

- A new boot sector is installed. Note that if you installed a previous version of UnixWare, you must install a new boot sector.
- The disk is not scanned for bad blocks
- The disk geometry is not reset

2.5.13 System profiles

System profiles are collections of software that simplify the installation process. When you select a system profile, you ensure that the correct software (services and packages) is installed to enable the type of system you want.

Choose from one of these system profiles:

Licensed-Based Defaults	The type of license that you purchased determines what software and services are installed. For example, if you purchased the “Enterprise Edition,” all packages and services are installed except for multiprocessing support and the kernel debugger.
Small Footprint Server	Includes “Extended OS utilities” and “Networking Services.” It does not include documentation or graphic support.
Full (All Packages)	Installs all packages except for multiprocessing support. Also installs software that is not licensed with a sixty (60) day evaluation license.
Customize Installation of Packages	Choose which services and packages to install. Use to install multiprocessing support.

2.5.14 System services

If you choose to customize your configuration, list one or more system services to install or remove from the services list. Use this checklist item during installation to note any customized services that you install or remove during installation.

2.5.15 System packages

If you configure services, you can select additional packages to install or remove from the package list. Use this checklist item during installation to note any customized services that you install or remove during installation.

2.5.16 Date, time, and timezone

Change the date and time if it displays incorrectly. You will also need to specify the:

- **Continent**—North America/South America, Europe, or Asia/Australia
- **Timezone**—Depending on which continent you choose, you can select from the appropriate timezone

2.5.17 Security profiles

Select one of the four security profiles:

Low	Provides limited security; recommended only for systems not publicly accessible, with a small number of cooperating users.
Traditional	Provides compatibility with other UNIX systems; recommended for systems that use traditional UNIX security.
Improved	Provides additional security features; recommended for systems accessed by groups of users who share information.
High	Used on highly confidential systems that do not allow remote access (including telnet , rlogin , and ftp).

2.5.18 System owner name

When you install the operating system, a user is designated as the *system owner*. The system owner:

- Is given the necessary permissions to perform the day-to-day system administration tasks using the **SCOadmin** managers desktop interface
- Cannot perform most administration tasks from the command line

Any user name (except those already reserved by the system, such as **uucp** and **bin**) can be assigned as the system owner.

The system owner information includes:

- System owner name
- System owner login name
- System owner user ID number
- System owner password

2.5.19 System root password

The **root** account has unlimited privileges to view files and run programs on an installed system. Assign the **root** user a password.

2.5.20 Save installation responses

After you answer installation prompts and before loading software, you can save your installation prompt responses to a floppy diskette for future use in reinstalling this system or another UnixWare 7 system.

2.5.21 Mouse type

After the installation of the software is complete and you reboot the system for the first time, you will need to enter the following information about your mouse:

- The type of mouse:
 - Bus Mouse
 - PS/2 compatible mouse
 - Serial Mouse—Select the serial port to which it is attached
 - No Mouse
- The number of buttons—two (2) or three (3)

2.6 Hardware Checklist

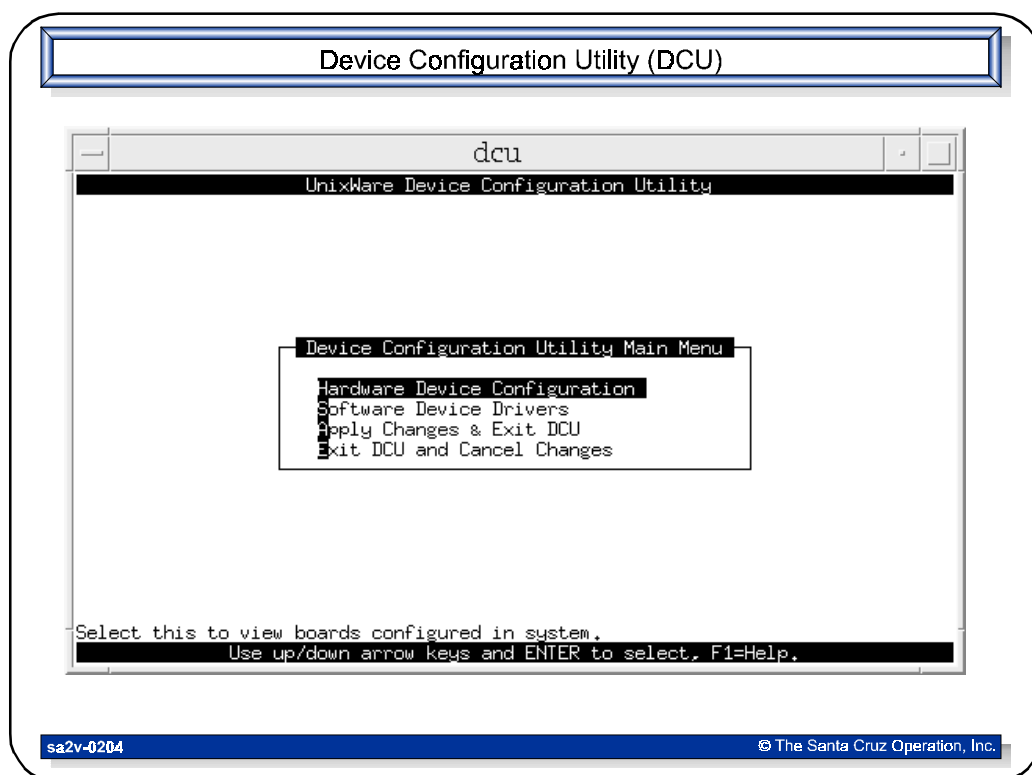
Recall that the UnixWare 7 installation disks contain Host Bus Adapter (HBA) drivers for most supported hardware. In addition, your hardware or software vendor may have supplied you with additional HBA diskettes. During the installation, you are prompted to enter your vendor supplied HBA diskette(s). Reference Section 2.5.4, “Host Bus Adapter (HBA) diskette(s),” on page 2-24 for details regarding the HBA insertion order.

UnixWare 7 automatically determines the correct hardware settings for EISA, MCA, and PCI hardware devices.

During installation, if your hardware is not detected or a setting is incorrect, you can use the **Device Configuration Utility** (DCU) to:

- Configure an ISA device
- View all detected and configured hardware on your system
- Modify hardware settings to resolve interrupt, I/O, and memory conflicts
- Activate or configure a software device driver needed by your hardware
- Deactivate a hardware device

AV 2-4



After installation, you can add hardware devices or change settings by logging in as **root** and entering:

/sbin/dcu

You can also run the DCU from **scoadmin(1M)**. Select **Hardware** ⇔ **Device Configuration Utility (DCU)**.

2.6.1 Entering the DCU

When you enter the DCU during installation, the main menu displays these options:

- **Hardware Device Configuration**

Use this option to view the hardware devices configured on your system. You can change the device driver parameters to match the installed hardware.

- **Software Device Configuration**

Use this option to view or change the software device drivers configured on your system.

- **Apply Changes and Exit the DCU**

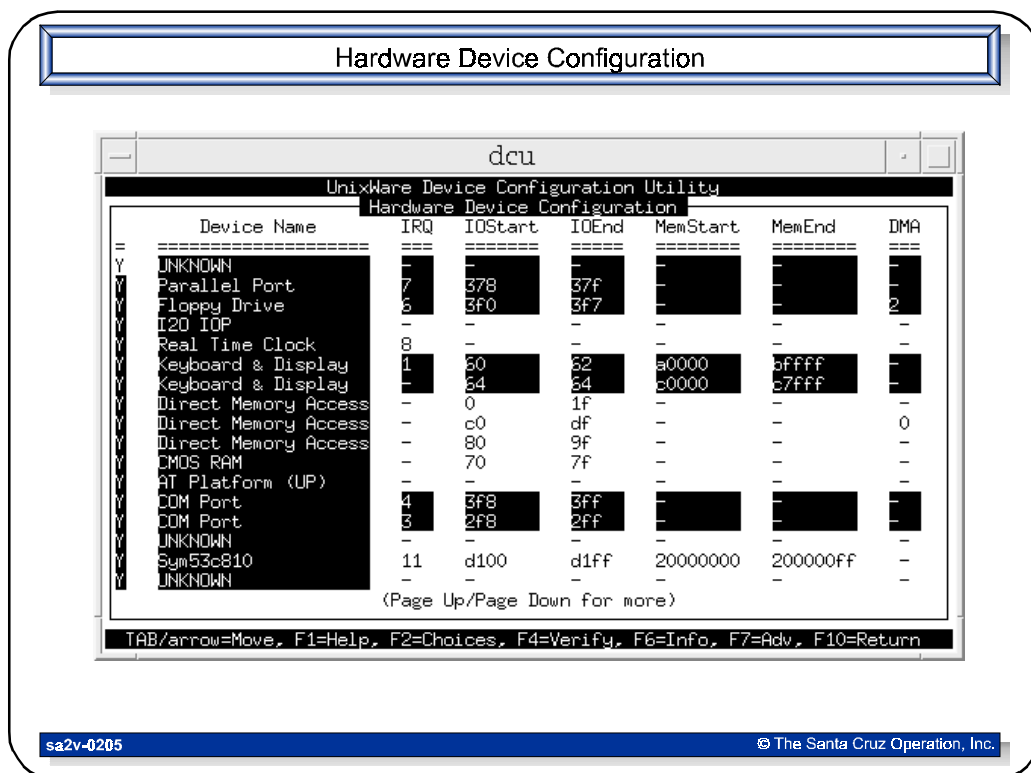
Use this option if you have made changes to any of your hardware or software device configurations.

- **Exit DCU and Cancel Changes**

Use this option to exit the DCU and cancel any changes.

Note that you cannot use your mouse to make selections and navigate the DCU. Use your cursor keys.

AV 2-5



2.6.2 Hardware device configuration

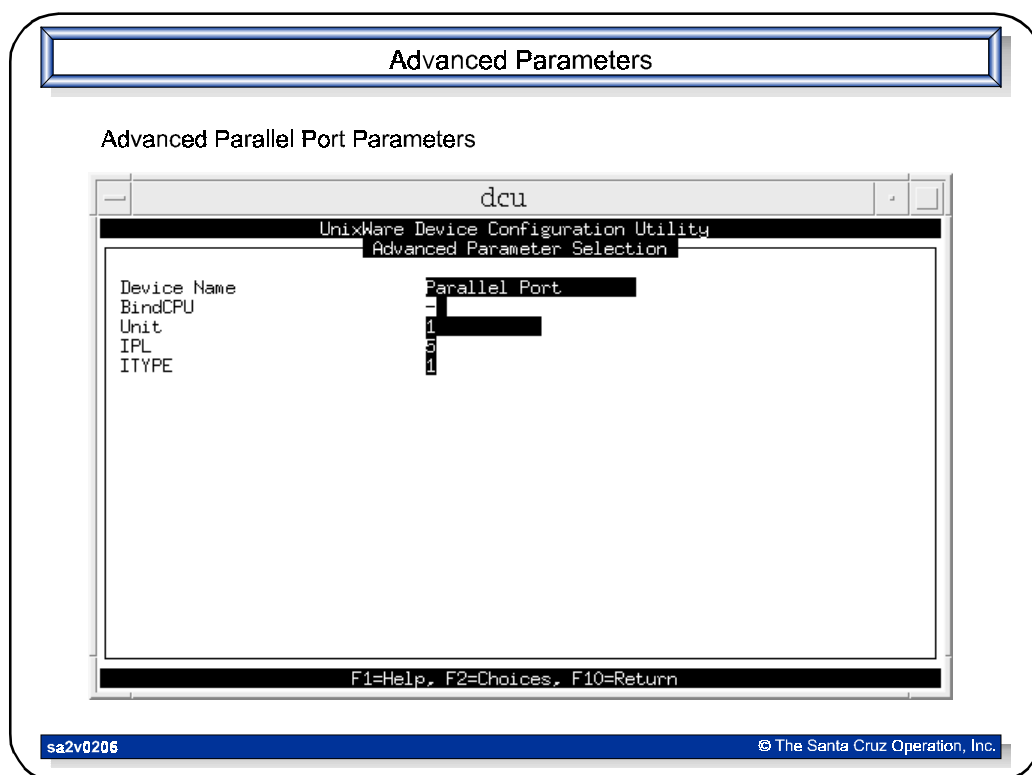
When the DCU is run during installation, you can view your detected hardware by selecting **Hardware Device Configuration** from the main menu. This window displays the hardware settings for each device configured on your system. The dash (-) character indicates that the entry does not require a value for that field because it is not required by the device driver.

These settings display:

Configuration Status	Indicates whether the hardware controller is configured into the kernel by displaying (Y) or (N).
Device Name	Displays the name of the device. The entry can appear as: • The device name such as <code>Parallel</code> or <code>COM</code> port. • The driver name such as <code>async</code> (the driver for the COM port) or <code>adsl</code> (the driver for an Adaptec 294x SCSI host adapter). • UNKNOWN. The DCU cannot assign a driver name. On an EISA, MCA, or PCI system, this is an entry from NVRAM (Non-Volatile RAM indicating that you have installed the hardware, but did not install the software driver. For example, if you have installed a network adapter, but have not installed the corresponding software for the adapter, it is listed as UNKNOWN. • UNUSED. Indicates that an ISA driver is disabled although the device settings are retained.
IRQ	Specifies the interrupt vector number used by the device. Valid values are 1 through the value of the highest interrupt vector number supported by the system.
IOWStart, IOWEnd	Specifies the range between the lowest and highest (inclusive) I/O port address through which the device communicates. Valid values are hexadecimal numbers from 0 to FFFF. For devices without controller memory, this field should contain the value of 0.

MemStart, MemEnd	Specifies the range between the lowest and highest (inclusive) address in memory through which the device can communicate. Valid values are hexadecimal numbers from 10000 through FFFFFFFF. For devices without controller memory, this field should contain the value of 0.
DMA	Specifies the channel number for hardware that use DMA channels. Valid values are 0 through 7. For hardware devices that do not use a DMA channel, this field should contain the value of -1.

AV 2-6



You can display **Advanced Parameters** settings by using your <Tab>  key to move to the device name and then pressing .

Advanced parameter settings are:

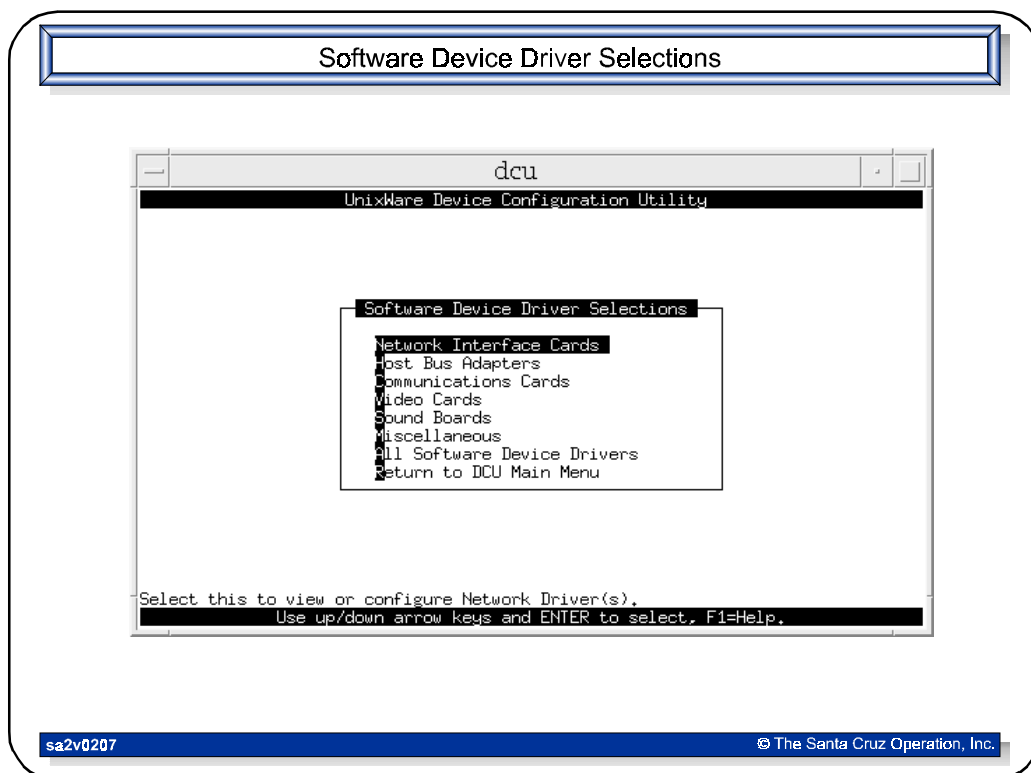
Unit	Specifies the unit number for a controller that supports multiple devices.
IPL	Specifies the interrupt priority level for those devices that have priority handling capability. Valid values are 1 (lowest priority) to 7 (highest priority). When multiple controllers share the same IRQ values, the software device drivers must operate at the same IPL.
ITYPE	The IRQ sharing type. If this field is not used, it must contain a 0.
BindCPU	The CPU to which the device binds. The driver code will only run on the specified CPU.

2.6.3 Software device drivers configuration

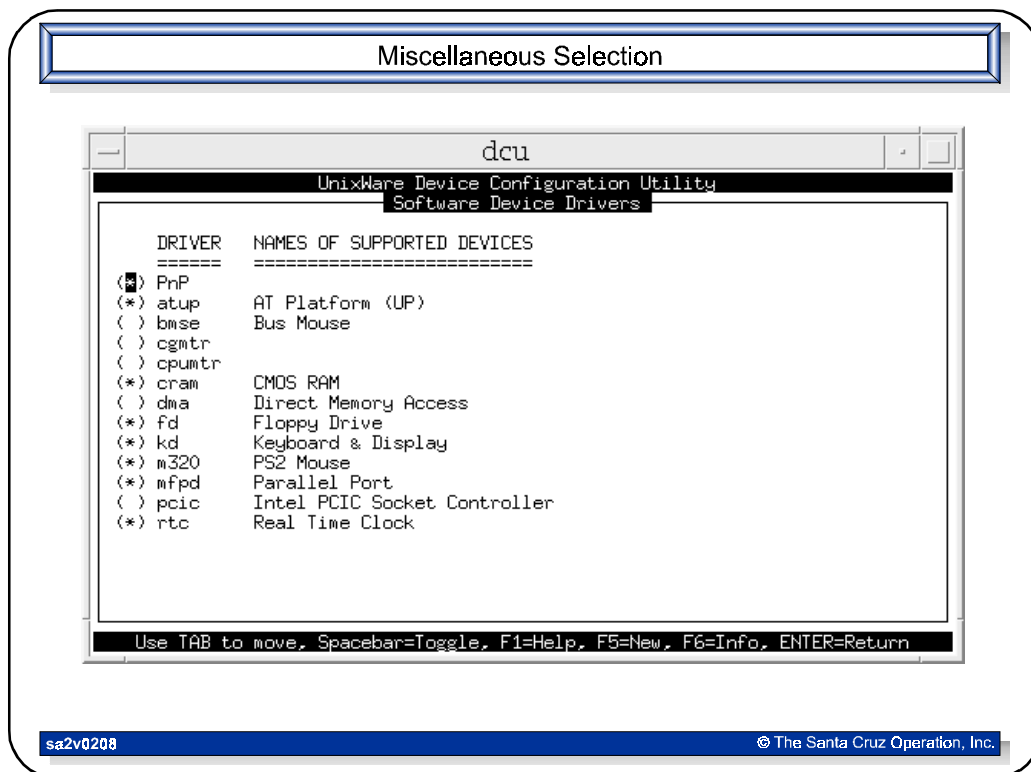
The Software Device Drivers screen of the DCU lists each available software device driver and the associated hardware device.

Use this option to activate a device driver, if your hardware device is manually configured and does not have software setup utilities that define the hardware settings to your system's NVRAM.

AV 2-7



AV 2-8



The Software Device Drivers selections menu classifies drivers into several categories:

- Network Interface Cards
- Host Bus Adapters
- Communication Cards
- Video Cards
- Sound Boards
- Miscellaneous

The last selection on the menu “All Software Device Drivers” combines all of the categories.

When you select a driver category, the following information displays:

* Indicates the driver is active and available for use assuming the hardware that it supports is installed and enabled. If this field is blank, the driver is not active. To activate a driver, press in this field.

DRIVER The name of the software driver

NAMES OF SUPPORTED DEVICES

Lists the hardware controllers the device driver supports

2.7 Networking Checklist

The following sections review the data required for completing the networking checklist.

2.7.1 Network adapter

If your network adapter is a “smart” bus adapter (for example, an EISA or PCI adapter), UnixWare 7 will detect and autoconfigure the appropriate network driver.

If it is an ISA or PCMCIA/PC adapter, you must manually enter the configuration parameters during the installation.

- Network adapter vendor, make, and model
- Interrupt Vector (IRQ)
- I/O Address Range
- ROM Address
- DMA Channel
- Memory (RAM) Address Range
- RAM size

You can also defer the installation of your network adapter and install it later using the **scoadmin(1M) Network Configuration Manager**.

2.7.2 Configuring TCP/IP networking

If you did not defer the installation of your network card, and the system is a server or a client of a TCP/IP network, you will need to provide this information:

System IP Address	Four decimal numbers separated by periods. For example:132 . 145 . 80 . 9
System Netmask	If you are not configuring a subnetwork, enter a netmask of 255 . 255 . 255 . 0
Broadcast Address	In keeping with the IP address in Bullet 1, the Broadcast Address is 132 . 147 . 80 . 255

Default Router	Enter the router IP address if your system is using another system to route packets. This field is optional.
Domain Name	The domain name is an abbreviation for the server site and a code for the site's organization or country. For example, <i>sco.com</i> is the domain name used by SCO. If your system is registered, use your domain name provided by your network administrator. If you are creating a new domain and are not certain that you will eventually connect to an outside network, use <i>companyname.com</i>, where name is the name of your company or organization.
Primary DNS Address	Enter the IP address of your preferred Domain Name System name server for your system. This field is optional.
Other DNS Address	Enter the IP address of an alternate Domain Name System name server. These two fields are optional.
Frame Format	For TCP/IP, choose ETHERNET_II

2.7.3 Configuring IPX networking

If your system is part of an IPX network, enter this information:

- IPX Net Number
- Frame Format

If it is not, you can defer the configuration.

2.7.4 NIS configuration

If you installed the Network Information Service (NIS) package, you can configure your system as a:

- Master server
- Slave server
- Client

AV 2-9

Disk Configuration

Disk No.	Character Node	Action
=====	=====	=====
1	/dev/rdisk/c0b0t0d0s0	Customize partitions
2	/dev/rdisk/c0b0t2d0s0	Do not modify

<F1>=Help <F2>=Choices <F9>=Back <F10>=NextArrow keys
or <Tab>=Move

Choose One

() Use whole disk for UNIX
(*) Customize partitions
() Re-load partition information from disk

sa2v0209© The Santa Cruz Operation, Inc.

AV 2-10

The Partition Table

Type	Status	Start	End	%	Cylinders	Size (MB)
===	=====	=====	=====	=====	=====	=====
1 UNIX System	Active	0	268	50	261	2047
2 unused		0				
3 unused		0				
4 unused		0				

Number of Cylinders: 522 Currently used: 261

<F1>=Help <F2>=Choices <F9>=Back <F10>=NextArrow keys or
<Tab>=Select

sa2v0210© The Santa Cruz Operation, Inc.

2.8 Partition and Filesystems Checklist

If you choose to customize the partitions, determine the following:

- Number of partitions
- The type of partition

UNIX System	Use for UnixWare 7, SCO OpenServer, or other UNIX systems
Pre DOS 5.0	Use for DOS systems 5.0 and earlier. It must be loaded in partition 1 and start at block 0
DOS	Use for DOS systems after release 5.0. It can be loaded onto any partition.
Other	Use for other operating systems, such as OS/2
System	Use for autodetected hardware information. If this partition is detected, do not remove or reconfigure. This partition contains information specific to your hardware setup.

2.8.1 Installing other partitions

If you plan to install Windows 95, Windows NT, or OS/2 partitions:

- Use the NT or OS/2 tools. These operating systems can be loaded onto any partition.
- Remember that the UnixWare 7 partition must begin below cylinder 1024.

If you plan to install DOS partitions, note the following:

- A Pre-5.0 DOS partition must be the first partition on the first disk and must lie within the first 1024 cylinders. It must not exceed 32MB.
- A Post-5.0 DOS partition can be loaded onto any partition.

All DOS partitions should have a minimum size of 3MB.

- The Status of the partition. The `Unix System` partition must be:
 - **Active**. Only one partition can be active.

- At least 500MB in size. If you do not define a large enough active partition, the installation will not continue until you create one.
- Any change in the size, type, and cylinder position for an existing partition results in the partition being removed and recreated. All data in the partition is lost. This is also true for any other partitions.
- The starting cylinder of each partition
- The percentage of the disk being used for the partition

2.8.2 Configure slices and filesystems

If you choose to **Customize filesystem and slices**, you can:

- Change the size of your root filesystem.
- Change the size of the swap slice.
- Create a dump slice.
- Create and specify the type of two user filesystems on the primary disk. If you created a partition on Disk 2, you can specify that these filesystems are created on Disk 2. Note that */var*, */tmp*, and */var/tmp* can also be created on Disk 2.

Filesystem sizes are entered in megabytes.

AV 2-11

Responding to Prompts	
To:	Press:
Get help	<F1>
Toggle or display choices	<F2>
Read/write to media	<F3>
Configure advanced parameters	<F6>
Defer configuration	<F8>
Move back one window	<F9>
Move forward one window	<F10>
Move down	↓(down arrow)
Move up	↑(up arrow)
Move through selections	<Tab>
Select items in a list	<Space>
Accept an item	<Enter>

sa2v0211

© The Santa Cruz Operation, Inc.

2.9 Installing UnixWare 7

During the installation, use the checklist you completed to respond to the installation prompts.

The installation of the operating system is divided into four stages:

Interview	The installation prompts you with questions and records your answers. Your replies determine which software components are installed, how system hardware is configured, and the disk division layout.
Media extraction	Files are extracted from your installation media. The installation program installs and configures your system using the parameters defined during setup. This stage is not interactive.
Software loading	The installation program installs and configures your system using the parameters defined during setup. This stage is not interactive.
Initial hard disk boot	First boot of the primary hard drive. Network, mouse, and language supplement configuration occur during this phase of the installation. At the end of this phase morepkgs is invoked to run the last phase of the installation.
Updates and layered products installation	Interview and extraction of updates and optional products.

2.9.1 Responding to prompts

Use the keys from the slide on the adjoining page to respond to information prompts during the setup phase of the installation.

If you make an error while entering information, use your <Backspace> key to erase and then retype the information.

2.10 Lesson Summary

The UnixWare 7 family of products consists of:

- The operating system (runtime)
- Development Kits (both software and hardware)
- Optional Services, Vision Products, and Evaluation Software

While some variation is possible, the minimum suggested hardware configuration for the UnixWare 7 operating system includes:

- A Pentium processor
- 64 MB RAM
- 1GB of free disk space
- CD-ROM drive
- Tape drive

You can access the **SCO Compatible Hardware Web Pages** to investigate whether specific system hardware is compatible with UnixWare 7.

The **UnixWare 7 Getting Started Guide and Release Notes** contain useful information pertinent to installation. All system installations begin with an analysis of the system. You must consider many points, including:

- Which software is to be run on the system:
 - Graphical interfaces
 - Networking software
- How the hard disk should be configured
- How many users will share the system resources
- Which security profile is needed
- What hardware resources are available

You should complete an installation checklist prior to the installation of any system.

Review questions

1. Which system profile do you have to choose so that multiprocessing support is installed on the Enterprise Edition?
2. What is the recommended minimum amount of memory to run UnixWare 7?
3. Which program is used to configure hardware devices during installation?
4. Which devices does SCO recommend that you configure during installation?
5. Why must you define the */tmp* filesystem as a memfs filesystem type?

2.11 Lab: Installation

Overview

This lab describes the procedure for installing UnixWare 7.

Procedure

These steps describe how to install the UnixWare 7 operating system software using the CD-ROM installation method:

- Step 1. Insert the floppy diskette labeled, “Installation Disk 1 of 2” into the primary floppy drive.
- Step 2. Turn on your computer.
- Step 3. The system performs a POST test displaying your hardware information. After this test completes, the message `Starting UnixWare...` and the animated logo appear. You are prompted to remove the first diskette, and insert the floppy diskette labeled “UnixWare 7 Installation Diskette 2 of 2.”
- Step 4. Choose the language you to use during installation.
- Step 5. Insert diskette 2. Press `ENTER`.

After the Welcome to UnixWare 7 screen displays, if you previously installed UnixWare 7 on this or another system, and saved your installation responses to a diskette, you can use those responses as the defaults for this installation by inserting the diskette and pressing `F9`.
- Step 6. Press `F10` to continue the installation. Use the information from your checklist to install your system.
- Step 7. Choose a zone for your system and press `F10`.
- Step 8. Choose a locale for your system and press `F10`.
- Step 9. Choose a keyboard for your system and press `F10`.
- Step 10. Enter the licensing information provided by your instructor. Press `ENTER` after each entry. Press `F10` to continue.

If you make an error while entering the licensing information, use your <Backspace> key to erase and then re-enter the information.

Step 11. Loading Host Bus Adapter (HBA) drivers.

If you do not have any vendor-supplied HBA diskettes, just load the SCO-supplied one.

If you have a vendor supplied HBA follow the recommended installation order described in Section 2.5.4, “Host Bus Adapter (HBA) diskette(s),” on page 2-24. Note that the SCO HBA is always inserted during HBA driver loading even if you are using a vendor supplied diskette.

Step 12. Choose **Proceed with installation**. The Host Bus Adapters (HBA) drivers are loaded onto your system. Press **F10**.

Step 13. Choose one of these selections when the Device Configuration Utility (DCU) screen displays:

- **Enter the DCU** (manually configure drivers). You may need to run the (DCU) to configure a device used during your installation, such as your CD-ROM drive.

SCO recommends that you only configure those devices that are needed to install UnixWare 7. After the installation completes, you can run the DCU and configure additional devices.

- **Select Do not enter the DCU** if UnixWare 7 can autoconfigure the drivers

Press **F10** to continue.

The system displays the detected or selected drivers as it installs them.

Step 14. Enter your system node name. Press **F10** to continue.

Step 15. Choose **Install from CD-ROM** as the installation method. Press **F10**. You are prompted to insert the CD-ROM.

If you are installing from CD-ROM and the CD-ROM installation method selection does not display, select **Cancel installation and shut down**. Restart the installation from the beginning. You must enter the Device Configuration Utility (DCU) in Step 11 and configure your CD-ROM.

Step 16. Configure the first hard disk.

Recall that up to two detected hard disks can be configured at installation time. You can configure your second hard drive after the initial system installation.

Step 17. Press **F2** to display these choices:

- **Use whole disk for UNIX.** A single disk partition is created and the partition is marked active. Any existing data is destroyed.
- **Customize partitions.** The disk partition table displays allowing you to create up to four partitions. Enter the information from your installation checklist. The “UNIX System” partition must be “Active.”
- **Do not modify.** This option skips the modification of the second disk.
- **Re-load partition information from disk.** This option restores the configuration on your hard disk to what it was when you started the installation.

Step 18. Choose **Use whole disk for UNIX**, for the first disk and “Do not modify” for the second disk. Press **F10** to continue.

Step 19. Choose one of these selections to configure slices and filesystems:

- **Use default filesystem and types**
- **Customize filesystems and slices**

Choose to **Customize filesystems and slices**. Press **F10**.

Step 20. Use your **Tab** **F4** key to move from field to field. Create a separate */home* filesystem from the information on your installation checklist. Remember that you will need to decrease the size of your *root* filesystem to add space in your */home* filesystem.

The 64K inode limit restricts the number of files and directories that you can create. Note that by default the 64K inode limit is not imposed. Restricting to 64K inodes may be necessary for legacy (backwards compatibility) applications. To reset this restriction, press **Tab** **F4** to advance to the **Size** field of the filesystem and then press **F6**. Answer “Yes” to the “64K inodes limit.” You can also change the default block size by pressing **F2** in the “Filesystem block size” field and choosing a new block size.

Step 21. Accept the default. Press **F10** to continue.

- Step 22. Perform disk operations. Choose **Use default disk options** from these selections.
- **Use default disk options.** A new boot sector installs, your disk is not scanned for bad blocks and the disk geometry is not reconfigured.
 - **Customize disk options.** Enter the options from your installation checklist.

Note that if you choose to customize disk options and select to **Reset disk geometry (destructive)** the data (including data in non-UNIX partitions) is destroyed. Use with extreme caution.

Press **F10**.

- Step 23. Choose your System Profile. Select **License-Based Defaults**. Press **F10**.

- Step 24. Choose one of these selections to confirm or install your network interface card:

- **Use the detected card shown above**
- **Select from a full list of supported cards**
- **Defer network configuration**

Select **Use the detected adapter shown above**. Press **F10**.

- Step 25. Continue to configure your networking hardware. Depending on your networking card, you may be asked to enter or confirm additional information. Press **F10**.

- Step 26. Enter your TCP/IP networking parameters and frame type from your installation checklist. Press **F10**.

If you are not configuring TCP/IP, you can defer the configuration by pressing **F8**.

- Step 27. Enter your **IPX Net Number** and **Frame Type** from your installation checklist. Press **F10**.

If you are not configuring IPX/SPX, defer the configuration by pressing **F8**.

- Step 28. Defer NIS. Press **F8**.

Note that if you press the **F9** key to move backward and then press **F0** key to move forward to re-enter a configuration window, for a service that you have already deferred, you must press **F8** again if you still want to defer the configuration.

- Step 29. Confirm or enter the date and time values. Press **F0**.
- Step 30. Select **Traditional** as the security level. Press **F0**.
- Step 31. Enter the system owner name, login name, user ID, and password. Press **F0**.
- Step 32. Enter the **root** password. Press **F0**.
- Step 33. Select the **Optional Service** and press **F1** to access more information about the service.” This is an information only screen. Press **F0** to continue.
- Step 34. Select **Accept** to acknowledge that your understand the License Agreements and agree to be bound by their terms and agreements. If not, select **Do not accept** and the installation will not continue.
- Step 35. Press **F0** to continue.

The configuration of the installation is now complete, you can:

- Press **F9** to view or change any of your choices
- Press **F3** to save your choices to a floppy disk
- Press **F0** to install software

- Step 36. Press **F0** to install the software.

Software loading is the longest stage of the installation and can proceed unattended. You can track the status of the installation by viewing the progress indicator.

- Step 37. When prompted, insert the UnixWare 7.1 HBA diskette. Press **ENTER** for final HBA configuration.
- Step 38. Remove your floppy disk and CD-ROM when the installation completes.
- Step 39. Press **ENTER** to shut down and restart your system.

After your system restarts and several non-interactive configurations take place, you are prompted to configure your mouse and complete the installation.

Follow these steps to configure your mouse and complete the installation:

- Step 1. Select the type of mouse. Press **F10**.
- Step 2. Confirm or change the mouse parameters. Press **F10**.
- Step 3. Press **e** to begin the fifteen (15) second mouse test.
- Step 4. Click your right mouse button. Press **e** to continue.
- Step 5. Enter the UnixWare 7 Operating System Updates CD-ROM (Disk 2 of 3),
when prompted

Follow the prompts to install the desired updates. Updates are highlighted based on the license you entered and on what packages you installed from the Installation CD-ROM. You are prompted to insert other CD-ROM volumes in the set.

Depending on your license and the software already installed on your system, you are then presented with a list of products you can install.

Default selections matching your license are automatically highlighted for installation. You can override these defaults by selecting or de-selecting products.

If you have insufficient disk space, you have the option of installing these products at a later time.

- Step 6. Enter the UnixWare 7 Optional Services CD-ROM (Disk 3 of 3), when
prompted.

Follow the prompts to install the desired services. Services are highlighted for installation based on your operating system license.

- Step 7. Reboot the system.

Usually, your next step is to create your emergency recovery media.

2.12 Relevant Commands

The following commands were presented in this lesson.

Relevant commands

dcu(1M)

The Device Configuration Utility

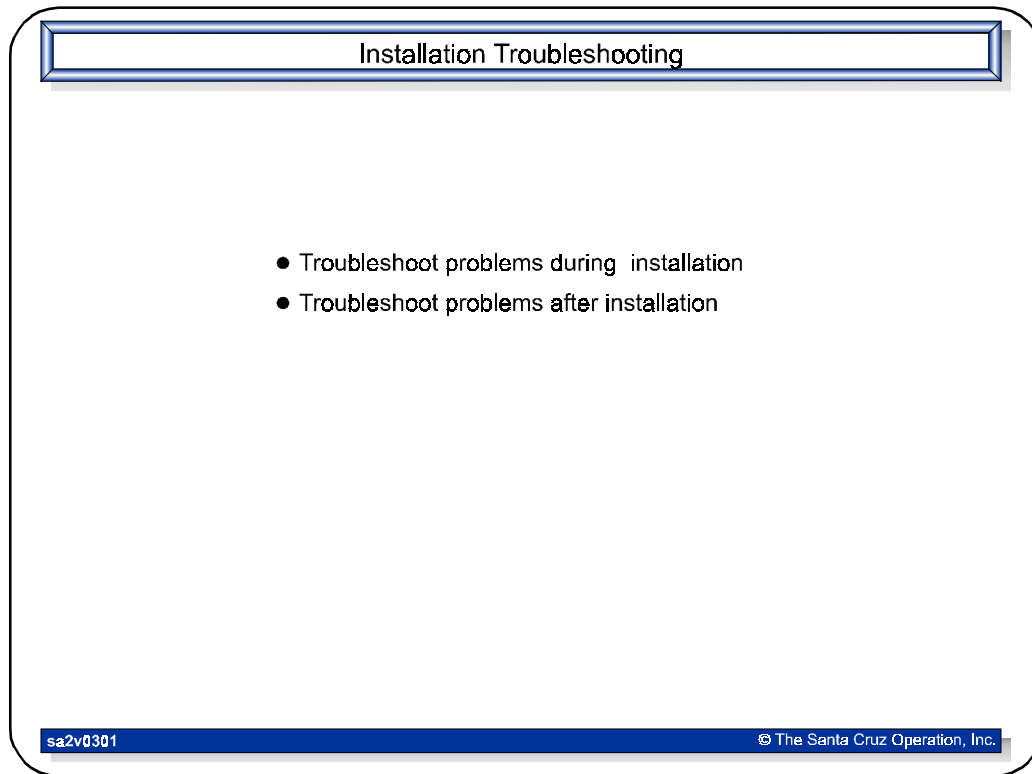
2.13 Answers to Review Questions

1. You have to choose the Custom Configuration system profile so that multiprocessing support is installed on the Enterprise Edition.
2. The recommended minimum amount of memory to run UnixWare 7 is 32MB.
3. The Device Configuration Utility **dcu**(1M)) program is used to configure hardware devices during installation.
4. The devices that SCO recommends that you configure during installation are ones that are needed to install the operating system.
5. You must define the */tmp* filesystem as a memfs filesystem type if you want to use the UnixWare 7 emergency recovery utilities.

3

Installation *Troubleshooting*

AV 3-1



Lesson 3

Installation Troubleshooting

Introduction

After you have completed the installation analysis and checklist for your system, you can install the operating system with confidence. The following lessons describes the procedures to follow if you encounter a problem during or immediately after the installation.

Objectives

The purpose of this lesson is to install the UnixWare 7 operating system on your system. By the end of this lesson, you will be able to:

- Troubleshoot problems during installation
- Troubleshoot problems after installation

3.1 Troubleshooting Problems During Installation

This section covers some of the common problems that might occur during your installation.

If you encounter a problem that is not specifically documented, do the following before calling for help from your software service provider:

- Confirm that your hardware is supported. Refer to the **UnixWare 7 Getting Started Guide** and **Release Notes**. If you have Internet access on another system, access the SCO Compatible Hardware Web Pages:

http://wdb1.sco.com/chwp/owa/hch_search_for

The SCO Compatible Hardware Web page includes:

- Listings of devices supported on SCO platforms
- Notice of new and updated drivers for supported devices
- Technical information concerning driver installation, configuration, and troubleshooting
- Make certain that your system devices are recognized during the boot process
- Verify that you installed any vendor supplied Host Bus Adapter (HBA) diskettes
- Check for hardware conflicts between components:
 - DMA
 - Interrupt vectors
 - Memory address
- Enter the Device Configuration Utility (DCU) and view, add, or change hardware settings if the installation completes, or if the installation did not complete, restart the installation and enter the DCU

3.1.1 System does not initially boot

If the system does not start the installation the first time you boot from the boot disk, make sure that:

- You are using the correct boot disk
- The disk is inserted correctly in the drive, and the drive door is closed

You may need to run the manufacturer's floppy drive diagnostic tests to determine that the floppy drive hardware is functioning properly.

These are some guidelines for troubleshooting boot disks and boot floppy drive problems:

- If the floppy drive works, you may have a damaged boot disk. Try booting the boot disk on another machine.
- If an error occurs while reading the floppy disk, make sure that the boot disk is inserted correctly in the drive, and the drive door is closed. Try removing and reinserting the floppy disk to make sure it is seated correctly in the drive.
- If the boot disk is damaged, and you do not have a backup copy, request a new one from your software provider

3.1.2 System hangs or resets during boot

If your system hangs or resets while booting with the UnixWare 7 Installation Diskette 1, the problem could be the result of not running a hardware setup program. You may be able to correct this problem by:

- Verifying that the hardware is recognized in the CMOS
- Running all programs provided by your hardware vendor and verify that your hardware is configured correctly

3.1.3 System cannot find the hard disk

The drivers that are supplied with your installation media may not contain the hardware driver for your primary hard disk. If so, check with the manufacturer's documentation for the required drivers.

Follow these guidelines when troubleshooting this problem:

- Restart the installation and insert the vendor's HBA diskette
- Enter the DCU and ensure that the controller settings are recognized and correct

3.1.4 System cannot find the CD-ROM

If your SCSI CD-ROM is not recognized during the installation, verify that the correct target ID is selected:

- Determine the target ID setting of the CD-ROM
- Refer to the manufacturer's documentation that the CD-ROM drive is terminated properly in relation to any other SCSI devices that may be on the SCSI bus

If your CD-ROM is located on a host bus adapter or controller other than your primary, restart the installation and enter the DCU to determine if the controller or host bus adapter is recognized.

3.1.5 Resolving other installation problems

Resolve other installation problems by referring to the ***UnixWare 7 Getting Started Guide and Release Notes***. This document contains many solutions related to specific hardware and bus types.

3.2 Troubleshooting Problems After Installation

Two of the most common problems after installation include the:

- Mouse not responding
- Proper configuration of your video display

The following topics discuss how to correct these problems.

3.2.1 Mouse not responding

The UnixWare 7 mouse software supports any mouse that is compatible with one of these mouse types:

- Any PS2 compatible/Keyboard mouse
- Logitech serial, bus, and Mouseman
- Microsoft serial and bus mouse
- Mouse System Corporation (MSC) compatible mouse

3.2.1.1 Bus mouse

Verify that the bus mouse controller board is recognized during the boot process and that there is no conflict with the IRQ and the I/O address range.

If you have installed a bus mouse, the driver package is configured to use an I/O address range of 23c through 23f. The installation will fail if this address range is not free.

Use the Device Configuration Utility (DCU) to view your current bus mouse controller board attributes. If there is an I/O port address or IRQ conflict, make the appropriate changes and reconfigure your mouse controller board.

3.2.1.2 PS2 compatible/keyboard mouse

If you have installed a PS2 compatible/Keyboard mouse, the mouse driver package is configured to use IRQ 12. The installation will fail if this IRQ is not available.

Note that the DCU will not display the IRQ and I/O address for a PS2/Keyboard mouse. Refer to your hardware manual for these parameters.

3.2.1.3 Serial mouse

Verify that the serial port or multiport board to which the mouse is attached is recognized during the boot process.

Enter the DCU and verify there are no IRQ or I/O address range conflicts.

3.2.1.4 Testing your mouse

If you have made changes in the DCU, after exiting the DCU and applying the changes, use the **mouseadmin(1)** utility to test your mouse.

Follow these steps to test your mouse:

- Step 1. Log in as **root** on the console.
- Step 2. Run the **mouseadmin** command.
mouseadmin
- Step 3. Select **T** to “Test your mouse configuration.”
- Step 4. Press  to begin the mouse test.
- Step 5. Press a mouse button to stop the test.
- Step 6. Press **E** to exit the **mouseadmin** utility.

3.2.1.5 Adding a new mouse

If your mouse test is not successful or your existing mouse does not work, you may have to add a different type of mouse. Use the **mouseadmin** command to install a new mouse.

Note that you must enter and exit the **mouseadmin** utility twice. Once to remove the existing mouse and second, to add the new mouse.

Follow these steps to remove a mouse:

- Step 1. Log in as **root**.
- Step 2. Run the **mouseadmin(1)** utility, enter:
mouseadmin

- Step 3. Select **R** to remove your mouse.
- Step 4. Select **U** to update your mouse configuration and quit.
- Step 5. Run the **mouseman** utility again. Enter:
- mouseadmin**
- Step 6. Select one of the following to add a mouse:
- **B**—Adds a Bus mouse. Enter the IRQ.
 - **P**—Adds a PS2 mouse. Enter the IRQ.
 - **S**—Adds a Serial mouse. Enter the associated terminal. Enter “console” if the mouse is attached to your console.
- Step 7. Select **U** to update your mouse configuration and quit.
- Step 8. Test your mouse by following the steps under Section 3.2.1.4 on page 3-8.

3.2.2 Configuring your video

When you first install the system, the X server attempts to create a default resolution of 800x600 with 256 colors using the SCO Video Electronics Standard Association (VESA) driver.

If your adapter cannot be configured with this driver, the X server defaults to standard IBM VGA mode (640x480 16 colors).

For more information on the default video settings, refer to the *UnixWare 7 Release Notes*, “Configuring Video Adapters,” “Understanding Resolutions.”

Most video adapters are automatically detected by the driver and are properly configured. If not, you can use the **scoadmin(1M) Video Configuration Manager** to select a different:

- Video adapter type— If you do not know what type of adapter is installed, run the program:

/usr/bin/X11/VideoHelp

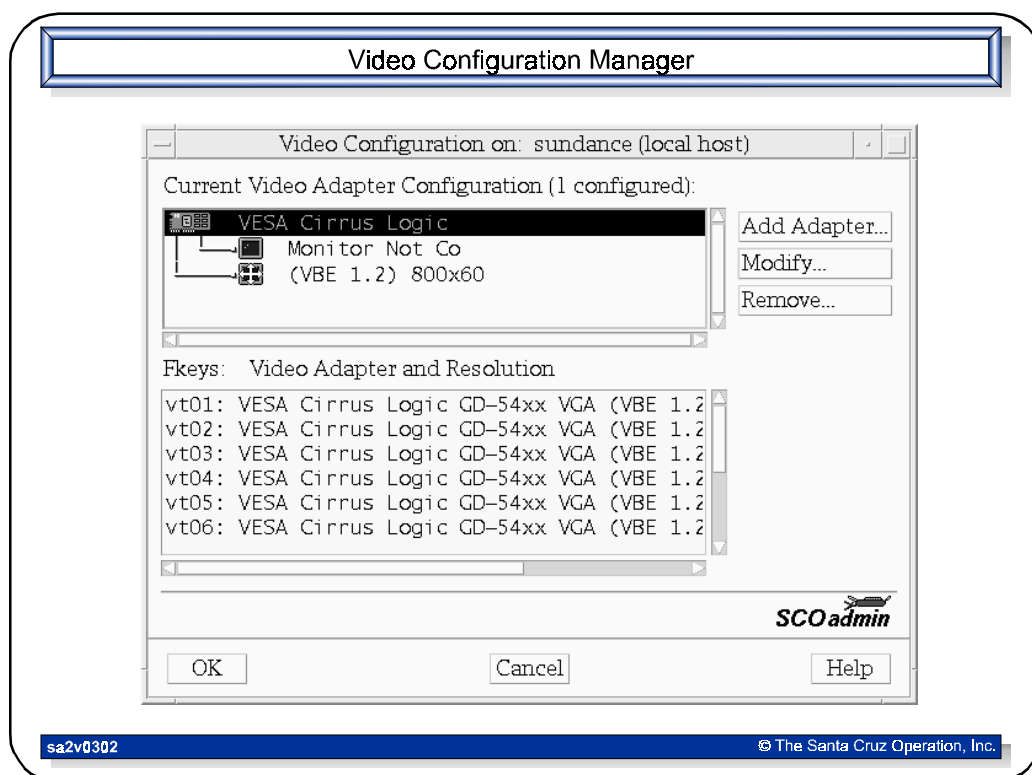
The **VideoHelp(X1M)** program attempts to identify graphic adapters by reading vendor-specific strings from the adapter’s ROM.

- Resolution—Some resolutions only work if you have enough video adapter memory. Ensure that you have at least the minimum DRAM or VRAM to support the number of colors at the specified resolution.
- Monitor type—Although your video adapter supports the higher resolutions, your monitor may not

The **scoadmin Video Configuration Manager** is usually run from the desktop, however if you are unable to access the desktop due to your video, log in as **root** on the console and enter:

scoadmin video configuration manager

AV 3-2



From the desktop login screen, access the **Video Configuration Manager** by following these steps:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(1M)**.
- Step 3. Select **Hardware** ⇒ **Video Configuration Manager**.

3.2.2.1 Changing the adapter type and monitor

When you enter the **scoadmin(1M) Video Configuration Manager**, if your video adapter type is not autodetected, this description displays:

```
IBM VGA
Monitor Not Configured
640x480 16 color
```

From the **Video Configuration Manager** window, change your video adapter by following these steps:

- Step 1. Select **Modify**.

The **Video Configuration Manager** will attempt to autodetect your video adapter type.
- Step 2. Select your video adapter type or a compatible if it is not autodetected. Click on **OK**.
- Step 3. Select **Change Monitor**. Select the monitor type.

If your monitor type is not listed, choose “Other...” or one that closely resembles your own.
- Step 4. Click on **OK**.
- Step 5. Complete your modifications and Click on **OK**.

AV 3-3

Memory Requirements					
		Colors			
		16 (4 bits)	256 (8 bits)	32K/64K (15/16 bits)	16M (24 bits) 16M (32 bits)
Resolutions					
1600 x 1200	N/A	2MB	4MB	6MB	8MB
1280 x 1024	1MB	2MB	4MB	4MB	6MB
1024 x 768	.5MB	1MB	2MB	4MB	4MB
800 x 600	.5MB	5MB	1MB	2MB	2MB
640 x 480	.5MB	5MB	1MB	1MB	2MB

sa2v-0303 © The Santa Cruz Operation, Inc.

3.2.2.2 Changing the resolution

Before changing the resolution, it is important to note that by selecting an incorrect resolution you can lose the ability to return the system to a sane state. If this should happen:

Press the   and <BACKSPACE> keys simultaneously to exit the X server. If your display is not returned to a sane state, you must reboot your system.

If your video adapter supports a resolution other than the default, continue with the steps from Section 3.2.2.1, “Changing the adapter type and monitor,” on page 3-13 to change the resolution:

- Step 6. Select the resolution icon. Select **Change Resolution**. Select the applicable resolution. Click on **OK**.

- Step 7. Select **Test** to test the resolution. Click on **Continue** after reading the “Warning Message.”

 A 30 second video test is performed.

- Step 8. Click on **Cancel** if the video test is not successful and repeat Steps 6 and 7.

 Click on **OK** if the video test is successful.

- Step 9. Click on **OK** to complete the resolution configuration.

 Your video configuration changes will take affect upon your next login.

- Step 10. Select **File** ⇒ **Exit** to exit the **Video Configuration Manager**.

3.3 Lesson Summary

During the setup stage of installation, you answer questions about customizing your installation. During the software loading stage of installation, the installation program transfers files to your system and configures it per your instructions.

There are two general categories for problems that can occur during installation:

- Problems that appear during the installation
- Problems that appear immediately after the installation

If you suspect you have defective installation media, try the media on another system.

If a problem occurs during the installation, follow these guidelines to troubleshoot the problem:

- Determine whether the problem is with the hardware or software
- Verify that the peripherals are set to the correct IRQ, Base RAM, and DMA
- Use the **scoadmin(1M) Video Manager** to make changes in your video configuration
- Use the **mouseadmin(1)** command to reconfigure your mouse

Review questions

1. Which UnixWare 7 guides would you use to resolve installation problems?
2. Which two programs would you run after the installation to install a serial mouse?
3. Which program can you run to help determine the type of video adapter installed on your system?

3.4 Relevant Commands

The following commands and files were presented in this lesson.

Relevant commands

dcu (1M)	Is the Device Configuration Utility
mouseadmin (1)	Is the mouse configuration utility
scoadmin (1M)	Invokes SCOadmin applications (in this lesson, the Video Configuration Manager)
VideoHelp (X1M)	Is the graphic adapter identification utility

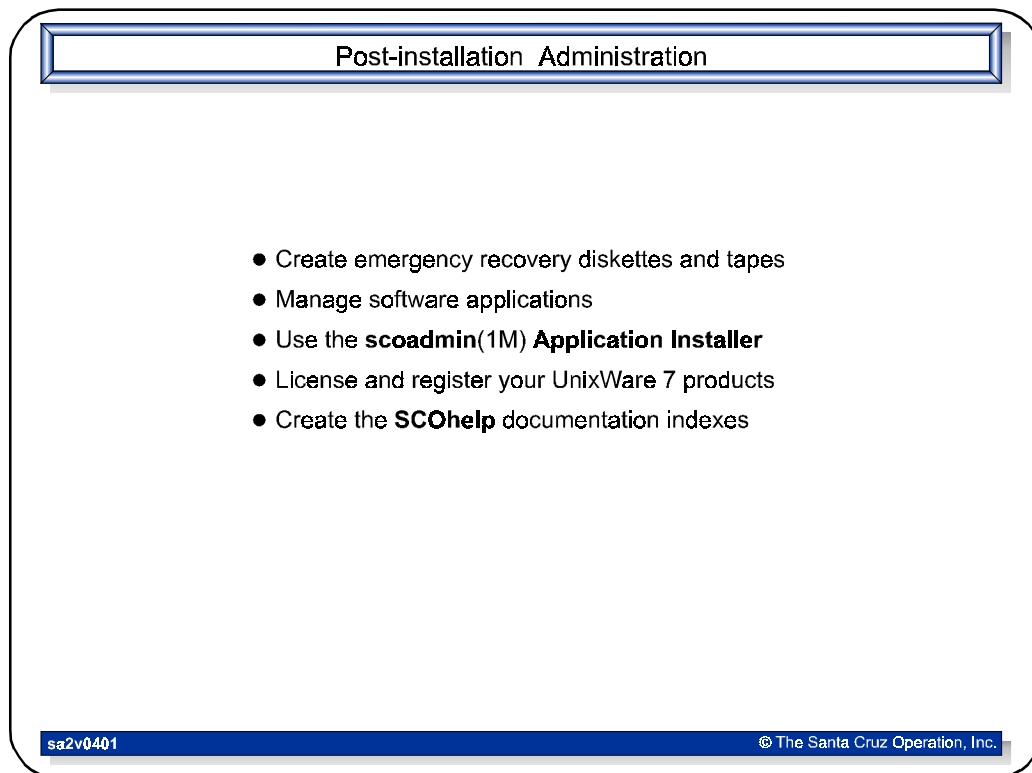
3.5 Answers to Review Questions

1. You would refer to the *UnixWare 7 Getting Started Guide and Release Notes* to resolve installation problems.
2. The two programs that you would run after the installation to install a serial mouse are **dcu(1M)** to verify the serial port settings and **mouseadmin(1)** to install and test your mouse.
3. You can run `/usr/bin/X11/VideoHelp` to help determine the type of video adapter installed on your system.

4

Post-installation Administration

AV 4-1



Lesson 4

Post-installation Administration

Introduction

The activities that you perform immediately after installation are critical.

With the emergency recovery diskettes and tapes you can solve system problems without taking the time and trouble to reinstall the complete product as well as recover your system if it should crash.

From the user's perspective, the installation of applications is extremely important.

Finally, you are required to license and register your UnixWare 7 product to stay in compliance with your License Agreement as well as to prevent recurring registration reminder messages.

Objectives

The purpose of this lesson is to identify and perform critical post-installation activities. By the end of this lesson, you will be able to:

- Create emergency recovery diskettes and tapes
- Manage software applications
- Use the **scoadmin(1M) Application Installer**
- License and register your UnixWare 7 products
- Create the **SCOhelp** documentation indexes

4.1 Creating Emergency Recovery Media

The emergency recovery media allows you to recover your system if it will not boot, your system software is corrupted beyond repair, or your hard disk has been reformatted or replaced.

After installing UnixWare 7 you should:

- Create two emergency recovery media sets (the second set allows for possible defective media)
- Test each set to make sure it works before storing them in a secure place

If you install system or application software at a later time, create new emergency recovery media.

Your emergency recovery media consists of:

- Two floppy diskettes. The emergency recovery diskettes contain a small, bootable version of the UnixWare 7 kernel from which you can boot your system.

Note: Emergency recovery diskettes are customized for a particular system and only work on that system.

- One or more tapes. The emergency tapes contain an image of:
 - Critical UnixWare 7 filesystems
 - NetWare Directory Services (NDS) master partitions, if installed
 - NetWare SYS: volume defined on your system, if installed

4.1.1 Requirements for creating emergency recovery diskettes

You can create the emergency recovery diskettes on:

- Two formatted 1.2MB 5.25-inch (96tpi, 15 sectors per track) floppy disks
- Two formatted 1.44MB 3.5-inch (135tpi, 18 sectors per track) floppy disks

4.1.2 Creating emergency recovery diskettes

Use the **emergency_disk**(1M) command to create emergency recovery diskettes by following these steps:

Step 1. Log into the system console as **root**.

Step 2. Format two floppy diskettes. For example, to format a 1.44MB floppy diskette, enter:

```
format -v /dev/rdisk/f03ht
```

Step 3. Label each diskette with the name of the system, date created, and sequence number.

Step 4. Verify that there are no active users on the system.

Step 5. Shut down the system to single-user mode.

```
shutdown -g10 -y -i1
```

Step 6. Determine in which filesystem you will create the emergency recovery media. The filesystem must have 30MB of free space. Determine how much free space is available by running:

```
/sbin/dfspace
```

Step 7. Create the first emergency recovery diskette by entering:

```
/sbin/emergency_disk -d pathname diskette
```

Where **pathname** is the name of the filesystem and **diskette** is diskette 1 or 2. For example, to create diskette1 in the */home* filesystem, enter:

```
/sbin/emergency_disk -d /home diskette1
```

Step 8. Insert a formatted disk into the floppy drive. Press .

Creating each recovery disks takes approximately fifteen minutes.

Step 9. Remove the diskette when prompted.

Step 10. Repeat Steps 8 and 9 for the second diskette.

Step 11. Store the diskettes in a safe place.

4.1.3 Creating emergency recovery tapes

Emergency recovery tapes let you restore your system and its data to the configuration stored on the media.

After creating the emergency recovery tapes, it is highly recommended that you perform periodic system backups using ARCserve/IT v6.6 from Computer Associates or **cpio**(1). Using these backups in conjunction with your emergency recovery tapes ensure that your system is restored to its most recent state.

Use the **emergency_rec**(1M) command to create your emergency tapes by following these steps:

- Step 1. Log into the system console as **root**.
- Step 2. Verify that no active users on the system.
- Step 3. Shut down the system to single-user mode.
- shutdown -g300 -y -i1**
- Step 4. Label each tape with the name of system, date created, and sequence number.
- Step 5. Insert the first tape.
- Step 6. Create the emergency recovery tapes using either Step 7 or 8.

Use Step 7 if:

- Only one hard disk is installed on your system. The entire UNIX partition is backed up, and
- The tape drive you are using has the capacity to backup your entire UNIX partition onto one tape

Use Step 8 if your system has multiple hard disks or a single disk system that does not meet the conditions in Step 7.

The **emergency_rec** command in Step 8, backs up:

- All UnixWare 7 partitions on your primary hard disk
- The */home*, */home2*, and */var* filesystems from the secondary disk (if they reside there)

If NetWare Services is installed:

- The SYS: volume of the NetWare Server
- Any master NDS partitions defined on your system

Step 7. Back up a single UNIX partition to the first tape drive. Enter:

/sbin/emergency_rec -e ctape1

Step 8. Back up multiple disks or a single disk that does not meet the criteria in Step 6, to the first tape drive. Enter:

/sbin/emergency_rec ctape1

Step 9. Remove the last tape from the drive.

Step 10. Store the emergency recovery tapes with the emergency recovery diskettes in a safe place.

After you create the emergency recovery tapes, you need to perform regularly scheduled backups. If you used Step 8, you need to specifically back up:

- Any UnixWare7 filesystems not on the primary hard disk
- The */home*, and */home2* filesystems, if they reside on a disk other than the primary or secondary disk
- Any NetWare volumes other than SYS: (including any residing on the primary hard disk)

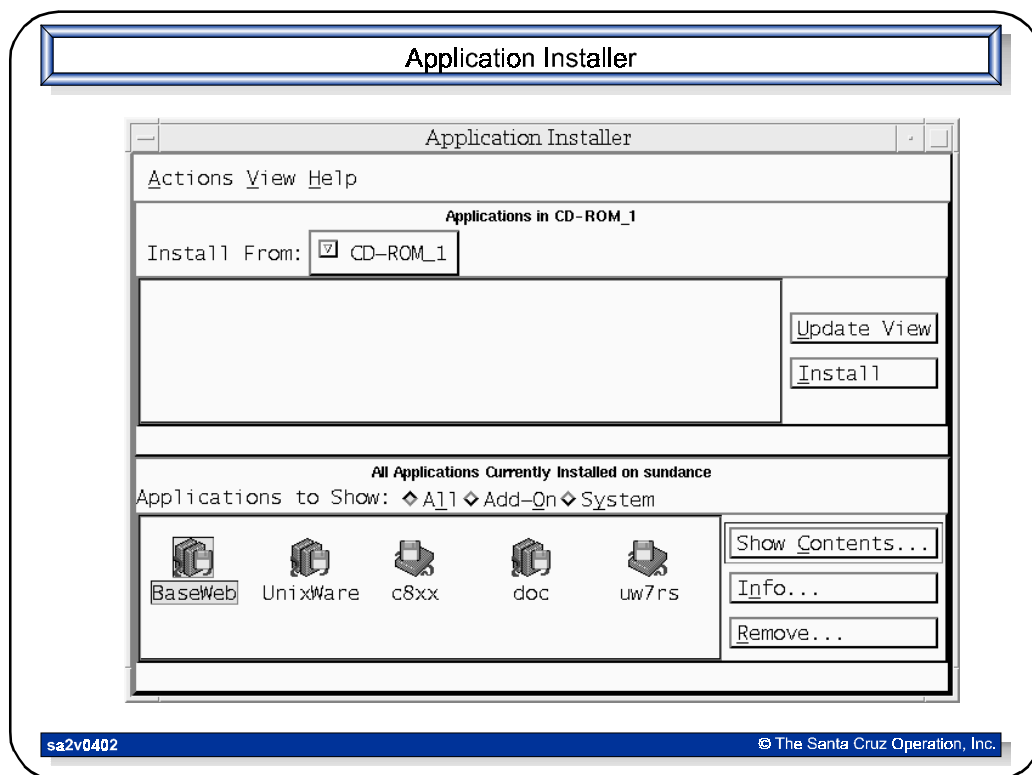
Exercise: Creating emergency recovery diskettes

Follow the steps under Section 4.1.2, “Creating emergency recovery diskettes,” on page 4-5 to create your emergency recovery diskettes.

When you are done, boot the system using the disks to verify that they work.

Remember to update your system log book.

AV 4-2



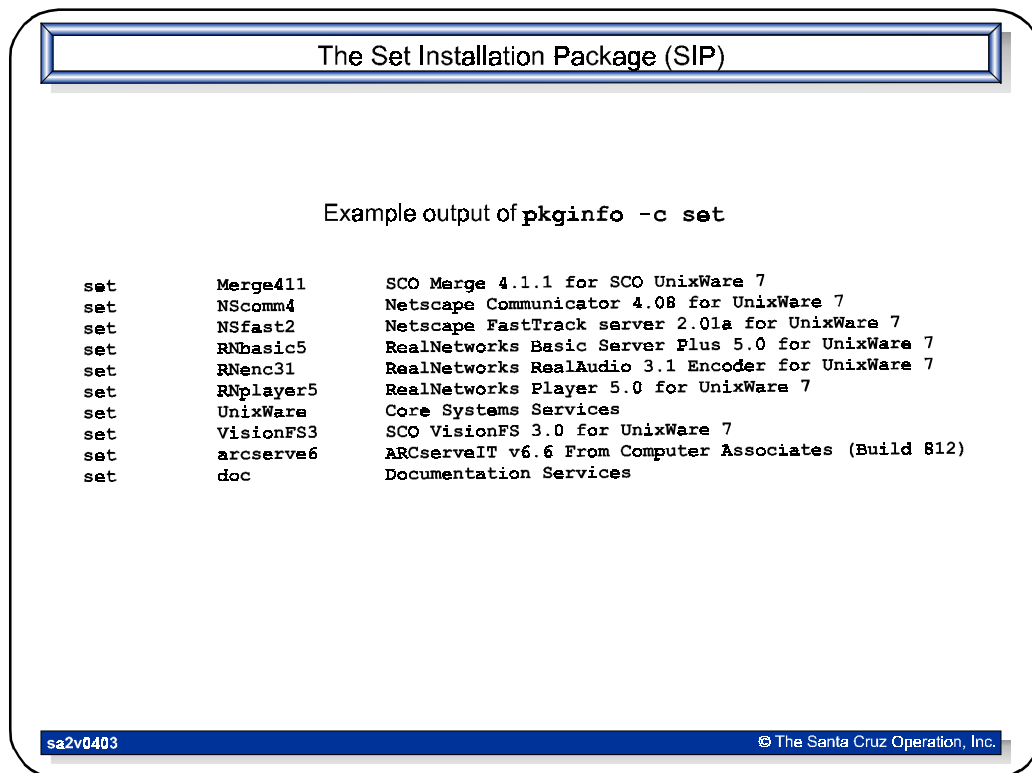
4.2 Managing Software Applications

There are two interfaces for installing, managing, and removing software applications on your UnixWare 7 system. You can use the:

- **scoadmin(1M) Application Installer** to add, view, and remove software packages in interactive mode
- **pkg*** commands in a non-interactive mode that include:
 - **pkgadd(1M)**—Adds software packages to the system
 - **pkginfo(1)**—Displays software packages
 - **pkgrm(1M)**—Removes software packages

You can install software packages from CD-ROM, disk, cartridge tape, install server, or from files stored in local or remote filesystems.

AV 4-3



4.2.1 Software management terminology

The following terminology is used to describe the system and application software that comprises your UnixWare 7 system:

- *Software package*. A software package is a collection of related files and executables that can be independently installed. For example, SCO VisionFS is a software package.
- *Set*. A set is a special-purpose package. This set is referred to as a Set Installation Package (SIP), and is a collection of one or more packages that are members of the same set. The SIP controls the installation of the set. For example, *doc* (which can be installed during installation) is a SIP.

Use the **pkginfo(1)** command to list the software packages and set contents. For example:

- **pkginfo -c set**
- **pkginfo doc**

To list the contents of a UnixWare 7 CD-ROM:

pkginfo -d cdrom1 | more

When prompted enter the UnixWare 7 Installation CD-ROM.

Exercise: Determining the type of software package

Run the **pkginfo -c set** command to determine which packages are SIPS (sets).

Run **pkginfo UnixWare** to determine which packages are members of the Core System Services SIP.

4.2.2 Package instances

Different releases of a software package can reside on your system simultaneously. Each release is known as an *instance* and is treated as a separate entity. Each entity is described by a *package identifier* (the name of the package) that is mapped to these parameters:

PKG	The package abbreviation that remains constant for every instance of the package
-----	--

VERSION The software package version

ARCH The software package architecture

You cannot install two instances of a package that have identical parameters on the same system.

4.2.3 The installation software database

Information about all packages and sets installed on your system is stored in the `/var/sadm/install/contents` file. You can use the **pkgchk**(1M) command to check the accuracy of installed files by comparing them to the contents of the `/var/sadm/install/contents` file.

Entries in this file include:

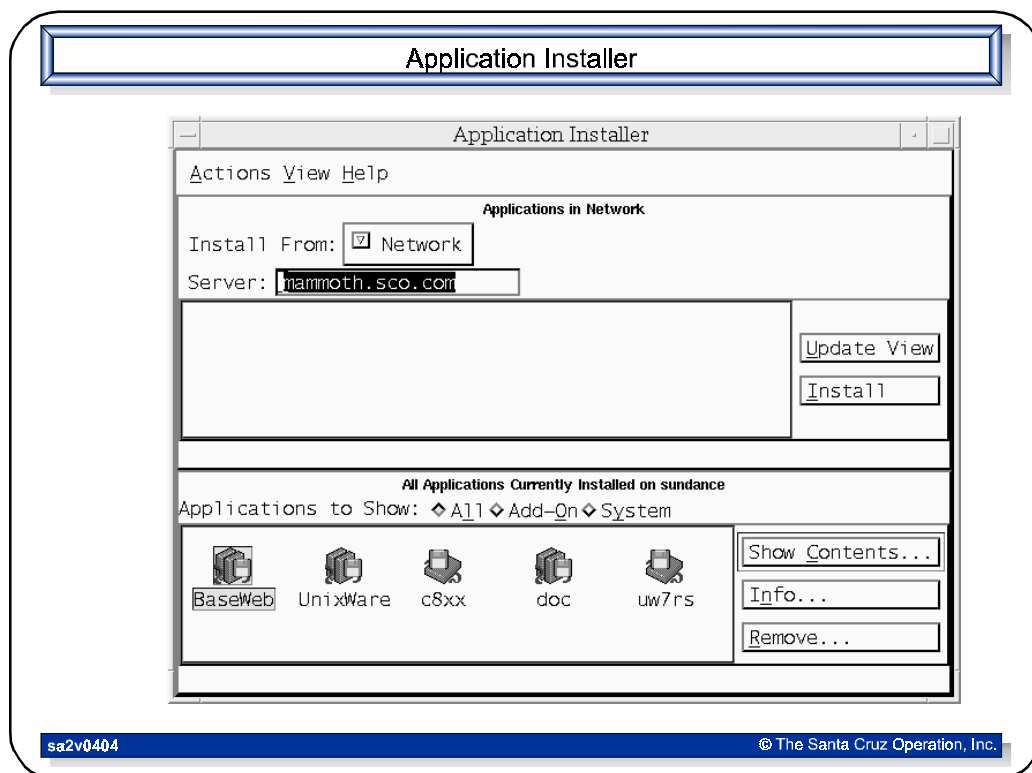
- A record of the package to which the component belongs
- The name of the component
- Where the component resides and its type
- A list of other packages that might reference the component
- Attribute information such as access permissions, ownership, file size, and time of last modification.

Refer to the **installf**(1M) command for a description of the layout of this file.

When adding or removing software, the system uses this database to determine:

- If an object is shared by more than one package
- If other packages are dependent on a package

AV 4-4



4.3 Using the *scoadmin(1M)* Application Installer

You can use the **scoadmin(1M) Application Installer** to install, remove, and check the properties of UnixWare 7 applications and application sets. You can also check third-party applications that have been installed using the **Application Installer**.

The **root** user and system owner can manage applications using the **Application Installer**.

To run the **Application Installer** from a graphical window, enter:

scoadmin application installer

or from **SCOadmin**:

Run **scoadmin(1M)** ⇨ **Software_Management** ⇨ **Application Installer**

The **Application Installer** uses a split screen:

- The top portion of your screen displays a catalog of applications contained on your removable media (if inserted) such as CD-ROM, disk, or tape. From this window you can add, remove, and check the properties of a package.
- The bottom portion of your screen displays a catalog of all the applications currently installed on your system

An application set consists of several application packages and is represented by an icon consisting of two boxes and a disk. A single application package is represented by a single box and disk. Note the red ribbon around the box(es); if it is cut, this indicates that the set or package is installed.

You can change the display of the **Application Installer** to show only:

Add-On	Applications that are not part of the operating system
System	UnixWare 7 packages
All	Add-On and System packages

Exercise: Running the Application Installer

Run the **Application Installer**. Which packages are installed on your system?

Select the doc package. Click on **Show Contents**. How many packages comprise the doc set?

4.3.1 Viewing properties and contents

You can view the properties of any application or application set installed on your system. These properties include:

Application Name	Abbreviated name of the application
Description	Full name of the application
Category	Type of application. The six (6) categories include application, graphics, system, utilities, set, and patch
Vendor	Name of company that designed the application
Version	Version number of the application
Architecture	Type of computer for which the application was written
Date Installed	Date and time the application was installed

Follow these steps to display properties using the **Application Installer**:

- Step 1. Select a package from the **Application Installer** bottom window.
- Step 2. Click on **Info**.

If the package is part of an installation set such as the **UnixWare** set, follow these steps to display individual package properties:

- Step 1. Select a package from the **Application Installer** bottom window.
- Step 2. Select **Show Contents**.
- Step 3. Select a package. Select **Properties**.

You can also view the programs that make up a package set by following these steps:

- Step 1. Select a set from the **Application Installer** bottom window.
- Step 2. Select **Show Contents**.
- Step 3. Select **Show Programs**.

Exercise: Viewing Properties

View the properties of the software packages that comprise the arcserve set.

You can also display package properties using the **pkginfo(1)** command. Display a detailed listing of all packages by entering:

```
pkginfo -l | more
```

Display a single package by entering:

```
pkginfo -l package_name
```

4.3.2 Installing packages

Use the **Application Installer** to install software packages. Follow these steps:

- Step 1. Insert the media (i.e., CD-ROM, tape, floppy diskette, etc.).
- Step 2. Run **scoadmin(1M)**.
- Step 3. Select **Software_Management**.
- Step 4. Select **Application Installer**.
- Step 5. Select the media to install from.
- Step 6. Select the package. Click on **Install**.
- Step 7. Select **Actions** ⇒ **Exit** to exit the **Application Installer**.

You can also use the **pkgadd**(1M) command to install packages or package sets. Add a package from your CD-ROM by entering:

```
pkgadd -d cdrom1 package_name
```

If you do not know the name of the package, enter:

```
pkgadd -d cdrom1
```

A listing of the packages displays. Select a number that corresponds to the package that you want to install.

4.3.3 Spooling packages

You may spool packages rather than installing them. Spooling a package copies its components from the installation media to a storage directory without invoking any installation actions, such as running installation scripts or updating the installation software database.

If you do not specify a target directory when spooling packages, the contents are copied to the default `/var/spool/pkg` directory.

Spool a package from your CD-ROM to the `/var/spool/pkg` directory by entering:

```
pkgadd -d cdrom1 -s package_name
```

Specify a different target directory by entering:

```
pkgadd -d cdrom1 -s /tmp package_name
```

You can not spool a set. You must spool the individual packages that are members of the set. Use the **pkginfo**(1) command to determine which packages are a member of a set.

You cannot spool packages using the **Application Installer**.

4.3.4 Package formats

Software packages and sets are delivered on media in one of two formats:

- *Filesystem format*—This format can only be read from block devices
- *Datastream format*—This format consists of a header and a series of **cpio** (1) archives that can be read from any raw device.

The **pkgtrans(1)** command is used to translate installable packages from one format to another. It depends on the integrity of the */etc/device.tab* file to determine whether a device can support a datastream and/or a filesystem format.

When spooling packages, if you receive the error:

```
UX:pkgadd:ERROR: unable to transfer package to filesystem format use the
-s option to transfer package in datastream format
```

Use the **pkgtrans(1)** command to transfer the package to filesystem format and copy it to the */var/spool/pkg* directory. For example, to spool the *nws* package from the CD-ROM to the */var/spool/pkg* directory, enter:

```
pktrans cdrom1 /var/spool/pkg nws
```

4.3.5 Removing packages

You can remove software packages or sets using the **Application Installer** or the **pkgrm(1M)** command. These commands can only remove software packages that were previously installed with the **Application Installer** or **pkgadd(1M)**.

Follow these steps to remove software packages using the **Application Installer**.

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **Software_Management** ⇌ **Application Installer**.
- Step 3. Select the icon that represents the software package that you want to remove.
- Step 4. Click on **Remove**.
- Step 5. Press .

The icon representing the software package or set no longer appears.

Use the **pkgrm(1M)** command to remove software packages or sets, enter:

```
pkgrm package_name
```

You can also remove multiple packages on the same command line.

If *package_name* is the name of a SIP, all packages that are members of the set, and the SIP itself, are removed.

4.4 Licensing and Registering UnixWare Products

You must license and register each SCO product. Many of the SCO products will prompt you for the license information during their installation procedure.

Licensing your SCO software is a two step process using the **scoadmin(1M) License Manager**:

1. Licensing your software
2. Registering your software

4.4.1 Licensing the software

Your license information might be included in the box by SCO or your software vendor, or a license you obtained from a web site when purchasing and downloading your software.

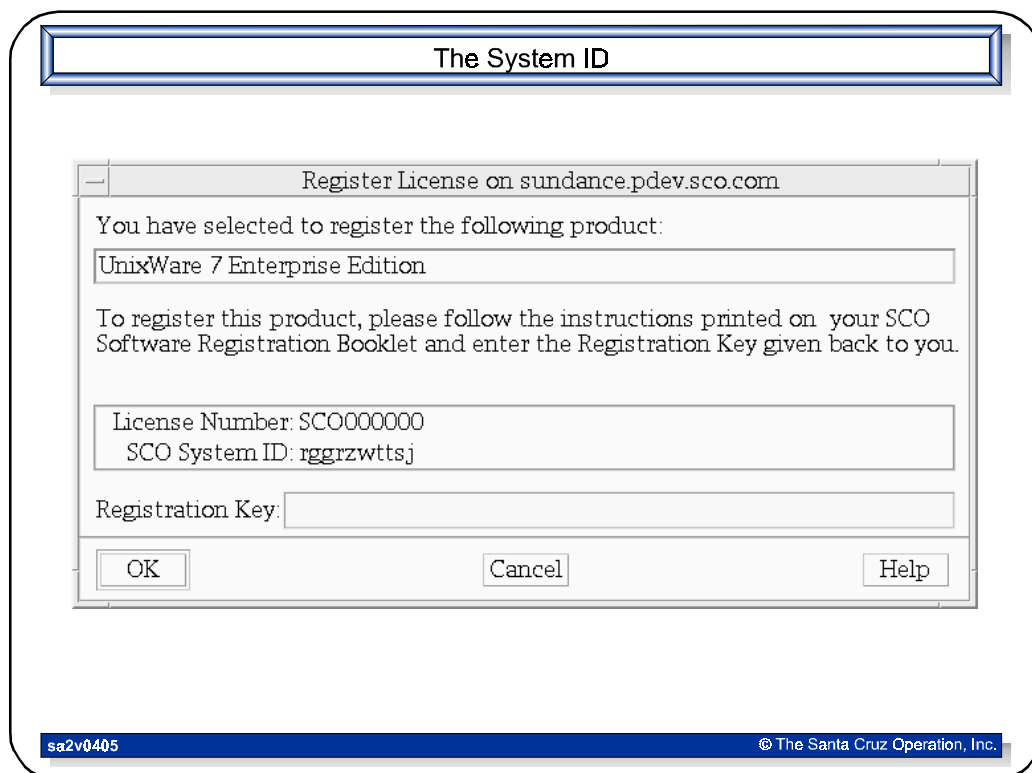
These three terms of the Certificate of License and Authenticity are your “license,” and are always used together during installation and licensing:

License Number	A unique number identifying each SCO product
License Code	A license key that activates the product
License Data	Additional license information that is needed to activate some products. It is a descriptor that further defines custom configurations. This field is not required for most products.

View licensing information for UnixWare 7 using the **License Manager**. Follow these steps to license software that is not licensed during installation:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(1M)**.
- Step 3. Select the **License Manager**.
- Step 4. Select a product for licensing.
- Step 5. Select **License** ⇒ **Add**.
- Step 6. Click on **License**. Enter the licensing information.
- Step 7. Select **Host** ⇒ **Exit** to exit the **License Manager**.

AV 4-5



4.4.2 Registering the software

After you install and license your UnixWare 7 product, register them as soon as possible. In addition to keeping you in compliance with your license agreement, this lets you purchase support contracts and future upgrades.

Two terms that are important in the process of registering your SCO product are the *System ID* and *Registration Key*.

System ID	A unique ID used to identify your UnixWare 7 installation. It is generated each time your SCO software is installed. Use the License Manager to view the System ID.
-----------	--

Registration Key	A key generated by the SCO Registration Center when you register your product with them. You will receive one Registration Key for each SCO product that your register.
------------------	---

If you perform a low-level format of your disk before reinstalling, a different System ID is generated. You must contact an SCO Registration Center to obtain a new Registration Key and repeat the registration process.

4.4.3 The registration process

After licensing your SCO products, registration reminder messages frequently display reminding you to register your software. These messages continue until the Registration Key is entered onto your system using the **License Manager**.

Before registering your software, complete the SCO Product Registration Form in the **SCO Software Registration** booklet provided with your software:

- Copy the System ID assigned to your system onto the SCO Product Registration Form
- Transfer the License Number label from the Certificate of License and Authenticity onto the SCO Product Registration Form
- Complete the information about your organization and your software supplier

4.4.3.1 Obtaining your Registration Key

You can obtain your registration key by:

- Accessing the online registration site. Point your browser to <http://www.sco.com/Products> and select the **Registration** icon. After you complete the registration, a Registration Key is generated.

This site contains links to an online registration form, as well as detailed information on how to register by email, fax, or mail.

- Fax the completed SCO Product Registration form to an SCO Registration Center using SCO's 24 hour fax service. A Registration key will be sent to you by fax within 24 hours. If SCO cannot fax the Registration key, it will be mailed to you.
- Mail the SCO Product Registration form to an SCO Registration Center

For complete instructions and a listing of the SCO Registration Centers, refer to:

scohelp ⇒ Installation and Licensing ⇒ Licensing and registering SCO products ⇒ Registering products

4.4.3.2 Using the License Manager to register

After you obtain your registration key, from the **License Manager**, enter the Registration Key by following these steps:

- Step 1. Select the product to register.
- Step 2. Select **License** ⇒ **Register**.
- Step 3. Enter the Registration Key. Click on **OK**.
- Step 4. Select **Host** ⇒ **Exit** to exit the **License Manager**.

The registration reminder messages no longer appear.

Exercise: Registering your software

Open the **License Manager** and find the System ID for the classroom system. List it below.

System ID:

Do you have a Registration Key listed? If so, what is it?

Registration Key:

4.4.4 Copy Protection Policy

A Copy Protection Policy is designed to detect duplicate serial numbers on the same network. The policy specifics are:

- At boot and every 24 hours after, every SCO system broadcasts all the serial numbers licensed to that system
- Once booted, all SCO systems listen for serial number broadcasts and compare those received with the licenses installed on it
- If an SCO system receives a serial number that is the same as one of its own serial numbers, a duplicate has been detected
- The SCO system detecting the duplicate then displays a message to */dev/console* and the */var/adm/syslog* file upon receipt of the broadcast

4.4.4.1 Duplicate message

If you have duplicate SCO serial numbers on your network, you will see a message similar to the following:

```
WARNING: Duplicate SCO serial numbers are detected on the
network. You may be violating the Software License Agreement
and Copyright Law by having illegally copied software on your
network. Please remove these duplicate copies.
```

```
Product: UnixWare 7
Serial Number: SC0123456
Host name: ahab.sco.com
IP address: 132.147.18.45
```

4.4.5 Copy Protection Policy daemons

The Copy Protection Daemon (**sco_cpd**) is launched during boot and runs continuously. If the policy manager daemon (**ifor_pmd**—which administers this policy) detects that the **sco_cpd** has been killed or tampered with, it will not grant any new licenses and will print a message to `/dev/console` and the `/var/adm/syslog` file. In this case, you will need to reboot the system.

You will see this warning message if **sco_cpd** is not running:

```
WARNING: The Copy Protection Daemon (sco_cpd) must be running
in order for your SCO software to function correctly. Please
reboot your system to restart this process.
```

4.4.5.1 Corrective user actions

If you install the same license on more than one machine, use the **License Manager** to remove the offending license and then reboot that machine.

4.5 Linux Compatibility

To an install and run Linux applications, UnixWare 7 includes a Linux package that allows you to run existing Linux binaries and benefit from the large base of Linux applications.

The **lrun** package contains the **lrun** utility and the **ld-xrun.so** shared object. Also included is the **lrunlib** package that contains Linux libraries needed to run Linux applications.

The most common Linux libraries are available in the runtime system and are stored in */usr/lib/linux*. This directory contains the *PATHMAP* file. It is used by the loaders to map Linux library calls to UnixWare 7 library calls.

4.5.1 Linux versions

Linux application support is limited to:

- Red Hat 4.2
- Rad Hat 5.1
- OpenLinux 1.2

4.5.2 Installing Linux applications

Install the Linux Run (lrun) package from the UnixWare 7 Installation *Update CD-ROM*. You can choose to install this during the initial installation or after UnixWare 7 is installed.

Important notes regarding the installation of Linux include:

- Always install the **lrun** package before installing any Linux applications.
- Some Linux applications depend on the **gtar** and **gzip** utilities being installed. These utilities are located on the Skunkware CD-ROM.
- Install Linux applications in the **LINUX_ROOT** directory, */usr/lib/linux*. This ensures that all pathnames are correctly mapped.

4.5.3 Running Linux applications

There are two ways to execute Linux binaries. The traditional way is to enter the **lrun** command followed by the binary. For example:

```
lrun hello
```

This causes the lrun loader to load the binary **hello**. This is the only way to run *a.out* format binaries.

You can also run Linux binaries using the *ld-lrun.so* loader. To do so, your Linux binary must be an *elf* format binary.

4.6 Creating the Documentation Search Indexes

After you install all software and documentation, you must create the search index using the **config_search(1M)** command. This process takes up to an hour, but the command can be run in the background. Enter the following in a terminal window:

```
/usr/man/bin/config_search -f &
```

If you are indexing English language documentation only, you can enter the following to speed the indexing process:

```
/usr/man/bin/config_search -f -L en_US.IS08859-1 &
```

4.7 Lesson Summary

You should create and test at least two sets of emergency recovery diskettes and tapes using the **emergency_disk**(1M) and the **emergency_rec**(1M) commands for each system that you install.

Use the **Application Installer** or the **pkg*** commands to view, install, or remove software on your system after the installation. The **Application Installer** cannot be run in character mode.

You may defer licensing your UnixWare 7 software during the installation and later use the **License Manager** to license the software. Obtain a Registration Key when you register your software with an SCO Registration Center.

After installation it is necessary to create the **SCOhelp** search indexes.

Review questions

1. How much free space must be remaining in the filesystem used to create the emergency recovery media?
2. Which option to the **emergency_rec**(1M) command do you use to back up a system that has a single primary UNIX partition?
3. Which command would you use to translate a software package from datastream format to filesystem format when installing a software package?
4. What takes place when you “spool” a software package rather than install the software package?
5. Describe three methods that you can use for obtaining your Registration Key from SCO.

4.8 Lab: Installing UnixWare 7 Software Packages

Overview

Use the **scoadmin(1M) Application Installer** to install an add-on software package from the CD-ROM. Your instructor will furnish you with the CD-ROM and the name of the software package..

Procedure

Use the **Application Installer** to install a software package from the CD-ROM provided by your instructor.

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(1M)**.
- Step 3. Select **Software_Management**.
- Step 4. Select **Application Installer**.
- Step 5. Insert the CD-ROM.
- Step 6. Select the media to install from.
- Step 7. Select the package. Click on **Install**.
- Step 8. Select **Actions** ⇒ **Exit** to exit the **Application Installer**.

4.9 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

cpio (1)	Copies file archives in and out
emergency_disk (1M)	Creates emergency recovery diskettes
emergency_rec (1M)	Creates emergency recovery tapes
pkgadd (1M)	Adds software package or sets to the system
pkginfo (1)	Displays software packages or sets on the system
pkgrm (1M)	Removes software packages or set from the system
pkgtrans (1)	Translates software packages format
scoadmin (1M)	Invokes SCOadmin applications (in this lesson, the Application Installer and the License Manager)

Relevant files

<i>/etc/device.tab</i>	Is used by the pkgtrans (1) command to determine whether a device can support a datastream and/or filesystem format
<i>/var/adm/syslog</i>	Contains system messages
<i>/var/sadm/install/contents</i>	Contains the software installation database
<i>/var/spool/pkg</i>	Contains spooled software packages

4.10 Answers to Review Questions

1. Thirty (30) megabytes of free space must be remaining in the filesystem used to create the emergency recovery media.
2. The `-e` option to the **emergency_rec**(1M) command is used to back up a system that has a single primary UNIX partition.
3. The **pkgtrans**(1) command is used to translate a software package from data stream format to filesystem format when installing a software package.
4. When you “spool” a software package rather than install the software package the following takes place: The software components from the installation media are copied to a storage directory without invoking any installation actions such as running installation scripts or updating the installation software database.
5. The three methods that you can use for obtaining your Registration Key from SCO are accessing the online registration site, fax, or mail.

5

Managing System Startup and Shutdown

AV 5-1

Managing System Startup and Shutdown

- Describe the default UnixWare 7 boot process
- Initiate the Boot Command Processor (BCP)
- Describe early initialization
- Initialize the multiuser and networking states
- Customize multiuser initialization
- Troubleshoot system startup
- Analyze system shutdown

sa2v0501 © The Santa Cruz Operation, Inc.

Lesson 5

Managing System Startup and Shutdown

Introduction

Starting your UnixWare 7 system requires more than just turning on the power. The system goes through a series of phases to ready the system for use. These phases include the actual boot process, choosing the operating mode, and initialization of user logins.

As a system administrator, understanding the different phases of system startup will enable you to customize your UnixWare 7 system to meet the needs of your organization.

Objectives

This lesson details the different phases of system startup. At the end of this lesson, you will be able to:

- Describe the default UnixWare 7 boot process
- Initiate the Boot Command Processor (BCP)
- Describe early initialization
- Initialize the multiuser and networking states
- Customize multiuser initialization
- Troubleshoot system startup
- Analyze system shutdown

5.1 Describing the Default Boot Process

The first phase of system startup is the boot process. The primary program that controls this initialization phase is **boot**(1M).

The **boot** program can be interactive and is used to load and execute the UnixWare 7 kernel. It can load and execute any program linked for stand-alone execution. When installed, operating systems in other disk partitions such as DOS and Windows NT can also be started from the UnixWare 7 boot prompt.

Recall that the UnixWare 7 partition is subdivided into *slices*. The two slices of interest to the boot procedure are the *boot* and *stand* slices:

- The *boot* slice is not a filesystem, but a special area of the disk that contains the boot programs. The only operation that takes place on this slice is the loading of the boot programs.
- The *stand* slice contains the bootable operating system *unix* and is a *bfs* filesystem type. It contains all of the bootable programs and data files used during the boot procedure.

5.1.1 Standard boot procedure

Before the boot process begins, the system first executes a Power-On Self-Test (POST). Once this completes, the actual booting process begins. When booting from a floppy disk, the boot procedure has two stages:

1. The boot block in sector 0 of the UNIX partition loads **boot**.
2. The **boot** program is loaded and executes the UnixWare 7 kernel.

When booting from the hard disk, the boot procedure has four stages:

1. The system firmware loads the masterboot block from absolute sector 0 on the hard disk.
2. The masterboot then loads the partition **boot** program from logical sector 0 of the active partition.
3. The remainder of **boot** is loaded from */dev/stand*, a *bfs* filesystem.
4. The **boot** program loads and executes the UnixWare 7 kernel.

As the boot program is loaded, this status message displays:

Starting UnixWare...

You can press any key to interrupt the boot process and initiate the *Boot Command Processor* (BCP). The prompt displays:

[boot]

If you do not interrupt the boot process, the boot program continues. Boot parameters are passed to **boot**(1M) from the */stand/boot* and */stand/bootmsgs* files. These files are discussed in detail later in this section.

5.2 Initiating the Boot Command Processor (BCP)

You can use the Boot Command Processor (BCP) to:

- Enter single-user state
- Load a different *unix* kernel
- Boot other operating systems such as MS-DOS and Microsoft Windows
- Display or change default boot parameters

While in the Boot Command Processor (BCP), you can list the available commands by entering **help** or **?** at the boot prompt.

5.2.1 Entering single-user state

It may be necessary to enter single-user state during the boot process to perform system maintenance or back up your system.

Perform these steps to enter single-user state during the boot process:

- Step 1. Boot your system.
- Step 2. Press  when the **Starting UnixWare...** message displays.
- Step 3. Enter these lines at the **[boot]** prompt:
- Step 4. **INITSTATE=S**
 boot
- Step 5. Enter your **root** password when you are prompted to enter single-user state.
- Step 6. Perform your task(s).
- Step 7. Continue the boot process to multiuser mode. Enter:

  **D**
- Step 8. Enter **3** when prompted for the run-level.

AV 5-2

Other Bootable Operating Systems	
DOS	Matches any DOS filesystem
DOS_12	DOS 12-bit FAT
DOS_16	DOS 16-bit FAT
DOS_32	DOS 32-bit FAT
DOS_EXT	DOS extended partition
NT	NT™
OS2	OS/2
UNIX	UnixWare 7
XENIX	XENIX operating system

sa2v0502 © The Santa Cruz Operation, Inc.

5.2.2 Loading a kernel other than the default kernel

By default, the **boot(1M)** program loads the *unix* kernel file, located in the */stand* directory. If a new kernel is created due to new hardware or software being added, the original *unix* kernel is copied to */stand/unix.old*.

If there is a problem and the new kernel does not load, you can load the *unix.old* kernel.

Load the *unix.old* kernel by performing these steps:

Step 1. Reboot your system.

Step 2. Press  when the **Starting UnixWare...** message displays.

Step 3. Verify that the *unix.old* file exists. List the contents of the */stand* directory.
Enter:

dir

Step 4. Boot the *unix.old* kernel. Enter:

BOOTPROG=unix.old
boot

5.2.3 Booting other operating systems

If you have another bootable operating system loaded into another partition on your primary hard disk such as DOS or Windows, at the boot prompt, enter:

b hd(0, DOS)

Refer to **boot(4)** for more information on booting other operating systems. Display or change default boot parameters

Boot parameters can be set or changed in the */stand/boot* file. You can display the current values while in the Boot Command Processor by entering:

show

at the **[boot]** prompt.

Exercise: Viewing parameters that are recognized by boot

View the `/stand/help.txt` file. Which option to **show** would list all of the boot parameters and messages?

View the **boot(4)** manual page for a description of these parameters and messages.

5.2.4 Describing boot parameters

There are three types of parameters that are used to boot the UnixWare 7 operating system to a usable state. They include:

- Parameters recognized by **boot(1M)** and passed to the kernel
- Read-only parameters generated by **boot**
- Parameters directly processed by the kernel
- Parameters that define boot messages

5.2.4.1 Parameters recognized by boot

Parameters recognized by **boot** can be added to or changed in the `/stand/boot` file. This information describes some of the important parameters:

BOOTPROG=filename Describes which kernel is loaded during boot. The default is `/stand/unix`. If other UNIX kernels are present in the `/stand` directory, you can change this parameter. For example, load the `/stand/unix.old` kernel, by entering:

BOOTPROG=unix.old.

BOOTDEV=device Names the device **BOOTPROG** is loaded from.

`hd(unit[partition][, slice])`

To boot DOS, enter: **b hd(0,dos).**

<code>AUTOBOOT=YES NO num</code>	Indicates whether boot (1M) loads the kernel immediately or enters an interactive boot session. The <code>num</code> field is the minimum time (in seconds) to wait while loading <code>BOOTPROG</code> . The default is five (5) seconds.
<code>TIMEOUT=number</code>	Specifies the number of <i>seconds</i> the boot program waits for input from the user before loading the kernel. The default is 0 seconds.
<code>CONSOLE=device</code>	Overrides the default configuration of the console device. By default this parameter is set to a PC console: <code>CONSOLE=kd (0)</code> You can change this parameter to indicate a terminal attached to a COM port. For example, to change this parameter to COM2, using no flow control, at 38400 baud, with no parity checking, and 1 stop bit, enter: <code>CONSOLE=iasy(5, B38400)</code> Note that you can only change the <code>CONSOLE</code> parameter to COM1 or COM2.

5.2.4.2 Parameters that are processed by the kernel

The following information describes some of the important parameters that are passed directly to the kernel and not recognized by **boot**:

<code>DISABLE_CACHE=(YES/NO)</code>	Setting this parameter to YES disables the use of internal and external memory caches on i486 and above CPUs.
<code>ENABLE_4GB_MEM=(YES/NO)</code>	Setting this parameter to YES enables detection of memory above 4GB.
<code>IGNORE_MACHINE_CHECK=(YES/NO)</code>	Setting this parameter to YES ignores hardware check conditions. This parameter allows invalid machine checks to be ignored rather than disabling the system.

INITSTATE=sysstate	Defines the parameter to specify the system state to enter after booting. Recall that entering INITSTATE=S at the boot prompt, booted the system to single-user mode. System states are discussed in detail later in this section.
--------------------	---

For a complete listing of parameters that are processed by the kernel, refer to **boot(4)**.

5.2.4.3 Parameters that define boot messages

The following information describes some of the important parameters that let you override default messages that are stored in */stand/bootmsgs*.

AUTOMSG=string	Changes the default message Automatic Boot Procedure when an automatic reboot is about to occur.
BOOTMSG1=string	Changes the default boot message Starting UnixWare... displayed during the booting of UnixWare 7.
BOOTMSG2=string	Changes the default string Bootstrap Command Processor Ready for boot commands... [? for help] that is used to indicate the beginning of an interactive boot session.

For a complete listing of boot messages and strings than can be changed, refer to **boot(4)**.

In addition to changing boot messages, you can also change other informational messages, including the:

COPYRIGHT=string	Changes the default copyright line
DEVICENAMES=string	Changes the listing of the devices attached to a SCSI bus
REBOOTMSG=string	Changes the Press any key to reboot ... message
STARTUPMSG=string	Changes the The system is coming up. Please wait... message

For a complete listing of boot messages and strings than can be changed, refer to **boot(4)**.

Exercise: Describing fields in the */stand/bootmsgs* file

Which parameter in the */stand/bootmsgs* file displays the amount of memory detected during system startup?

5.3 Describing Early Initialization

After the operating system is loaded into primary memory using the boot programs, the **init**(1M) daemon is created. The **init** daemon immediately scans the */etc/inittab* file and performs the following:

- Sets the default or initial *run-level* of your system
- Checks and mounts the root filesystem
- Mounts all other non-root filesystems
- Prepares login sessions
- Starts file and print sharing if applicable networking packages are installed
- Initializes processes that can be delayed until after the login sessions are started

5.3.1 Run-levels

A run-level (also referred to as a system state) is a system software configuration under which only a selected group of processes exist. At any given time the system can be in one of eight possible run-levels:

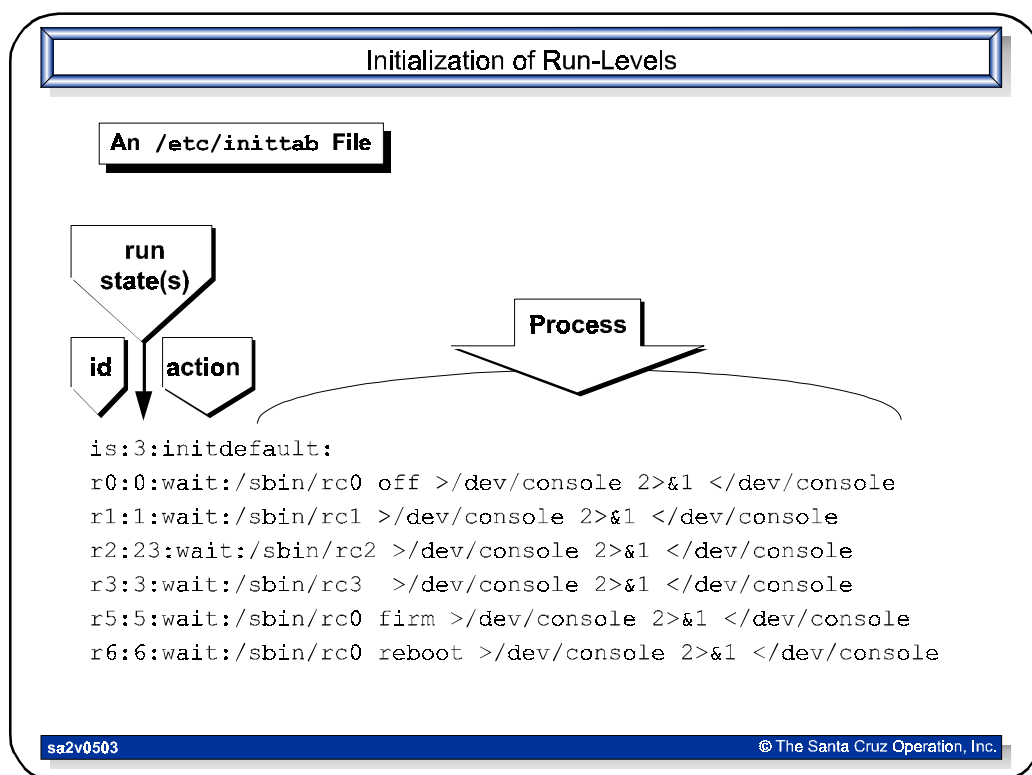
- | | |
|---|---|
| 0 | Shutdown; it is safe to remove power. |
| 1 | Administrative; all filesystems are mounted. Only a small set of essential kernel processes run. |
| 2 | Multiuser; all filesystems are mounted and all processes are running |
| 3 | Extended multiuser or networking state; start the remote file sharing processes and daemons. Mount and advertise remote daemons. This is the default run-level. |
| 4 | An alternative multiuser environment that can be defined. This run-level is not necessary for normal system operations and is not used. |
| 5 | Shutdown with the option to “Hit any key to reboot.” |
| 6 | Shutdown and reboot |

- s and S Single-user mode; note that the set of filesystems mounted and the list of processes killed when a system enters single-user state are not always the same. When the system is:
- Brought up to single-user state using **init(1M)**, the only filesystems mounted are */stand*, *root*, */var*, */proc*, and */dev/fd*. User filesystems are not mounted. Only essential kernel processes are kept running.
 - Brought down to single user state using **init**, all mounted filesystems remain mounted and all processes started by **init** in multiuser state are killed.
 - Changed to single-user state using the **shutdown(1M)** command, the system is restored to the state in which it was running when the system was initially booted.
 - All processes spawned by **init** are terminated. When entered from other than the console, system states s,S also reassign the system console. Refer to the **init(1M)** manual page for more information on reassigning the console device.

The **init** system states a, b, and c are pseudo-states that can be defined to run certain commands; however, they do not cause the current run-level to change.

The **init** state q,Q is used to re-examine */etc/inittab* if any changes have been made to the file.

AV 5-3



5.3.2 The */etc/inittab* file

The */etc/inittab* file specifies the programs that **init** uses for system initialization. It also defines at what run-level certain programs can exist, and begins the login process at the specified run-level.

The */etc/inittab* file contains four fields:

<code>label</code>	A unique identifier of up to four characters
<code>run-level</code>	The system state in which init executes an entry
<code>action</code>	A keyword indicating the action to be taken
<code>process</code>	The program that init executes upon entering the specified run-level

AV 5-4

Inittab(4) Action Definitions	
respawn	If the process does not exist, start the process, and when it dies restart the process.
wait	Upon entering the matching run-level, start the process and wait for its termination.
once	Upon entering the matching run-level, start the process; do not wait for its termination. When it dies, do not restart the process.
boot	The entry is to be processed only at boot-time read of the file.
bootwait	The entry is to be processed the first time when the system goes from single-user to multiuser state after the system is booted.

sa2v0504 © The Santa Cruz Operation, Inc.

AV 5-5

More Inittab(4) Action Definitions	
powerfail	Execute the process associated with this entry only when it receives a power fail signal.
powerwait	Execute the process associated with this entry only when it receives a power fail signal and wait until it terminates before continuing.
off	Ignore this process.
ondemand	This is the same as respawn, used with run-levels a, b, c and e.
initdefault	This is used to determine the run-level.
sysinit	Entries are executed before init(1) accesses the console.

sa2v0505 © The Santa Cruz Operation, Inc.

The action field in */etc/inittab* tells **init**(1M) when and how the program executes. The process field tells **init** which program it has to execute.

The action **sysinit** tells **init** to start the program the first time **init** reads */etc/inittab*, and wait for its termination before executing the next entry. Because there is no run state associated with the action **sysinit**, the entry is executed before **init** tries to access the console.

The action **bootwait** tells **init** to execute the program when initially going from single to multiuser state, wait for its termination, and not restart the program.

When a terminal is enabled for login, **respawn** appears in the action field. If a program does not exist, the action **respawn** tells **init** to start the program, and when it terminates, to restart the program.

Refer to the **inittab**(4) manual page for a detailed description of action definitions.

Exercise: Determining at what run-levels a program is executed

Use **grep**(1) to search for the word *sac* in the */etc/inittab* file. Determine in which run-levels the Service Access Controller (**sac**(1M)) is started.

5.3.3 Entering the default run-level

When you power up or reboot your system, the system enters the default run-level that appears in the **initdefault** entry of the */etc/inittab* file.

The run-level:

- | | |
|---|---|
| 3 | Indicates that Network File System (NFS) is installed and the system is put into networking state. |
| 2 | Indicates that the NetWork File System (NFS) is not installed and the system is put into multiuser state. |

The **initdefault** entry appears as:

is:3:initdefault:

5.3.4 Changing system run-levels

When a system run-level changes, any program that was started from */etc/inittab* that does not have an entry for the new run-level receives a terminate signal (15). If the process does not terminate in a few seconds, it receives a kill signal (9).

If you are using the **init**(1M) command to go from the networking state to a single-user or shutdown state, warnings are not issued to the users.

Use the **wall**(1M) command to warn your users before issuing the **init** command. For example:

```
wall   
System will shut down in 15 minutes  
 d
```

To change to single-user state, enter:

```
init 1
```

The **init** program terminates any processes that were started from */etc/inittab* at run-level 3.

Exercise: Using the **wall**(1M) command

Send the message, “System will shut down in 15 minutes” using the **wall**(1M) command.

You can also use the **shutdown**(1M) command and specify a new system state. For example, to enter single-user mode in five (5) minutes, enter:

```
shutdown -g300 -y -i1
```

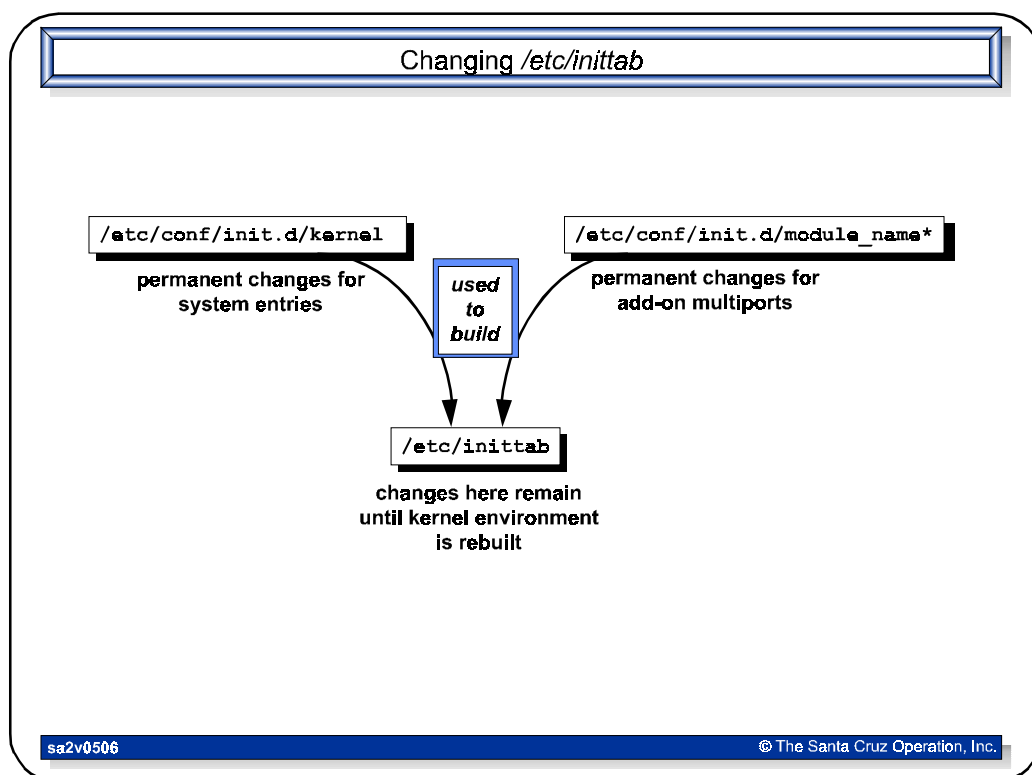
Determine your current system state, by using the **-r** option to the **who**(1) command. Enter:

```
who -r
```

Exercise: Determining your current system state

Enter **who -r** to determine your current system state. Access the online manual pages and explain the output of this command.

AV 5-6



5.3.5 Changing the */etc/inittab* file

There are times when entries in the */etc/inittab* file need to be changed. For example, you may want your console login prompt to display differently or if you do not have Network File Systems (NFS) installed, you may want to change the `initdefault` entry to multiuser state (2).

When making changes, do not edit the */etc/inittab* file. This file is recreated from the following files each time a new kernel is rebuilt. Make changes to the files:

/etc/conf/init.d/kernel Contains entries that are associated with system startup and the console.

/etc/conf/init.d/module_name Contains entries associated with any module's Driver Software Package (DSP)

For example, to change the Console login prompt entry in the */etc/conf/init.d/kernel* file, perform these steps:

- Step 1. Copy the */etc/conf/init.d/kernel* file to */etc/conf/init.d/kernel.safe*.
- Step 2. Edit the */etc/conf/init.d/kernel* file and change the words **Console Login** in this entry:

```
co:12345:respawn: /usr/lib/saf/ttymon -g -p
"Console Login:" -d /dev/console -l console
```

- Step 3. Save your changes.
- Step 4. Create a new *inittab* file in the */etc/conf/cf.d* directory. Enter:

```
/etc/conf/bin/idmkinitt
```

- Step 5. Copy */etc/conf/cf.d/inittab* to */etc/inittab*.

Exercise: Listing the files that make up */etc/conf/init.d* directory

How many files are in the */etc/conf/init.d* directory?

5.4 Initializing the Multiuser and Networking States

Recall that run-level 2 (multiuser state) is the operating state of the system when network file sharing is not installed. Users can:

- Log into the system
- Use system resources simultaneously

In run-level 3 (the networking state) in addition to starting all run-level 2 processes, the system:

- Starts file sharing and connects your computer to a file-sharing network
- Advertises your resources to remote computers
- Mounts remote resources on your computer

5.4.1 Multiuser state (run-level 2)

When the default run-level is 2, two entries in */etc/inittab* instruct **init**(M) to initialize multiuser state. They include:

- `r2:23:wait:/sbin/rc2 > /dev/console ...` This entry instructs **init** to execute the files stored in the */etc/rc2.d* directory.
- `d2:23:wait:/sbin/dinit >/dev/console ...` This entry instructs **init** to execute the files in */etc/dinit.d* directory after the login process.

Initializing run-level 2 also starts the Service Access Facility (SAF) that manages access to your system through ports and other communication devices. The Service Access Facility (SAF) is discussed in detail in Lesson 9, *Installing and Managing Serial Ports*.

AV 5-7

The /etc/rc2.d Directory

```

total 136
-r--r--r-- 5 root sys 2668 Feb 3 13:23 K20nfs
-r--r--r-- 3 root sys 873 Feb 3 13:20 S01MOUNTFSYS
-r--r--r-- 2 root sys 222 Feb 3 13:20 S02PRESERVE
-r--r--r-- 3 root audit 1137 Feb 3 13:22 S02audit
-r--r--r-- 4 root sys 879 Feb 3 13:20 S02mse
-r--r--r-- 2 root sys 1795 Feb 3 13:20 S05RMTMPFILES
-rwxr-xr-x 1 root sys 757 Feb 3 13:20 S15mkdtab
-r-xr-xr-x 5 root sys 5392 Feb 3 13:19 S15nd
-rw-r--r-- 3 bin bin 206 Feb 3 13:21 S42ls
-r--r--r-- 2 root sys 282 Feb 3 13:19 S65loopback
-r--r--r-- 1 bin bin 667 Dec 5 15:17 S69ARCserve@
-r--r--r-- 4 root sys 4381 Feb 3 13:23 S69inet
-r--r--r-- 4 root sys 462 Feb 3 13:25 S70Nnwip
-r--r--r-- 4 root sys 8279 Feb 3 13:25 S70Pnw
-r--r--r-- 4 root sys 2764 Feb 3 13:25 S70Rnuc
-rwxr-xr-x 1 root sys 1260 Feb 3 13:20 S70pf
-r--r--r-- 2 root sys 939 Feb 10 05:09 S70unixtsa
-rwxr-xr-x 1 root sys 193 Feb 3 13:23 S71ppp@
-r-x----- 3 root sys 2847 Feb 3 13:25 S73snmp
-r--r--r-- 2 root sys 1450 Feb 3 13:19 S75rpc
-r-xr--r-- 2 root sys 574 Feb 3 13:19 S79sr
-r--r--r-- 4 root sys 2064 Feb 3 13:22 S80nis
-r-xr-xr-x 1 bin bin 1460 Feb 3 13:20 S81sendmail@
-rwxrwxr-x 4 root sys 408 Feb 3 13:25 S88ldap
-rwxr-xr-x 2 root sys 1576 Jan 16 21:12 S90sysinfo2html
-rwxr-xr-x 1 root bin 855 Feb 3 12:33 S93scohelphttp
-r-xr-xr-x 2 root sys 2494 Feb 3 13:21 S95vxvm-recover
-r-xr-xr-x 4 root sys 3486 Feb 10 05:19 S99dtlogin

```

sa2v0507

© The Santa Cruz Operation, Inc.

5.4.1.1 Analyzing the */etc/rc2.d* scripts

The */etc/rc2.d* directory contains default startup scripts for several programs. Scripts are also included that clean up temporary and log files.

This is a partial listing of important scripts executed by **init**(1M) when the system enters multiuser mode.

<i>S01MOUNTFSYS</i>	Mounts non-root filesystems
<i>S02audit</i>	Enables system auditing
<i>S05RMTMPFILES</i>	Removes temporary files
<i>S69inet</i>	Starts TCP/IP
<i>S70Pnw</i>	Starts NetWare services
<i>S93scohelphttp</i>	Starts the SCOhelp daemon
<i>S99dtlogin</i>	Starts the graphical login

5.4.1.2 Analyzing the */etc/dinit.d* scripts

Recall that */etc/dinit* is executed from */etc/inittab*. The */etc/dinit* script is responsible for completing portions of initializations that were traditionally performed by */etc/rc2*. Files are executed from the */etc/dinit.d* directory after the login process starts. Doing so results in faster performance from startup to multiuser state.

Programs started by */etc/dinit* include:

<i>S70uucp</i>	Cleans up uucp locks, status, and temporary files
<i>S75cron</i>	Starts the cron (1M) daemon
<i>S80lp</i>	Starts the lpsched (1M) daemon

5.4.2 The networking state (run-level 3)

When the default run-level is 3, in addition to executing */sbin/rc2* and */etc/dinit*, the */etc/rc3* script is also executed from this entry in */etc/inittab*:

```
r3:3:wait:/sbin/rc3 >/dev/console ...
```

The */sbin/rc3* files executes the */etc/rc3.d/S22nfs* and *S90visionfs* files that start the network file sharing services.

5.5 Customizing Multiuser Initialization

It may be necessary to add your own initialization procedures in the `/etc/rc2.d` directory or edit an existing script. If you do so, there are certain rules that you should follow:

- Do not place an entry in `/etc/inittab` for your script. Instead, place the script file in the `/etc/rc2.d` directory where it will automatically be executed with the other files.
- Test the script that performs the desired function from the command line
- Name the file so that it begins with the uppercase letter *S* or *K*. The *S* or *K* indicates whether the process should be started (*S*) or killed (*K*) upon entering the new system states.

5.5.1 How scripts are executed from `/etc/rc2.d`

When creating your own scripts, it is also important to know that:

- The order in which files are executed is important:
 - The first character must be *S* or *K*
 - Characters 2 and 3 must be digits and cause the files to be executed in order from the lowest number to the highest number, i.e., *S01xxx* is executed before *S02xxx*.
- When entering run-level 2, files that begin with *S* are executed with the *start* option
- Files are started in ASCII sorted order
- When entering run-level 0, scripts that begin with *K* for Kill are executed with the *stop* option. *K* files are found in the `/etc/rc0.d` directory with the exception of the *K20nfs* file which is in the `/etc/rc2.d` directory.
- Files that begin with characters other than *S* or *K* are ignored

Refer to **rc2(1M)** for more information on the executing of **rc** scripts.

5.5.2 Executing commands at system startup

In addition to writing customized startup scripts, individual commands can be started in the background during multiuser initialization.

To execute a command during multiuser initialization, create the file */etc/rc2.d/S89USRDEFINE*.

Follow these steps to add the command:

- Step 1. Create the */etc/rc2.d/S89USRDEFINE* file.
- Step 2. Add the following to the end of the script:


```
command &
```

(where *command* = the command name)
- Step 3. Save your changes and quit.
- Step 4. Activate the command by shutting down your system and rebooting.

5.5.3 Defining other run-level initialization scripts

When you switch to single-user state, **init**(1M) scans the */etc/inittab* file for the run-level and takes the following actions:

- When changing to single-user mode, **init** scans for the entry:

```
r1:1:wait:/sbin/rc1 > /dev/console 2>&1 </dev/console
```

The *sbin/rc1* file contains instructions to execute the scripts stored in the */etc/rc1.d* directory.

- When you shut down the system, **init** scans for the entry:

```
r0:0:wait:/sbin/rc0 off > /dev/console 2>&1 < /dev/console
```

The */sbin/rc0* file contains instructions to execute the scripts stored in the */etc/rc0.d* directory.

5.5.4 Deactivating rc scripts

To deactivate any script found in the */etc/rc[0,2 or 3].d* directories, SCO recommends that you rename the file so that it begins with a lowercase letter. Files that do not follow the S, and K naming convention will not execute. Do not remove the file.

5.6 Troubleshooting System Startup

This section discusses reasons why a system that has booted successfully in the past may not boot. These situations usually occur as a result of a power failure or system panic that corrupts the boot or root filesystem. Configuration changes, hardware failure, and human error can also cause these situations.

If you cannot boot your system, check the following:

- Are any cables loose, disconnected, or improperly connected?
- Are the cables terminated properly?
- Does the floppy drive contain a floppy disk that is not a boot disk?
- Has your hard disk developed a bad track or a bad block?

Refer to your *UnixWare 7 Getting Started Guide and Release Notes* for solutions to specific system boot problems.

Recovering an unusable system is covered in the *Disaster Recovery* lesson.

AV 5-8

The /etc/rc0.d Directory

The /etc/rc0.d Directory

total 68

-r--r--r--	3	root	sys	761	Feb	3	10:20	K00ANNOUNCE
-r--r--r--	3	root	audit	1137	Feb	3	10:22	K02audit
-r--r--r--	4	root	sys	879	Feb	3	10:20	K02mse
-r-xr-xr-x	4	root	sys	3486	Mar	5	07:12	K10dtlogin
---x-----	4	root	sys	562	Feb	3	10:20	K20lp
-r--r--r--	5	root	sys	2668	Feb	3	10:23	K40nfs
-rwxr-xr-x	1	root	sys	193	Feb	3	10:23	K40ppp@
-rwxrwxr-x	4	root	sys	408	Feb	3	10:25	K44ldap
-r-xr--r--	4	root	sys	3121	Feb	3	10:19	K68netbios
-r-xr-xr-x	1	bin	bin	1460	Feb	3	10:20	K68sendmail@
-r--r--r--	4	root	sys	4381	Feb	3	10:23	K69inet
-r--r--r--	4	root	sys	2064	Feb	3	10:22	K80nis
-r-xr-xr-x	5	root	sys	5392	Feb	3	10:19	K85nd

sa2v0508

© The Santa Cruz Operation, Inc.

5.7 Shutting Down the System

The primary function of the **shutdown**(1M) command is to terminate all currently running processes in an orderly and cautious manner. This command also:

- Allows an interactive shutdown
- Sends a broadcast message, warning users logged on the system to log off
- Allows only **root**, **system owner**, or a user with the *shutdown* role to execute **shutdown**(1M). The */sbin/shutdown* script checks to see if **root** or an authorized user is performing the command. If not, the shut down does not proceed.
- Runs **init**(1M)

5.7.1 Analyzing shutdown

When performing the **shutdown**(1M) command, the actual change of run-levels and killing of processes is performed by **init**(M). The default instructions for **init** are supplied from the following entry in */etc/inittab*:

```
r0:0:wait:/sbin/rc0 off 1> /dev/console ...
```

The */sbin/rc0* file performs the following:

- Echoes that the system is coming down
- Runs scripts stored in the */etc/rc0.d* directory
- Kills off all processes
- Unmounts all filesystems

5.8 Lesson Summary

You can customize the default UNIX boot process by editing the parameters in */stand/boot*. Multiuser initialization also can be customized by adding to, or editing scripts stored in the */etc/rc2.d* directory. The */etc/rc0* file points to the scripts stored in the */etc/rc0.d* directory to ensure an orderly shutdown of your UnixWare 7 system.

Review Questions

1. Which two slices of the UNIX partition are used to boot UnixWare 7?
2. What tasks can you perform using the Boot Command Processor (BCP)?
3. Which boot files are accessed by the **boot(1M)** program during system startup?
4. Describe some of the purposes of the */etc/inittab* file.
5. Which directory contains the default startup scripts for several programs used to initialize multiuser mode?
6. What do the letters S or K indicate at the beginning of each initialization script stored in */etc/rc2.d* directory?
7. When performing a shutdown, what does the */etc/rc0* file accomplish?

5.9 Lab: Creating an */etc/rc2.d* Script

Overview

Create an executable script in the */etc/rc2.d* directory named *S02awelcome*. This script will banner the word “WELCOME” to the console during system startup.

Procedure

- | | |
|---------|---|
| Step 1. | Log in as root . |
| Step 2. | Change to your student home directory. |
| Step 3. | Create a file named <i>S02awelcome</i> . |
| Step 4. | Enter the following line:

banner WELCOME |
| Step 5. | Save your file. |
| Step 6. | Copy the file to the <i>/etc/rc2.d</i> directory. |
| Step 7. | Change permissions so that the file is read and executable by root . |
| Step 8. | Test the results by shutting down and restarting your system. |

5.10 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

boot (1M)	Loads the UNIX system
init (1M)	Changes run-levels
shutdown (1M)	Terminates all processes gradually
wall (1M)	Sends a message to all users currently logged in
who (1)	Lists the users currently logged in to the system; with the -r option, indicates the current run-level of the init (1M) process

Relevant files

<i>/stand/boot</i>	Contains default boot parameters
<i>/etc/inittab</i>	Contains instructions for init (1M)
<i>/etc/rc0.d</i>	Scripts executed when entering single-user mode and state of shutdown
<i>/etc/rc2.d</i>	Scripts executed when entering multiuser mode
<i>/etc/rc3.d</i>	Scripts executed when entering the networking state
<i>/etc/shutdown</i>	Script that shuts down the system in an orderly fashion

5.11 Answers to Review Questions

1. The two slices of the UNIX partition that are used to boot the UnixWare 7 operating system are the boot and stand slices.
2. The tasks that you can perform using the Boot Command Processor (BCP) include entering single-user state, loading a different UNIX kernel, booting other operating systems, and displaying or changing default boot parameters.
3. The `/stand/boot` **boot(4)** file and `/stand/bootmsgs` are accessed by the **boot(1M)** program during system startup.
4. The `/etc/inittab` file specifies the programs that **init(1M)** uses for system initialization. It also defines at what run-levels certain programs can exist, and begins the login process at the specified run-level.
5. The `/etc/rc2.d` directory contains the default startup scripts for several programs that are used to initialize multiuser mode.
6. The letters S or K at the beginning of each initialization script stored in the `/etc/rc2.d` directory, indicate whether the process should be started (S) or killed (K) upon entering the new system state.
7. During a shutdown, the `/etc/rc0` file echoes that the system is coming down, runs scripts

6

Filesystem Configuration and Disk Management

AV 6-1

Filesystem Configuration and Disk Management

- Describe the UNIX filesystem structure
- Manage the `/dev` directory
- Describe important vxfs filesystem features
- Prepare the filesystem for access
- Manage filesystems
- Add large file support
- Check and repair filesystems
- Configure additional hard disks
- Manage the Volume Table of Contents (VTOC)
- Troubleshoot hard disks

sa2v0601 © The Santa Cruz Operation, Inc.

Lesson 6

Filesystem Configuration and Disk Management

Introduction

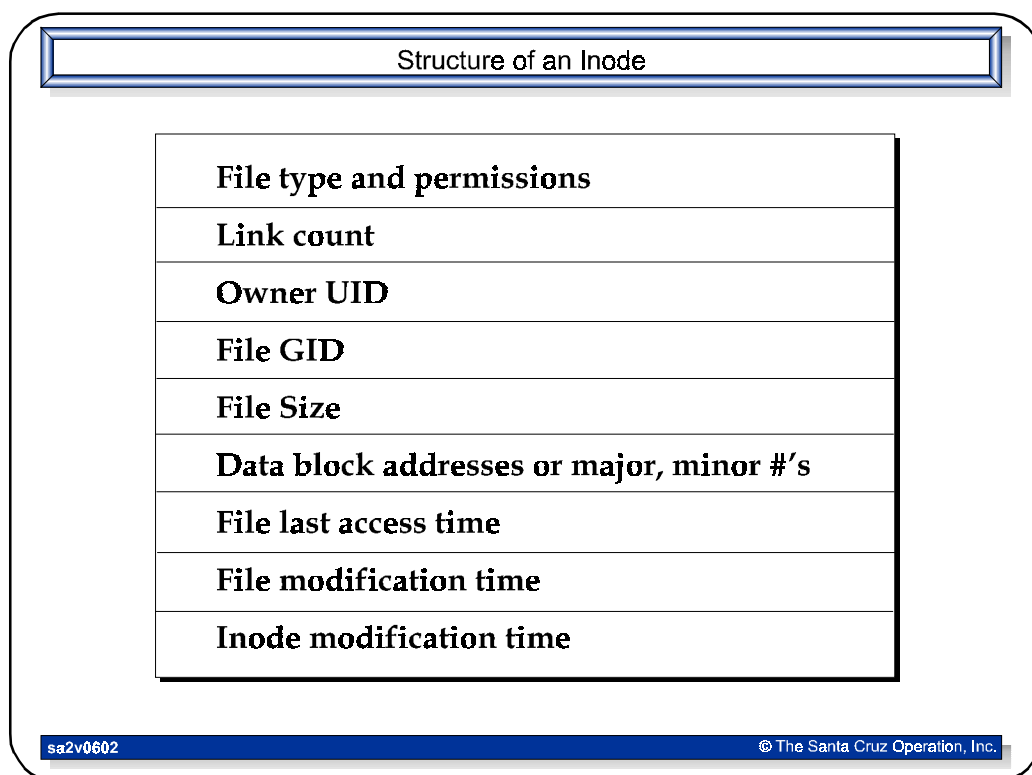
As the system administrator, you are responsible for the integrity and security of user data. Once you understand the structure of the UNIX filesystem, you can configure, maintain, and troubleshoot filesystems and hard disks.

Objectives

This lesson provides an in-depth discussion of the *vxfs* filesystem and special device files. It also describes the steps and commands you use to configure and troubleshoot hard disks and filesystems. By the end of this lesson, you will be able to:

- Describe the UNIX filesystem structure
- Manage the */dev* directory
- Describe important *vxfs* filesystem features
- Prepare the filesystem for access
- Manage filesystems
- Add largefile support
- Check and repair filesystems
- Configure additional hard disks
- Manage the Volume Table of Contents (VTOC)
- Troubleshoot hard disks

AV 6-2



6.1 Describing the UNIX Filesystem Structure

UnixWare 7 filesystems contain special structures that allow the operating system to access and maintain the files and data stored on the filesystem. These structures include data blocks, inodes, and a superblock.

6.1.1 Data blocks

A block is a unit of data stored on the disk. Some filesystems use variable block sizes to maximize the use of space. A data block can contain either directory entries or file data. A directory entry consists of an inode number and a filename.

6.1.2 Inodes

The filesystem code in the kernel accesses files by their inode numbers. When a file is created, the system assigns an inode number to that file. Inodes are *information nodes* that contain the following information about a file:

- Mode (file type and permissions)
- Link count (number of names used to access this file)
- Owner User ID (UID) and Group ID (GID)
- Size in bytes
- Pointers to disk addresses, hardware devices, and pathnames
- Access time (last time the file was opened for reading)
- Modification time (last time the file was opened for writing)
- Change time (last time the inode was changed)

Users access files by their filenames. Filenames are stored in directory files along with their associated inode number.

6.1.2.1 Reserved inodes

The system uses:

- Inode 0 to indicate deleted files and directories

- Inode 1 is not used
- Inode 2 (the root directory) as the top level of each mounted filesystem's hierarchical structure

6.1.2.2 Viewing inode numbers

You can display inode numbers associated with filenames. Enter the **-i** option with the **ls(1)** command.

You can also display inode numbers associated with directory files. Enter the **-ld** options with the **ls(1)** command.

Exercise: Associating file names with inode numbers

Determine the inode number of the root (/) and */stand* directories.

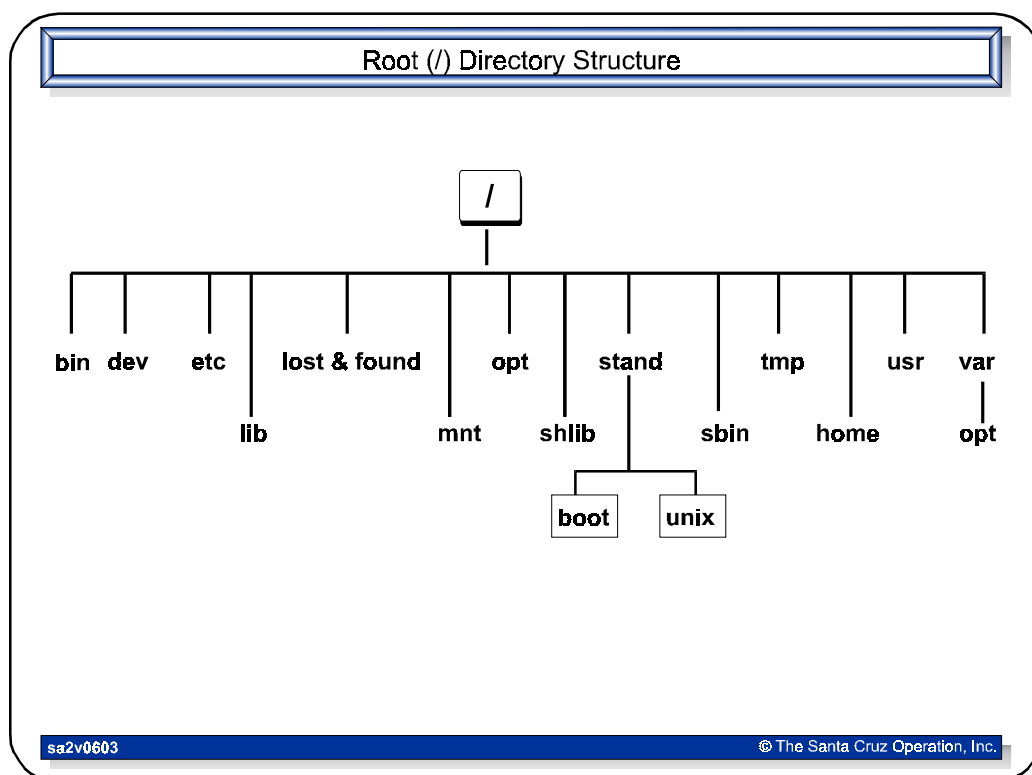
6.1.3 Superblock

One special data block, the *superblock*, contains overall information about the filesystem, just as the inode contains information about a specific file. The superblock contains the information necessary to mount a filesystem and access its data, including the:

- Size of the filesystem
- Number of free inodes
- Free space available

When the filesystem is mounted, the system reads information from the disk version of the superblock into memory.

AV 6-3



6.2 Describing the Directory Hierarchy

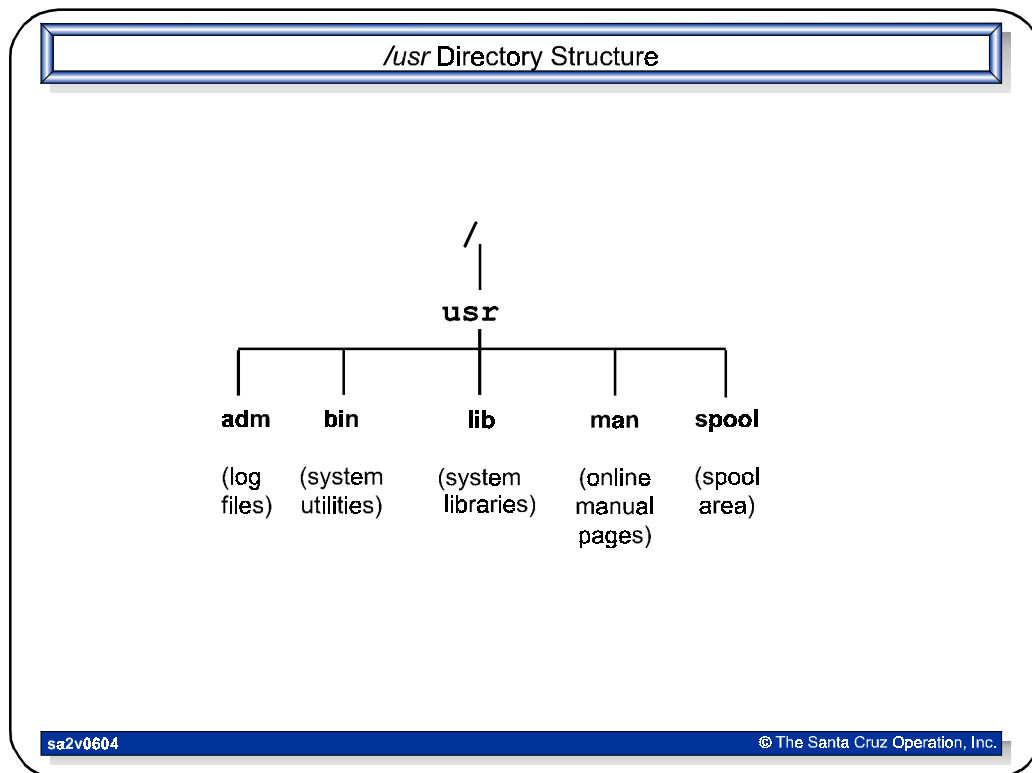
Knowing about some of the system directories on a UnixWare 7 system will help you remember where to find specific tools and configuration files.

The root (/) directory is the starting point of the hierarchical files and directories structure. A brief description of some its key directories is listed below:

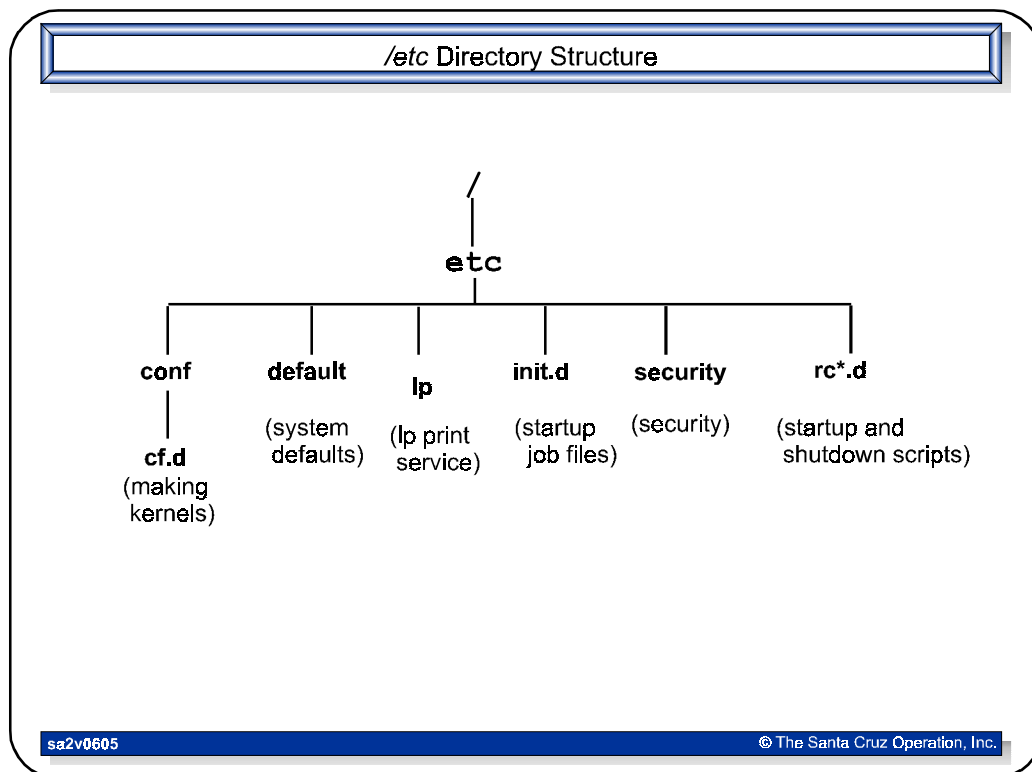
<i>/bin</i>	System binaries. This directory is symbolically linked to <i>/usr/bin</i> .
<i>/dev</i>	System device files
<i>/etc</i>	System administration files and tools, including startup and shutdown scripts, and kernel generation
<i>/home</i>	Default location for user accounts
<i>/lib</i>	C programming libraries. This directory is symbolically linked to <i>/usr/lib</i> .
<i>/lost+found</i>	Unreferenced files when cleaning the root filesystem. Each filesystem has its own <i>lost+found</i> directory.
<i>/mnt</i>	Empty directory used for mounting filesystems
<i>/opt</i>	Directory to which add-on application packages are installed
<i>/OpenServer</i>	Contains SCO OpenServer binaries and libraries for compatibility purposes
<i>/sbin</i>	Contains executables used in the booting process and in manual recovery from a system failure
<i>/proc</i>	Mount point which provides information on system processes
<i>/shlib</i>	Shared libraries. This directory is symbolically linked to <i>/usr/lib</i>
<i>/stand</i>	Directory to access the filesystem that contains the kernel and boot files
<i>/tmp</i>	Temporary files created by programs
<i>/usr</i>	Contains application and system-specific information
<i>/var</i>	Contains files and directories that vary from machine to machine, such as <i>tmp</i> , <i>spool</i> , and <i>mail</i>
<i>/var/adm</i>	Contains important system accounting and log files

<i>/var/opt</i>	Created and used by application packages
<i>/var/spool</i>	Contains various directories for storing files to be mailed, printed, or passed through networks. The <i>/usr/spool</i> directory is linked to <i>/var/spool</i> .

AV 6-4



AV 6-5



6.2.1 Key directories in the */usr* directory structure

The */usr* directory contains many working files and directories for UNIX utilities and includes:

<i>/usr/adm</i>	A symbolic link to <i>/var/adm</i> , a storage directory for important log files
<i>/usr/bin</i>	Contains binaries that are UNIX application specific, including uucp (1bnu), vi (1), and mailx (1).
<i>/usr/lib</i>	Contains configuration files for many UNIX application utilities, including <i>/usr/lib/mail/mailrc</i>
<i>/usr/man</i>	Contains the online man pages and links to the flat text files used by catman (1Mbsd)
<i>/usr/spool</i>	A symbolic link to <i>/var/spool</i> , a storage directory for spooling programs

6.2.2 Key directories in the */etc* directory structure

These commands and files in the */etc* directory structure contain system-specific configuration files and system administration databases:

<i>/etc/conf</i>	Contains files and programs used to configure the UnixWare 7 kernel
<i>/etc/default</i>	Contains system default files
<i>/etc/lp</i>	Contains configuration files and interface programs for the LP Print Service
<i>/etc/mail</i>	Contains mail configuration files and programs
<i>/etc/security</i>	Contains files and directories related to security and system auditing

Important files that also relate to system security are:

<i>/etc/security/ia/master</i>	Contains the same information as the <i>/etc/passwd</i> and <i>/etc/shadow</i> file. This file is machine readable only.
<i>/etc/security/ia/index</i>	Contains a list of user accounts. This file is machine readable only.
<i>/etc/security/tfm/users/*</i>	Contains the authorizations assigned to each user
<i>/etc/security/tfm/roles/*</i>	Contains the authorizations assigned to each administrative role

AV 6-6

Special Device Files

Example of block, character, and raw devices for a 3.5 floppy

files in /dev/dsk

brw-rw-rw-	1	root	sys	1,180	Apr	7	1998	/dev/dsk/f03c
brw-rw-rw-	1	root	sys	1,176	Apr	7	1998	/dev/dsk/f03ct
brw-rw-rw-	1	root	sys	1,100	Apr	7	1998	/dev/dsk/f03d
brw-rw-rw-	4	root	sys	1, 96	Apr	7	1998	/dev/dsk/f03dt

▲

files in /dev/rdisk

crw-rw-rw-	1	root	sys	1,180	Apr	7	1998	/dev/dsk/f03c
crw-rw-rw-	1	root	sys	1,176	Apr	7	1998	/dev/dsk/f03ct
crw-rw-rw-	1	root	sys	1,100	Apr	7	1998	/dev/dsk/f03d
crw-rw-rw-	4	root	sys	1, 96	Apr	7	1998	/dev/dsk/f03dt

▲

sa2v0606© The Santa Cruz Operation, Inc.

6.3 Managing the /dev Directory

Programs, applications, and commands (running on the UNIX system) do not directly access system hardware devices such as hard disks, terminals, and printers. Instead, they use files that represent system hardware.

The files that represent hardware do not contain data, but rather point to code that is located within the kernel itself. These empty pointer files are known as *special device files*.

Special device files require an inode, but do not use data blocks. They are either *character* or *block* files.

By convention, special device files are stored in /dev, or in subdirectories of /dev such as /dev/dsk or /dev/rdsk. The /dev/dsk and /dev/rdsk subdirectories contain the UNIX filenames for floppy and hard disks.

These common naming conventions are used to identify multiple device files associated with a single device:

/dev/dsk/f*	Floppy devices
/dev/term/*	Serial devices
/dev/lp*	Parallel devices

Exercise: Listing device files

Perform a long listing and display the contents of the /dev directory.

Note the subdirectories that are contained within the /dev directory.

6.3.1 Block device special files

You can identify a block device special file by running the `ls -l` or just the `l` command and locate a *b* in the first column before the file permissions.

Filesystems are examples of block special devices.

6.3.2 Character device special files

You can identify a character device special file by running the **l** command and locating a *c* in the first column before the file permissions.

Examples of character special devices include serial ports, parallel ports, disk drives, and tape drives.

6.3.3 Raw interfaces

Most block special devices also have a character device interface called the “raw” interface. Raw devices are character devices; that is, a character interface to an existing block device.

You can identify a raw interface if you run the **ls(1)** command and see an *r* prefix to the filename or subdirectory (for example, the raw interface to the floppy disk is */dev/rdisk/f03h*). Raw interfaces also have a *c* in the first column before the file permissions.

Raw interfaces allow for efficient transfers of data larger than 1KB.

AV 6-7

Major and Minor Device Numbers

				Major	Minor				
brw-rw-rw-	1	root	sys	1,180	Apr 7 1998	/dev/dsk/f03c			
brw-rw-rw-	1	root	sys	1,176	Apr 7 1998	/dev/dsk/f03ct			
brw-rw-rw-	1	root	sys	1,100	Apr 7 1998	/dev/dsk/f03d			
brw-rw-rw-	4	root	sys	1, 96	Apr 7 1998	/dev/dsk/f03dt			

crw-rw-rw-	1	root	sys	1,180	Apr 7 1998	/dev/dsk/f03c			
crw-rw-rw-	1	root	sys	1,176	Apr 7 1998	/dev/dsk/f03ct			
crw-rw-rw-	1	root	sys	1,100	Apr 7 1998	/dev/dsk/f03d			
crw-rw-rw-	4	root	sys	1, 96	Apr 7 1998	/dev/dsk/f03dt			

sa2v0607

© The Santa Cruz Operation, Inc.

6.3.4 Major and minor device numbers

Each special device file has a major and minor device number associated with it.

The kernel uses special device files to access hardware. Together, the major and minor numbers represent a section of device driver code that the kernel requires to access a specific hardware device as follows:

- Major numbers determine which device driver is accessed
- Minor numbers allow you to choose how the device driver will handle a given device to allow different modes of operation

6.3.4.1 Major device numbers

Each device driver of a given type has a unique major number. For example:

- Hard drives have a major number of 7679
- Floppy drives have a major number of 1
- Tapes have a major number of 111

The major number identifies which device driver the kernel should use to access a particular device.

Note that the major number can differ from system to system.

Exercise: Viewing major numbers

Identify the major numbers for hard disks, floppy disks, and tapes.

Minor device numbers

The major device number uses the minor device number to further identify the physical device being accessed.

A minor device number specifies which routine is being used by the device. For example, the tape device has a major number of 111. However, additional device files exist that access the same device but perform a different function.

Major and minor numbers that represent tape devices include:

111,0 —A non-retentioning rewinding device (*/dev/rmt/ctape1*)

111,1 —A non-retentioning, non rewinding device (*/dev/rmt/ntape1*)

111,4—A device that unloads on close (*/dev/rmt/utape1*)

A minor number can also represent a second device when multiple types of the same device are installed.

To view the major and minor numbers note above, enter:

```
l /dev/rmt | more
```

6.3.5 Creating special device files

Most special device files are automatically created as part of the installation procedure for the device. If a special device file is erased, you have to recreate the missing special device file.

Use the **mknod**(1) command to create special device files:

```
mknod device_name [c | b] major minor
```

For example, to recreate */dev/rmt/ctape1*, enter:

```
mknod /dev/rmt/ctape1 c 111 0
```

You will often have to change the permission, ownership, and group to that of the original file.

To print a copy of your */dev* directory, enter:

```
l -iR /dev/ | lp
```

Place it in your **System Log Book** so that the you can easily recreate device files if necessary.

6.3.6 The Device Database

The Device Database, */etc/device.tab*, is a database in which the system keeps a list of attributes about each device on the system.

As new devices are added, the device database is updated automatically. You can use the **getdev**(1M) to get a list of the devices defined in the Device Database. This list will give you a list of devices that can be used with the **devattr**(1M) command.

You can use the **devattr**(1M) to list the attributes of any device.

To view the attributes for the first hard disk, enter:

```
devattr -v /dev/rdisk/c0b0t0d0s0
```

The **devattr** command also accepts alias device names as arguments. For example, the alias device name for the first:

- Hard disk is *disk1*
- Floppy disk is *diskette1*
- CD-ROM is *cdrom1*
- Tape is *ctape1*

You can use the **prtconf**(1M) command to list the amount of memory and configured devices on your system.

Exercise: Using the prtconf and devattr command

Use the **prtconf** command to list the devices that are configured on your system.

Use the **getdev**(1M) command to list the devices in the Device Database.

Use the **devattr** command with the first floppy device. From the information displayed, what is the command that you would use to format this device?

6.3.7 Alternate names for special device files

Use links to create alternative names for frequently used devices. For example, to link the name *DAT_tape* to */dev/rmt/ctape1*, enter:

```
ln /dev/rmt/ctape1 /dev/DAT_tape
```

You can now access the tape drive by either device name.

6.3.8 Troubleshooting the */dev* directory

Potential sources of problems in the */dev* directory can include:

- The removal of special device files, either by accident or by program malfunction
- Misspelled device names that leave extraneous, ordinary files

If a legitimate device file is removed, it can be replaced with a text file with the same name.

To locate a list of “suspect” files, enter one of the following:

```
find /dev -type f -exec lr -li {} \;
```

```
lr -li /dev | grep “^.”
```


6.4 Describing Important vxfs Filesystem Features

One of your most important responsibilities as a system administrator is creating and maintaining filesystems. Filesystem maintenance includes keeping the filesystems clean, repairing damaged filesystems, and ensuring adequate space for all users.

The UnixWare 7 system has at least two filesystems on the primary hard disk:

root	Contains the programs and directories that comprise the operating system. The root filesystem is a <i>vxfs</i> filesystem type.
stand	Contains the information needed to boot your system, the boot(4) program, and the kernel, <i>/stand/unix</i> . The stand filesystem is a <i>bfs</i> filesystem type.

The *vxfs* default filesystem type is an extent based filesystem. It is a high-integrity filesystem that provides enhanced performance and reliability and includes these features:

- Largefile support
- Extent based allocation
- Supports logical block sizes of 1KB, 2KB, 4KB, and 8KB
- Quotas
- Enhanced data integrity
- Fast filesystem recovery

6.4.1 Large file support

By default, file sizes are limited to 2GB. To create files up to 1TB, when creating a *vxfs* filesystem with the **mkfs(1M)** command, you must specify the **largefile** option. This is discussed in detail in Section 6.5, “Adding Large file Support,” on page 6-35.

6.4.2 Extent-based allocation

An extent is defined as one or more adjacent blocks of data within a filesystem. When storage is added to a file on a *vxfs* filesystem, it is grouped in extents, as opposed to being allocated a block at a time (as is done by the *s5*, *sfs*, and *ufs* filesystem).

Using extents means that:

- Disk I/O to and from a file can be done in units of multiple sequential blocks
- For sequential I/O, multiple block operations are considerably faster than block-at-a-time operations. Almost all modern disk drive controllers accept I/O operations in multiple blocks.

6.4.3 Logical block size

Logical block size is the size of the blocks that the system kernel uses to read and write files. The logical block size usually differs from the physical block size.

Logical block size defaults for vxfs are based on the size of the filesystem created. They are:

- 1KB for up to 8GB
- 2KB for up to 16GB
- 4KB for up to 32GB
- 8KB over 32GB

There are no fragments with vxfs because storage is allocated in extents that are collections of one or more blocks.

Because vxfs uses extent based allocations, keeping the logical block size as small as possible increases performance and reduces wasted space for most workloads.

6.4.4 Quotas

The vxfs filesystem lets you place a per-user quota limit on the use of inodes and data blocks. A quota consists of two limits:

- Hard—The user cannot exceed this limit on data blocks and inodes under any circumstances
- Soft—This limit is smaller than the hard limit and allows a user to temporarily exceed the soft limits as long as the user reduces their resource use before the time limit expires

Warnings are issued to users before additional resources are denied.

Use the **edquota(1M)** command to set quotas.

For additional information on quotas, refer to **quotaon_vxfs(1M)**.

6.4.5 Enhanced data integrity

To improve performance, most filesystems do not synchronously update data and inode changes to disk. If the system crashes, files that have been updated within the past minute are in danger of losing data.

When mounting a vxfs filesystem you can specify the `mincache=clasesync` mode. The `mincache=clasesync` mode filesystem is somewhat slower as any changes to the file are flushed to the disk when the file is closed.

6.4.6 Fast filesystem recovery

The vxfs filesystem provides recovery only seconds after a system failure by utilizing a tracking feature called intent logging. Intent logging is a scheme that records pending changes to the filesystem structure. These changes are recorded in a circular *intent log*.

The intent log is a circular activity log with a default size of 1024 blocks. If the filesystem is less than 4MB, the log size is reduced to avoid wasting space.

The vxfs intent log only records changes to the filesystem structure. File data changes are not logged.

When the system is halted unexpectedly, the **fsck(1M)** command only checks logged transactions rather than checking the entire filesystem for inconsistencies.

AV 6-8

The Disk1 VTOC				
#SLICE	TAG	FLAGS	START	SIZE
0	0x5	0x201	61	1044198
1	0x2	0x200	237473	785009
2	0x3	0x201	42517	194956
3	0x0	0x0	0	0
4	0x0	0x0	0	0
5	0x0	0x0	0	0
6	0x0	0x0	0	0
7	0x1	0x201	61	34
8	0xd	0x201	95	942
9	0x0	0x0 0	0	
10	0x9	0x200	1037	41480
11	0x0	0x0 0	0	
12	0x0	0x0 0	0	
13	0x0	0x0 0	0	
14	0x0	0x0 0	0	
15	0xf	0x201	1022482	2074

sa2v0608 © The Santa Cruz Operation, Inc.

AV 6-9

Description of the VTOC		
/* Slice tags */		
V_BOOT	0x01	/* Boot slice */
V_ROOT	0x02	/* Root filesystem */
V_SWAP	0x03	/* Swap filesystem */
V_USR	0x04	/* Usr filesystem */
V_BACKUP	0x05	/* full disk */
V_ALTS	0x06	/* alternate sector space */
V_OTHER	0x07	/* non-unix space */
V_ALTRK	0x08	/* alternate track space */
V_STAND	0x09	/* Stand slice */
V_VAR	0x0a	/* Var slice */
V_HOME	0x0b	/* Home slice */
V_DUMP	0x0c	/* dump slice */
V_ALTSCTR	0x0d	/* Alternate sector/track */
V_MANAGED1	0x0e	/* Volume management public slice */
V_MANAGED2	0x0f	/* Volume management private slice */
/* Slice permission flags */		
#define V_UNMNT	0x01	/* Unmountable partition */
#define V_READONLY	0x10	/* Read only */
#define V_VALID	0x200	/* Partition is valid to use */

sa2v0609 © The Santa Cruz Operation, Inc.

6.4.7 Describing VTOC

The first level of organization of the disk by vtoc is the partition table. The partition table divides the disk into pieces (called partitions) that serve as logical disks. There are a maximum of four partitions for each disk. A partition has four characteristics:

- Start sector
- Length
- Operating system type (for example, UNIX, DOS, or Extended DOS)
- Active flag that identifies the current bootable partition

6.4.8 The fdisk(1M) command

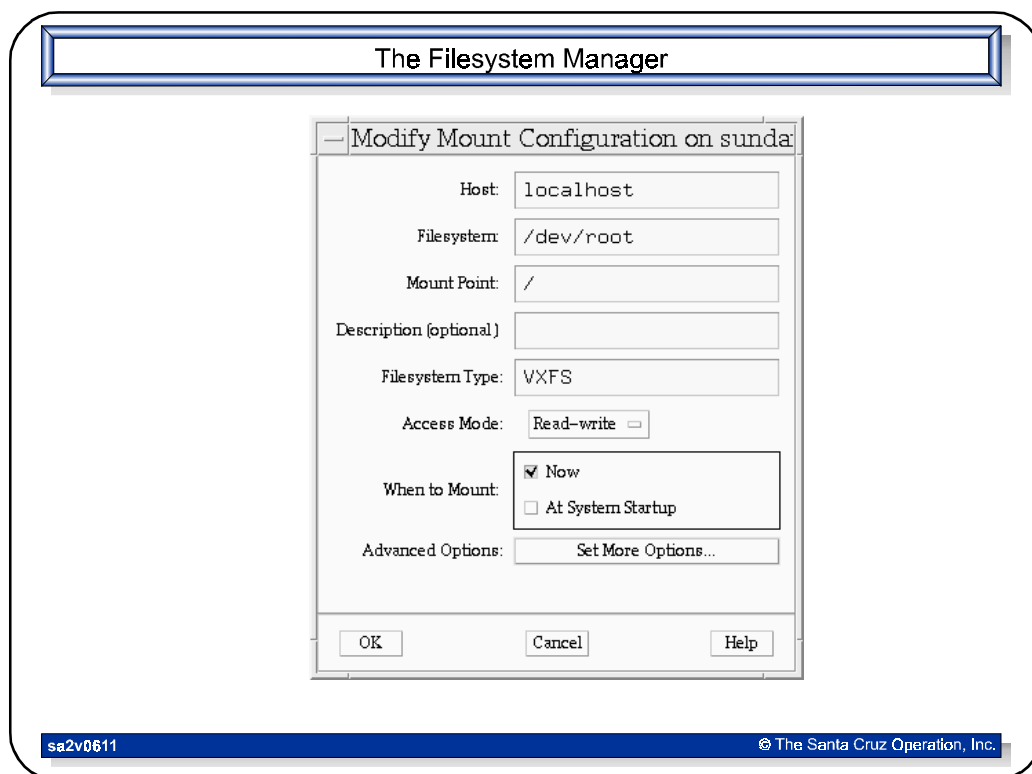
The **fdisk** command is called first to partition the hard disk. You can create up to four (4) partitions per hard disk.

6.4.9 The disksetup(1M) command

AV 6-10



AV 6-11



The **disksetup**(1M) command:

- Executes surface analysis, creates/writes the pdinfo, VTOC and alternates information to the disk
- Issues the **mkfs**(1M) calls to create a specific type of filesystem on the slice. The defaults used to create the filesystem are taken from the */etc/default/vxfs* file. Do not edit this file.
- Creates directories on which the filesystems are mounted
- Mounts the filesystems
- Updates the */etc/vfstab* file so that filesystems are remounted during subsequent reboots of the system

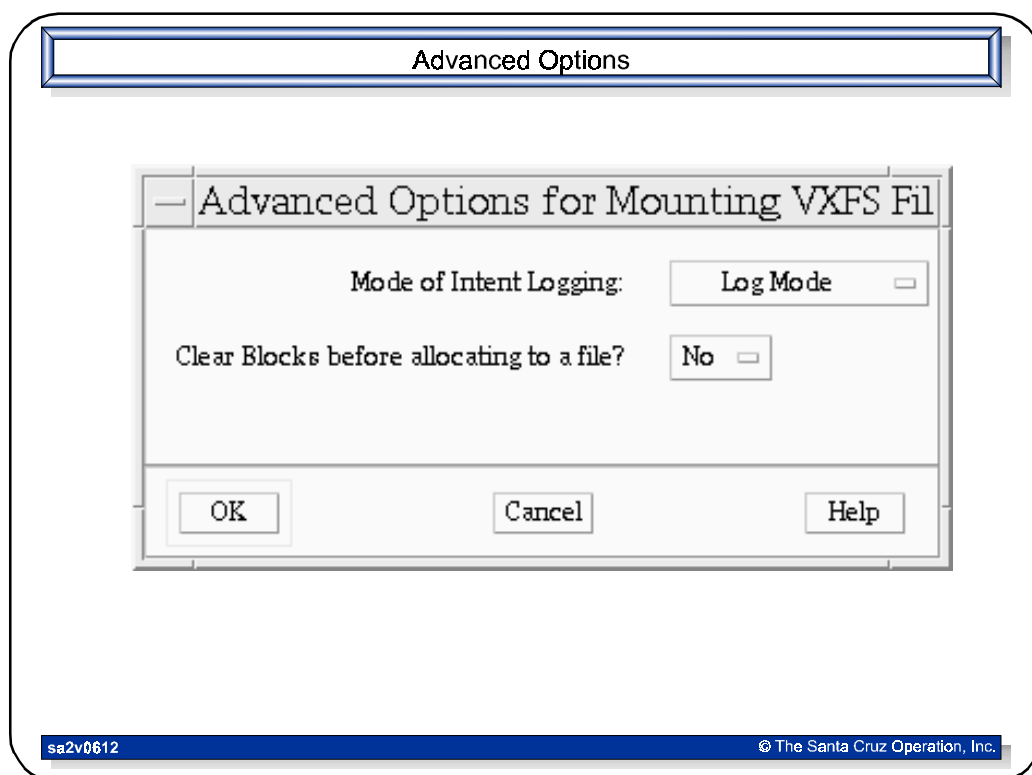
6.4.10 Preparing the Filesystem for Access

Use the **Filesystem Manager** to view filesystems and add or change the mount configuration.

You can change:

- The mount point; that is, the directory name to which the filesystem is mounted (this can be an existing empty directory, or the utility will create the directory if it does not exist)
- A description of the filesystem (optional)
- The access mode (read-write or read-only)
- When to mount the filesystem (now or at startup)

AV 6-12



You can also use the **Filesystem Manager** to specify filesystem-specific **Advanced Options** when mounting a filesystem. For example, you can change the mode of intent logging or clear blocks before allocating a file for the vxfs filesystem.

The available advanced options depend on the type of filesystem you are adding or changing (vxfs, s5, sfs, ufs, and so forth).

The changes you make with the **Filesystem Manager** are written to the `/etc/vfstab` file.

6.4.10.1 Changing the mode of intent logging

In the **Advanced Options** field of the **Filesystem Manager** you can change the mode of intent logging.

Recall that by default, vxfs filesystems log file transactions before they are committed to disk, reducing the time spent checking and repairing filesystems. You can configure this behavior as:

- **Log Mode**—Log all transactions before the system call returns to the application. This incurs a slight performance penalty while ensuring fast recovery.
- **Delayed Mode**—Allow some system calls to return to the application before being logged. This increases system throughput at the expense of data integrity.
- **Temporary Mode**—Delay almost all transaction logging. This has the highest penalty for data integrity and is intended for use with temporary filesystems.

6.4.10.2 Clear blocks before allocating file

Also as part of the Advanced Features Set of the **Filesystem Manager** you can choose to **Clear blocks before allocating file**. Answering yes ensures that existing data is always overwritten with zeroes before the disk space is released.

6.4.11 The `/etc/vfstab` file

The `/etc/vfstab` file contains default information when mounting filesystems. Default options for a vxfs filesystem include:

<code>/dev/dsk/device_name</code>	The block special device file
<code>/dev/rdsk/device_name</code>	The raw device used by fsck (1M)

<i>/directory_name</i>	The mount directory to which the filesystem is mounted
vxfs	The name of the filesystem type
1	A numeric entry indicates to fsck(1M) that the filesystem is to be checked automatically.
yes/no	A yes entry indicates that the filesystem should be mounted automatically at system startup by mountall(1M)

These default mount options are also included:

mincache=close	Specifies that data is written to the disk immediately when a file is closed. This feature can affect system performance because of the additional time writing to the disk. To optimize system performance, you can remove this entry from <i>/etc/vfstab</i> . Unmount your filesystem before doing so. After making your changes, remount the filesystem.
-----------------------	--

Although you cannot add fields to the */etc/vfstab* file, you can add mount options to the seventh (7th) field. For example, so that the filesystem is mounted with the **largefiles** capability enabled, enter a comma (,) and the word **largefiles** after the word:

mincache=close

The line should now read:

mincache=close, largefiles

Refer to the **mount(1M)** and **mount_vxfs(1M)** reference pages for a complete definition of mount options.

6.5 Adding Large file Support

By default, vxfs filesystems are created to support file sizes of up to 2GB. The vxfs filesystem can support file sizes greater than 2GB. Use the **mkfs(1M)** command to determine if your vxfs filesystems supports **largefiles**.

For example, if your */home* filesystem resides on */dev/rdisk/c0b0t0d0s4*, enter:

```
mkfs -m /dev/rdisk/c0b0t0d0s4
```

If **largefiles** are:

- Supported, the string **largefiles** is displayed
- Not supported, the string **no largefiles** is displayed

Largefiles are only supported for the vxfs Version 4 disk layout. Be careful when implementing largefile support, some third party backup programs may not be able to back up large files.

Follow these steps using the **mkfs** command to recreate a filesystem and add **largefiles** support.

- | | |
|---------|---|
| Step 1. | Back up the data in the filesystem. |
| Step 2. | Run mkfs -m to determine the syntax to recreate the filesystem. |
| Step 3. | Unmount the filesystem. |
| Step 4. | Run mkfs to recreate the filesystem changing the no largefiles options to largefiles . For example to recreate the <i>/home</i> filesystem, enter: |

```
mkfs -F vxfs -o ninode=65536,bsize=1024,version=4,  
ino size=256,logsize=1024,largefiles /dev/rdisk/c0b0t0d0s4  
7936110
```

- | | |
|---------|-----------------------|
| Step 5. | Mount the filesystem. |
| Step 6. | Restore your data. |

6.6 Checking and Repairing Filesystems

The **fsck**(1M) command checks and repairs the structure of a filesystem. The checking is performed in a series of passes, and each pass checks for different inconsistencies within a filesystem.

In most instances of filesystem corruption, **fsck**(1M) automatically repairs a filesystem without interaction. If you need to interact with **fsck**, you need to know the definition of passes, prompts, and error messages.

Use these guidelines when running **fsck**:

- You must be in single-user mode to run **fsck** on the root filesystem
- Non-root filesystems must be unmounted
- If you do not specify a device when running **fsck**, all character special files listed in the */etc/vfstab* file that have an entry in the **fsckdev** and the **fsckpass** (with a numeric entry) fields are checked.

6.6.1 Specifying a filesystem type

If you specify a device when running **fsck**, you must also specify a filesystem type because each filesystem can have a different structure. Not all functions that **fsck** performs are applicable to every filesystem.

Use the **-F** option to specify a filesystem type. For example, to specify the *vxfs* filesystem type for a filesystem residing on slice 4, enter:

```
fsck -F vxfs /dev/rdisk/c0b0t0d0s4
```

This lesson covers running **fsck** on a *vxfs* filesystem. For more information on the functions that **fsck** performs on different filesystem types, refer to **scohelp** ⇒ **Filesystems** ⇒ **Solving filesystem problems**.

6.6.2 Full vs. fast checking

Recall that the *vxfs* filesystem updates transactions to an intent log. Intent log records are written to disk before actual changes are made to the filesystem.

The intent log is “replayed” when changes are made to the filesystem. The **fsck** command runs an intent log replay, also known as a “fast” filesystem check, on a *vxfs* filesystem by default.

Because of the intent log replay, a “full” filesystem check should be performed only if there is a disk hardware failure or **fsck** indicates that a full **fsck** is necessary.

6.6.2.1 Fast filesystem check

When you run a fast filesystem check, if a log replay is not required, this message displays:

```
file system is clean - log replay is not required
```

If a log replay is required, this message displays:

```
log replay in progress  
replay complete - marking superblock as clean
```

Recall that a fast filesystem check is run by default. For example to run a fast check on a *vxfs* filesystem, enter:

```
fsck -F vxfs /dev/rdisk/c0b0t0d0s4
```

If the filesystem is not marked clean, you must perform a full filesystem check.

6.6.2.2 Full filesystem check

When you run a full check on a *vxfs* filesystem, the **fsck(1M)** command scans and examines the filesystem structures and reports on its progress. Each pass compares components and checks that these components agree with each other. These passes are:

pass 0 - Checking structural files

Performs a fast check on the filesystem if you did not request a full check.

pass 1 - checking inode sanity and blocks

Reads the inode table to determine sizes and locates the blocks used by each file. Inodes are checked for inode type, zero link counts, inode sizes, and bad or duplicate blocks. Scans the filesystem a second time if duplicate blocks were found.

pass 2 - checking directory linkage

Cleans up error conditions caused by improper inode status, out-of-range inode pointers, and directories that point to bad inodes.

pass 3 - checking reference counts

Checks for directory connectivity and reconnects files that were severed from the directory structure. Any files that are unreferenced but still valid are placed in the *lost+found* directory.

When the directory entry is deleted, the name of the file is lost, so **fsck** assigns the inode number as the new name of the file.

pass 4 - checking resource maps

Checks the link count of each entry that survived Phase 2 and 3.

Refer to **scohelp** ⇒ **Filesystems** ⇒ **Solving filesystem problems** for more information on checking filesystems with **fsck**.

6.6.3 Running fsck(1M)

When you boot the system after an abnormal shutdown, the **fsck(1M)** command runs automatically on the root filesystem. Other filesystems are checked when the system enters multiuser mode.

When running **fsck** on the root filesystem, you should first unmount all other filesystems.

For example, to check and repair the */home* filesystem that is installed on */dev/rdsk/c0b0t0d0s4*, do one of the following:

- Run **fsck** and assume a yes response to all questions asked by **fsck(1M)**:

fsck -F vxfs -y /dev/rdsk/c0b0t0d0s4

When you specify the **-y** option, **fsck** will (if necessary) perform a full check after running the intent log replay.

- Run a full check:

```
fsck -F vxfs -ofull /dev/rdisk/c0b0t0d0s4
```

If **fsck**(1M) finds that the root filesystem has been modified, it will return values that indicate to either remount the root filesystem or reboot the system automatically. The root filesystem is only remounted if the damage to it is minor.

Exercise: Repairing the */home* filesystem

Unmount the */home* filesystem and run **fsck**.

6.6.4 Error messages

Error messages may appear in any pass. For example, if an inode was not connected to a directory in pass 4, **fsck**(1M) would reconnect it to the *lost+found* directory. If you are running **fsck** in interactive mode, the following message appears, prompting you to answer *y(es)* to reconnect the file to the *lost+found* directory:

```
UNREF FILE I=782 OWNER=230 MODE=0777 SIZE=5046 MTIME=March 8
1995 (RECONNECT?)
```

The following list describes the fields in this message:

UNREF FILE I=782	The inode number of the unreferenced file
OWNER=230	The user ID of the owner of the file
MODE=0777	The permissions on the file
SIZE=5046	The size of the file in bytes
MTIME=March 8 1997	The last time the file was modified

Refer to the **fsck**(1M) manual page for a complete definition of all error messages reported.

6.6.5 Running **ncheck(1M)**

If **fsck(1M)** reports file errors, it is best to exit the program and run the **ncheck(1M)** command. This command associates inode numbers with filenames.

Sometimes the problem is with the inode, not the actual file. To resolve this problem, you can copy the file to create a new inode number and then remove the original file.

To run the **ncheck** command on the root filesystem, enter:

```
ncheck -i inode_number /dev/root
```

If you are checking other filesystems, specify the filesystem's raw device name.

If you do not specify a filesystem, **ncheck** checks all filesystems that are listed in the */etc/vfstab* file.

Exercise: Associating inode numbers with filenames

Use the **-i** option to the **ls(1)** command to determine the inode number for the */etc/default/tar* file. Search the root filesystem with **ncheck(1M)** and display the results.

6.7 Configuring Additional Disks and Filesystems

During the installation of UnixWare 7, you can install up to two hard disks. If you did not install a second disk during installation, you can use the **diskadd(1M)** command to add any additional disks.

This section covers the installation of a hard disk attached to a SCSI host adapter and an IDE controller. Before adding disks, you will need to:

- Configure the hardware according to your vendor's documentation
- Verify that the hard disk has been formatted and passes the vendor's diagnostic tests

6.7.1 Possible configurations for secondary hard disks

You can add many types of hard disks to your system, provided the disk controller is compatible with the disk. This list provides some guidelines for possible configurations for systems with multiple hard disks:

- Boot disk on the primary SCSI host adapter

You can add only other SCSI host adapters.

- Boot disk on an IDE, EIDE, or ESDI controller

Systems can support up to two IDE or EIDE controllers. Each controller can support two IDE devices. You can also add and configure SCSI host adapters.

- Boot disk on a Compaq IDA-III controller

You can add a maximum of five or more IDA-III controllers. Each controller can support up to 14 disk drives.

6.7.2 The Storage Device Interface (SDI)

The Storage Device Interface (SDI) is a framework for driver software for many of the hardware devices on a system. The SDI interface:

- Ensures compatibility between current drivers and future UnixWare releases
- Organizes, simplifies, and standardizes the way device drivers are written
- Makes it easier to administer a wide variety of both devices and interfaces

The Storage Device Interface (SDI) is a versioned driver interface that provides a framework for developing device drivers for mass storage devices. SDI also is a driver architecture that uses a multilevel I/O stack. In SDI versions prior to SDI 4, the I/O stack contained two layers: a controller-specific portion (the HBA driver) and a device-specific portion (the target driver).

Beginning with SDI 4 and UnixWare 7, a third layer is added for a new type of I/O driver called a layered driver, which is stacked in the path of the I/O operation. Every layer implements a well-defined piece of functionality and, as such, modifies the I/O operation. The UnixWare 7 I/O stack has three layers:

- Target drivers
- Host Bus Adapter (HBA) drivers
- Layered devices

SDI forms the connection between these layers, providing a consistent and complete interface for each. SDI is based on the SCSI Device Interface, but has been expanded to also support non-SCSI devices. This architecture/framework has been developed to be broad in its functionality and flexibility, so that the majority of SDI driver source is generic and portable. The enhancements to SDI 4 and the UnixWare 7 I/O stack provide the flexibility required to accommodate new technologies such as ccNUMA architectures, multiported devices, and high availability configurations such as multipath I/O.

The mass storage target drivers in UnixWare 7 include:

<code>sd01(7)</code>	Hard disks
<code>st01(7)</code>	Tape devices
<code>sc01(7)</code>	CD-ROM devices
<code>sw01(7)</code>	WORM devices
<code>mc01(7)</code>	Medium changers

The target drivers support both SCSI and non-SCSI controllers. Non-SCSI controllers include ESDI, IDE, and ST506. For example, if you have two different types of hard disk controllers installed in your system, both would be supported by the `sd01(7)` target driver.

For more information on the SDI and associated devices, refer to **scohelp** ⇒ **Hardware** ⇒ **Mass storage devices overview**.

6.7.3 Hard disk naming conventions

The hard disk naming conventions for SCSI and non-SCSI disk include both a block and character device file. Entries for block devices are in the `/dev/dsk` directory and character devices are in the `/dev/rdisk` directory.

Disk device filenames have either of these two forms:

`/dev/[r]dsk/c#b#t#d#s#`

`/dev/[r]dsk/c#b#t#d#p#`

Where:

- `c#` is the occurrence of the host adapter board or IDE controller
- `b#` is the HBA bus number. `#` is always 0 for IDE disks.
- `t#` is the SCSI target controller (ID). For the first IDE disk, `#` is 0, and the second `#` is 1.
- `d#` is the logical unit number. `#` is always 0 for IDE disks.
- `s#` is the slice number
- `p#` is the partition number. `p0` corresponds to the whole disk.

6.7.4 Multi-Path Input/Output (MPIO)

Newer storage processors and RAID boxes have multiple independent ports to access the same media. Also, some new configurations have multiple Host Adapters on the same bus, allowing independent access to the media through each card.

Multipath I/O allows multiple Host Bus Adapters (HBAs) to be connected to the same bus and multiported devices. This serves two purposes: MPIO provides high availability to the devices when an HBA fails. Upon controller failure, the operating system will redirect I/O through another I/O path to the device.

MPIO also provides load balancing of I/O to the device across multiple controllers. Currently, disks are the only devices supported by multipath I/O.

Multipath I/O is implemented as a device driver in the operating system kernel and as such is able to detect and discover all known paths for devices on system boot. By default the multipath I/O (MPIO) driver will attempt to use all paths and load balance I/O across controllers.

Traditionally, devices are represented through special files such as `/dev/rdisk/c0b0t1d0s0`. This name represents a disk that is connected to the first I/O card, on the first bus off that card, at an Id of 1 and logical unit 0. In a system with MPIO, it represents a handle to the storage device and in fact the first path discovered to the media.

For example, a disk at target 0 would be known as `/dev/rdisk/c0b0t0d0s0`. If the first path were to be lost, the system would remember the association of this name and a particular stamp. This name would not be reused until some administrative command is run to acknowledge the change.

The **sdipath(1M)** command manages paths in MPIO.

As the system boots, physical disks are discovered and brought online. Each disk has a stamp associated with it that is used to determine if this is the first occurrence of this disk. If it is the first occurrence, a device node is created in `/dev/rdisk` and `/dev/dsk`, then an entry is made into the resource manager database.

All subsequent occurrences of this disk will only get an entry in the **resmgr(1M)** database under an "sdol" entry. The pathname associated with each path to the disk is created by appending the resource manager key value to the word path. Thus if a resource manager entry is made for a path having a resmgr key of 29, the name of this path would be "path29".

6.7.5 Configuring IDE disks and disk controllers

The IDE interface design consists of an adapter (usually embedded in the motherboard) connected to a maximum of four IDE hard drives. These drives are connected to the adapter by a single 40-conductor cable that carries both control and data information.

Configure IDE disks and disk controllers by:

- Setting the jumpers on the drives according to the number of drives installed. If two or more drives are connected, jumper one for *master* mode, the others for *slave* mode. Jumper a single drive connected to an IDE pattern for *single drive* mode.
- Selecting the ST506 interface on the drive (if you can set this using jumpers; refer to the drive documentation for details)

The embedded controller on most IDE drives can perform automatic sector translation. This allows you to configure drives with several different head/cylinder/sectors-per-track settings.

You may want to use settings other than the factory defaults if your drive has more than 1024 cylinders or if the new settings more closely match some drive types in the BIOS. This is important if you are also installing another operating system, such as DOS.

6.7.6 Using **diskadd(1M)** to configure disks

Configure disks with **diskadd(1M)**. If you invoke **diskadd** with a disk number argument of 1 or without an argument **diskadd(1M)** will attempt to add a second disk with the appropriate name.

For any additional disks, the argument must be in the form of *c#b##d#* as described in Section 6.7.3, “Hard disk naming conventions,” on page 6-43.

The **diskadd** command performs these tasks associated with configuring a hard disk:

- Invokes **fdisk(1M)** to create a partition(s)
- Invokes **disksetup(1M)** to:
 - Perform a surface analysis
 - Create the VTOC (Volume Table of Contents).
 - Write the bad track and bad sector (alternate) tables out to the drive
 - Create the layout of slices
 - Issue **mkfs(1M)** calls to create the filesystem and the *lost+found* directory
 - Create mount points
 - Mount the filesystems
 - Update the */etc/vfstab* file

6.7.7 Adding a hard disk

After you physically install your hard disk and verify that it is recognized by UnixWare 7 using **sdiconfig(1M)**, follow these steps to add an additional hard disk:

Step 1. Log in as **root**.

Step 2. Enter **diskadd 1**.

If you are adding the second hard disk on the system, use **1** for the disk number.

For subsequent disks, you must use the full disk name in the form of /dev/dsk/c#b#t#d#. For example to add a third SCSI disk on the first host adapter, enter:

diskadd /dev/dsk/c0b0t2d0

Enter **y** to continue with the **diskadd** utility.

The **fdisk(1M)** command is invoked by **diskadd(1M)**.

Step 3. Create your partition(s).

- Enter **y**. A single UNIX partition is created and it is marked active.
- Enter **n**. You create the partition(s). It is marked active.

Step 4. Perform a surface analysis.

- Enter **y** to skip a surface analysis.
- Enter **n** to perform a surface analysis

Step 5. Enter the number of slices (filesystems) to be created on this disk.

Step 6. Enter the directory name to which this filesystem will be mounted. You must enter the absolute pathname. If the directory does not exist, it is created.

Step 7. Enter the filesystem type or press to select the default of vxfs.

Step 8. Select the block size or press to use the default block size of 1024 bytes.

Step 9. Press to have the filesystem automatically mounted on reboot or enter **no** to override.

Step 10. Perform Steps 6-9 again if you choose to install more than one filesystem.

Step 11. Specify the size of the filesystem in cylinders (1 megabyte of disk space is approximately 1 cylinder). If you choose to install more than one filesystem, you are prompted for the size of the additional filesystem(s).

Enter **y** to confirm the size of the filesystems. If you answer **n**, you must repeat Steps 5 through 11 to recreate your filesystems.

Step 12. Enter the number of inodes.

- Press  to allocate the default number of inodes
- Enter the number of inodes. You can now create or move files to the new filesystem.

6.8 Managing the Volume Table of Contents (VTOC)

Recall that the Volume Table of Contents (VTOC) is created when you add a disk.

The **vtoc** target driver uses the VTOC to manage the organization of the disk that includes physical disk and slice information:

The PDINFO (Physical Disk Information) is part of the VTOC that is stored in the boot slice of each disk. This physical information is used as a sanity check against what is found at driver initialization time (boot).

Use the **prtvtoc(1M)** command to print the PDINFO. Although you can run this command on any disk, it must always be run on slice 0.

prtvtoc -p /dev/rdisk/c0b0t0d0s0

Exercise: Printing the PDINFO

Print the PDINFO for your primary disk. What is the number of heads and cylinders for your disk?

The VTOC (Volume Table of Contents) also contains the slice information. Recall that a slice can be a filesystem, swap space, or can be left to be organized by an application such as a database. Sixteen (16) slices are created by default.

The slice information includes the:

- Starting sector and cylinder
- Length of the filesystem
- Slice type (for example, root, boot, swap, and stand)
- Permissions

Although it is somewhat difficult to read, the **prtvtoc** command prints the slice information by default:

prtvtoc /dev/rdisk/c0b0t0d0s0

Using the **-f** option and specifying a filename redirects the output to a file in a different format:

```
prtvtoc -f /tmp/vtoc /dev/rdisk/c0b0t0d0s0
```

Exercise: Printing the slice information

Print the slice information on the primary disk. Which slice contains the boot slice? What is its starting sector?

6.9 Troubleshooting Hard Disks

Although hard disks are very reliable, they suffer from physical defects like any other physical media. Sometimes data loss is recoverable and in rare cases the disk or disk controller needs to be replaced. This section deals with recoverable data loss.

6.9.1 Mapping of bad blocks

Mapping of bad blocks is performed dynamically and automatically by the **sd01** disk driver without user intervention. Defective blocks are reassigned to an area of the disk reserved for that purpose.

The **sd01** target driver can map both marginal (readable with some difficulty and actual (unreadable) bad blocks.

If the bad blocks contain data, you may need to restore files or filesystems from a backup.

6.10 Lesson Summary

UnixWare 7 systems include filesystem types that incorporate the latest in filesystem technology while ensuring compatibility with earlier releases of the operating system. Filesystems include a superblock, inodes, and data blocks.

Applications and programs do not directly access system hardware devices such as hard disks, terminals, and printers. Instead, they use special device files that represent system hardware. These files are usually stored in `/dev`, or in subdirectories of `/dev`, such as `/dev/dsk` or `/dev/rdisk`.

UnixWare 7 has many utilities for configuring and maintaining filesystems and hard disks. You can access these utilities using the **Filesystem Manager** or by issuing commands at the command line.

Review questions

1. Which inode represents the root directory of each mounted filesystem?
2. Which command do you use to create special device files?
3. Which filesystem type supports intent logging?
4. Which file contains the information used when mounting a filesystem?
5. Which option to the **fsck(1M)** command performs a full check on a filesystem?

-
- 6-52
- UnixWare® 7 System Administration II: Installation, Configuration, and Maintenance*

6.11 Lab: Configuring a Second Hard Disk

Configure a second hard disk attached to the first controller. If the hard disk is not attached to the first controller, your instructor will provide you with the necessary information to install the disk.

Overview

Run the **diskadd(1M)** command to configure a second hard disk. Create a single partition. Do not run a surface analysis. Create a single filesystem, name the filesystem *sales*, and configure it using the default block size. The filesystem will be mounted on */sales*. This filesystem should mount when going to multiuser mode.

Your instructor will provide you with the hard disk device name.

Procedure

- Step 1. Log in as **root**.
- Step 2. Enter **diskadd 1** or the name that is provided by your instructor.
- Step 3. Enter **y** to continue with the **diskadd** utility.

The **fdisk(1M)** command is invoked by **diskadd(1M)**.

- Step 4. Create a single partition.
- Enter **y**. A single UNIX partition is created and marked active.
- Step 5. Do not perform a surface analysis.
- Enter **y** to skip a surface analysis.
- Step 6. Enter **1** for the number of slices (filesystems) to be created on this disk.
- Step 7. Enter the */sales* directory name on which this filesystem will be mounted. You must enter the absolute pathname. If the directory does not exist, it is created.
- Step 8. Enter the vxfs filesystem type or press to select the default of vxfs.
- Step 9. Press to use the default block size of 1024 bytes.

- Step 10. Press  to have the filesystem automatically mounted on reboot.
- Step 11. Specify the size of the filesystem in cylinders (1 megabyte of disk space is approximately 1 cylinder). Your instructor will furnish you with the size of the filesystem.
- Step 12. Enter **y** to confirm the size of the filesystem.
- Step 13. Press  to allocate the default number of inodes.
- Step 14. Run the **mount(1M)** command to verify that the filesystem is mounted.
- Step 15. Copy the */etc/default/tar* file to the *sales* filesystem.

6.12 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

diskadd(1M)	Configures additional hard disks
disksetup(1M)	Configures hard disks at installation time
edvtoc	Edits the Volume Table of Contents (VTOC)
fdisk(1M)	Displays, creates, or deletes partitions
fsck(1M)	Checks and repairs filesystems
grep(1)	Searches for a pattern
mkfs(M)	Constructs a filesystem on a special device
mknod(1)	Creates a directory entry and a corresponding inode for a special file
mount(1M)	Mounts a filesystem
ncheck(1M)	Generates a pathname and inode number list for all files in the specified filesystem
prtvtoc(1M)	Displays or prints the Volume Table of Contents (VTOC)
scoadmin(1M)	Invokes SCOadmin applications (in this lesson, the Filesystem Manager)
umount(1M)	Unmounts a filesystem

Relevant files

<i>/etc/vfstab</i>	Contains filesystem defaults
<i>/etc/default/vxfs</i>	Contains filesystem defaults used by diskadd(1M)

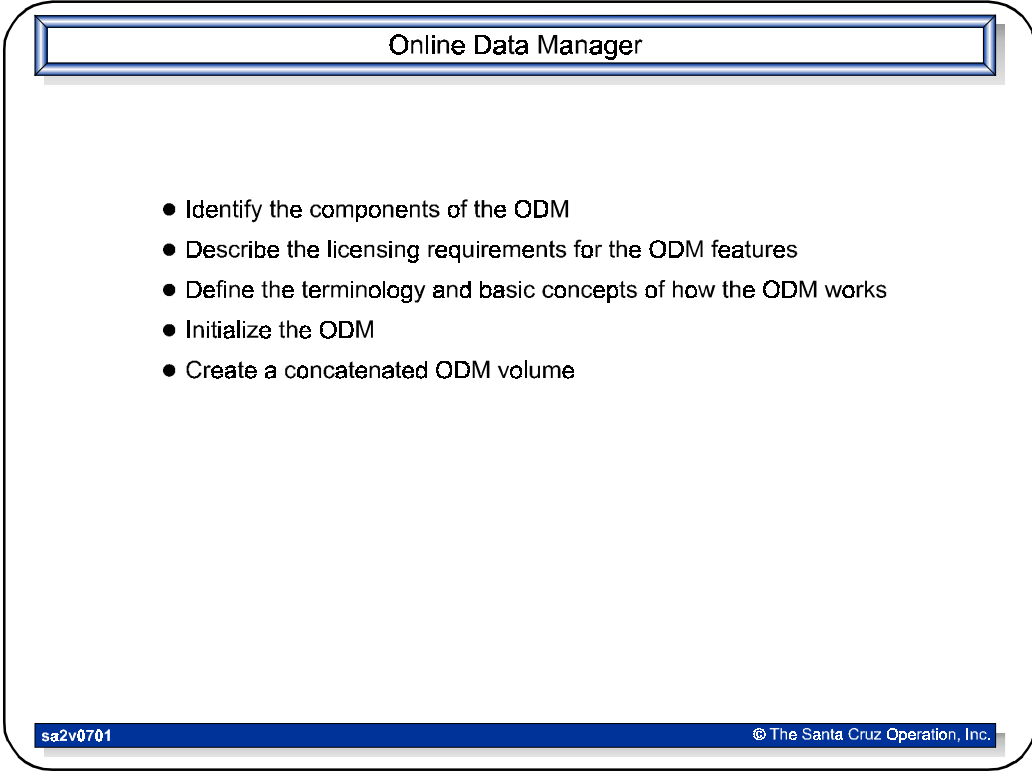
6.13 Answers to Review Questions

1. Inode 2 represents the root directory of each mounted filesystem.
2. You can create special device files with the **mknod**(1) command.
3. The vxfs filesystem supports intent logging.
4. The */etc/vfstab* file contains the information used when mounting a filesystem.
5. The -o full option to the **fsck**(1M) command performs a full check on a filesystem.
6. The order in which programs are invoked when you run **diskadd**(1M) to configure a hard disk are **fdisk**(1M) and **disksetup**(1M).
7. The information about a filesystem slice stored in the Volume Table of Contents (VTOC) includes the starting sector and cylinder, length of the filesystem, slice type, and permissions.

7

Online Data Manager

AV 7-1



The screenshot shows a window titled "Online Data Manager". Inside the window, there is a list of five tasks:

- Identify the components of the ODM
- Describe the licensing requirements for the ODM features
- Define the terminology and basic concepts of how the ODM works
- Initialize the ODM
- Create a concatenated ODM volume

At the bottom of the window, there is a blue status bar with the text "sa2v0701" on the left and "© The Santa Cruz Operation, Inc." on the right.

Lesson 7

Online Data Manager

Introduction

The UnixWare 7 Online Data Manager (ODM) is an enterprise-class storage management solution for high availability and online volume management. It provides software RAID Levels 0, 1, 5, 10 (striping, mirroring, striping with distributed parity and striped mirroring) as well as disk-spanning capabilities. The following products are included in the ODM:

- VERITAS Filesystem (VxFS)
- Volume Manager (VxVM)
- Visual Administrator (VxVA)

Objectives

This lesson provides an in-depth discussion of the UnixWare 7 Online Data Manager (ODM). By the end of this lesson, you will be able to:

- Identify the components of the ODM
- Describe the licensing requirements for the ODM features
- Define the terminology and basic concepts of how the ODM works
- Initialize the ODM
- Create a concatenated ODM volume

7.1 UnixWare 7 Online Data Manager (ODM)

The next sections describe the features available in the ODM. The following products are detailed:

- VERITAS Filesystem (VxFS)
- Volume Manager (VxVM)
- Visual Administrator (VxVA)

7.1.1 VERITAS Filesystem (VxFS)

All editions of the UnixWare 7 operating system include the 'basic' version of the VxFS with the following functionality:

- Extent-based allocation
- Extent attributes
- Fast filesystem recovery
- Enhanced security
- Support for BSD-style quotas

The Enterprise and Data Center UnixWare 7 editions include the full SCO Online Data Manager product which adds the following advanced VxFS features:

- Online administration for defragmentation and resizing of filesystems
- Snapshot mounts for online backup
- Enhanced application interface
- Enhanced mount options
- Improved synchronous write performance
- Enhanced I/O performance
- DMAPI (Data Management API)

Support for the advanced VxFS features on UnixWare 7 editions other than the Enterprise and Data Center, can be purchased with an add-on license for the SCO Online Data Manager.

7.1.2 Volume Manager (VxVM)

To understand the features of the VxVM, it is necessary to define VxVM terminology and the principles of how it works.

7.1.2.1 How the VxVM Works

The VxVM builds virtual devices, called “volumes” on top of physical disks. Volumes are accessed by a UNIX filesystem, a database, or other applications in the same way physical disk partitions would be accessed.

Volumes are composed of other virtual objects that can be manipulated to change the volume’s configuration. Volumes and their virtual components are referred to as “VxVM objects”.

VxVM objects can be manipulated in a variety of ways to:

- optimize performance
- provide redundancy of data
- perform backups or other administrative tasks on one or more physical disks

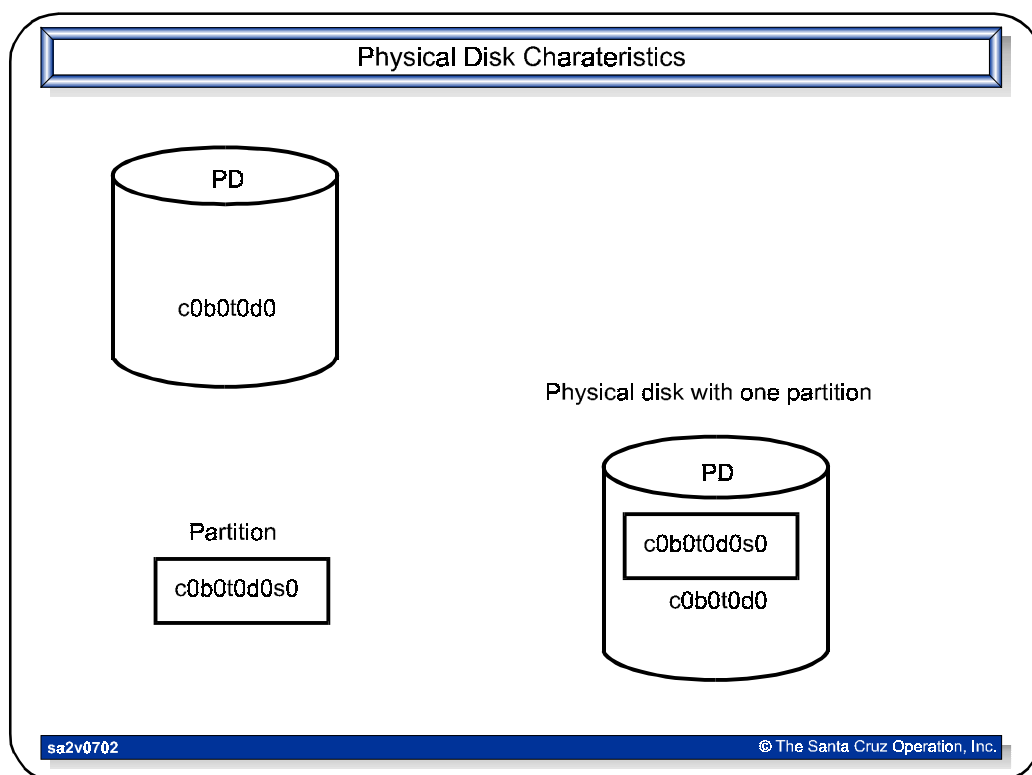
To understand the VxVM, you must understand the relationships between physical objects and VxVM objects.

7.1.2.2 Physical Objects

To perform disk management tasks using the VxVM, you must understand two physical objects:

- Physical disks
- Partitions

AV 7-2



7.1.2.3 Physical Disks

A physical disk is the underlying storage device (media). A physical disk can be accessed using a device name such as `c#b#t#d#`, where `c#` is the controller, `b#` is the bus, `t#` is the target ID, and `d#` is the disk number. The disk in AV 7-2 is disk number 0 with a target ID of 0, and it is connected to controller number 0 in the system.

7.1.2.4 Partitions

A physical disk can be divided into one or more partitions. The partition number, or `s#`, is given at the end of the device name. Note that a partition can take up an entire physical disk, such as the partition shown.

The relationship between physical objects and VxVM objects is established when you place a partition from a physical disk under VxVM control.

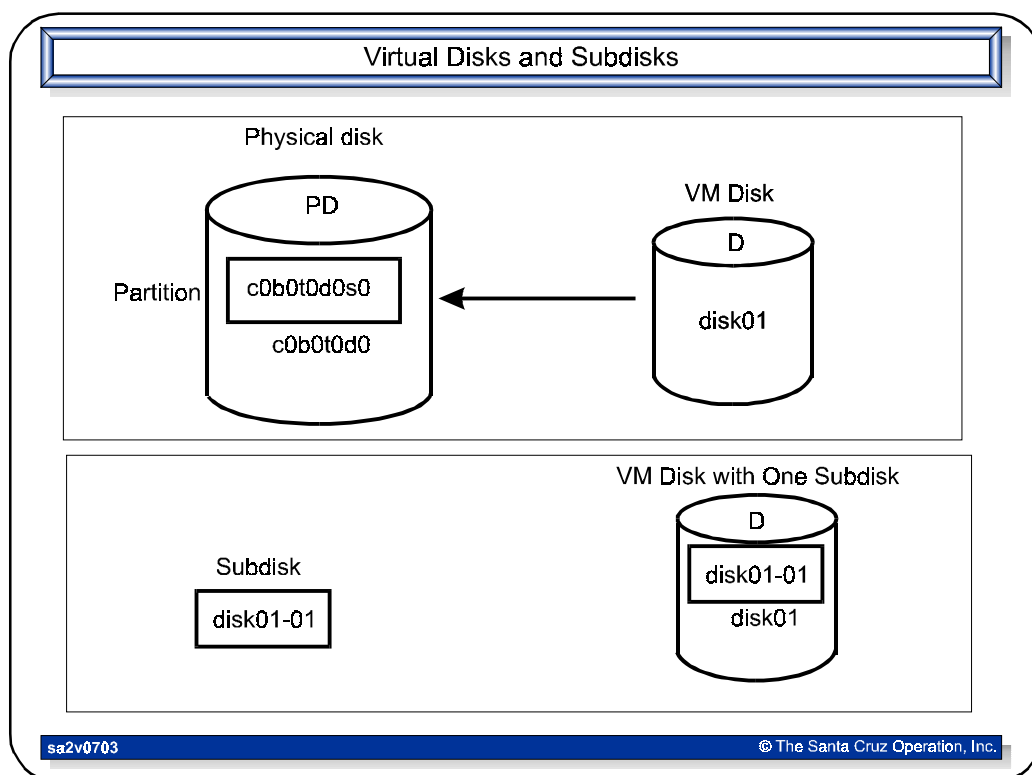
7.1.2.5 VxVM Objects

There are several VxVM objects that allow you to perform disk management tasks:

- VM disks
- Disk groups
- Subdisks
- Plexes
- Volumes

The VxVM objects are described in the sections that follow. Before you can use the VxVM to perform disk management tasks, you must understand the definitions of these objects and their relationships to each other and physical disk devices.

AV 7-3.



7.1.2.6 VM Disks

A VM disk is a contiguous area of disk space from which the VxVM allocates storage. When you place a partition from a physical disk under VxVM control, a VM disk is assigned to the partition.

VM Disk properties include:

- Each VM disk corresponds to at least one partition.
- A VM disk is typically composed of a public region (from which storage is allocated) and a private region (where configuration information is stored).
- A VM disk is accessed using a unique disk media name, which you can supply (or else the VxVM assigns one that typically takes the form `disk##`). The AV7-3 shows a VM disk with a disk media name of `disk01` that is assigned to the partition `c0b0t0d0s0`.

With the VxVM, applications access volumes (created on VM disks) rather than partitions.

7.1.2.7 Disk Groups

A disk group is a collection of VM disks that share a common configuration.

A configuration consists of a set of records containing detailed information about existing VxVM objects, their attributes, and their relationships.

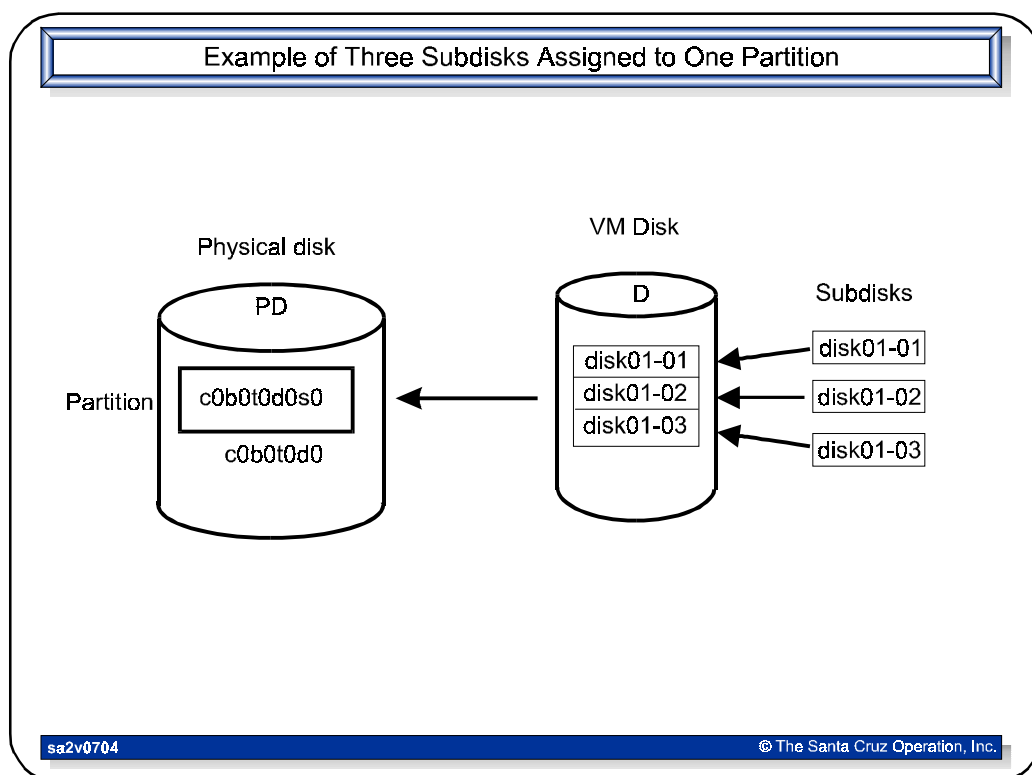
The default disk group is **rootdg** (the root disk group). Additional disk groups can be created, as necessary. Volumes are created within a disk group; a given volume must be configured from disks belonging to the same disk group.

Disk groups allow the administrator to group disks into logical collections for administrative convenience. A disk group and its components can be moved as a unit from one host machine to another.

7.1.2.8 Subdisks

A subdisk is a set of contiguous disk blocks. Subdisks are the basic units in which the VxVM allocates disk space. A VM disk can be divided into one or more subdisks. AV 7-3 shows an example of a subdisk

AV 7-4



Any VM disk space that is not part of a subdisk is considered to be free space, which can be used to create new subdisks.

Each subdisk represents a specific portion of a VM disk, which is mapped to a specific region of a physical disk.

Since the default name for a VM disk is disk## (such as disk01), the default name for a subdisk is disk##-##. So, for example, disk01-01 would be the name of the first subdisk on the VM disk named disk01.

A VM disk may contain multiple subdisks, but subdisks cannot overlap or share the same portions of a VM disk. AV 7-4 shows a VM disk, with three subdisks, that is assigned to one partition.

7.1.2.9 Plexes

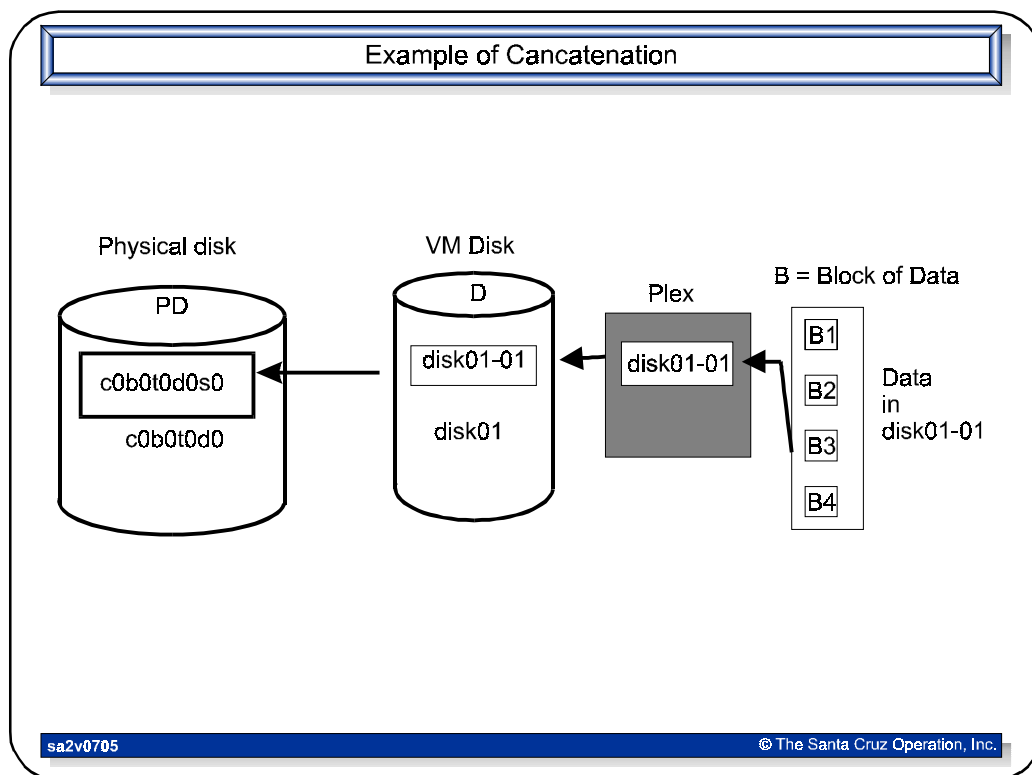
The VxVM uses subdisks to build virtual entities called “plexes” (also referred to as “mirrors”). A plex consists of one or more subdisks located on one or more disks. There are three ways that data can be organized on the subdisks that constitute a plex:

- Concatenation
- Striping (RAID-0)
- RAID-5

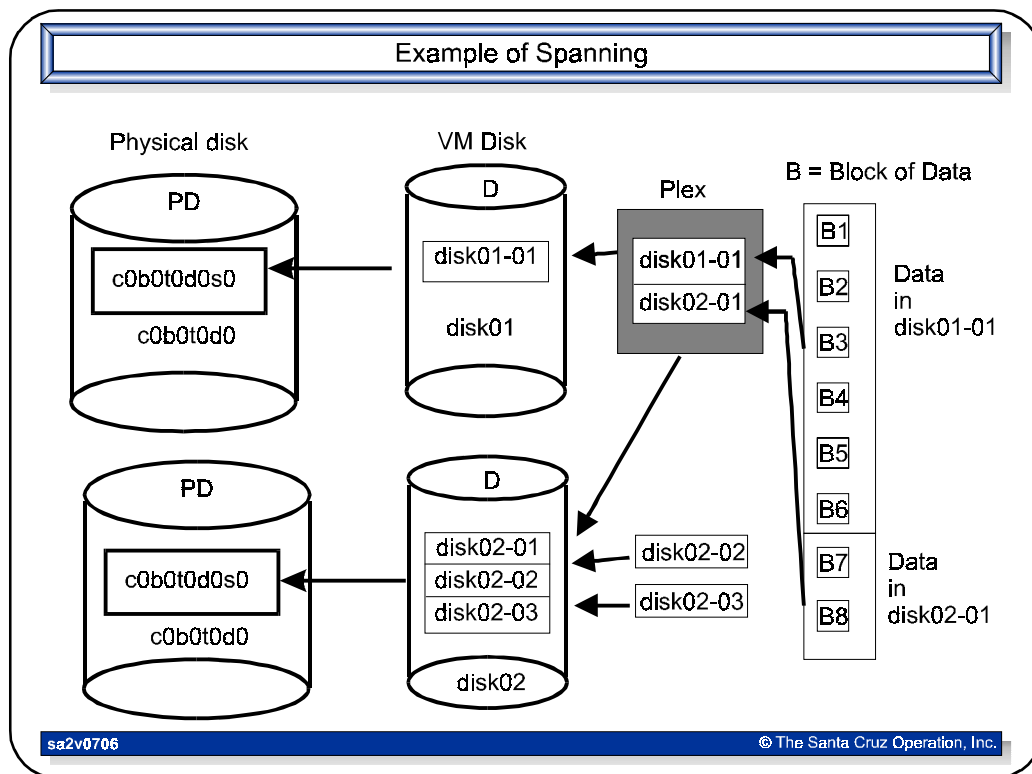
Concatenation is discussed in this course.

For details on striping and RAID-5, refer to **scohelp** ⇒ **Filesystems** ⇒ **VERITAS Online Data Manager** ⇒ **Veritas Volume Managers (TM) User's Guide** ⇒ **Description of the Volume Manager** ⇒ **Mirroring (RAID-1)**

AV 7-5



AV 7-6



7.1.2.10 Concatenation

Concatenation maps data in a linear manner onto one or more subdisks in a plex. If you were to access all the data in a concatenated plex sequentially, you would first access the data in the first subdisk from beginning to end, then access the data in the second subdisk from beginning to end, and so forth until the end of the last subdisk.

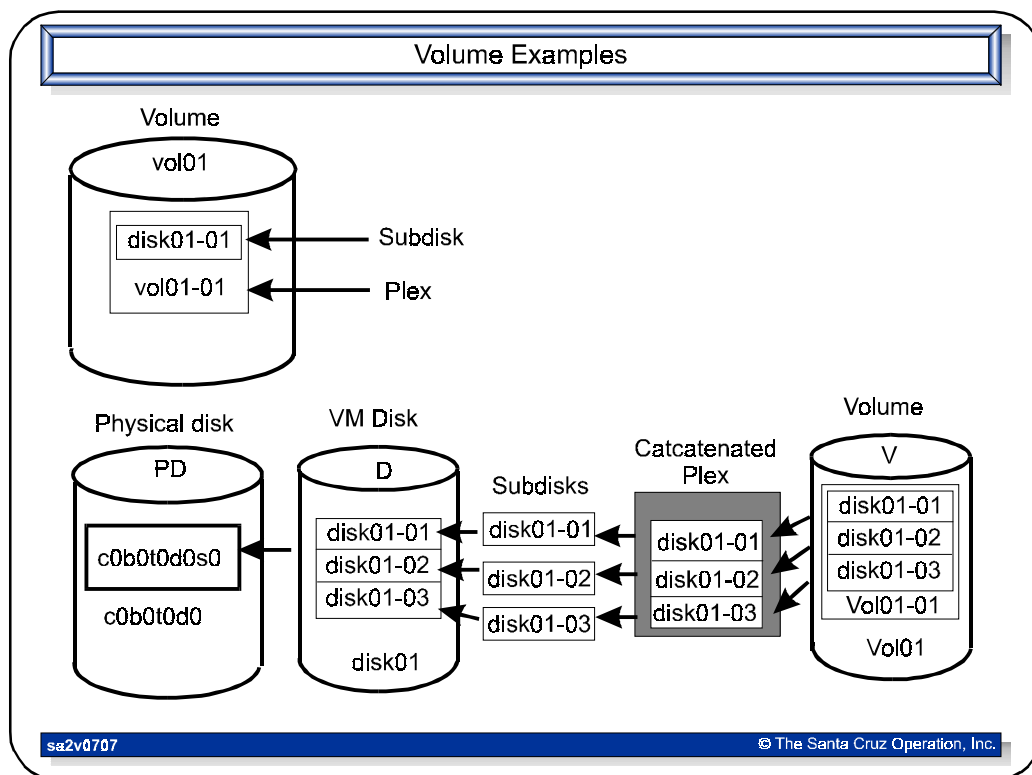
The subdisks in a concatenated plex do not necessarily have to be physically contiguous and can belong to more than one VM disk. Concatenation using subdisks that reside on more than one VM disk is also called “spanning”.

Concatenation with multiple subdisks is useful when there is insufficient contiguous space for the plex on any one disk. Such concatenation can also be useful for load balancing between disks, and for head-movement optimization on a particular disk. AV 7- 5 shows an example of concatenation. The diagram shows data that is spread over two subdisks in a spanned plex.

Since the first six blocks of data (B1 through B6) consumed most or all of the room on the partition that VM disk disk01 is assigned to, subdisk disk01-01 is alone on VM disk disk01. However, the last two blocks of data, B7 and B8, take up only a portion of the room on the partition that VM disk disk02 is assigned to. That means that the remaining free space on VM disk disk02 can be put to other uses. In this example, subdisks disk02-02 and disk02-03 are currently available for some other disk management tasks.

CAUTION! Spanning a plex across multiple disks increases the chance that a disk failure will result in failure of its volume. Use mirroring or RAID-5 (both described later) to substantially reduce the chance that a single disk failure will result in volume failure.

AV 7-7



7.1.2.11 Volumes

A “volume” is a virtual disk device that appears to applications, databases, and filesystems like a physical disk partition, but does not have the physical limitations of a physical disk partition.

A volume consists of one or more plexes, each holding a copy of the data in the volume. Due to its virtual nature, a volume is not restricted to a particular disk or a specific area on a disk. The configuration of a volume can be changed, using the VxVM interfaces, without causing disruption to applications or filesystems that are using the volume.

For example, a volume can be mirrored on separate disks or moved to use different disk storage. A volume can consist of up to 32 plexes, each of which contains one or more subdisks. In order for a volume to be usable, it must have at least one associated plex with at least one associated subdisk. Note that all subdisks within a volume must belong to the same disk group.

The VxVM uses the default naming conventions of vol## for volumes and vol##-## for plexes in a volume. Administrators are encouraged to select more meaningful names for their volumes. A volume with one plex contains a single copy of the data.

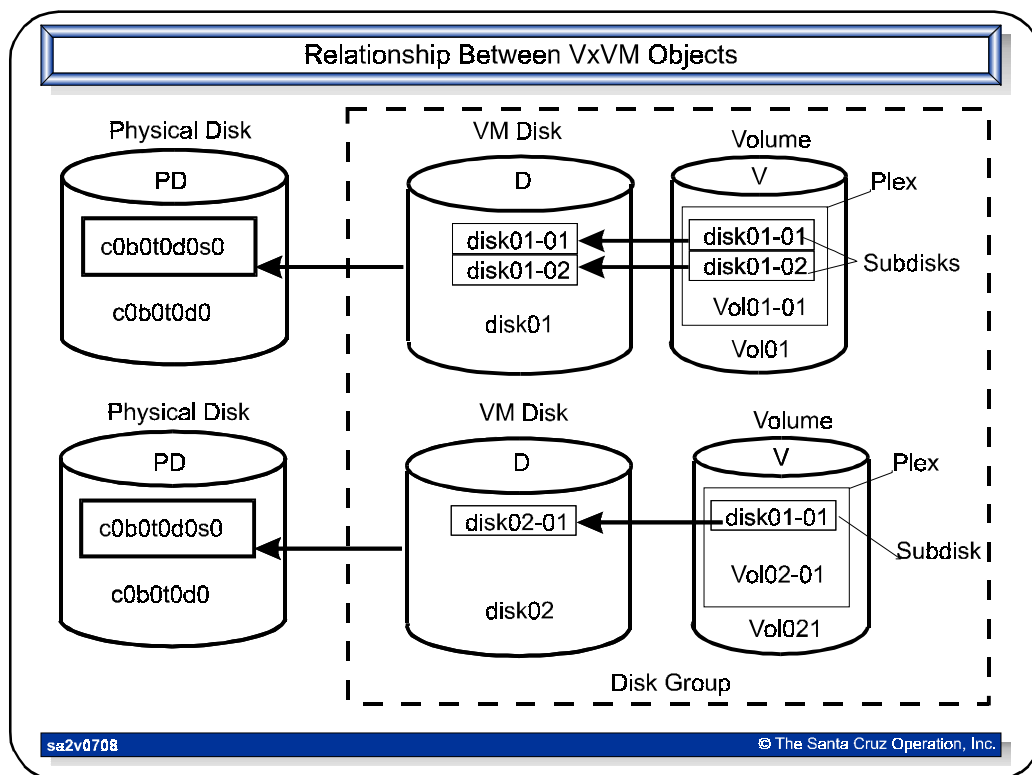
AV 7-7 shows an example of a volume with one plex. Note that volume vol01 in the diagram has the following characteristics:

- Contains one plex named vol01-01
- Plex contains one subdisk named disk01-01
- Subdisk disk01-01 is allocated from VM disk disk01

The diagram also shows how a volume would look if it were set up with a simple, concatenated configuration.

Refer to **scohelp** ⇒ **Filesystems** ⇒ **VERITAS Online Data Manager** ⇒ **Veritas (TM) User's Guide** ⇒ **Description of the Volume Manager** for information regarding the setup of volumes with striping and mirroring.

AV 7-8



7.1.2.12 Relationships between VxVM objects

VxVM objects are of little use until they are combined to build volumes. VxVM objects generally have the following relationship:

- VM disks are placed under VxVM control and grouped into disk groups. Subdisks (each representing a specific portion of a disk) are combined to form plexes.
- A volume is composed of one or more plexes.

AV 7-8 illustrates the relationships between (virtual) VxVM objects, as well as how they relate to physical disks. This illustration shows a disk group containing two VM disks (disk01 and disk02). disk01 has a volume with one plex and two subdisks and disk02 has a volume with one plex and a single subdisk.

7.1.2.13 VxVM feature licensing

The "basic" version of the VxVM includes the following functionality:

Concatenation	Data is mapped linearly onto one or more subdisks in a plex
Spanning	Concatenation using subdisks that reside on more than one VM disk

The UnixWare 7 Enterprise and Data Center editions include the SCO Online Data Manager which licenses the following advanced VxVM features:

- Striping
- RAID-5
- Mirroring

For support of the VxVM advanced features on UnixWare 7 editions other than the Enterprise and Data Center, you must purchase an SCO Online Data Manager add-on license.

7.1.3 Visual Administrator (VxVA)

The VxVA is graphical user interface for the VxVM. Although all VxVM functionality is presented in all editions of UnixWare 7, without the additional licensing of the SCO Online Data Manager, the interface prevents you from executing tasks that apply to product features you have not licensed.

7.1.4 Initializing the VxVM

Before starting the VxVM, you must use **vxinstall** to initialize the VxVM. In addition, the VxVM configuration daemon (**vxconfigd**) must be running and enabled.

If you attempt to run the VxVA from **scoadmin(1M)** after initial system installation without initializing the VxVM, you will get the system message noted below.

Cannot connect to vxconfigd maybe vxconfigd isn't running.

Click on the window to exit.

The next sections describe the process for initializing the VxVM including:

- Pre-initialization instructions
- Using **vxinstall**
- Checking VxVM processes

7.1.4.1 Pre-initialization instructions

Consider these items in preparation for VxVM initialization:

Path	Most of the commands used in the initialization are in the <i>/sbin</i> , <i>/usr/sbin</i> , <i>/usr/bin/X11</i> , or <i>/opt/vxvm-va/bin</i> directory. You should add these directories to your PATH environment variable. For example, using the Bourne or Korn Shell, modify your <i>.profile</i> with:
------	--

```
PATH=/sbin:/usr/sbin:$PATH:/usr/bin/X11:/opt/vxva/bin export PATH
```

Disk Space	Any disk to be managed by VxVM should have two free partitions and a small amount of free space (1024 sectors). The free space should be at the beginning or end of the disk and should not belong to a partition. This space is used for storing disk group configurations and a disk label that ensures that the VxVM can identify the disk, even if it is moved to a different address or controller. It is also used to keep track of disk configuration and to ensure correct recovery. The VxVM allocates approximately 1024 sectors (512K) from each disk for the disk group configurations and the disk label. This space is sufficient to handle normal disk group configurations for up to approximately 20 disks.
------------	--

The boot (root) disk is a special case; if no other space is available, the VxVM will attempt to allocate space usually reserved for swap by shrinking the swap slice. This process is known as swap relocation and, if necessary, will happen automatically during root disk encapsulation.

Although it is possible to put disks with no free space under VxVM control, this should only be used as a migration strategy because many VxVM capabilities based on disk identity are available only for disks with the required free space.

Disk Contents	Before initializing the VxVM, make sure you are aware of the contents of each disk.
Disk Control	Do you want to place all or just some of the disks on the system and/or controller under VxVM control?
Root Disk Control	Do you want to place the system's root disk under VxVM control? (This is necessary in the case of a one-disk system.)
Encapsulation	Encapsulation is a way of placing a disk under Volume Manager control that preserves any existing data on the disk. When you place disks under VxVM control, do you want to preserve data in any or all existing filesystems and partitions through encapsulation? With encapsulation, existing partitions are converted to volumes and existing filesystems are mounted on volumes. With encapsulation, the <i>/etc/vfstab</i> and <i>/etc/swaptab</i> files are updated to ensure that filesystems previously mounted on disk partitions will be mounted on volumes instead. If you plan to encapsulate any disks, make sure that those disks have the required minimum free space (1024 sectors).
Initialization	This destroys all data and partitions on each disk and makes the disk available as free space for allocating new volumes, or mirrors of existing volumes. If you are running vxinstall on new disks that have never been placed under VxVM control before, choose this option so that the disks will be repartitioned according to the Volume Manager's specifications.
Install Type	During the initialization you will need to decide whether to do a Custom Installation or a Quick Installation.

- Custom Installation—encapsulate, unitized, or leave disks alone
- Quick Installation—encapsulate or initialize all disks on a controller together

7.1.4.2 Using **vxinstall** to initialize the VxVM

The following describes how to initialize the VxVM using the **vxinstall** program. The **vxinstall** program places all specified disks under VxVM control. These disks are automatically placed in the default disk group, **rootdg**.

At least one disk must remain in **rootdg** at all times while VxVM is running.

Disks that are not initially placed under VxVM control by **vxinstall** can be added later using a VxVM interface (such as **vxdiskadm**). However, you must add at least one disk to VxVM control with **vxinstall** before you can add disks using other utilities or interfaces.

Initialize the VxVM as follows:

- Step 1. Log in as **root**.
- Step 2. Setup any disk or controller exclude files that you require. To exclude one or more disks from VxVM control, add the names of those disks to the */etc/vx/disks.exclude* file. **vxinstall** ignores any disks listed in this file and leaves them untouched. The following is an example of the contents of a *disks.exclude* file (which you must create this file manually):

```
c0b0t0d0
```

```
c0b0t1d0
```

Similarly, you can exclude all disks on an entire controller by adding the name of that controller to the */etc/vx/cntrls.exclude* file.

The following is an example of the contents of the *cntrls.exclude* file:

```
c0
```

```
c1
```

- Step 3. Enter the following to start **vxinstall**:

```
vxinstall
```

The **vxinstall** program then examines all controllers it finds attached to the system and then lists them:

Generating list of attached controllers....

Volume Manager Installation

Menu: Volume Manager/Install

The VxVM names disks on your system using the controller and disk number of the disk, substituting them into the following pattern:

c<controller>b<bus>t<target>d<disk>

Examples:

c0b0t0d0 - first controller, bus 0, first target, first disk

c1b0t0d0 - second controller, bus 0, first target, first disk

c1b0t1d0 - second controller, bus 0, second target, first disk

The Volume Manager has detected the following controllers on your system:

c0:

c1:

c2:

Step 4. Press  to continue.

Next, **vxinstall** displays a brief introduction to the initialization process:

Volume Manager Installation

Menu: VolumeManager/Install

You will now be prompted to use Quick Installation or Custom Installation. Custom Installation allows you to select how the VxVM will handle the installation of each disk attached to your system.

Quick Installation examines each disk attached to your system and attempts to create volumes to cover all disk partitions that might be used for filesystems or for other similar purposes. If you do not want to use some disks with the VxVM, or if you want to reinitialize some disks, use the Custom Installation option. Otherwise, we suggest that you use the Quick Installation option.

Step 5. Press to continue.

vxinstall then displays a menu with the following options:

```
1 Quick Installation
2 Custom Installation
? Display help about menu
?? Display help about menuing system
q Exit from menus
```

Step 6. Select 1 to perform a quick installation:

Quick Installation will bring every disk attached to your system under VxVM control unless you have chosen to exclude some disks (using the *disks.exclude* or *.cntrl.exclude* file, described in step 2). The Quick Installation option is the easiest to use. For each controller, Quick Installation gives you the option of either initializing or encapsulating all disks. If you want to encapsulate some disks on a given controller and initialize others, use Custom Installation.

Quick installation allows you to initialize disks on a controller, even if valid partitions are found on those disks. However, you should avoid selecting the initialize option for any disks containing valid data.

With encapsulation, the */etc/vfstab* and */etc/swaptab* files are updated to ensure that filesystems previously mounted on disk partitions will be mounted on volumes instead.

vxinstall asks you a series of questions for each controller and disk it finds connected to your system. When a default response is displayed in parentheses, press to accept that default. At any of the **vxinstall** prompts, you can select **q** to completely abandon the initialization and start over.

All disks are encapsulated or initialized (according to your instructions) at the end of the **vxinstall** procedure. If you quit **vxinstall** before it enters its final phase of actually initializing or encapsulating the disks, all disks will be left as they were before **vxinstall** started.

- Step 7. At the following prompt, indicate whether to encapsulate your boot disk (this example assumes your boot disk is named `c0b0t0d0`).

The `c0b0t0d0` disk is your Boot Disk. You cannot add it as a new disk. If you encapsulate it, you will make your root filesystem and other system areas on the Boot Disk into volumes. This is required if you want to mirror your root filesystem or system swap area.

Encapsulate Boot Disk [y,n,q,?] (default: n)

Enter n, to leave your root and other filesystems will be left unaltered.

- Step 8. **vxinstall** now goes through each controller and asks you how to handle the disks contained on that controller. **vxinstall** begins this process for each controller by identifying the controller and generating a list of its disks:

Generating list of attached disks on c1....

The Volume Manager has detected the following disks on controller c1:

`c1b0t0d0 c1b0t1d0 c1b0t2d0 c1b0t3d0 c1b0t4d0`

Press  continue.

If any disks are listed in the `/etc/vx/disks.exclude` file, they are listed here separately as excluded disks.

- Step 9. At the following prompt, indicate whether you want to initialize all the disks on the controller:

Volume Manager Quick Installation For Controller c1 Menu:

VolumeManager/Install/QuickInstall/c1

Initialize all disks on this controller?

(destroys data on these disks) [y,n,q,?] (default: n)

- If you enter n, **vxinstall** assumes that you want to encapsulate all disks on the controller instead and displays the following:

Volume Manager will now try to encapsulate all the disks on this controller. Disks not having valid partitions will be initialized.

Press to continue.

- If you enter **y**, **vxinstall** prompts you for confirmation and warns you that this will destroy any existing data on the disks.

Are you sure (destroys data on these disks)

[y,n,q,?] (default: n)

NOTE: Be careful when using this option as it can result in data loss.

- Step 10. Enter **y** to continue with the disk initialization. The following message appears:

Volume Manager will now initialize all the disks on this controller including disks having valid partitions.

Press continue.

- Step 11. At the following prompt, indicate whether you want default disk names to be assigned to all of the disks on the controller.

Use default disk names for these disks? [y,n,q,?]
(default: y)

Enter **y**, **vxinstall** automatically assigns and lists default disk names for each disk:

The c1b0t0d0 disk will be given disk name disk01

The c1b0t1d0 disk will be given disk name disk02

The c1b0t2d0 disk will be given disk name disk03

The c1b0t3d0 disk will be given disk name disk04

(If you enter **n**, **vxinstall** prompts you for a disk name for each disk on the controller individually)

Press to continue.

If encapsulation was selected, this list also includes the following message for each disk that can be encapsulated

The c1b0t0d0 disk has been configured for encapsulation.

or the following message is displayed empty disks that will be initialized instead of encapsulated:

The c3b0t2d0 disk appears to be empty.

Adding as a new disk.

Press to continue.

Step 12. When all of the disks on the current controller have been named, press to continue and repeat the naming process for the remaining controllers on your system.

Step 13. When you have completed the **vxinstall** procedure for all controllers on your system, **vxinstall** displays a summary of the disks you have designated for initialization (New Disk) or encapsulation (Encapsulate) on each controller:

The following is a summary of your choices.

c0b0t5d0 New Disk

c1b0t0d0 New Disk

c1b0t1d0 New Disk

c1b0t2d0 New Disk

c2b0t2d3 New Disk

c3b0t0d0 Encapsulate

c3b0t1d0 Encapsulate

c3b0t2d0 New Disk

Is this correct [y,n,q,?] (default: y)

This is your last chance to review and alter your choices about how to handle any of the disks to be placed under VxVM control.

Step 14. Encapsulate or remove disks from VxVM control.

- Enter **y**, **vxinstall** will encapsulate all disks listed with **Encapsulate** and initialize (destroying any existing data) all disks listed with **New Disk**.
- If you enter **n**, **vxinstall** prompts you for the name of a disk to be removed from the list and excluded from VxVM control:

Step 15. After running **vxinstall**, shut down and reboot your machine

Exercise: Initializing the VxVM

Follow the steps under Section 7.1.4, “Initializing the VxVM,” on page 7-18 to place a second disk under VxVM control.

Perform a quick installation and encapsulate any existing data on the disk.

7.1.5 Checking VxVM processes

After the VxVM has been successfully installed and you have rebooted your system, you can use the **ps(1)** command to confirm that key VxVM processes **vxconfigd(1M)**, **vxnotify(1M)**, and **vxsparecheck(1M)** are running:

```
ps -ef | grep vx
```

These processes must be running before you can successfully use the VxVA.

Exercise: Checking the VxVM processes

Use the **ps(1)** command to verify that the VxVM process are running.

7.1.6 Starting up VxVA

You can invoke the VxVA Visual Administrator either by using **SCOadmin** ⇒ **VERITAS Volume Manager** or from a "terminal" window. From a terminal window, enter:

```
/sbin/tfadmin vxva&
```

There is also a command-line interface to the VxVM. For information about using the command-line options see **scohelp** ⇒ **Filesystems** ⇒ **VERITAS Online Data Manager** ⇒ **VERITAS Volume Manager(TM) Users' Guide** ⇒ **Part 3: Using the Command Line Interface**.

7.1.7 Performing VxVA basic operations

The VxVA basic operations are provided to perform administrative operations relatively quickly and easily. The basic operations use free space management techniques to automatically locate available space for the desired operation, unless you specify some suitable disk space.

The Basic-Ops menu provides cascading menus for the following operations:

File System Operations	Perform operations involving general filesystem maintenance. Filesystems can be created, mounted and unmounted, resized, and backed up.
Volume Operations	Perform operations involving general volume maintenance. Volumes can be created from VxVM disks, removed, mirrored, resized, and backed up.
Disk Operations	Perform operations involving general disk maintenance. Disks can be initialized, removed, or replaced.

Filesystem operations and basic disk operations are not simulated by the Visual Administrator in demo mode, so are only accessible in real mode. The advanced disk operations are, however, accessible in demo mode.

7.1.8 Creating a concatenated volume

Volumes must be created to take advantage of the VxVM concept of virtual disks. Once a volume exists, a filesystem can be placed on the volume to organize the disk space with files and directories. Alternatively, applications such as databases might be used to organize data on volumes.

The creation of a volume involves the creation of plex and subdisk components. With the basic approach to volume creation, the underlying plexes and subdisks are handled automatically. You simply indicate the desired volume characteristics.

You can create various types of volumes using either the basic or the advanced approach:

- | | |
|----------|--|
| Basic | The basic approach takes information about what the user wants to accomplish and then performs the necessary underlying tasks. This approach requires only minimal input from the user, but does permit more detailed specifications. Basic operations are accessed using the Basic-Ops menu. |
| Advanced | The advanced approach consists of a number of fairly complicated commands that typically require the user to specify detailed input. These commands use a "building block" approach that requires the user to have a detailed knowledge of the underlying structure and components in order to manually perform the sequences of commands necessary to accomplish a certain task. Advanced operations are accessed via the Advanced-Ops menu in the VxVA. |

Volumes can be created with various layout types:

- | | |
|---------|---|
| Simple | A concatenated volume whose subdisks are arranged both sequentially and contiguously within a plex. Concatenation allows a volume to be created from multiple regions of one or more disks if there is not enough space for an entire volume on a single region of a disk. |
| Striped | A volume with data spread fairly evenly across multiple disks. Stripe units are equal sized fragments that are allocated alternately and evenly to the subdisks of a single plex. There should be at least two subdisks in a striped plex, each of which should exist on a different disk. Throughput increases with the number of disks across which a plex is striped. Striping helps to balance I/O load in cases where high traffic areas exist on certain subdisks. |
| RAID-5 | A volume that uses striping to spread data fairly evenly across multiple disks in an array, and allows independent access. It also stripes parity across all the disks in the array, rather than being concentrated on a single parity disk. Each stripe contains parity stripe units and data stripe units. In comparison to the performance of striped volumes, throughput of RAID-5 volumes decreases since parity information needs to be updated each time data is written. However, in comparison to mirroring, the use of parity reduces the amount of space required. |

Mirrored	A volume with multiple plexes that duplicate the information contained in a volume. Although a volume can have a single plex, at least two are required for true mirroring (redundancy of data) to be in effect. Each of these plexes should contain disk space from different disks, in order for the redundancy to be useful.
Striped and Mirrored	A volume with a striped plex and another plex that mirrors the striped one. This requires at least two disks for striping and one or more other disks for mirroring (depending on whether the plex is simple or striped). A striped and mirrored volume is advantageous because it both spreads data across multiple disks and provides redundancy of data.

7.1.8.1 Creating a concatenated volume with the basic approach

To create a simple concatenated volume with the basic approach:

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **VERITAS Volume Manager**.
- Step 3. Select the **View** window corresponding to the disk group in which you want to perform this operation (View of **rootdg**, by default)
- Step 4. Select **Basic-Ops** ⇒ **Volume Operations** ⇒ **Simple**.
- Step 5. Fill out the information required in the **Simple Volume/FS Create Form**.

The following information is found in the **Simple Volume/FS Create Form**:

Volume name	Either use the default or enter a different name for the volume to be created. The name must be unique within this disk group. The maximum length of this field is 31 characters.
Volume size	Enter the desired volume size. The size should be entered as a number followed immediately by the letter k, m, s, or g to indicate kilobytes, megabytes, sectors, or gigabytes, respectively. If no unit is specified, the default is sectors. The volume size should be less than or equal to the available free space of the disks.

Usage Type	Select the desired usage type. fsgen is the filesystem generic usage type that assumes the volume is being used by a filesystem. gen is the generic usage type that makes no assumptions regarding the data content of the volume. The default is fsgen .
Create log subdisk	Indicate whether a log subdisk is to be created. The default is No .
Create file system	Indicate whether a filesystem is to be created. When this form is invoked from the Volume Operations menu, the default is not to create a filesystem (No). All fields below this field are only accessible when Yes is specified here.
FS type	Select the desired filesystem type.
Impose inode limit	Specify whether to force the filesystem to have a limit of 65,536 inodes (files). This feature is necessary when making filesystems that are compatible with applications for some UNIX systems (such as System V releases prior to System V Release 4). Specifying No in this field instructs the filesystem to create as many inodes as appropriate for the filesystem's size. The default is Yes .
Mount file system:	Specify whether or not you want the filesystem to be mounted after creation. If the answer is Yes (the default), you must specify a mount point in the mount point field. All fields below this field are only accessible when Yes is specified here.
Mount point:	Enter the desired mount point for the new filesystem. If the specified mount point does not already exist, the Visual Administrator will automatically create it. This field is required if the filesystem is to be mounted.
vxfs mount option:	If vxfs is selected as the filesystem type, various parameters that affect filesystem performance can be specified when mounting a filesystem. Select one of the filesystem configurations provided, each of which represents a certain type of environment. Mount options suitable for the chosen environment will be run. This field is ignored if vxfs is not selected as the filesystem type.
Mount automatically:	Indicate whether this filesystem should be mounted every time the system comes up (by placing an entry in <i>/etc/vfstab</i>). The default is Yes .

Step 6. Click on **Apply**.

Exercise: Create a simple concatenated volume

Follow the steps in Section 7.1.8.1, “Creating a concatenated volume with the basic approach,” on page 7-29 to create a simple concatenated volume.

7.2 Lesson Summary

Basic Online Data Manager features are included with all UnixWare 7 editions. The Enterprise and Data Center editions also include a license for ODM advanced features. A separate license is required to enable advanced features of the ODM on other UnixWare 7 editions.

To understand the features and utilize the capabilities of the ODM, you must learn the basic terminology and operation principles of the ODM.

To use the VxVM, you must run **vxinstall** on your system. After running **vxinstall**, you can use the VxVM basic and advanced options to create volumes that leverage the features of the system including:

- Simple concatenated
- Striped
- RAID-5
- Mirrored
- Striped and Mirrored

Review questions

1. Which two UnixWare 7 editions include a license for the advanced features of the SCO Online Data Manager?
2. List and describe the two “basic” features of the VxVM available in all editions of UnixWare 7.
3. Which commands must you verify are running before you can successfully run the VxVA?

7.3 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

scoadmin(1M)	Invokes SCOadmin applications (in this lesson, the VERITAS Volume Manager)
tfadmin vxva	Command line to run the Visual Administrator in real mode
vxinstall	Initializes and/or encapsulates disks for use by VxVM

Relevant files

<i>/etc/swaptab</i>	Lists all the swap devices to be configured in the system
<i>/etc/vfstab</i>	Table of filesystem defaults
<i>/etc/vx/disks.exclude</i>	File you can create to exclude disks during vxinstall
<i>/etc/vx/cntrls.exclude</i>	File you can create to exclude disk on designated controllers during vxinstall

7.4 Answers to Review Questions

1. The Enterprise and Data Center UnixWare 7 editions include a license for the basic and advanced features of the SCO Online Data Manager product.

2. The "basic" version of the VxVM includes the following functionality:

Concatenation	Data is mapped linearly onto one or more subdisks in a plex
---------------	---

Spanning	Concatenation using subdisks that reside on more than one VM disk
----------	---

3. The VxVM processes **vxconfigd**(1M), **vxnotify**(1M), and **vxsparecheck**(1M) must be running before you can run the VxVA.

8

Installing and Managing Serial Ports

AV 8-1

The slide is titled "Installing and Managing Serial Ports" in a blue-bordered box at the top. Below the title is a bulleted list of eight tasks. At the bottom of the slide, there is a blue footer bar containing the text "sa2v0801" on the left and "© The Santa Cruz Operation, Inc." on the right.

- Identify standard serial ports
- Install a serial card
- Use multiscreens
- Describe the Service Access Facility (SAF)
- Configure serial ports
- Install serial terminals
- Manage logins
- Troubleshoot terminals

sa2v0801 © The Santa Cruz Operation, Inc.

Lesson 8

Installing and Managing Serial Ports

Introduction

Before other users can access your system, you must install the hardware and configure the software for serial ports and terminals.

To successfully administer and troubleshoot these ports and terminals, you also need to understand the login process and be familiar with the associated configuration files and utilities.

This lesson includes some background information on serial ports. For example, the first two sections discuss standard serial ports and describe how the terminal hardware communicates with the operating system. It also provides the information you need to manage and maintain your serial ports after installation and configuration.

Objectives

This lesson describes how to install, configure, manage, and troubleshoot serial ports and terminals. By the end of this lesson, you will be able to:

- Identify standard serial ports
- Install a serial card
- Use multiscreens
- Describe the Service Access Facility (SAF)
- Configure serial ports
- Install serial terminals
- Manage logins
- Troubleshoot terminals

AV 8-2

Standard Serial Port Naming Conventions			
	UnixWare 7	UnixWare2	OpenServer Release 5
COM1	/dev/term/00	/dev/term/00	/dev/tty1a
COM2	/dev/term/01	/dev/term/01	/dev/tty2a
COM3	/dev/term/02	/dev/term/02	/dev/tty3a
COM4	/dev/term/03	/dev/term/03	/dev/tty4a

sa2v0802 © The Santa Cruz Operation, Inc.

8.1 Identifying Standard Serial Ports

By default, UnixWare 7 systems include support for four standard serial ports. A serial port is used to connect serial devices such as terminals, modems, printers, and mice to your computer.

Standard serial ports with their device names include:

COM1	<i>/dev/term/00</i>
------	---------------------

COM2	<i>/dev/term/01</i>
------	---------------------

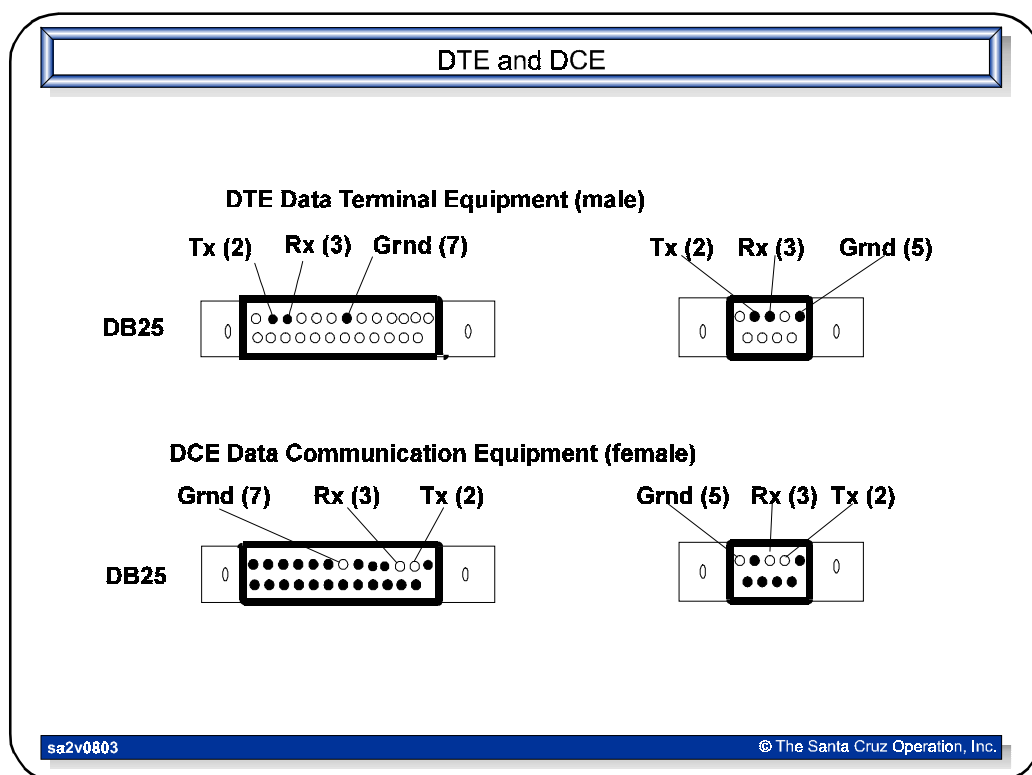
COM3	<i>/dev/term/02</i>
------	---------------------

COM4	<i>/dev/term/03</i>
------	---------------------

Note that most computers come equipped with only two (2) standard serial ports, COM1 and COM2.

The following information includes several factors that you need to consider before configuring serial ports.

AV 8-3



8.1.1 Serial port configuration

There are two types of serial ports configurations:

Data Terminal Equipment (DTE)	This type of serial port is usually found on the back of most computers, dumb terminals, and some printers.
Data Communications Equipment (DCE)	This type of serial port is found on the back of modems and some printers.

8.1.2 Flow control

Flow control (or handshaking) is a method used by a device that is sending data to another device faster than the other device can process the data. Flow control allows the sending device to pause while the receiving device catches up and prevents data loss.

Serial devices use two types of flow control:

- Hardware flow control (such as RTS/CTS) consists of control signal lines between the serial device and the computer. These signal lines are used to control the flow of data between the devices.
- Software flow control (such as XON/XOFF) use special control characters embedded in the data stream to control the flow of data between devices.

8.1.3 Asynchronous data communications

The serial port uses asynchronous communications. There is no predefined interval between sending characters. Therefore, the receiving device needs to be told when to expect the data. This is achieved by *framing* the data bits with start and stop bits. Start and stop bits indicate the beginning and ending of a character.

AV 8-4

Serial Port Speeds and UARTs	
300	
1200	
2400	
4800	
9600	Highest speed for 8250
19200	Highest speed for 16450 when used with fast modems
38400	
57600	
115200	Highest speed for 16550
9600	Highest speed for 8250
19200	Highest speed for 16450 when used with fast modems
38400	
57600	
115200	Highest speed for 16550

sa2v0804

© The Santa Cruz Operation, Inc.

The actual data bits usually number 7 or 8 in total. In addition, a *parity* bit is also added before the stop bit to provide error detection. There are three types of parity settings:

- No parity— No error detection is enabled
- Even parity
- Odd parity

8.1.4 UARTS

A serial port is controlled by a Universal Asynchronous Receiver and Transmitter (UART) chip. The UART:

- Converts parallel data into a serial data stream that can be transmitted over a serial line
- Converts received serial data to parallel data for further processing by the CPU
- Generates a hardware interrupt when a character is received and when the UART is ready for another character. The *trigger level* refers to the number of characters required in the UART's receive buffer before an interrupt is generated.

The serial drivers in UnixWare 7 support the 8250, 16450, 16550, 16650, and 16750 UART chips.

Earlier systems may have 8250 or 16450 UARTs. These UARTs do not support high speed modems. If your system is installed with one of these UARTs, you may want to consider upgrading your serial port.

8.1.5 Serial device naming conventions

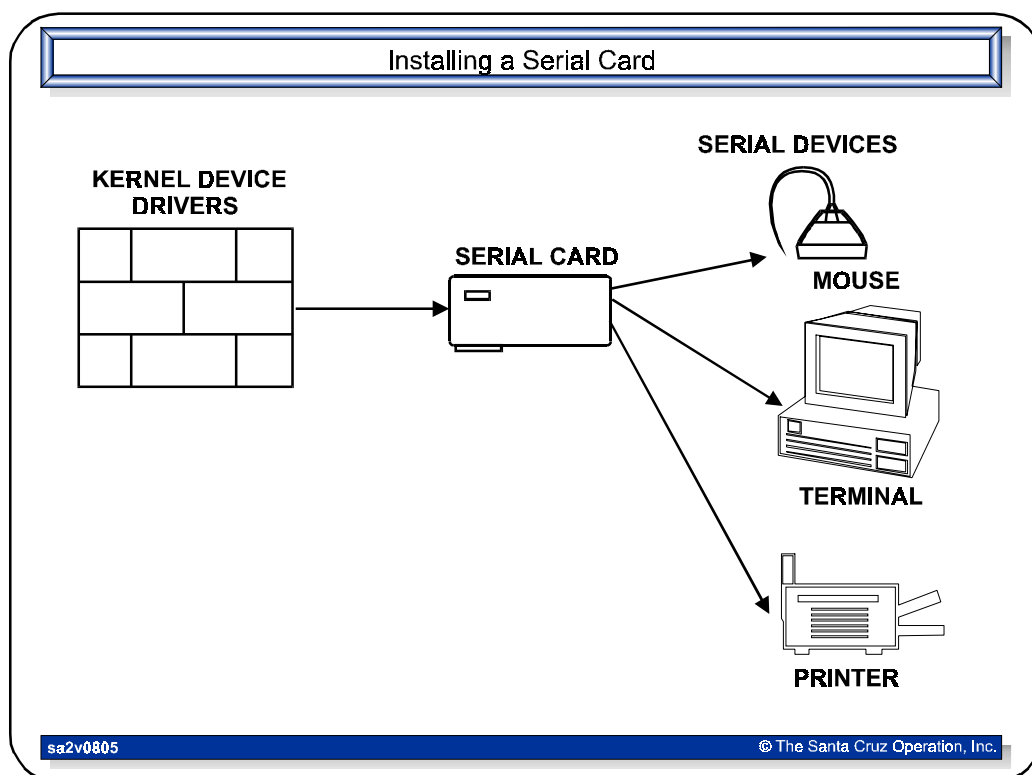
Each serial port is identified by four character device files. Different kinds of serial devices are associated with different device files. Previous releases of SCO UnixWare and SCO OpenServer device names are also supported.

Serial device files are stored in the `/dev/term` directory and include:

<code>/dev/term/0?s</code>	Software flow control (XON/XOFF) is enabled. Use this port when DCD is to be high and data integrity is not crucial, for example, with a mouse.
<code>/dev/term/0?h</code>	Hardware flow control (RTS/CTS) is enabled. Use to configure some serial printers.

<code>/dev/term/0?m</code>	No flow control. Used by the scoadmin(1M) Modem Manager to configure modems.
<code>/dev/term/0?t</code>	No flow control initially when opened. Allows port to be opened without DCD to be high. Use to configure dumb terminals that use software flow control.

AV 8-5



8.2 Installing a Serial Card

Most systems are equipped with two standard serial I/O ports: COM1 and COM2. Some systems also use ports COM3 and COM4.

You can connect serial devices such as a mouse, printer, or terminal to these ports.

To connect additional serial devices to your system, you must install at least one serial adapter card that provides more I/O ports. Serial adapter cards are also known as *multiport adapters*.

The number of serial ports that you can add depends on the card you use. Many cards can add between 8 and 16 ports.

8.2.1 Serial adapter cards

The two basic types of serial adapter cards are *dumb* cards and *smart* cards.

8.2.1.1 Dumb (standard or non-intelligent) cards

These cards:

- Usually convert one serial channel into many ports
- Have minimal on-board processing capabilities
- Rely on server CPU for I/O processing
- Are typically inexpensive
- Are good options for servers with 4 to 32 concurrent users
- Must be configured so that the kernel recognizes the card as COM1 or COM2 (usually, you can only install two dumb cards on a system)

8.2.1.2 Smart (intelligent) cards:

These cards:

- Have on-board processors and memory and use these resources to manage serial I/O through the server

- Include a Host Bus Adapter (HBA) diskette and configuration software that are installed with vendor-specific instructions. The method of installation is manufacturer-dependent.
- May or may not use interrupts, depending on the manufacturer

8.2.2 Installing the hardware

As with the installation of all adapter cards, make sure that there are no conflicts between the settings of this card and the following settings of any other installed card:

- DMA
- I/O address
- Base RAM
- Shared memory (for some cards)

Recall that a dumb serial card **MUST** duplicate one of the standard serial ports.

When you have installed a card, reboot your system. Enter the **Device Configuration Utility** (DCU) to verify that your card is recognized.

Serial device names are created automatically and placed in the */dev* directory.

Be sure to record the configuration information in your system administration log book.

8.3 Using Multiscreens

With the multiscreen feature you can access up to eight (8) different screens by default, each corresponding to a separate device file. Each one can be accessed one at a time through the console.

You can access the multiscreen by entering:

  <Fn>

where *Fn* is the number of one of the function keys located on the console.

Associated device names in the */dev* directory are */dev/vt01* through */dev/vt08*.

8.4 Describing the Service Access Facility (SAF)

Before you configure ports, you need to understand how UnixWare 7 monitors serial ports and multiscreens.

As an administrator, you are responsible for making your system accessible to authorized users. These users may be local (logged on to your system directly) or remote (logged on to your system through a network).

Both local and remote users gain access to a system through *ports*. These ports can be physical (serial) or symbolic (network).

The Service Access Facility (SAF) provides a mechanism for uniform access to port monitor services. Each physical port on a computer running UnixWare 7 has a monitor associated with it.

The monitor associated with each port:

- Grants or denies access
- Keeps track of usage
- Provides information about availability to administrative programs that need to use the port

Three port monitors are provided on UnixWare 7 systems:

inetd (1Mtcp)	Listens for connections on the Internet addresses of the services specified in its configuration file.
listen (1M)	Monitors ports when network users log in or access the system.
ttymon (1M)	Monitors ports when local users log in.

This lesson covers the **ttymon** port monitor. Refer to **scohelp** ⇌ **Manual pages** for information on the **listen**(1M) and **inetd**(1Mtcp) port monitors.

You can use the **sacadm**(1M) command to view the port monitors running on your system, enter:

sacadm -l

Exercise: Viewing port monitors

Enter the **sacadm** command to view the port monitors running on your system.

The PMTYPE field lists the port monitor.

The Service Access Facility (SAF) also provides a set of tools for installing, configuring, and querying port monitors through a utility called the Service Access Controller (SAC).

8.4.1 The Service Access Controller (SAC)

The Service Access Controller (SAC) is the administrative point of control for all port monitors. Its job is to maintain the port monitors on the system through:

Customization	During initialization, the SAC customizes its own environment by invoking the per-system configuration script supplied with the SAF.
Startup	Next, the SAC reads its administrative file to determine which port monitors are to be started.
Polling	After the SAC is running, it has two ongoing functions: (1) it polls its port monitors periodically, and (2) initiates recovery procedures if necessary.

The SAC is started by the **init(1M)** command when the system enters multiuser or the networking state from this entry in */etc/inittab*:

```
sc:234:respawn:/usr/lib/saf/sac -t 300
```

AV 8-6

The `/etc/saf/_sactab` File (Partial Extraction)

An extraction from the `/etc/saf/_sactab` file:

```
tcp:listen::3:/usr/lib/saf/listen -m inet/tcp0 tcp 2>/dev/null#  
inetd:inetd::0:/usr/sbin/inetd  
contty:ttymon::0:/usr/lib/saf/ttymon
```

sa2v0806

© The Santa Cruz Operation, Inc.

8.4.2 The **ttymon**(1M) port monitor

This section describes the role of **ttymon** in monitoring local access to the system using dumb terminals and console multiscreens.

The **ttymon** port monitor:

- Initializes and monitors ports
- Sets terminal modes, baud rates, and line disciplines for each port it monitors
- Writes a login prompt on the terminal or multiscreen and waits for user input
- Identifies and authenticates users

The **ttymon** port monitor is started by **sac**(1M) from the `/etc/saf/_sactab` file.

After **ttymon** is started, it monitors serial ports and multiscreens that contain an entry in the `/etc/saf` directory.

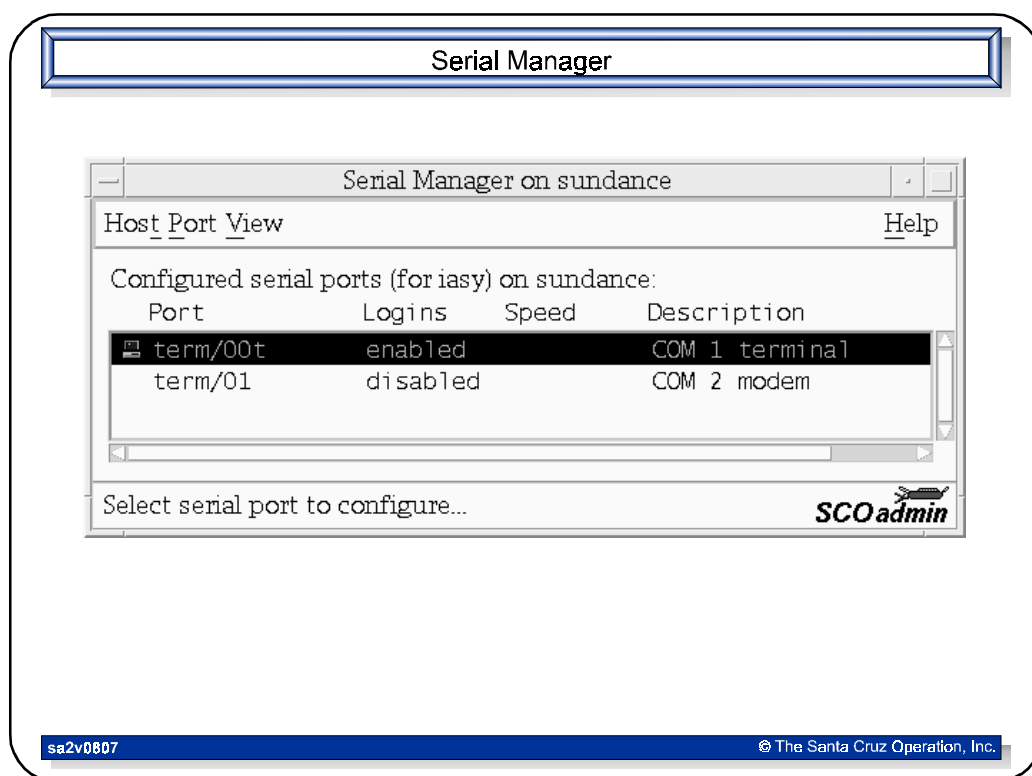
You can use the **pmadm**(1M) command to view the ports being monitored. Enter:

pmadm -l

Exercise: Viewing ports

Enter the **pmadm** command to view the ports being monitored on your system.

AV 8-7



8.5 Configuring Serial Ports

The **scoadmin(1M) Serial Manager** configures serial ports after they are installed. The **Serial Manager** is also a front end to the Service Access Controller (SAC).

You can use the **Serial Manager** to configure serial ports for use with:

- Dumb terminals
- Mice
- Modems
- Serial printers

After verifying that your serial board is recognized in the DCU, follow these steps to configure a serial port:

Step 1. Run **scoadmin(1M)**.

Step 2. Select **Hardware** ⇒ **Serial Manager**.

The **Serial Manager** displays the available serial ports. If a serial port is configured, an icon representing the device associated with the port displays. Note that you cannot modify a serial port that is associated with a mouse.

You can also view the serial boards that are installed in your system by selecting **View** ⇒ **Boards**.

Step 3. Select **Port** ⇒ **Modify** to modify a serial port.

Step 4. Select the **Port type** from one of the following:

- `software flow control (XON/XOFF)` if you are adding a serial mouse.
- `hardware flow control` if you are adding a serial printer that requires hardware flow control
- `terminal` if you are adding a dumb terminal
- `modem` if you are adding a modem

Step 5. Select how you want the port configured:

- Select **outgoing only** if the port is to be used for a dialout modem or serial printer.
- Select **incoming only** if the port is to be used with a dumb terminal for login purposes.
- Select **incoming and outgoing** if the port is to be used with a modem that is used for dialin and dialout purposes.

Step 6. Select the speed (bps) of the port from one of the following:

- If you select a baud rate, it should not be greater than the capabilities of the UART chip that controls the port
- If you select **AUTO**, the **ttymon(1M)** port monitor automatically detects and sets the port configuration

If you selected **incoming only** or **incoming and outgoing** in Step 5, and a baud rate in Step 6, proceed to Step 7.

If you selected **outgoing only**, proceed to Step 9.

Step 7. Select **Port Settings** to set the number of data bits per character and type of parity:

- Data bits— Choose from 7 or 8 bits. The default is 8 bits.
- Parity—Choose from **None**, **Even**, or **Odd**. The default is **None**.

Step 8. Select **OK** to accept the number of data bits and type of parity.

Step 9. Select **Receive Buffer** to set the trigger level of the UART chip. Recall that the trigger level refers to the number of characters required in the UART's receive buffer before an interrupt is generated.

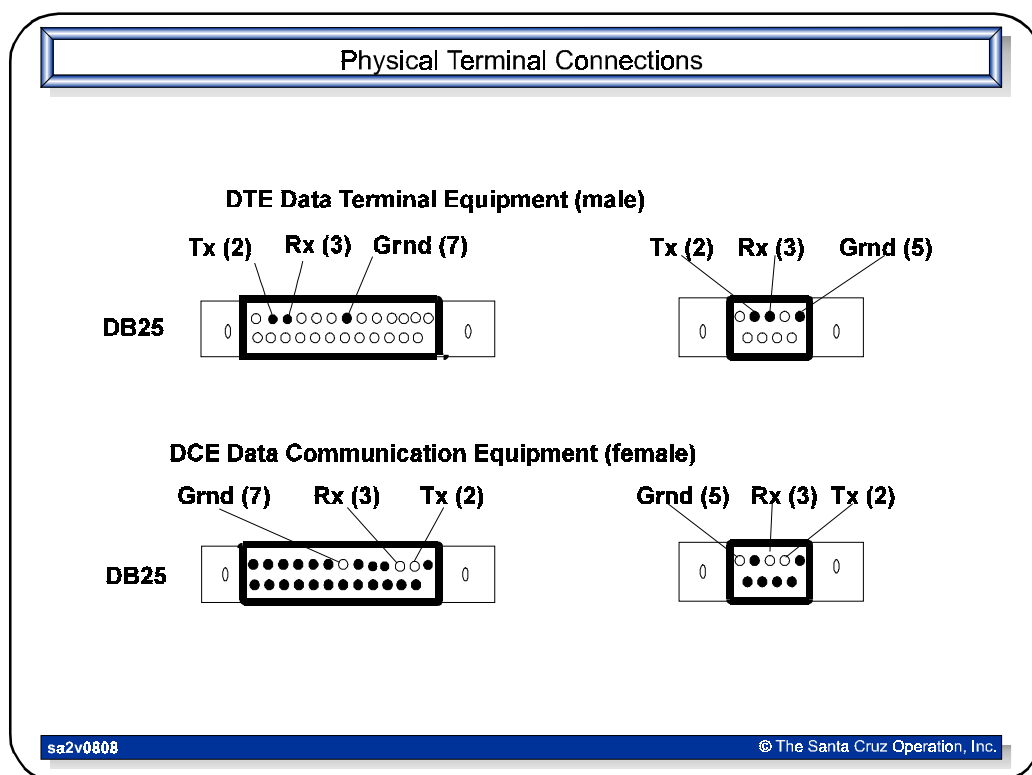
Do one of the following to set the trigger level. Move the sliding bar to:

- Select a lower value to alleviate data overruns (characters are being lost in incoming data)
- Select a higher value to improve performance by reducing the number of interrupts that a UART generates

If the **Serial Manager** is unable to read the current trigger level of a UART, a value of eight (8) is shown as the default.

- Step 10. Select **OK** to accept the trigger level.
- Step 11. Select **OK** to accept the serial port configuration.
- Step 12. Select **Host** ⇒ **Exit** to exit the **Serial Manager**.

AV 8-8



8.6 Installing Serial Terminals

Now that you know how to configure your serial port, you need to identify to the operating system the type of terminal that you are using.

This section describes terminal configuration and cabling.

8.6.1 About serial terminal hardware

Before installing a terminal, do the following:

- Examine the hardware manual for instructions on connecting the terminal to a serial line
- Refer to **scohelp** ⇒ **Manual pages** ⇒ **terminfo(4)** for a complete listing of supported terminal capabilities and characteristics

Support for terminals is provided through the **terminfo** database that contains the definitions and classifications of keystrokes and control sequences that vary from terminal to terminal.

8.6.2 Ports and cables

The two types of serial (RS-232) ports used for terminals are:

- DTE (Data Terminal Equipment)—for example, terminals
- DCE (Data Communications Equipment)—for example, COM ports

The manufacturer of the hardware determines the type of serial port.

Common types of connectors are:

- 25-pin (DB-25)
- 9-pin (DB-9)
- RJ-11 phone jack
- RJ-45
- Other vendor-specific tables

The physical connection between a terminal and a server usually requires three signals:

- Pin 2 (Tx—transmit data)
- Pin 3 (Rx—receive data)
- Pin 7 (Gnd—ground)

All other pins are not used by the operating system. For DTE terminals, the cable must be set up so that the Tx and Rx pins (pins 2 and 3) are reversed at one end. The transmit data pin at the computer end of the cable must connect to the receive data at the terminal end.

Some multiport board ports and terminal connections may be configured to allow the use of a straight through cable.

If you cannot reconfigure the port or terminal, you can add a *null-modem* device to the line to reorder the signals.

AV 8-9

Example Terminal Setup Screen

F10 - TOP LEVEL	ARROWS - NEW FIELDS	SPACE - NEXT CHOICE
BAUD RATE=9600	DATA/STOP BITS=8/1	PARITY=NONE
RCV HNDSHAKE=XON/XOFF	XMT HNDSHAKE=XON/XOFF	XPC HNDSHAKE=OFF
AUX BAUD RATE=9600	AUX DATA/STOP BITS=8/1	AUX PARITY=NONE
AUX RCV HNDSHAKE=DSR	AUX XMT HNDSHAKE=NONE	
F1 DISP	F2 GENERL	F3 KEYBRD
F4 COMM	F5 MISC	F6 TABS
F7 A/BACK	F8 F/KEYS	F9 LABELS
F10 EXIT		

sa2v0800

© The Santa Cruz Operation, Inc.

8.6.3 Terminal settings

Use the terminal setup screens to select specific settings for the:

- Baud rate
- Number of data bits
- Stop bits
- Parity
- Duplex

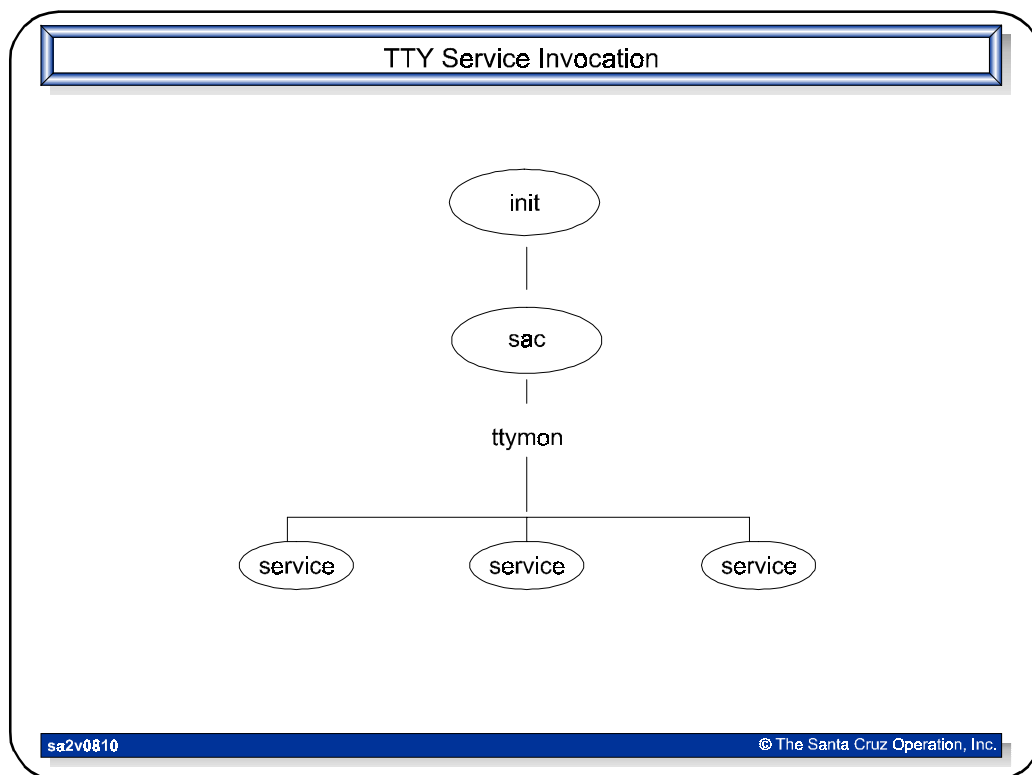
For example, you might be required to set your classroom terminal to:

- 9600 baud
- 8 data bits
- 1 stop bit
- no parity
- full duplex
- XON/XOFF handshaking

Exercise: Connecting your terminal

Plug in your terminal and turn it on. Plug the serial cable into the terminal and into the available serial port. Go to the setup screen for the terminal and make sure the settings are appropriate for your classroom setup, as presented by your classroom instructor.

AV 8-10



8.7 Managing Logins

Recall that **ttymon**(1M)

- Initializes and monitors ports
- Sets terminal modes, baud rates, and line disciplines for each port it monitors
- Writes a login prompt on the terminal and waits for user input
- Identifies and authenticates users

After **ttymon**(1M) invokes **login**(1), and the user logs in, it starts a service.

8.7.1 Logging in

When the Service Access Controller (SAC) starts **ttymon**, **ttymon** reads the (*/etc/saf/ttymon[1-9]/_pmtab*) file that contains information about which ports to monitor and what service (that is, process) is associated with each port.

The **shserv**(1M) service that enables users to login is started by **ttymon**. When a user logs in, the **shserv** process:

- Sets up the basic environment
- Changes directory to the user's home directory
- Execute's the user's shell

8.7.2 Setting the terminal type

Terminal types are set after the shell executes and reads the user startup files. To assign a terminal type, you can change the **TERM** variable:

- In the user's *.profile* for the Bourne and Korn shells, enter:

```
TERM=termttype;export TERM
```

- In the user's *.login* for the C shell, enter:

```
setenv TERM termttype
```

Setting the terminal type with the **TERM** variable assumes that the user is going to always log in at the same type of terminal.

The **tset(1)** command can use the terminal type entered in the */etc/ttytype* file. If a terminal type from */usr/lib/terminfo[a-z]* does not exist in */etc/ttytype*, the terminal type is set to **UNKNOWN**, and the user is prompted to enter the terminal type. The following is an example of the **tset** command entered in the user's *.profile*:

```
eval 'tset -m ansi:ansi -m: :\?${TERM:-wy60} -r -s -Q'
```

This line instructs the **tset** command to check for an **ansi** to label match first in the */etc/ttytype* file, and if there is not a match, prompt the user to enter a terminal type. If the user presses  at the prompt for a terminal type, the **wy60** label is set.

8.7.3 Disabling or enabling logins

If you need to disable or enable a terminal for login, follow these steps:

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **Hardware** ⇒ **Serial Manager**.
- Step 3. Select the serial port to disable or enable.
- Step 4. Select **Port** ⇒ **Modify**.
- Step 5. Select one of the following:
 - Select **outgoing only** to disable the port.
 - Select **incoming only** to enable the port.
- Step 6. Select **OK**.
- Step 7. Select **Host** ⇒ **Exit** to exit the **Serial Manager**.

You can also use the **pmadm(1M)** command.

For example, to disable multiscreen () for login, enter:

```
pmadm -d -p contty -s 8
```

To enable multiscreen () , enter:

```
pmadm -e -p contty -s 8
```

8.8 Troubleshooting Terminals

This section explains how to solve problems associated with terminals. The following are common types of problems associated with terminals.

- Terminal is completely non-functional

Login prompt is not displayed

Keyboard does not respond

If the above are true, the problem is usually hardware-related.

- Terminal is hung

Previous session is still visible

There is no session activity or keyboard response

If the above are true, the problem can be hardware-related, software-related, or both.

- Display is scrambled

Keyboard is responsive

Session is still active

If the above are true, the problem is usually software-related.

AV 8-11

Hung or Scrambled Terminal	
Hung Terminal	Scrambled Terminal
1. Check physical connections. 2. Type <code>CTRL q</code> 3. Press <code>DEL</code> or <code>CTRL \</code> 4. Kill process. 5. Turn off and on. 6. Check terminal configuration. 7. . Type <code>CTRL jstty sane CTRL j</code>	1. Log out and log back in. 2. Check terminal type. 3. Check terminal setup. 4. Type <code>CTRL jstty sane CTRL j</code>.

sa2v0811

© The Santa Cruz Operation, Inc.

8.8.1 Troubleshooting non-functional terminals

The following steps describe how to troubleshoot non-functional terminals:

- Step 1. Check the power and cable connections at the terminal and CPU.
- Step 2. Check the brightness control.
- Step 3. Enter the terminal's setup mode and verify the configuration, making sure that all settings match those in the **scoadmin(1M) Serial Manager**.
- Step 4. Verify the hardware connection by redirecting output from another terminal. For example, enter:

```
yes > /dev/term/00t
```

Redirects the **yes(1xnx)** command to COM1.

- Step 5. Run the **ps(1)** command and verify that the **ttymon(1M)** port monitor is running.

8.8.2 Troubleshooting hung or scrambled terminals

The following are some troubleshooting guidelines for hung or scrambled terminals:

- Press **DEL** or **CTRL ** to terminate the process or issue the **kill(1)** command from another login session to terminate the process
- Recycle power to the terminal to reset the terminal hardware
- Check the terminal configuration and compare to the **Port settings** in the **Serial Manager**
- Check the user's terminal type using the **env(1)** command. If the **TERM** variable is incorrect, either manually set **TERM** or source the user's *.profile* file. After **TERM** is set correctly, run **tset(1)** with no arguments to reinitialize the terminal.
- Regain a reasonable set of serial-line characteristics with **stty sane**:

```
CTRL j stty sane CTRL j
```

8.9 Lesson Summary

Terminals or modems attached to serial ports allow other users to access your system. By default, UnixWare 7 provides support for the console and up to four standard serial ports.

If you want to install additional ports, you must install a serial multiport card. There are two types of multiport cards, dumb and intelligent.

Serial ports are configured using the **scoadmin(1M) Serial Manager**.

Review questions

1. Which communications flow control uses special control characters embedded in the data stream to control the flow of characters?
2. Which serial device file represents the COM1 device file that has software flow control enabled?
3. Which programs can you enter to confirm that UnixWare 7 recognizes the installed serial card?
4. What is the role of the **ttymon(1M)** port monitor?
5. Which **Port type** would you select in the **scoadmin(1M) Serial Manager** for a modem that is used for dialout purposes only?
6. Which service is started by **ttymon(1M)** after a user logs in?

8.10 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

dcu (1M)	Device Configuration Utility (DCU)
init (1M)	Process control initialization
listen (1M)	Network listener port monitor
pmadm (1M)	Port monitor administration
sac (1M)	Service Access Controller (SAC)
sacadm (1M)	Service access controller administration
scoadmin (1M)	scoadmin (1M)Invokes SCOadmin applications (in this lesson, the Serial Manager)
ttymon (1M)	Port monitor for terminal ports

Relevant files

<i>/etc/saf</i>	Directory that contains an entry for the serial ports that are monitored by ttymon (1M)
<i>/etc/saf/_sactab</i>	File from which the ttymon (1M) port monitor is started

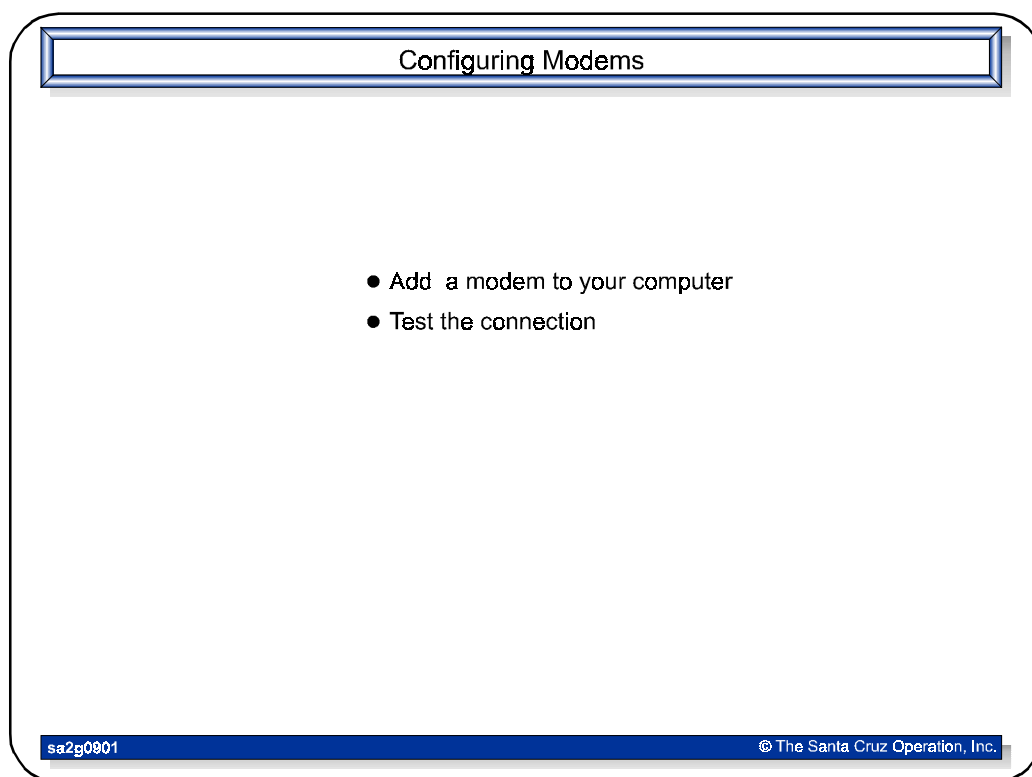
8.11 Answers to Review Questions

1. The XON/XOFF software communications flow control uses special control characters embedded in the data stream to control the flow of characters.
2. The `/dev/term/00s` serial device file represents the COM1 device file that has software flow control enabled.
3. You can enter the Device Configuration Utility (DCU) and the Serial Manager to confirm that UnixWare 7 recognizes the installed serial card.
4. The role of the **ttymon**(1M) port monitor is to initialize and monitor ports; set terminal modes, baud rates, and line disciplines for each port; write a login prompt on the terminal and wait for input; and identify and authenticate users.
5. You would select the Port type “outgoing only” in the **scoadmin**(1M) Serial Manager for a modem that is used for dialout purposes only.
6. The **shserv**(1M) service is started by **ttymon**(1M) after a user logs in.

9

Configuring Modems

AV 9-1



Lesson 9

Configuring Modems

Introduction

As terminals let local users communicate with your system, modems enable you to communicate over phone lines and ISDN channels. Because many modem types are supported under UnixWare 7, the **scoadmin(1M) Modem Manager** lets you easily configure your modem.

Objectives

This lesson describes how to connect your modem to your computer and test the connection. By the end of this lesson, you will be able to:

- Add a modem to your computer
- Test the connection

9.1 Adding a Modem

Modems enable you to communicate over phone lines with remote systems. You must configure and install a modem before you can set up your system for use with serial communications or networking software.

To set up a modem on your system, you need to:

- Choose a serial port for connecting the modem
- Configure the modem
- Install the modem

9.1.1 Choosing a serial port

Choose the `/dev/term/??m` device file when configuring modem ports. If necessary, you can change serial port settings when you install the modem.

Intelligent (smart) multiport serial cards are supplied with their own drivers and have a variety of device naming schemes. Consult the documentation that came with your intelligent card for a list of device names.

9.1.2 Modem support

Any external 100% Hayes-compatible modem should work provided that it is connected to a serial port that is capable of supporting modem control protocols. Some third-party multiport smart cards cannot be used as they do not support modem control.

Several Hayes-compatible internal and PCMCIA modems are also supported. Some models of older internal modems have compatibility problems that prevent them from working properly. For example, they can flood the system bus with spurious interrupts, or they may not support the quiet, no-echo, or auto-answer modes.

Refer to the SCO Hardware Compatibility Web pages for a list of supported modems.

9.1.3 Connecting the modem

The following two sections describe how to connect your modem to your computer.

9.1.3.1 Pins and Cables

Recall that a three wire cable is used to connect a dumb terminal. These types of cables cannot be used to connect a modem. Use a straight through cable between your modem and computer to connect:

Pin 2 (Tx—Transmit Data)

Pin 3 (Rx—Receive Data)

Pin 4 (RTS—Request To Send)

Pin 5 (CTS—Clear To Send)

Pin 7 (Gnd—Ground)

Pin 8 (CD—Carrier Detect)

Pin 20 (DTR—Data Terminal Ready)

Physical connections between a device and the system may depend on the hardware configuration. For specific information, refer to the hardware manuals provided with your modem and computer.

9.1.3.2 Modem setting guidelines

The factory default settings are usually suitable for installing a modem. Do not adjust these settings until you have tested to see if the modem operates successfully with your system.

If you should have to change settings, follow these guidelines:

- Set the modem to auto-answer
- Set the modem so that it does not answer when Data Terminal Ready (DTR) is low, and so that it drops the current connection when DTR goes from high to low
- Set Data Carrier Detect (DCD) to go high when a carrier is present, low when a carrier is not present
- Turn off echoing of characters sent to the modem
- Select quiet mode so that the modem sends no response codes

9.1.4 Configuring the modem

Use the **scoadmin(1M) Modem Manager** to configure your modem.

After you have connected your modem, follow these steps to configure your modem:

Step 1. Turn on your modem.

Step 1. Run **scoadmin(1M)**.

Step 2. Select **Hardware** ⇒ **Modem Manager**.

Step 3. Select **Modem** ⇒ **Add**.

Step 4. Choose one of the following ways to install your modem:

- **Automatic Detection**—The **Modem Manager** can automatically detect the type of modem that is connected to your system. Proceed to Step 5.
- **Manual Detection**—You select the type of modem that is connected to your modem. Proceed to Step 6.

Step 5. Select the serial port to which the modem is attached:

- If you selected **Auto Detection** in Step 4, select **ALL** to have the **Modem Manager** to search all serial ports.
- If you did not configure the serial port with the **scoadmin(1M) Serial Manager**, select **Configure port**.

After you have configured your serial port, select **Detect** ⇒ **OK**. Proceed to Step 7.

Step 6. Select from the following:

- **Modem Vendor**—Select the manufacturer of your modem. If it is not listed, select **Standard Modem Types**.
- **Modem Model**—Select the modem model. If you selected **Standard Modem Types** for the modem vendor, select the appropriate **Standard speed bps Modem**.
- **Modem Port**—Select the serial port to which the modem is attached. If the serial port is not configured, select **Configure port**.

Step 7. Click on **OK** to accept the configuration.

Step 8. Select **Host** ⇒ **Exit** to exit the **Modem Manager**.

You can also modify the modem setup or remove the modem using the **Modem Manager**.

If your modem requires a special configuration, refer to the modem scripts stored in the */etc/uucp/default* directory.

9.2 Testing the modem connection

Follow these steps to test whether your modem responds and that you can dial out.

Step 1. Change the ownership of the serial port to **uucp**:

```
chown uucp /dev/term/??m
```

```
chgrp uucp /dev/term/??m
```

where ?? is the number of the corresponding line.

Step 2. Test the modem to ensure that it dials correctly. For example, use the **cu(1bnu)** command to dial using */dev/term/00m*:

```
cu -l /dev/term/00m
```

You should see a message indicating that you are connected.

Step 3. Enter: **AT**

The screen should echo **OK** (if your screen is set to return result codes as text) or **O** (if your screen is set to return results as numeric codes).

Step 4. Test the modem for dial-out:

```
ATDTphonenum (for tone dial phone connection)
```

```
ATDPphonenum (for pulse dial phone connection)
```

where *phonenum* is the number of the machine to which you are trying to dial out.

Step 5. Enter **~.** to disconnect.

Refer to **scohelp** ⇔ **Hardware** ⇔ **Adding and configuring modems** ⇔ **Configuring UUCP for modems** for information on how to configure **uucp(1bnu)**.

9.3 Lesson Summary

Use the **scoadmin(1M) Modem Manager** to configure your modem. Verify the modem's connection using the **cu(1)** command.

Review questions

1. Which serial device file is used when you attach a modem?
2. Which option to the **cu(1bnu)** command lets you directly access the serial port?

9.4 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

chown (1)	Changes the owner ID of a file
chgrp (1)	Changes the group ID of a file
cu (1bnu)	Calls up another UNIX system
scoadmin (1M)	Invokes SCOadmin applications (in this lesson, the Modem Manager)

Relevant files

<i>/etc/uucp/default</i>	Contains modem dialer scripts
--------------------------	-------------------------------

9.5 Answers to Review Questions

1. The `/dev/term/??m` serial device file is used when you attach a modem.
2. The `-l` option to the `cu(1bnu)` command lets you directly access the serial port.

10

Installing and Managing Printers

AV 10-1

Installing and Managing Printers

- Install a parallel card
- Install parallel and serial printers
- Add local and remote printers
- Describe the UNIX print service
- Manage print request directories
- Set print request priorities
- Manage printer interface scripts
- Troubleshoot the UNIX print service

sa2v1001© The Santa Cruz Operation, Inc.

Lesson 10

Installing and Managing Printers

Introduction

After terminals and modems are installed and configured on your system, users can log in and access its data. It is likely that these users will want to print out their data. You will need to install at least one printer for them to do so.

The UNIX print service is a collection of utilities that allow you to configure, monitor, and control the printers on your system. Installing and managing both local and network printers on your UnixWare 7 system is an important part of your duties as system administrator.

Objectives

This lesson describes how to install and troubleshoot printers and manage the UNIX print service. By the end of this lesson, you will be able to:

- Install a parallel card
- Install parallel and serial printers
- Add local and remote printers to your system
- Describe the UNIX print service
- Manage the print request directories
- Set print request priorities
- Manage printer interface scripts
- Troubleshoot the UNIX print service

10.1 Installing a Parallel Card

Most systems come with one parallel port installed, but UnixWare 7 supports up to two. The supported ports are:

- Serial/parallel adapter #1:
 - Device special filename: `/dev/lp0` or `/dev/lp`
 - Address: 0x378-37f
 - IRQ 7
- Serial/parallel adapter #2:
 - Device special file name: `/dev/lp2`
 - Address: 0x278-27a
 - IRQ 5 (or IRQ 7 on MCA)
- Parallel adapter:
 - Device special file name: `/dev/lp1`
 - Address: 0x3bc-3be
 - IRQ 7

As with the installation of all adapter cards, make sure that there are no conflicts between the settings of this card and any other installed card. When possible, select the default device interrupts and settings.

Add a parallel card by following these steps:

- | | |
|---------|--|
| Step 1. | Shut down your system. |
| Step 2. | Install your parallel card. |
| Step 3. | Boot your system to single-user mode. |
| Step 4. | Enter the Device Configuration Utility (DCU) and verify that the card is recognized. |

If the card is not recognized:

- Check for IRQ or address conflicts in the DCU
- Shut down your system and verify that the card is seated and that cables are connected properly, etc.
- Repeat Steps 1 through 4

If the card is recognized, proceed to Step 5.

Step 5. Continue booting your system to multiuser mode.

10.2 Installing Parallel and Serial Printers

After you install a parallel or serial port for a printer, you are ready to install your printer. Recall that you learned how to install serial ports earlier in Lesson 9, “Installing and Managing Serial Ports.”

If you are installing a parallel printer, you must use a standard parallel interface cable. If you are not sure if the printer is a parallel printer, refer to the printer documentation.

If you are installing a serial printer, note that serial printers must be capable of supporting software flow control XON/XOFF or hardware flow control RTS/CTS.

These steps describe how to install a parallel or serial printer.

- Step 1. Power down your system.
- Step 2. Connect the printer:
 - For a parallel printer, plug the parallel cable into the printer port and the system parallel port. Be careful not to bend any pins.
 - For a serial printer, connect the RS-232 cable from the selected serial port on your system to the serial port on your printer
- Step 3. Turn the printer on.
- Step 4. Power up the system.

Exercise: Installing parallel or serial printer hardware

If you have a parallel or serial printer to install, follow the directions above and install it now.

10.2.1 Verifying the hardware installation of a parallel or serial printer

Verify the hardware installation of a parallel or serial printer as follows:

- For a parallel printer, enter:

```
cal > /dev/lpn
```

where n is the number of the parallel port you are using, for example, `/dev/lp0`.

- For a serial printer, enter:

```
cal > /dev/term/??s
```

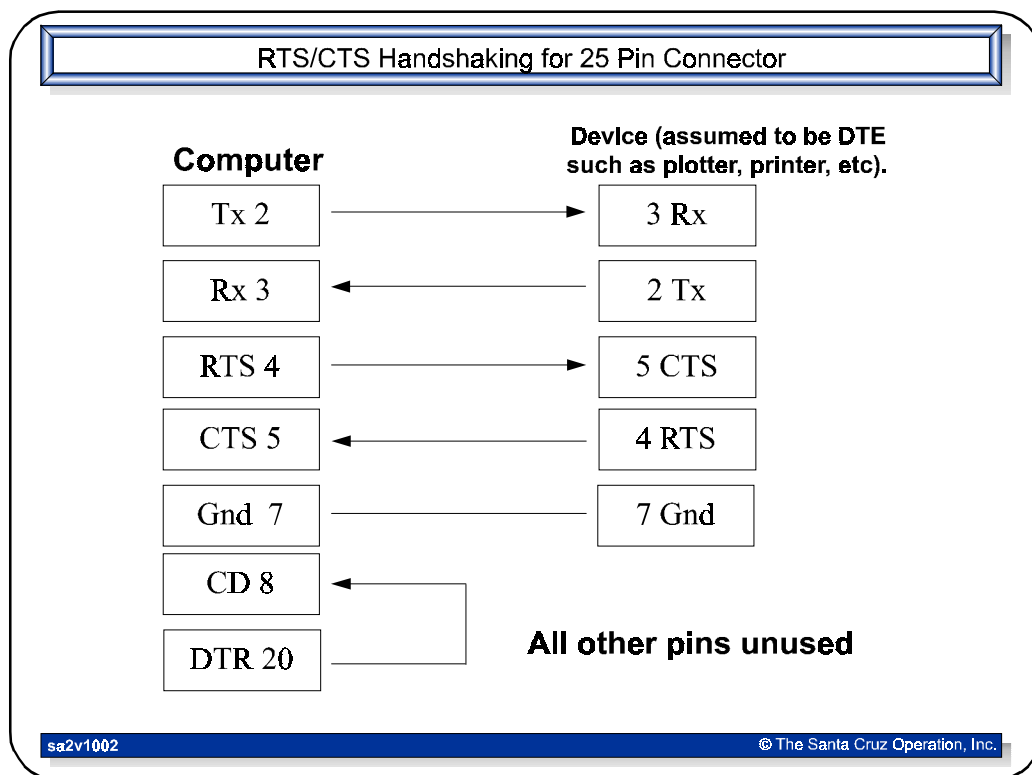
where ?? is the number of the serial port you are using, for example, `/dev/term/00s`.

If the calendar prints on the printer, the physical connection is good.

Exercise: Verifying the parallel printer hardware connection

If you installed a parallel or serial printer, follow the steps outlined in “Installing Parallel and Serial Printers” and verify that the printer hardware connection and cable are correct.

AV 10-2



10.2.2 Setting up RTS/CTS protocol serial printers

If you are setting up an RTS/CTS protocol serial printer, use a hardware flow control port such as */dev/tty01h*.

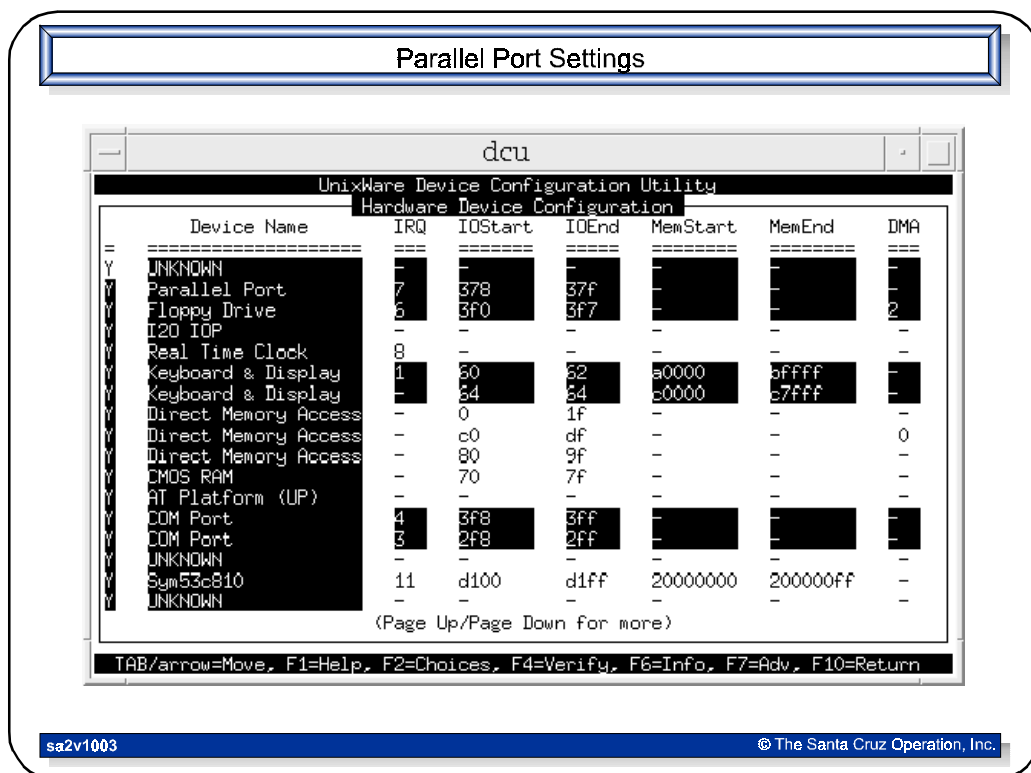
To verify the hardware installation of an RTS/CTS protocol serial printer, enter:

```
cal > /dev/term/??h
```

where ?? is the number of the serial port you are using, for example, */dev/term/00h*.

If the calendar prints on the printer, the physical connection is good.

AV 10-3



10.2.3 Troubleshooting Printer Installations

Use the following guidelines for troubleshooting printer installations:

- Verify that your cable is securely connected at both ends
- Test the cable on a parallel printer by connecting it to another server and parallel printer that are currently working
- Recheck your printer configuration by verifying its switch settings; you must use the printer hardware manual to do this
- Recheck the switch settings on your card—it must be recognized in the Device Configuration Utility (DCU)
- If the card is recognized, the interrupt vector that the card is using is listed in the third column. Make sure that it does not conflict with any other hardware in the system.
- Check and verify that you selected the correct serial device file and port settings in the **scoadmin(1M) Serial Device Manager**.

10.3 Adding Printers

Use the **Printer Setup Manager** to add a local printer to your system's print services.

10.3.1 Adding a local printer

Add a local printer (parallel or serial) by performing these steps:

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **Printer Setup Manager** ⇨ **Printer** ⇨ **Add Local Printer**.
- Step 3. Enter the printer name.
- Step 4. Select the printer model. Click on the **Connection Type**.

If it is a parallel printer, click on **Parallel**.

If it is a serial printer, click on **Serial** ⇨ **Serial Settings**. If your serial settings differ from the displayed settings, change the following:

- **BAUD**—Select the applicable baud rate. The default is 9600 baud.
- **Parity**—Select **Even**, **Odd**, or **None**. The default is None.
- **Stop Bits**—Select **1** or **2**. The default is 1.
- **Character Size**—Select **8** or **7**. The default is 8.

- Step 5. Click on **OK**.
- Step 6. Select **Yes** to receive a mail message if the printer fails.
- Step 7. Select **Yes** to allow users to print without a banner page.
- Step 8. Select **Advanced Options** if you are installing a non-PostScript® printer. You can set the:
- **Page Length**
 - **Page Width**
 - **Character Pitch**

- Line Pitch

You can select a unit value in inches, centimeters, or characters.

- Step 9. Click on **OK** to accept the settings entered in Step 8.
- Step 10. Click on **OK** to accept all printer configuration settings.
- Step 11. Click on **OK** to add the printer.

The printer appears in the **Printer Setup** list. In addition, the printer is automatically enabled. If you want to enable remote print services from the **Printer Setup** window, you must also do the following:

Select **Printer** ⇒ **Set Remote Access** ⇒ **Enter** or select the host(s) that can have access to this printer. Click **OK**.

- Step 12. Select **Printer** ⇒ **Make Default** to make this printer the default printer.
- Step 13. Select **Host** ⇒ **Exit** to exit the **Printer Setup Manager**.

10.3.2 Adding a TCP/IP printer

If your system is connected to a TCP/IP network and you want to access a printer that is also connected to the network, add a TCP/IP printer by performing these steps:

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **Printer Setup Manager** ⇒ **Printer** ⇒ **Add TCP/IP Printer**.
- Step 3. Enter the printer name.
- Step 4. Select the printer model.
- Step 5. Select the **Protocol** (spooler type) on the remote host. Spooler types include:
- **lpd (BSD)**—The default networking printing protocol
 - **System V (Legacy)**—Provided for backwards compatibility with older UNIX systems

For HP printers configured **On Network**, System V is automatically selected.

- Step 6. Select **On Network** if the printer is directly connected to a network. Or if the printer is connected to a computer on the network, select **On Remote Server**.
- Step 7. Enter the name of the host that controls the remote printer or Click on **Select** to choose from a list. The **Remote System** field does not display if you selected **On Network** in Step 6.
- Step 8. Enter or select the name of the remote printer if you selected **On Network** in Step 6.
- Step 9. Click on **OK**.
- The new remote printer appears in the **Printer Setup** list on the local host.
- Step 10. Select **Host** ⇒ **Exit** to exit the **Printer Setup Manager**.

Your local host must be listed in the */etc/hosts.lpd* or the */etc/hosts.equiv* files on the remote host before the remote printer can accept jobs from your local host. The printer on the remote host must also be configured to accept remote jobs, and UNIX remote print service must be enabled on both the local and remote hosts.

10.3.3 Adding a NetWare printer

If your system is connected to a Novell NetWare network and you want to access a printer that is also connected to the network, add a NetWare printer by performing these steps:

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **Printer Setup Manager** ⇒ **Printer** ⇒ **Add NetWare Printer**.
- Step 3. Enter the printer name.
- Step 4. Select the printer model.
- Step 5. Enter the name of the NetWare server that controls the remote printer or click on **Select** to choose from a name of existing NetWare servers.
- Step 6. Enter the name of the printer queue on the remote system or click on **Select** to choose from a list of existing printer names.
- Step 7. Enter the name of the host that controls the remote printer or click on **Select** to choose from a list.
- Step 8. Click **OK** to accept the configuration.

10.4 Describing the UNIX Print Service

The background daemon responsible for the print service is **lpsched**(1M), also known as the print scheduler. The print scheduler starts each time the system goes into multiuser mode. There is an **lpsched** process for each print request currently printing.

10.4.1 Who can administer the print service?

These users can administer the print service:

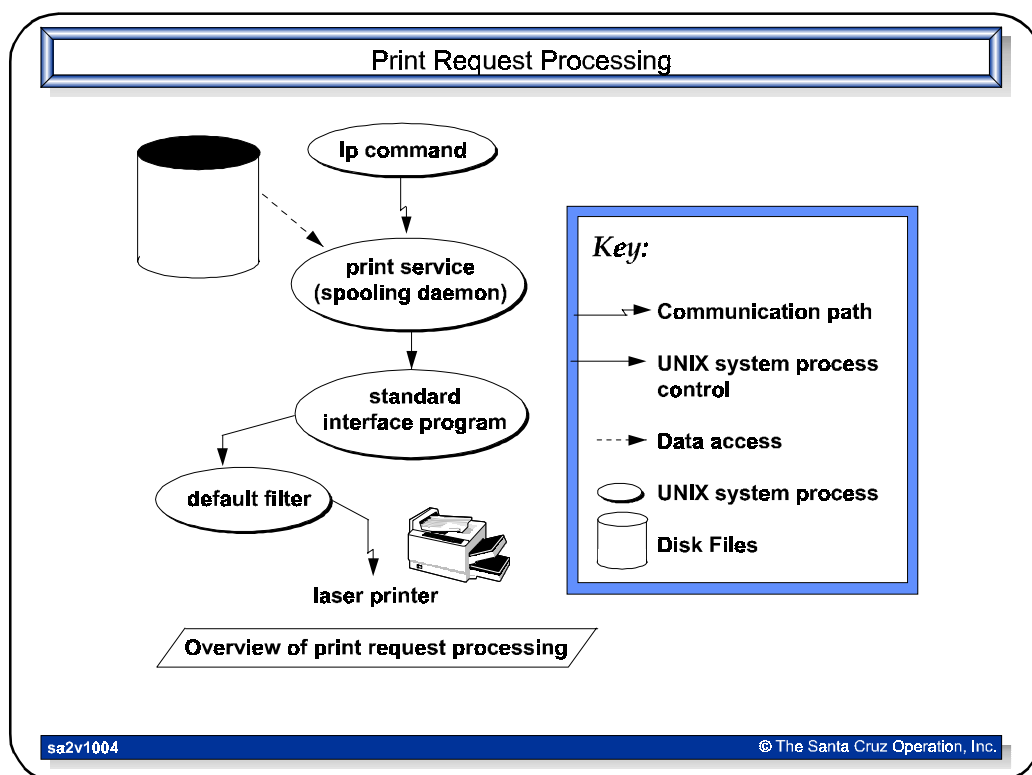
- The **lp** account
- The **root** account
- The **system owner** account

10.4.2 Print scheduler functions

The print scheduler does the following:

- Accepts files for printing
- Schedules the work of one or more printers
- Starts programs that interface with the printer
- Monitors the status of print jobs
- Issues error messages when problems arise

AV 10-4



10.4.3 Print request process

The following describes the flow of a UNIX print request process:

1. A user requests to print a file.
2. The print scheduler assigns the request a unique name and *request ID*. The request ID contains the name of the designated printer and a unique number identifying it in the print queue, such as *laser-17*.
3. The pointers to the print job are then spooled (stored on disk) with other print job pointers where they await their turn in line to be printed.

AV 10-5

Viewing the Status of the Print Scheduler with `lpstat -t`

```
scheduler is running
system default destination: laser
device for printer: /dev/lp0
laser accepting requests since Fri Feb 3 13:12:22 1995
printer laser enabled since Wed Feb 8 08:16:29 1995. available.

laser-5      root      1246      Feb  8  08:16  on laser
laser-6      johnsp    1246      Feb  8  08:16
laser-7      aaronm    1246      Feb  8  08:17
```

sa2v1005

© The Santa Cruz Operation, Inc.

10.4.4 Viewing the print scheduler

Use the **lpstat(1)** command with the **-t** option to view the status of the print scheduler. The following is a description of information displayed:

scheduler is running	The print service is operational
system default destination	The name of default printer
device for printer	The physical serial or parallel port to which the printer is attached
printer is accepting requests	Indicates that users can submit print requests to the printers listed
printer is enabled	The printers listed will print accepted requests
printer is idle	The printers listed are not currently printing but are available to print
printername-ID	The print jobs waiting to print
on printername	The jobs currently printing

Exercise: Viewing the print scheduler

Run **lpstat -t**.

What is the name of your default printer?

10.5 Managing the Print Request Directories

The `/var/spool/lp/tmp/machine_name` and the `/var/lp/logs` directories describe each request that has been submitted to the print service. You can display the information that is stored in the `/var/spool/lp/tmp` directory with the **lpstat(1)** command.

The user who submitted the job has access to the request file in `/var/spool/lp/tmp`; only **lp** or **root** has access to the file in `/var/lp/logs`.

For example, a print request assigned the job number 17 would have the following entries:

<code>/var/spool/lp/tmp/machine_name/17-0</code>	Contains the absolute pathname of the file to be printed and scheduling information.
<code>/var/spool/lp/tmp/machine_name/17</code>	Contains error messages encountered while trying to print.
<code>/var/spool/lp/tmp/machine_name/F17-1</code>	Contains a filtered file, e.g., PostScript code.
<code>/var/spool/lp/tmp/machine_name/17-1</code>	Contains an unfiltered print file when source is from stdin or another host.
<code>/var/lp/logs/requests.1</code>	After the print job has completed printing, files are removed from <code>/var/spool/lp/tmp/machine_name</code> and the information is appended to <code>/var/spool/lp/logs/requests.1</code> .

AV 10-6

Partial Request Log Entries

Letter	Content of line
C	Number of copies printed
D	Printer or class destination or the word "any"
F	Filename in <code>/user/spool/lp/temp</code> printed. This line repeated for each file printed, and files are printed in the order given.
N	How the print service notified the user after printing the file. • M by an electronic mail message • W by a message written to the user's terminal
O	Any -o options given to lp(C)
P	Priority of the print request
s	Outcome of the job, expressed as a combination of individual bits in hexadecimal form. The important bits used internally by the spooler are: 0x0004 Slow filtering finished successfully. 0x0010 Printing finished successfully. 0x0040 Request was cancelled. 0x0100 Request failed filtering or printing.
U	Name of the user who submitted the print request.

sa2v1006

© The Santa Cruz Operation, Inc.

Examining the print request log

The request log has a simple structure that makes it easy to extract data using UNIX system commands. For example, to evaluate whether a second printer is needed for a group of users, use the **grep**(1) command and search for the “gid” to determine the number of print requests submitted by that group.

The request log entries are listed in the order in which they were printed, containing the request ID, the user and group IDs of the user, the total number of bytes in the file, and the time the request was queued. Other important entries include:

- C The number of copies printed
 - D The printer destination
 - F The name of the file printed
 - s The outcome of the request
- 0x0010 - Printing finished successfully
- 0x0040 - The request was canceled
- 0x0100 - The request failed printing

Refer to **scohelp** ⇒ **Printers** ⇒ **Advanced printer configuration** ⇒ **About the print request log** for a complete definition of all print request log entries.

Exercise: Determining the contents of a print request log

1. Disable your classroom printer. Your instructor will provide the printer name.
2. Print the */etc/default/tar* file. The file remains in the queue because the printer is disabled.
3. Change directory to */var/spool/lp/tmp/machine_name*.
4. Examine the contents of your print request ID file.
5. Change the directory to */var/lp/logs*.
6. Examine the contents of the *requests* file.
7. Cancel the print request.
8. Re-examine the contents of */var/lp/logs/requests.1*

10.6 Setting Print Request Priorities

The print service provides a simple priority mechanism that lets users submit print requests with a new priority or change the priority of a print request residing in the queue.

A user can be assigned a priority limit between 0 and 39. A lower number indicates a higher priority. To ensure that the priority scheme works, **root** or the **lp** administrator controls how high users can set the priority when submitting or changing a print request.

10.6.1 Default priority level and limits

You can use the **lpusers(1M)** command to set print priorities and limits.

To determine how the current priorities and limits are set, enter:

```
/usr/sbin/lpusers -l
```

If the default priority is 20, any print job submitted without requesting a new priority has a priority of 20.

If the default priority limit is 0, any user can submit a print job requesting a priority between 0-39.

10.6.2 Changing the default priority for all users

You can assign a default priority for the balance of users not assigned a personal limit:

```
/usr/sbin/lpusers -d new_priority
```

For example:

```
/usr/sbin/lpusers -d 30
```

Users not assigned a personal priority limit can submit print jobs requesting priorities between 30-39.

Exercise: Changing the default priority

Change the default priority so that all users can request a print priority between 25 and 39 when submitting print requests.

10.6.3 Setting a personal priority limit for a user

You can set personal priority limits for a user or group of users:

```
/usr/sbin/lpusers -q priority_limit -u username
```

For example:

```
/usr/sbin/lpusers -q 10 -u kellyk
```

The user **kellyk** can submit a print request with a priority between 10-39.

10.6.4 Restoring the default priority limit

You can restore the default priority limit for a user:

```
/usr/sbin/lpusers -u username
```

For example:

```
/usr/sbin/lpusers -u kellyk
```

10.6.5 Submitting a print job with a different priority

You can submit a print request using a different priority than the default:

```
lp -q new_priority filename
```

For example:

```
lp -q 10 /etc/default/tar
```

10.6.6 Changing the priority of a print job residing in the queue

You can reassign a new priority for print requests that are still waiting to print:

```
lp -i request-ID -q new_priority
```

For example:

```
lp -i laser-17 -q 10
```

This repositions the request in the queue, putting it ahead of lower priority requests, but behind any others at the same higher priority.

10.6.7 Bypassing all other awaiting print requests

The **root** user can bypass all awaiting print requests.

To bypass all awaiting print requests residing in the queue, enter:

```
lp -H immediate filename
```

For example:

```
lp -H immediate /etc/default/tar
```

If the printer is currently printing, the request is the next job printed.

10.7 Managing Printer Interface Scripts

A printer interface script is a program the print service uses to manage the printer each time a file is printed. The interface script:

- Initializes the physical printer port
- Initializes the physical printer (restores the printer to a normal state in case a previously printed file has left it in an unusual state, that is, restores character or line pitch, page size, and character sets)
- Prints a banner page, if required
- Processes options to the **lp(1)** command
- Reports any errors encountered by the print service

The system contains model scripts; however, the interface script *standard* can be chosen for most industry-standard serial and parallel printers. Printer interface scripts are assigned when the printer is installed.

10.7.1 Assigning an interface script

Model interface scripts are stored in */etc/lp/model*.

Exercise: Model interface scripts

List the model interface scripts for your classroom system.

During the installation of a printer, the selected script is copied to:

/etc/lp/interfaces/printername

There is one interface script for each printer installed.

Exercise: Installed printer interface scripts

List the installed printer interface scripts for your classroom system.

10.7.2 Customizing the standard interface script

The *standard* interface script defines minimal capabilities common to most printers. However, there may be instances where you need to change the settings rather than developing your own printer interface script. Changes can include:

- Baud rates
- Serial line settings
- Character pitches
- Restricting the printer from printing a banner page

To make these changes, edit the script in */etc/lp/interfaces*. Do not edit the scripts in */etc/lp/model*.

10.7.2.1 Restricting banner pages

A banner is a page describing the print request (printer name, user, date, and so forth). You can restrict banner pages from printing as follows:

- At the shell prompt, edit the printer's interface script. Change the line that reads:

```
nobanner="no"
```

to read:

```
nobanner="yes"
```

- From the **Printer Setup Manager**, select **Printer** ⇒ **Properties** ⇒ Deselect **Allow Banner Page Override**.

10.7.2.2 Turning off breaks

When printing multiple data files, to turn off page breaks between files, if you have used the model interface script *standard*, change the line that reads:

```
nofilebreak=no
```

to read:

```
nofilebreak=yes
```

10.7.2.3 Changing a baud rate

It may be necessary to change a serial printer's baud rate if it is producing garbled printouts.

The following steps describe how to change the baud rate for a printer:

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **Printer Setup Manager**.
- Step 3. Select the printer.
- Step 4. Select **Printer** ⇒ **Properties** ⇒ **Serial Settings**.

A window displaying the current serial communications settings for that printer appear.
- Step 5. Click on the applicable BAUD rate.
- Step 6. Click on **OK**.
- Step 7. Select **Host** ⇒ **Exit** to exit the **Printer Manager**.

Exercise: Changing a printer's baud rate

Use the steps outlined in "Changing the baud rate" on this page to change the baud rate for your printer.

10.8 Troubleshooting Common Printer Problems

This section covers problems you might encounter while using your printer(s).

If you do not see the expected print output, the most likely cause is either a missing form feed or some type of hardware problem.

10.8.1 Printer does not print

If your printer is not producing printouts, do the following:

- Step 1. Verify the printer has power, is online, and has no loose cabling. Run a printer self-test. Refer to your printer manual for instructions.
- Step 2. Run **lpstat -t** to determine if the scheduler is running, and the printer is enabled and is accepting requests.
- Step 3. Enter the Device Configuration Utility (DCU) and determine if the printer port is recognized.
- Step 4. Redirect the output from the **cal(1)** command to the appropriate **tty** or **lp** port:

```
cal > /dev/term/??s
```

or

```
cal > /dev/lp?
```

- If the printer prints the calendar, but output is garbled, make sure the printer is configured properly
- If it is a serial printer, check the baud rate and see if the printer setup matches the printer interface script.

10.8.2 Restarting a printer

Sometimes just disabling and re-enabling a printer can fix a printer problem.

To enable the printer, enter:

```
enable printername
```

To disable the printer, enter:

disable *printername*

10.8.3 Laser printers and form feeds

Note that some laser printers must be sent a form feed (either a `\f` or, less commonly, a `\014`) at the end of a print job telling them to eject the page. For these printers, use the following test:

(cal; echo "\f\c") > /dev/lp0

An alternative is to take the printer off line and manually press the Form Feed button on the printer itself, which should cause whatever is in the printer buffer to print out.

10.8.4 Testing printers in PostScript mode

Laser printers that are in PostScript mode can be tested by sending them a PostScript file (Postscript files typically end in `.ps`):

cat /usr/lib/hnp/testfiles/ps > /dev/lpn

If the printer does not print, check the cable connections both at the printer and the server by disconnecting and reconnecting the cable.

10.8.5 Determining if the print scheduler is active

If the print service stops in the middle of a print request or does not start any new print jobs, use **lpstat(1)** to determine if the **lp sched(1M)** daemon is running.

lpstat -r

If it is not running, the system displays: `scheduler is not running`.

Restart the print scheduler. Enter:

/usr/lib/lpsched

Any jobs that are shown to be on the printer when the scheduler stopped will be printed in their entirety when the scheduler is restarted.

If you need to stop the print scheduler, enter:

/usr/lib/lpshut

You can also use the **scoadmin(1M) Printer Setup Manager** to **Start** or **Halt** the print scheduler by selecting the **Server** option.

10.8.6 Redirecting output

If you redirect output directly to the parallel or serial port and nothing happens or the system displays the “cannot create” message, check the following:

Step 1. Verify the device file exists in the */dev* directory.

ls /dev/lp0

The output should be similar to this:

```
crw-rw-rw- 1 root root 103,0 June 13 1993 /dev/lp0
```

If the device file is not a character file, remove it:

rm /dev/lp0

Step 2. Recreate the */dev/lp0* file. For example, enter:

mknod /dev/lp0 c 103 0

Step 3. Change the file permissions, link count, ownership, and group to that of the original device file:

```
chown root /dev/lp0
chgrp root /dev/lp0
chmod 666 /dev/lp0
```

Step 4. Enter the Device Configuration Utility (DCU) and select **Hardware Device Configuration** to verify that the port is recognized.

Step 5. Redirect the **date(1)** command to the printer:

date > /dev/lp0

10.8.7 Printer output spacing is wrong

If the printer output is legible, but spacing is wrong:

- Correct double spacing by adding either the **-onlcr** or the **-tabs** option to the *stty* statement in the printer interface script
- Correct “zig-zag” output by entering the **onlcr** option in the printer interface script

10.9 Lesson Summary

Install your parallel card and verify that it is recognized using the Device Configuration Utility (DCU).

You can install printers that have serial or parallel hardware interfaces. Supported parallel printers adhere to standard interfaces, and must use standard parallel cables.

Because there are many kinds of supported serial printers (with differing software controls and handshakes), cabling options for serial printers are typically more complex than parallel printer cabling.

Serial printers must support XON/XOFF or hardware flow control.

When you verify a printer installation, redirect output to the printer device to confirm the hardware connection.

The UNIX print service is managed by using the command-line interface or the **Print Setup Manager** in **scoadmin(1M)**. You can set print request priorities for individual or all users. In addition to installing and maintaining printers, you should also manage the print request directories on a regular basis.

10.9.1 Review questions

1. Which two parallel adapters support IRQ 7?
2. Outline the steps for installing a parallel printer.
3. Which **scoadmin(1M)** manager would you use to change a serial printer's baud rate?
4. List three common printer installation problems and their resolutions.

5. Name four functions of the UNIX print scheduler.
6. Which two directories contain print request information?
7. Which user has the authority to bypass all waiting print requests?
8. Which tasks are performed by the printer's interface script?
9. What command could you use to determine if the print scheduler is running and enabled to accept requests?

10.10 Lab: Printer Troubleshooting

Your instructor will create a failure in your classroom server and printer setup.

Overview

The instructor has created a common failure in your system and printer setup. Your task as system administrator of this system is to identify the failure and correct it.

There is no step-by-step procedure for this lab. Feel free to refer to all classroom and UnixWare 7 documentation to find the problem and correct it as quickly as possible.

When you believe you have solved the problem, call the instructor over to your system and explain what you did to solve the problem.

If you find and resolve a problem quite quickly, ask your instructor to create another failure for you to isolate and repair.

10.11 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

disable(1)	Activates the specified printer(s) and allows them to accept lp(1) print requests
enable(1)	Activates the specified printer(s) and allows them to accept lp(1) print requests
lp(1)	Sends print requests to a printer
lpsched(1M)	Starts the UNIX print service
lpshut(1M)	Stops the UNIX print service
lpstat(1)	Prints lp print service status information
lpusers(1M)	Sets printing queue priorities
stty(1)	Sets the options for a serial terminal or printer
scoadmin(1M)	Invokes SCOadmin applications (in this lesson, the Printer Setup and Print Job Managers)

Relevant files

/dev	The device special file directory
/dev/term/00	Sample device special file for the first serial port
/dev/tty00h	Sample device special file for the first serial port that supports hardware flow control
/dev/lp0	Device special file for the first parallel port on a system
/dev/lp2	Serial/parallel adapter #2
/dev/lp1	Parallel monochrome adapter
/etc/inittab	A serial device control file used by init(M)

<i>/etc/lp/interfaces</i>	Contains interface scripts for existing printers
<i>/var/lp/logs</i>	Completed print request log files
<i>/etc/lp/model</i>	Contains model printer interface scripts
<i>/var/spool/lp/tmp/machine</i>	Contains print log information while printing

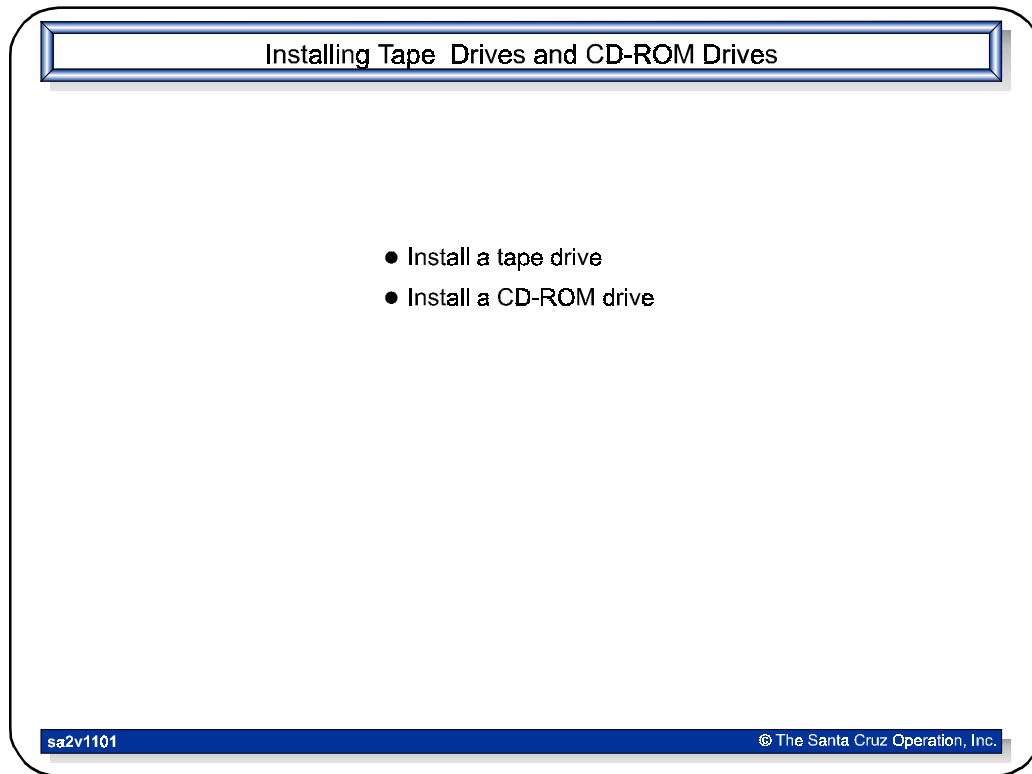
10.12 Answers to Review Questions

1. The serial/parallel and parallel adapters support IRQ 7.
2. To install a parallel printer, connect the hardware (use the correct parallel cable), configure the parallel port, and test the connection.
3. You would use the **scoadmin(1M) Printer Setup Manager** to change a serial printer's baud rate.
4. Three common printer installation problems include: incorrect cable (serial printers—either pins not correctly configured, or incorrect combination of DTE and DCE cable ends)—use the correct printer cable; port not recognized at boot time—port must be correctly installed so that the kernel will recognize it; printer is not online—put printer online; printer cables loose—tighten cable ends.
5. Functions of the UNIX print scheduler include: accepting files for printing, scheduling the work of one or more printers, starting programs that interface with the printer, monitoring the status of print jobs, and issuing error messages when problems arise.
6. The `/var/spool/lp/tmp/` machine_name and the `/var/lp/logs` directories contain print request information.
7. The root user has the authority to bypass all waiting print requests.
8. A printer interface script initializes the physical port, initializes the physical printer, prints a banner page, processes options to the **lp(1)** command, and reports any errors encountered by the print service.
9. Run `lpstat -t` to determine if the scheduler is running and the printer is enabled and accepting requests.

11

Installing Tape Drives and CD-ROM Drives

AV 11-1



Lesson 11

Installing Tape Drives and CD-ROM Drives

Introduction

Although your tape and CD-ROM drive are detected automatically during the UnixWare 7 installation, it may be necessary to add tape and CD-ROM drives.

Objectives

This lesson includes the necessary information to install additional tape and CD-ROM drives. By the end of this lesson, you will be able to:

- Install a tape drive
- Install a CD-ROM drive

11.1 Describing Supported Tape Devices

A tape drive (whether for a cartridge, 9 track, 8-millimeter, or a DAT tape) is a character device and uses device names stored in the `/dev/rmt` directory that include:

<i>ctape1</i>	Non-retentioning rewinding device
<i>ntape1</i>	Non-retentioning non-rewinding device
<i>rtape1</i>	Retentioning rewinding device
<i>nrtape1</i>	Retentioning non-rewinding device
<i>utape1</i>	Unload on close device

Note that the above device names represent the first tape device. If you had multiple tapes, *ctape2* represents the second tape and so on.

The mass storage target driver that supports tape controllers in UnixWare 7 is *st01(7)*.

Non-SCSI cartridge tape controllers require the *ictha* HBA or a device driver developed by the manufacturer for use with UnixWare 7. The *ictha* HBA is on your UnixWare 7 Installation CD-ROM. You can install it using the **scoadmin(1M) Application Installer**.

Exercise: Viewing device files

List the files in the `/dev/rmt` directory.

11.2 Adding a Tape Drive

ISA bus tape drive controllers use jumpers or switches to set the interrupt vector, the DMA channel, and so on. EISA and MCA bus settings are configured using a utility provided with the system.

Verify that the controller card settings do not conflict with the settings of other controller cards already plugged into the bus.

Hardware settings are vendor-specific. Cartridge tape drives typically require settings for:

- DMA
- IRQ
- Base I/O address

SCSI tape drives may require settings for:

- Host adapter type
- Host adapter number
- Target ID
- Bus number
- Logical unit number (LUN)
- SCSI version

You can use the **sdiconfig(1M)** command to determine which host adapters are installed on your system.

If the SCSI host adapter is not yet installed, you will need to install it before continuing with the SCSI tape installation.

Exercise: Viewing installed host adapters

Run the **sdiconfig(1M)** command to determine which host bus adapters are installed on your system.

11.2.1 Adding a SCSI tape drive

Follow these steps to install a SCSI tape drive:

- Step 1. Shut down your system.
- Step 2. Verify that the host adapter is installed.
- Step 3. Configure the tape drive. For example, you may need to set the SCSI ID and termination. Follow your manufacturer's instructions.
- Step 4. Attach (physically) the new tape drive to the host adapter.
- Step 5. Reboot your system.
- Step 6. Access your tape drive using the **tapecntl(1)** command.

11.2.2 Adding a non-SCSI tape drive

Follow these steps to install a non-SCSI tape drive:

- Step 1. Install the device driver (HBA) software.
- Step 1. Shut down your system.
- Step 2. Set the tape controller to these settings:
 - IRQ 5
 - I/O address 300
 - DMA channel 1
- Step 3. Install the tape controller.
- Step 4. Attach (physically) the new tape drive to the tape controller.
- Step 5. Reboot your system.

11.3 Adding a CD-ROM Drive

CD-ROM (whether a SCSI or IDE) character device names are stored in the `/dev/cdrom` and `/dev/rcdrom` directories:

- `cdrom1`—The device alias for the first CD-ROM
- `cdrom2`, `cdrom3`, and so on—Subsequent CD-ROM drives

The block device is `/dev/cdrom/c0b0t#10`.

The mass storage target driver that supports CD-ROMs in UnixWare 7 is `sc01(7)`.

UnixWare 7 supports both SCSI and IDE CD-ROMs. The IDE device driver provides support for:

- IDE Advanced Technology Attachment (ATA)
- ATA Packet Interface (ATAPI)

Follow these steps to add a CD-ROM drive:

- | | |
|---------|--|
| Step 1. | Shut down your system. |
| Step 2. | Verify that the SCSI host adapter or IDE controller is installed. |
| Step 3. | Configure the CD-ROM drive. For example, you may need to set the SCSI ID and termination. Follow your manufacturer's instructions. |
| Step 4. | Attach (physically) the new tape drive to the host adapter or IDE controller. |
| Step 5. | Reboot your system. |

Exercise: Mounting the installation CD-ROM

Mount your installation CD-ROM to the `/mnt` directory.

Change to the `/mnt` directory and list the files on your CD-ROM.

Unmount your CD-ROM.

Refer to the **mount_cdfs** manual pages for instructions on how to mount your CD-ROM. Use the device name that is stored in your `/dev/cdrom` directory.

11.4 Lesson Summary

UnixWare 7 supports both SCSI and IDE CD-ROMS. When installing tape drives and CD-ROM drives on your system, you must verify that the controller card for the drive does not conflict with other installed devices.

Review questions

1. Which command can you use to determine the host adapters installed on your system?
2. Do you use the block or character device name when mounting a CD-ROM?

11.5 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

mount	Mounts a filesystem
sdconfig(1M)	Determines which disk, tape, and SCSI controllers are present

Relevant files

<i>cdrom1</i>	First CD-ROM device
<i>ctape1</i>	Non-retentioning rewinding tape device
<i>/dev/cdrom</i>	Directory in which the CD-ROM block device names are stored
<i>/dev/rcdrom</i>	Directory in which the CD-ROM character device names are stored
<i>/dev/rmt</i>	Directory in which tape device names are stored
<i>ntape1</i>	Non-retentioning non-rewinding tape device
<i>rtape1</i>	Retentioning rewinding tape device
<i>nrtape1</i>	Retentioning non-rewinding tape device
<i>utape1</i>	Unload on close tape device

11.6 Answers to Review Questions

1. You would use the **sdconfig(1M)** command to determine the host adapters installed on your system.
2. You would use the block device name when mounting a CD-ROM.

12

Controlling User and System Processes

AV 12-1

Controlling System and User Processes

- Describe how processes are created
- Distinguish between user- and system-generated processes
- Check process activity
- Manage process scheduling
- Manage process accounting

sa2v1201 © The Santa Cruz Operation, Inc.

Lesson 12

Controlling User and System Processes

Introduction

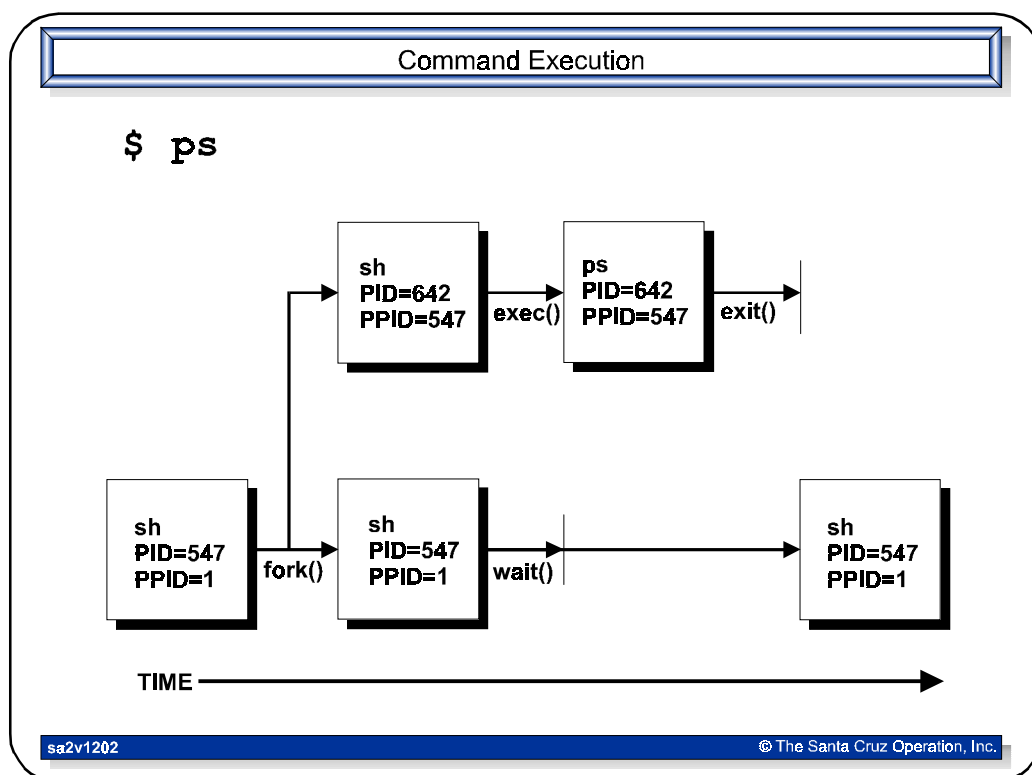
The UnixWare 7 operating system provides many programs to monitor processes, including those processes that were started during system initialization. However, before monitoring can take place, it is important to know what types of processes are running on your system, how process priorities are calculated, and what tasks are automated.

Objectives

This lesson describes the monitoring of processes and how to gather information about user and system processes. By the end of this lesson, you will be able to:

- Describe how processes are created
- Distinguish between user- and system-generated processes
- Check process activity
- Manage process scheduling
- Manage process accounting

AV 12-2



12.1 Creating Processes

Processes are generated by users, applications, and system programs to handle a variety of tasks. Except for built-in shell commands, the execution of a command usually creates a new process. This is accomplished as follows:

1. The shell reads the command line and prepares to execute the command.
2. The shell issues a **fork(2)** system call that creates a new process that is nearly identical to the calling (parent) shell process.
3. The new child process issues an **exec(2)** system call that overlays the calling process with a new program to execute, which is the program specified on the command line.
4. The child process runs the new program and eventually finishes by issuing an **exit(S)** system call. At this point, the child process is said to have “terminated.” It then enters a “zombie” (defunct) state until the parent process handles this.
5. In the meantime, the parent process issues a **wait(2)** system call that causes the parent to wait until the child process has finished and terminated.

12.2 Distinguishing Types of Processes

There are many types of processes that exist on a UNIX system. Processes are typically associated with a terminal or started by the kernel during system startup.

12.2.1 Light Weight Processes (LWP)

The UnixWare 7 kernel supports Light Weight Processes (LWPs). Every process can have one or more LWPs, each supported separately by the kernel. Each LWP:

- Shares the same process ID
- Shares the same address space and other resources of the process
- Makes system calls and blocks I/O for resources
- Is scheduled independently

On a multiprocessor system each LWP can be dispatched to run on a different processor because processes can run in parallel.

To print the status of LWPs within a process, enter:

ps -eL

In addition to viewing LWPs with the **ps(1)** command, you can also change to the */proc* filesystem.

Light Weight Processes can be either system or user processes.

Exercise: Viewing Light Weight Processes

Run the **ps -eL** command. What is the Process ID for **sysproc**? Change to the */proc* filesystem. Change to the *0/lwp* directory and view the running LWPs.

AV 12-3

System Security Profiles - Passwords				
Security Parameters	Low	Traditional	Improved	High
Passwords				
Minimum days between changes	0	0	0	2
Expiration time (days)	Infinite	Infinite	1	6
Lifetime (days)	Infinite	Infinite	24	12
Minimum Length	1	3	6	8
Checked for obviousness	-	No	No	Yes
Password required to login	No	Yes	Yes	Yes
Single User Password required	Yes	Yes	Yes	Yes

sa2v1303

© The Santa Cruz Operation, Inc.

12.2.2 System processes

System processes, also known as daemon processes, are not associated with a terminal or user. They perform system tasks on a periodic basis and sleep between those tasks. A question mark (?) appears in the *tty* field of the **ps(1)** command output.

Exercise: Identifying daemon processes

Use **ps(1)** with **grep(1)** to list the daemon processes running on your classroom system.

This is a partial listing of system daemon processes:

sysproc—PID 0

- Is the first process created by the UNIX kernel
- Creates other daemon processes at system startup
- Swaps processes out of memory into the swap area and back
- Has multiple LWPs

init—PID 1

- Initializes the boot and login processes
- Is the ancestor of all user processes
- Is the ancestor of non-**sysproc** daemon processes

ifor_pmd

- Implements the license policy manager
- Spawns other processes associated with implementing the policy manager

syslogd—Reads and forwards system messages to the appropriate log files and users

12.2.3 User Processes

User processes are generally associated with terminals and are also identified with a unique User ID. Users manage their own processes and can normally terminate them without the help of the system administrator.

12.3 Checking Process Activity

You can use several commands to check process activity on your system. Various combinations of these commands provide information about resource usage, process priorities, and the state of a process or groups of processes.

12.3.1 The **ps(1)** command

This command obtains information about active processes and gives a “snapshot” picture of what processes are executing.

To view the most detailed information about every process running on your system, use the **-elf** options.

Exercise: Viewing all active processes

Use **ps(1)** to view all currently active processes on your classroom system. What command could you use in conjunction with **ps** to quickly display the number of active processes?

AV 12-4

Process States										
F	S	UID	PID	PPID	CLS	PRI	NI	C	TIME	COMD
3	S	0	0	0	SYS	79	0	0	0:04	sysproc
42	S	0	1	0	TS	70	0	0	0:01	init
2	S	0	74	73	TS	80	0	0	0:00	ifor_pmd
42	S	0	76	74	TS	80	0	0	0:00	ifor_sld
2	S	0	458	1	TS	70	0	0	0:00	pppd

sa2v1204

© The Santa Cruz Operation, Inc.

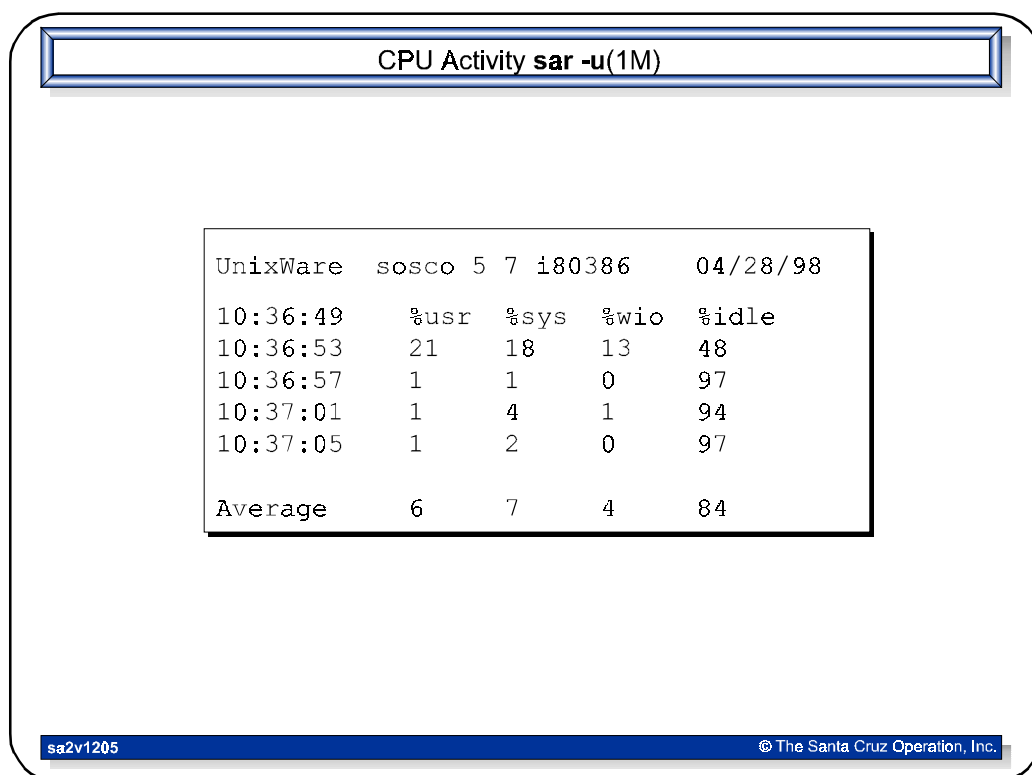
Understanding the output of the **ps(1)** command enables you to analyze the status of all active processes. The following is a listing of important definitions for the **ps** command.

F (Flag)	The number in this field represents a combination of one or more flags. For example: ● The sysproc process has a setting of 3, which is the sum of flags 1 and 2 ● This means that the sysproc process is a system process that is always loaded in primary memory (1) and is being traced by its parent (2)
S (State of the process)	A process can be in one of several states: ● S) the process is sleeping ● R) the process is ready to run ● O) the process is currently running on the processor ● X) the process is waiting for more primary memory ● Z) the process is a zombie (defunct) ● I) the process is being created; and ● T) the process is being traced
CLS	The scheduling class for the process.
PRI (Priority of a process)	Process priorities range from the lowest priority of 0 to the highest priority of 127.
NI (Nice value of the process)	This value directly affects the calculation of a process's priority when it is being scheduled to run on the processor. The values range from 0-39; the default nice value is 20.
TIME	Shows the minutes and seconds of CPU time used by processes. This is useful for seeing if any processes are CPU "hogs," or if they are runaway processes, taking up large amounts of CPU time.

Exercise: Determining the CPU time of a process

Use **ps(1)** to determine which processes are receiving the most CPU time.

AV 12-5



12.3.2 The **sar(1M)** command

The **sar(1M)** command reports system activity for a specified processor or system-wide activity. By default **sar** figures displayed are average across all CPU's on multiprocessor systems. You can use the **-P** option to display information for each CPU. Although the **sar(1M)** program reports on many different kinds of system activity, the options discussed in this lesson are related to monitoring process activity.

To determine if the system is bogged down by a runaway process, use the **-u** option with **sar**:

%usr	Indicates the percentage of time the system is executing user processes
%sys	Indicates the percentage of time the system is executing system processes
%wio	Indicates the percentage of time the operating system is idle with processes that could run if they were not waiting for I/O to complete
%idle	Indicates the percentage of time the operating system is idle with no runnable processes

If %idle falls consistently below 30%, the critical resource is processor power. Run the **ps(1)** command to check if the excessive CPU usage is due to a runaway process.

AV 12-6

Examining the Run Queue `sar -q(1M)`

```
UnixWare honey 5 7.1.1.0 i386    03/26/99

00:00:00      prunq %prunocc      runq %runocc      swpq %swpocc
01:00:00
02:00:00        1.0         0        2.0      100
03:00:00        1.0         0        2.0      100
04:00:00        1.0         0        2.0      100
05:00:00        1.0         0        2.0      100
Average        1.0         0        2.0      100
```

sa2v1206

© The Santa Cruz Operation, Inc.

To determine if there is an unacceptable buildup of processes waiting to run on the processor use the **-q** option to the **sar(1M)** command.

prunq	size of processor private queue of process in memory and runnable
%prunocc	percentage of time processor private run queue is occupied
runq-sz	size of run queue of processes in memory and runnable
%runocc	percentage of time run queue is occupied
swpq-sz	the average number of processes in the swap queue when there were processes in the queue. If there were no processes in the swap queue, this field is blank.
%swpocc	the percent of time during the sample that there were processes in the swap queue. If there were no processes in the swap queue, this field is blank.

If the **-P** option has not been specified, then the **prunq** and **%prunocc** columns will be blank.

If **runq-sz** is greater than 2 and **%runocc** is greater than 90%, the CPU is heavily loaded and response time may be degraded. Upgrading to a more powerful CPU or a multiprocessor system might improve response time.

12.3.2.1 Running the **sar(1M)** command

The **sar** command (system activity reporter) can either gather system activity data at the present time, or extract historic information collected in data files created by the system.

The **sar** command reports on many different types of activities including CPU utilization, buffer usage, and disk I/O activity.

For example, to collect processor data at 60-second intervals, for 10 minutes, enter:

```
sar -q 60 10
```

12.3.2.2 Running the **sa1(1M)** and **sa2(1M)** commands

Two programs have been set up for you to collect and report system activity data. These programs are scheduled to be run by **cron** from entries in a **crontab(1)** file. They are started automatically on your system.

The information is stored on a daily basis in the `/var/adm/sa` directory. The **sa1**(1M) command collects and stores data in the binary file `/var/adm/sa/sadd`, where *dd* is the current day of the month. For example, the files *sa10* and *sa11* represent the 10th and 11th day of the current month.

It can be started from the sys **crontab** in `/var/spool/cron/crontabs/`.

The **sa2**(1M) command writes a readable daily report to the file `/var/adm/sa/sar $dd$` , where *dd* is the current day of the month. It can also be started from the sys **crontab** in `/var/spool/cron/crontabs/sys`.

To manage system activity reporting:

- Change the running time of these programs. Edit `/var/spool/cron/crontabs/sys`.
- Display CPU usages from the files. For example, enter this command to display the information in the file `/var/adm/sa/sa10`:

```
sar -q -f /var/adm/sa/sa10
```

- Disable the collection of data by removing the entries from the sys **crontab** file.

12.3.3 The **timex**(1M) command

When you run **timex**(1) without options, it reports the amount of real time in seconds that expired while the command was executing and the amount of CPU time (user and system) that was devoted to the program. For example, to find out how much CPU time was taken to search the `/etc/termcap` file for an *ansi* label:

```
timex grep 'ansi' /etc/termcap
```

The output is written to standard error.

Refer to **timex**(1) manual page for additional information.

Exercise: Recording the output of the **timex command**

How would you record the output from **timex**(1) into a file?

12.4 Managing Process Scheduling

The UnixWare 7 kernel bases its process scheduling on the Light Weight Processes (LWPs) not the actual process. When scheduling system calls to accept a process on which to operate, the operation is applied to each LWP in the process.

The system scheduler determines when the process is run and maintains priorities based on:

- Configuration parameters
- Process behavior
- User requests

12.4.1 Scheduling classes

The scheduler uses four scheduling classes to determine when to run a process:

Time Sharing (TS)	The time-sharing class adjusts priorities dynamically in an attempt to provide good response time to interactive LWPs and good throughput to CPU-intensive LWPs. This is the default class used by the scheduler.
Fixed Class (FC)	This class is similar to the Time Sharing (TS) class, but it does not adjust priorities.
Fixed Priority (FP)	Fixed priority scheduling, previously known as real-time scheduling, allows users to set fixed priorities on a per-process or LWP basis. In the default configuration, the highest-priority fixed priority LWP always gets the CPU as soon as it is runnable.
System (SYS)	The system class uses a fixed priority policy to run kernel processes such as sysproc . You cannot change any process class to SYS, nor can you change SYS to any other class.

By default, all fixed priority processes or LWPs have higher priorities than any kernel process, and kernel processes have higher priorities than any time-sharing process.

Use the **ps(1)** command to list processes and LWPs with their associated class:

ps -ec | more

12.4.2 Controlling the nice value

The **nice**(2) command is used to execute a program at a different scheduling priority than usual.

The default nice value of a user's process is 20. An ordinary user can increase this value to 39. An increased nice value decreases the process's priority. Processes with low nice values will, on average, get more CPU time because of the effect the values have on the scheduling algorithm.

The **nice** command reduces the priority of a new process. Only **root** can set a lower nice value which results in a higher priority.

The **renice**(1) command reduces the nice value of a process that is already running. Only **root** can set a lower nice value which results in a higher priority.

For example, to change the nice value so that processes with IDs 987 and 32 have a lower scheduling priority, enter:

```
renice -n 5 -p 987 32
```

Refer to **nice** and **renice** manual pages for information on increasing or decreasing the nice value of a process.

You can also use the **scoadmin**(1M) **Process Manager** to change the priority of a process.

12.5 Managing Process Accounting

On your UNIX system, programs are executed, processes are swapped in or out, and files are accessed in an instant. This “invisible” activity is summarized by process accounting. Although process accounting was originally designed for large UNIX systems that billed individuals for using system time, it is still a useful tool for:

- Detailing analysis of system and user processes
- Determining possible causes of system errors
- Monitoring the system for security reasons

As with any program running over a period of time, process accounting may require a significant number of resources in terms of CPU usage and disk space.

12.5.1 Enabling process accounting

Use the **scoadmin Reports Manager** to start process accounting and run process accounting reports.

When you start process accounting from the **Reports Manager**, the following takes place:

- The `/usr/lib/acct/startup` and `/usr/lib/acct/shutacct` script that start and stop process accounting when going to and from multiuser mode is linked to `/etc/rc2.d/S02acct` and `/etc/rc0.d/K02acct` respectively.

Process accounting records are written to the `/var/adm/pacct` file.

- Entries are also placed in the `/var/spool/cron/crontabs/adm` file so that **cron** executes:
 - **ckpacct**(1M) to check the size of the `/var/adm/pacct` file. When this file becomes 500 blocks, the file is moved to the `/var/adm/pacct?` file. A new `pacct` file is created.
 - **dodisk**(1M) to gather information about disk usage by user
 - **runacct**(1M) to generate reports used by the **Reports Manager**
 - **monacct**(1M) to create a monthly report based on the daily reports created by **runacct**

Follow these steps to start process accounting:

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **System** ⇒ **Reports Manager** ⇒ **Accounting** ⇒ **Enable**.
- Step 3. Select **Host** ⇒ **Exit** to exit the **Reports Manager**.

12.5.2 Generating Reports

Accounting takes place at the termination of each process. One record is written to the accounting file for each command executed.

Recall that accounting reports are generated by the **runacct(1M)** program and are printed using the **Reports Manager**. These reports include the:

- Daily Report—Reports on terminal utilization
- Daily Usage Report—Reports usage of system resources by users listed in order of UID
- Daily Command Summary—Reports of system resources by commands executed. These commands are listed in descending order according to memory usage.
- Last Login Report—Reports the last time each user logged in. It is arranged in chronological order.

Generate process accounting reports by running **scoadmin(1M)** ⇒ **System** ⇒ **Reports Manager** ⇒ **Accounting Summary**.

You can also use the **acctcom(1M)** command to generate process accounting reports. Enter:

acctcom [/var/adm/pacct]

Process accounting files include eight fields of information:

- Command name
- User login name
- Name of the tty device
- Starting time of the process
- Ending time of the process

- Real time in seconds
- CPU time in seconds
- Mean Size

The command record is prefixed with a pound sign (#) indicating the command was executed with **root** privileges.

Commands not associated with a terminal are indicated by a question mark (?) in the *tty* field.

Exercise: Reading the process accounting file

On your classroom system, run several commands as a regular user. Run additional root commands. Read your process accounting file. Can you distinguish between **root** commands and **user** commands? If so, how?

You can generate many different types of accounting reports. These reports can be printed by user, by a selected group of users, or by terminal. In addition to sending these reports to your terminal, you can send them to a printer or a data file. To send:

- By user:

```
acctcom -u login_name /var/adm/pacct
```

- By group:

```
acctcom -g group_name /var/adm/pacct
```

- By terminal:

```
acctcom -l /dev/term/00t /var/adm/pacct
```

- By the most recent process:

```
acctcom -b /var/adm/pacct
```

12.5.3 Disabling process accounting

Use the **Reports Manager** to disable process accounting. Run **scoadmin(1M) ⇒ System ⇒ Reports Manager ⇒ Accounting ⇒ Enable/Disable**.

When you disable process accounting with the **Reports Manager**, all entries are removed from the `/var/spool/cron/crontabs/adm` file and the `/etc/rc?.d` directories.

Exercise: Generating process accounting reports

Beginning with the most recent process, display all records for the **root** user.
When completed, disable process accounting.

12.6 Lesson Summary

Use the **ps**(1) and **sar**(1) commands to monitor process activity and to distinguish between user- and system-generated processes. You can change process priorities using the **nice**(1) and **renice**(1) commands. Although process accounting is usually started at boot time, it can be stopped and started as necessary. Automate system processes using the **crontab**(1) command.

Review questions

1. When a command is executed, which system call creates a new process that is almost identical to the original process?
2. Which system process is responsible for swapping data to the hard disk if memory is not available?
3. Name two programs that you can use to identify how much time programs are taking up on the CPU.
4. What is the default nice value of a user's process?
5. Which **scoadmin**(1M) manager is used to enable process accounting?

12.7 Relevant Commands and Files

The following commands and files were presented in this lesson.

12.7.1 Relevant commands

acctcom (1)	Prints process accounting files
chkpacct (1M)	Checks the size of the Process Accounting file
dodisk (1M)	Gathers information about disk usage by user
exec (2)	Executes a file
fork (2)	Creates a new process
monacct (1M)	Creates monthly process accounting reports
nice (2)	Changes the priority of a time-sharing process
ps (1)	Reports process status
renice (1)	Sets the system scheduling priorities of running processes
runacct (1M)	Generates accounting reports used by the scoadmin (1M) Reports Manager .
sa1 (1M), sa2 (1M)	Is the system activity report package
sar (1M)	Reports system activity
timex (1)	Reports process data and system activity
wait (2)	Waits for a child process to stop or terminate

12.7.2 Relevant files

<i>/var/adm/pacct</i>	Default process accounting file
<i>/var/adm/sa</i>	Directory that contains system activity reports

12.8 Answers to Review Questions

1. The fork system call creates a new process that is almost identical to the original process when a command is executed.
2. The sysproc process swaps data to the hard disk if memory is not available.
3. You can use the **ps**(1) and **sar**(1M) commands to identify how much CPU time programs are taking.
4. The default nice value for a user process is 20.
5. The **scoadmin**(1M) Reports Manager is used to enable process accounting.

13

Managing System Security

AV 13-1

Managing System Security

- Describe trusted system concepts
- Describe and select a security profile
- Change your system's security profile
- Review the account database files
- Delegate system administration responsibilities
- Configure a user account to execute **root** commands
- Manage Access Control Lists (ACLs)
- Use advanced protection bits
- Configure additional security measures

sa2v1301© The Santa Cruz Operation, Inc.

Lesson 13

Managing System Security

Introduction

Every computer system needs protection from people accessing the computer disks and system files without permission. UnixWare 7 systems provide built-in security features both compatible to, and beyond, other UNIX systems. These features apply to all users of the system and are maintained by you, as the system administrator.

Objectives

This lesson provides the tools needed for you to select the best security measures for your system. By the end of this lesson, you will be able to:

- Describe trusted system concepts
- Describe and select a security profile
- Change your system's security profile
- Review the account database files
- Delegate system administration responsibilities
- Configure a user account to execute **root** commands
- Manage Access Control Lists (ACLs)
- Use advanced protection bits
- Configure additional security measures

13.1 Describing Trusted System Concepts

Because no computer system is completely free from risk, systems are referred to as “trusted” rather than “secure.” A trusted system is one that achieves a greater level of control over access to information, providing mechanisms to prevent (or at least detect) unauthorized access, along with additional means to confirm that these mechanisms are functioning properly.

In addition to traditional UNIX security, passwords, and permissions, UnixWare 7 incorporates a trust, known as Trusted Facility Management (TFM). The TFM is a collection of software that maintains parts of the operating system related to security. When combined with traditional UNIX security, these features are provided:

- Discretionary Access Control (DAC) determines the objects a user can access, including programs, data files, and directories
- Identification and Authentication (I&A) offers greater control over login attempts and the entire login process
- Accountability traces an action to an individual user
- Authorizations specify user attributes required to perform a certain action
- Protected Subsystems offer related collections of files, devices, and commands that serve particular functions. The security features provide control of the users who maintain these functions.
- The TFM implements the security policy of the system. The security policy is a set of rules to oversee and guard interaction between “subjects” (processes running on the system) and “objects” (files and devices).
- Object Reuse ensures that information is cleared in RAM before reallocating the resource to another process

AV 13-2

System Security Profiles - Logins				
Security Parameters	Low	Traditional	Improved	High
Logins				
Maximum unsuccessful attempts (account/terminal)	99	99	5	3
Delay between login attempts (secs) - terminal only	0	10	20	20
Time to complete login (terminal only)	300	60	60	60
System Security Profiles (Other)				
Default umask	022	022	027	077
UIDs reusable	yes	yes	no	no
Home directory permissions	755	755	750	700
Remote printing allowed	yes	yes	no	no

sa2v1302 © The Santa Cruz Operation, Inc.

AV 13-3

System Security Profiles - Passwords				
Security Parameters	Low	Traditional	Improved	High
Passwords				
Minimum days between changes	0	0	0	2
Expiration time (days)	Infinite	Infinite	1	6
Lifetime (days)	Infinite	Infinite	24	12
Minimum Length	1	3	6	8
Checked for obviousness	-	No	No	Yes
Password required to login	No	Yes	Yes	Yes
Single User Password required	Yes	Yes	Yes	Yes

sa2v1303 © The Santa Cruz Operation, Inc.

13.2 Describing a Security Profile

A security profile is a set of pre-configured values for parameters that control your system's security behavior such as password expiration, login controls, and what privileges are assigned to users. Once you select a profile, you can switch to another profile, or change any one of the parameters on an individual basis.

13.2.1 The high security profile

The High security profile is recommended for systems containing confidential information accessed by many users. With a high security profile:

- Passwords and login controls are strictly enforced. Networking services are limited. Users can transfer information *to* but they cannot receive information *from* other systems.
- User accounts can be deleted but UIDs cannot be reused
- All security features are active from installation

13.2.2 The improved security profile

The improved security profile is recommended for systems accessed by groups of users who can share information. With an improved security profile:

- Login controls and password expiration are more lenient. Most networking services are enabled
- User accounts can be deleted and the UID reused
- Security features are active, using parameters that are less restrictive than the High security profile

AV 13-4

System Security Profiles - Logins				
Security Parameters	Low	Traditional	Improved	High
Logins				
Maximum unsuccessful attempts (account/terminal)	99	99	5	3
Delay between login attempts (secs) - terminal only	0	10	20	20
Time to complete login (terminal only)	300	60	60	60
System Security Profiles (Other)				
Default umask	022	022	027	077
UIDs reusable	yes	yes	no	no
Home directory permissions	755	755	750	700
Remote printing allowed	yes	yes	no	no

sa2v1304 © The Santa Cruz Operation, Inc.

AV 13-5

System Security Profiles - Passwords				
Security Parameters	Low	Traditional	Improved	High
Passwords				
Minimum days between changes	0	0	0	2
Expiration time (days)	Infinite	Infinite	1	6
Lifetime (days)	Infinite	Infinite	24	12
Minimum Length	1	3	6	8
Checked for obviousness	-	No	No	Yes
Password required to login	No	Yes	Yes	Yes
Single User Password required	Yes	Yes	Yes	Yes

sa2v1305 © The Santa Cruz Operation, Inc.

13.2.3 The traditional security profile

The Traditional security profile provides compatibility with other UNIX systems. It is recommended for systems that use traditional UNIX security and want to incorporate some of the security features.

With a Traditional security profile:

- Passwords do not expire. They are not required, but are recommended. All networking services are enabled.
- User accounts can be deleted and the UIDs reused
- Additional security features can be engaged

13.2.4 The low security profile

The Low security profile is recommended for systems not publicly accessible, with a small number of cooperating users. With a Low security profile:

- Passwords are not required, but are recommended
- User accounts can be deleted and the UIDs reused
- Additional security features are not active

13.3 Viewing and Changing your System Security Profile

Exercise: Accessing additional security information

Refer to **scohelp** ⇒ **Security** ⇒ **Changing the security profile** ⇒ **security profile**.

Which security profiles logs the use of the **su(1)** command?

You are asked to select a security profile at installation time. Later, however, it is possible to select a different profile by using the **scoadmin(1M) Security Profile Manager** or running the **relax(1M)** command.

13.3.1 Determining your security profile

You can determine your current security profile using **scoadmin(1M)**.

Exercise: Determining your security profile

To determine your security profile:

Run **scoadmin(1M)**.

Select **System** ⇒ **Security Profile Manager**.

You can also determine your security profile by entering the **secdefs(1M)** command:

secdefs

Other options to the **secdefs** command are discussed later in this lesson.

13.3.2 Selecting a security profile

Files that define each security profile are stored in the **/etc/security/seclevel** directory. For example, the file that defines Improved security is **/etc/security/seclevel/improved/etc_def**.

When you change a security profile, parameters are written to these individual default files:

<i>/etc/default/login</i>	Controls login and password restrictions
<i>/etc/default/passwd</i>	Controls password aging parameters
<i>/etc/default/su</i>	Records the use of the su (1M) command
<i>/etc/default/audit</i>	Controls when auditing is enabled and disabled
<i>/etc/default/useradd</i>	Defines which events are audited and the default permissions of a user's home directory when it is created.

These steps describe how to select *Improved* security:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin**(1M).
- Step 3. Select **System** ⇒ **Security Profile Manager**.
- Step 4. Select **Current security profile**.
- Step 5. Select **Improved** ⇒ **Host** ⇒ **Save**.
- Step 6. Click on **OK** to confirm that your system has been successfully updated.
- Step 7. Select **Host** ⇒ **Exit** to exit the **Security Profile Manager**.
- Step 8. Reboot your system to invoke the new parameters:

shutdown -i6 -g10 -y

You can also use the **relax**(1M) command to select a new security profile. For example, to change to Improved security, do the following:

- Step 1. Log in as **root**.
- Step 2. Enter:

relax improved
- Step 3. Reboot the system:

shutdown -i6 -g10 -y

13.3.3 Changing a parameter in your security profile

Instead of selecting a new security profile, you can change individual parameters, including password expiration information and the maximum number of login attempts, within a security profile.

Make your changes by editing the appropriate file in the */etc/default* directory. Note that these changes will affect all users.

13.3.4 Using the **secdefs(1M)** command

If you suspect that individual parameters have been changed in your system's security profile, use the **secdefs(1M)** command to compare the current settings of the system-wide security defaults with the security profiles defined in the */etc/security/seclevel/** directories.

Exercise: Running the secdefs(1M) command

Your classroom system is installed with the traditional security profile. Run **secdefs -v** to determine if any changes have been made to the traditional security profile.

To determine if any parameters have been changed from the default, enter:

secdefs -v

The command displays:

- The name of the default file
- The name of the parameter
- The value of the changed parameter
- The value of the default parameter

Exercise: Determining security profile differences(1M)

Edit the `/etc/default/useradd` file. Change the HOME_MODE permissions to 750.

Run **secdefs -v** to verify the change.

13.4 Reviewing the Account Database Files

When you change parameters in the security profile, several security files are affected. These files comprise what is known as the Account Database.

An important distinction between UNIX systems is how account information is stored. To ensure compatibility with other UNIX systems, these files fall into two categories: UNIX system files (those defined in the System V Interface Definition) and the trusted facility database that extends System V Security.

13.4.1 System V files

“System V” files are based on the System 5 Release 4.2 (SVR4.2) version of UNIX developed at AT&T Bell Laboratories.

System V files include the following:

<i>/etc/passwd</i>	Contains both account data and (on some systems) an encrypted password
<i>/etc/group</i>	Contains group names, IDs, and names of users belonging to each group
<i>/etc/shadow</i>	Contains encrypted passwords. Access to this file is restricted; it is maintained for compatibility with other UNIX systems.
<i>/etc/default/passwd</i>	Contains default account information such as the minimum length of a password and number of repeated attempts to successfully change a password
<i>/etc/default/login</i>	Contains default account information, for example, if a password is required upon login and if pertinent information displays when a user logs in

Refer to the **passwd**(1) and **login**(1) manual pages for more information on password restrictions and login controls.

13.4.2 Trusted facility database files

Trusted facility database files include:

<i>/etc/security/ia/master</i>	Contains the same information found in <i>/etc/passwd</i> and <i>/etc/shadow</i> . It is a system file and cannot be read or edited.
<i>/etc/security/ia/index</i>	Contains a listing of user accounts. It is a system file and cannot be read or edited.
<i>/etc/security/tcb/privs</i>	Contains the file privilege database
<i>/etc/security/tfm/users/*</i>	Contains a listing of privileged users
<i>/etc/security/tfm/roles/*</i>	Contains a listing of administrative roles

Exercise: Reading the */etc/security/ia/master* file

Because the */etc/security/ia/master* file is a binary file, use the **strings(1)** command to view the contents of this file.

13.5 Delegating System Administration Responsibility

Your role of system administration can be limited to a single **root** login or you can delegate certain responsibilities to individual users. Delegating responsibility eliminates the need for a **root** login to perform tasks associated with system administration.

13.5.1 The system owner account

When you install the operating system, a user is designated as the *system owner*. Any user name (except those reserved by the system such as **uucp** and **bin**) can be assigned as the system owner.

The system owner can only perform system administration from the desktop. After an owner leaves the desktop (by opening a terminal window or logging in as a non-desktop user) they lack the power to perform administration tasks from the command line.

13.5.2 Limited system owner authorizations

You can also assign a limited set of authorizations to a user. After assigned, a user can perform these tasks using **SCOadmin**. These authorizations include:

Shutdown System	Shut down the system using the Shutdown Manager .
Administer Intranet	Configure TCP/IP and manage network connections to remote systems using the Network Configuration Manager .
Administer Filesystems	Add, configure, and manage filesystems using the Filesystem Manager .
Administer Printers	Add, configure, and remove printers using the Printer Manager .
Change System Tunables	Change kernel tunable parameters using the System Tuner Manager .
Monitor System Resources	View how system resources are used using the System Monitor .

You can assign a user any or all of the authorizations.

When a user lacks authorization to run a **SCOadmin** manager, when accessing that manager, this message displays:

You are not authorized to run ...

13.5.2.1 Assigning authorizations

Use the **scoadmin(1M) Account Manager** to assign authorizations to a user:

- Step 1. Run **scoadmin(1M)**.
- Step 2. Select **Account Manager**.
- Step 3. Select an existing user ⇒ **Users** ⇒ **Authorizations**.
- Step 4. Select an authorization.
- Step 5. Click on **Add**.

You can continue adding authorizations. When you are finished:

- Step 6. Click on **OK**.
- Step 7. Select **Host** ⇒ **Exit** to exit the **Account Manager**.

13.5.2.2 Removing authorizations

Use the **scoadmin(1M) Account Manager** to remove authorizations from a user:

- Step 1. Follow Steps 1-4 under Section 13.5.2.1, “Assigning authorizations,” on page 13-18.
- Step 2. Click on **Remove**.
- Step 3. You can continue removing authorizations. When you are finished:
- Step 4. Click on **OK**.
- Step 5. Select **Host** ⇒ **Exit** to exit the **Account Manager**.

13.5.2.3 The make-owner(1M) command

You can also use the `/usr/lib/scoadmin/account/make-owner` command to assign or remove authorizations. All authorizations are assigned or removed. You cannot selectively assign or remove authorizations using **make-owner(1M)**.

13.5.2.4 Important files

Authorizations to be assigned are stored in the */usr/lib/scoadmin/account/PrivTable* file. Additional authorizations can appear if you have installed add-on software packages.

Account names with associated authorizations are stored in */usr/lib/scoadmin/account/Users/**.

13.6 Allowing users to execute superuser commands

You can assign users the ability to execute a command that is only executable by the **root** user. When executing the command, the user does not have to log in or **su**(1M) to the **root** user, ensuring a more secure system.

Use the **tfadmin**(1M) to execute **root** only commands.

Before a user can execute a **root** only command, several actions must take place:

1. Create a new administrative role using the **adminrole**(1M) command or use an existing administrative role.
2. Add the **root**-only command(s) and the privilege *allprivs* to the administrative role using the **adminrole**(1M) command.

The *allprivs* privilege is required to execute a **root**-only command. This privilege is what differentiates the **root** user from an ordinary user.

3. Assign the user to the administrative role using the **adminuser**(1M) command.

13.6.1 Adding an administrative role

To create a new administrative role, enter:

```
adminrole -n ROLE_NAME
```

For example, to create the role of ADMIN, enter:

```
adminrole -n ADMIN
```

If it is an existing role, to view the existing commands associated with the role, enter:

```
adminrole ROLE_NAME
```

To remove a role, enter:

```
adminrole -d ROLE_NAME
```

13.6.2 Adding commands to the role

Use the **adminrole -a** command to add commands to an administrative role, enter:

```
adminrole -a rootcmd:path:priv role
```

For example, to add the **useradd** command to the ADMIN role, enter:

```
adminrole -a useradd:/usr/sbin/useradd:allprivs ADMIN
```

Use the **adminrole -r** command to remove a command from a role, enter:

```
adminrole -r rootcmd role
```

13.6.3 Assigning the user the administrative role

Use the **adminuser(1M)** command to assign the user to an administrative role. A user can be assigned to one or more roles.

To assign the user **ivey** to the ADMIN role use the **-n** option. Enter:

```
adminuser -n -o ADMIN ivey
```

If the user has previously been assigned a role, do not use the **-n** option.

To delete the user from the role you must either delete the user from the TFM database with the **-d** option or use the **-o** option to supply the user with a new role list that does not contain the role that you want to delete from the user.

A database of users and their associated roles is stored in */etc/security/tfm*.

13.6.4 Executing the tfadmin(1M) command

After the role is assigned to the user, the user performs the following to execute the command:

Step 1. Log into the system.

Step 2. Enter the **tfadmin(1M)** command with the **root** only command:

```
$ /sbin/tfadmin rootcmd
```

For example, to run the **useadd(1M)** command, enter:

```
$ /sbin/tfadmin useradd [username]
```

Exercise: Delegating system administration responsibility

Create a user named **ivey**. You can use the **useradd(1M)** command or the **scoadmin(1M) Account Manager**.

Add the administrative role ADMIN using the **adminrole(1M)** command.

Add the **useradd** command to the ADMIN role using the **adminrole** command. Do not forget to assign the *allprivs* privilege.

Assign the ADMIN role to the user account **ivey** using the **adminuser(1M)** command.

Log in as **ivey** and execute the **useradd** command.

Refer to Section 13.6, “Allowing users to execute superuser commands,” on page 13-20 if you should need help on creating and assigning the role.

13.6.5 Updating the file privilege database

Recall that when you assign a user an administrative role, you also assign the *allprivs* privilege. The system maintains the file privilege database */etc/security/tcb/privs*.

If users are unable to run the **tfadmin(1M)** command, it is possible that file privileges have been modified and the changes are not reflected in the file privilege database.

This can be caused by:

- Running the **pkgchk(1M)** command. This command goes through the */var/sadm/install/contents* file and compares file attributes in the file with those on the filesystem.

If there is a discrepancy between the two, **pkgchk** changes the file attributes to match the */var/sadm/install/contents* file.

After the file is modified, the file privilege database is out of sync with the actual file.

- Restoring system files. This causes the “last update” time of the file to be modified and does not match the last update time in the file privilege database.

You can use **initpriv(1M)** to check the validity of */etc/security/tcb/privs*. If **initpriv** returns an error, you can run */etc/security/tools/setpriv* to synchronize the file privilege database with the filesystem.

To determine if the file privilege database is out of sync:

Step 1. Log in as **root**.

Step 2. Enter:

initprivs

Step 3. Synchronize the file privilege database if errors are returned by **initprivs**.
Enter:

/etc/security/tools/setpriv -x

AV 13-6

Access Control Lists					
total 8					
-rw-r--r--	1	suees	training	9 Mar 30 14:22	curriculum
drwxr-xr-x+	2	suees	training	96 Mar 30 14:05	myprojects
-rw-r--r--+	1	suees	training	27 Mar 30 13:46	proposal
drwxr-xr-x	2	suees	training	96 Mar 30 14:21	sales
drwxr-xr-x	2	suees	training	96 Mar 30 14:21	training

sa2v1306

© The Santa Cruz Operation, Inc.

13.7 Managing Access Control Lists (ACLs)

Access Control Lists (ACLs) give you a more precise way to control access to files. The ACL contains one-line entries naming specific users and groups and what access is granted to each.

ACLs are associated with each file on the Secure (sfs) and Veritas (vxfs) filesystems.

A plus sign (+) appears at the end of the permission bit characters. It shows that additional access permissions, beyond those shown by the three sets of three bits, have been granted or denied through the ACL mechanism.

There are at least four (4) entries in an ACL and correspond to the permissions on the file. For example, if a file's permissions were 644, the ACL entries are:

```
user: :rw-  
group: :r--  
class: :r--  
other: :---
```

The `class` entry is the maximum permission that can be granted to another group, however, it can be overridden.

The **root** user and the owner of a file can set or unset ACLs.

You can view ACLs with the **getacl(1)** command, enter:

```
getacl filename
```

AV 13-7

ACL Entries

Before: ACL

```
# file: proposal
# owner: suees
# group: training
user::rw-
group::r-
class:r--
other:r--
```

After: ACL

```
# file: proposal
# owner: suees
# group: training
user::rw-
user:jamesb:rw- #effective:r--
group::r--
class:r--
other:r--
```

sa2v1307© The Santa Cruz Operation, Inc.

AV 13-8

Default ACLS

Before:

```
# file: myprojects
# owner: suees
# group: training
user::rwx
group::r-xclass:r-x
other:r-x
```

After:

```
# file: myprojects
# owner: suees
# group: training
user::rwx
group::r-xclass:r-x
other:r-x
default:user:jamesb:rw-
```

sa2v1308© The Santa Cruz Operation, Inc.

13.7.1 ACL entries for individual files

You can specifically grant and/or deny access to specific users and/or groups on your system by adding user and group entries with the **setacl(1)** command.

For example, to add a read and write entry for the user **jamesb** on the *proposal* file, enter:

```
setacl -m u:jamesb:rw- proposal
```

The **-r** option adds the ACL.

To add a read and write group entry for the **product** group, enter:

```
setacl -m g:product:rw- proposal
```

13.7.2 Default ACLs

All files that are created in a directory can have specific ACL entries. For example, you might want another person to write to any file in your directory because you are working together on a project.

You can add an ACL entry for every file in the directory, however when you create a file you have to add the entry each time. Using *default* ACL entries, you can get the system to automatically create these ACLs when the file is created.

Default ACLs can be placed on directory files only.

To configure a default ACL for the user **jamesb** on the *myprojects* directory, enter:

```
setacl -m d:u:jamesb:rw- myprojects
```

13.7.3 Deleting ACLs

You can delete any ACL entry without specifying the permissions. For example, to delete a group entry for *product* on the *proposal* file, enter:

```
setacl -d g:product proposal
```

Exercise: Managing ACLs

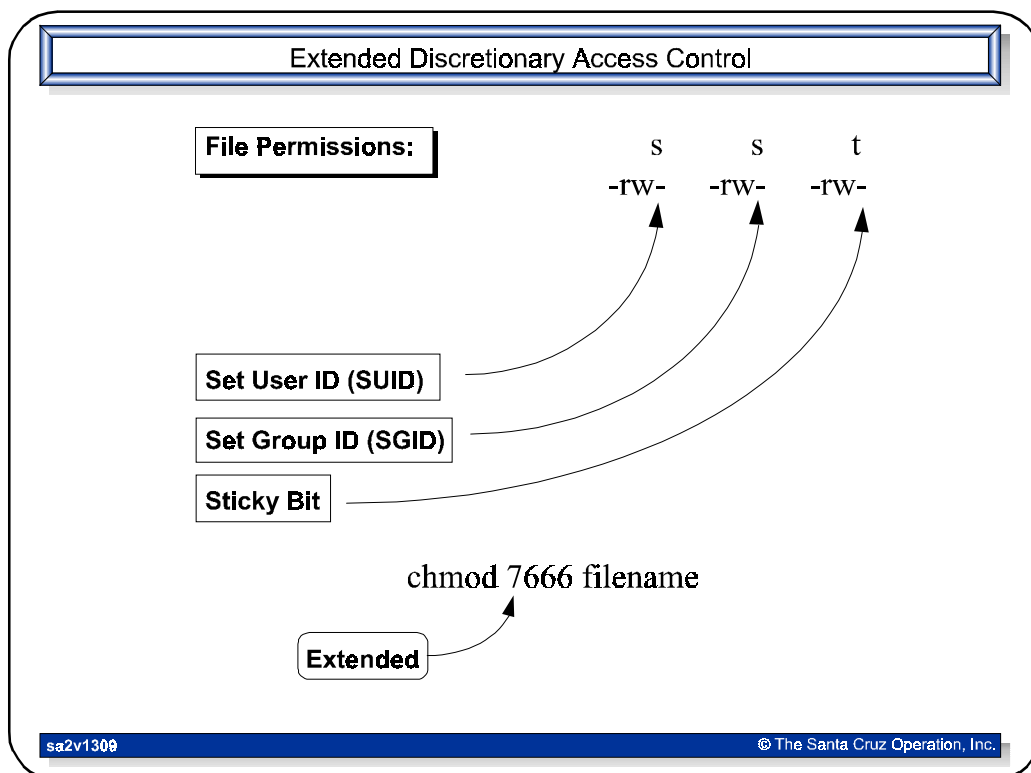
Log in using your student account. Create a file named *proposal*. Use the **setacl(1)** command and give the user **ivey** read and write permission on the *proposal* file.

13.8 Advanced Protection Bits

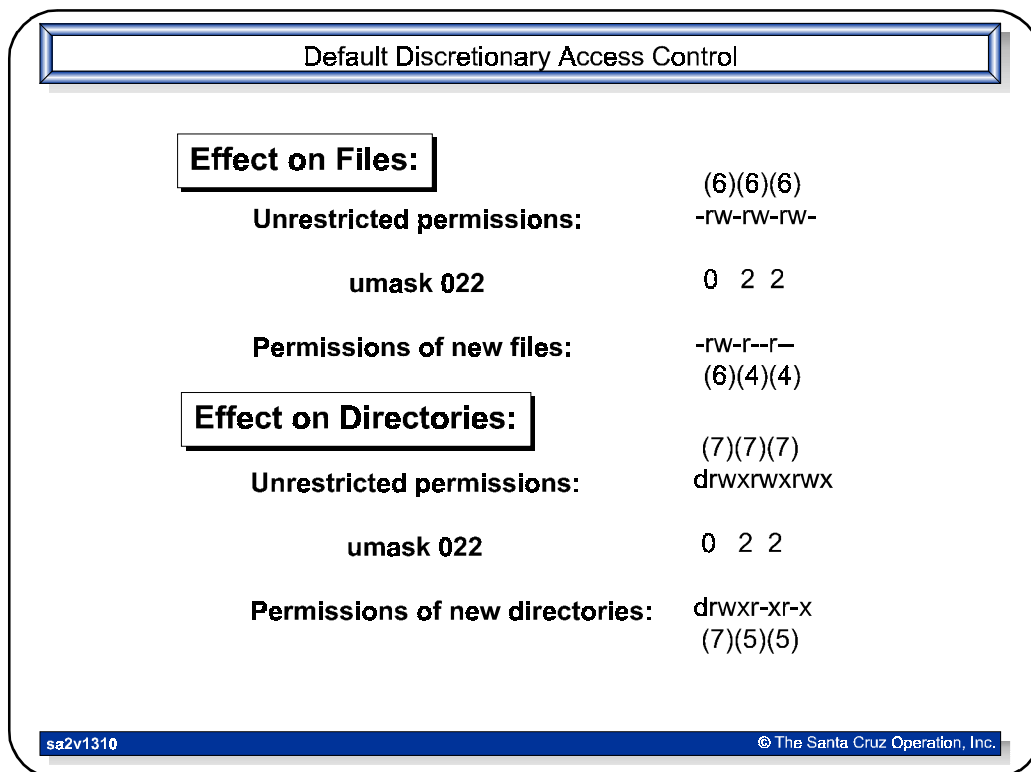
The primary data protection on your system is the use of standard UNIX system permissions on files and directories. Your UnixWare 7 system also includes features that extend this protection. These features greatly enhance the security of the system as follows:

- When SUID (set user ID) and SGID (set group ID) bits are set on the permissions of a *binary* file, the program executes with the UID or GID of the owner rather than that of the person executing the program
- A SUID/SGID *binary* has access to all the files, processes, and resources belonging to the owner or group ownership of the binary file
- SUID and SGID do not work on shell scripts

AV 13-9



AV 13-10



13.8.1 Examining SUID/SGID bits

When the SUID/SGID is set, a lowercase “s” appears in the user or group permission bits on a file. In the following example, the permissions and ownership indicate that only **root** and **sys** can modify */etc/system_file*. Since the SGID bit has been set on */bin/command*, when the user executes */bin/command*, the user assumes the group identity **sys**. This enables the user to change the contents of */etc/system_file*.

```
---x--s--x 1 root sys ... /bin/command  
-rw-rw---- 1 root sys ... /etc/system_file
```

The SUID bit appears as:

```
---s--x--x 1 root sys ... /bin/command
```

To set the SUID bits, use the **chmod(1)** command:

```
chmod 4111 binary_file
```

To set the SGID bit:

```
chmod 2111 binary_file
```

To remove the SUID and SGID bits:

```
chmod 111 binary_file
```

13.8.2 Setting the sticky bit

The third advance protection bit is known as the sticky bit. This bit allows users to add files to a “public” directory and remove only files that they own. The sticky bit appears as:

```
drwxrwxrwt 4 bin bin 224 Mar 27 10:14 /tmp
```

To add a sticky bit to a directory:

```
chmod 1777 dirname
```

To remove a sticky bit from a directory:

```
chmod 777 dirname
```

Users can only set and remove sticky bits from directories that they own.

The **root** user can set and remove sticky bits from any directory. Additional Security Measures

The most basic security requirement is to prevent unsupervised physical access to the system itself. This is accomplished by the following rules:

- Keep your system under lock and key when an operator is not present
- Organize and lock up all backup media

Your UnixWare 7 system provides additional tools to monitor system security.

13.8.3 Restricting root logins

You can restrict the **root** login to the console, eliminating entry from any other port. The following steps describe how to do this:

- Step 1. Log in as **root**.
- Step 2. Edit the file `/etc/default/login`.
- Step 3. Add the entry:
- `CONSOLE=/dev/console`

Restricted root logins are configured automatically if you select the Improved or High security profiles.

13.8.4 Monitoring the use of the su(1M) command

Users are able to switch to another account, including the **root** account, if they know the password. The following steps describe how to log both successful and unsuccessful **su(1M)** attempts to the `/var/adm/sulog` file:

- Step 1. Log in as **root**.
- Step 2. Edit the file `/etc/default/su`.
- Step 3. Verify that the entry `SULOG=/var/adm/sulog` exists. If not, add the entry.
- Step 4. Uncomment the entry `CONSOLE=/dev/console`.

These attempts are automatically monitored if you select the Improved or High security profiles.

Exercise: Reporting su attempts to the console

Edit the `/etc/default/su` file and add an entry to report **su**(1M) attempts to the console.

Log into your student account and access the **root** account using **su**.

View `/var/adm/sulog`.

View the message on the system console with   .

13.9 Lesson Summary

The UnixWare 7 system provides more than traditional UNIX security. As the system administrator, you select the security profile and can entrust users with certain tasks by assigning authorizations. Additional security can be provided by restricting **root** logins and configuring ACLs.

Review Questions

1. Which security profile(s) allows the removal of user accounts?
2. Using the command-line interface, how would you change your security profile from Improved to Traditional?
3. In which file is the encrypted password stored when using traditional security on a UnixWare 7 system?
4. Which command executes a root command for an ordinary user?
5. Which default file specifies whether passwords are required for users when logging into a system?
6. Which system owner authorization would you assign to a user so the user is able to view how system resources are used?

13.10 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

adminrole(1M)	Add, change, delete, or display roles in the TFM database
adminuser(1M)	Add, change, delete, or display administrators in the TFM database
getacl(1)	Display ACLs on files
initpriv(1M)	Checks the validity of the security database
make-owner(1M)	Add, change, or remove the ID of the system owner
relax(1M)	Change system security defaults
secdefs(1M)	Report settings of security defaults
setacl(1)	Set ACLs on files
strings(1)	Find printable strings in an object file or binary
su(1M)	Switch to another user
tfadmin(1M)	Execute a root -only command

Relevant files

<i>/etc/default/audit</i>	Controls when auditing is enabled or disabled
<i>/etc/default/login</i>	Contains default login parameters
<i>/etc/default/passwd</i>	Contains default password parameters
<i>/etc/default/su</i>	Contains options for the su(1M) command
<i>/etc/group</i>	Contains group names, IDs, and names of users belonging to each group
<i>/etc/passwd</i>	Contains default account information

<i>/etc/security/seclevel</i>	Is the directory that contains the default security level
<i>/etc/security</i>	Is the directory that contains trusted facility database files
<i>/etc/security/tfm</i>	Contains a database of users and their associated roles
<i>/etc/security/tcb/privs</i>	Contains the file privilege database
<i>/etc/default/useradd</i>	Defines which events are audited and the default permissions of a user's home directory when it is created
<i>/usr/lib/scoadmin/account/PrivTable</i>	
	Contains authorizations to be assigned to user accounts
<i>/usr/lib/scoadmin/account/Users/*</i>	
	Contains user account names with associated authorizations
<i>/var/adm/sulog</i>	Contains a log of successful and unsuccessful su (1M) attempts

13.11 Answers to Review Questions

1. All security profiles allow the removal of user accounts.
2. To change your security profile from Improved to Traditional from the command line, enter: `relax traditional`.
3. The encrypted passwords are stored in `/etc/shadow`.
4. The `tfadmin(1M)` command executes a root command for an ordinary user.
5. The `/etc/default/login` file specifies if passwords are required for users when logging into a system.
6. You would assign the “Monitor System Resources” system owner authorization to a user so the user is able to view how system resources are used.
7. The + character at the end of the permission bit indicates that additional access permissions have been granted or denied through the ACL mechanism.
8. The `/var/adm/sulog` file contains both successful and unsuccessful `su(1)` attempts.

14

Performance Tuning

AV 14-1

Performance Tuning

- Describe factors that influence performance
- Set performance goals
- Collect performance data
- Run the **scoadmin**(1M) System Monitor
- Configure the **scoadmin**(1M) System Monitor
- Autotune kernel parameters
- Interpret kernel error messages
- Configure kernel parameters

sa2v1401 © The Santa Cruz Operation, Inc.

Lesson 14

Performance Tuning

Introduction

In addition to controlling processes, your job as a system administrator is to manage the sharing of the available system resources equally among all users.

As delivered, UnixWare 7 is configured to provide optimal performance for most configurations and uses. As system administrators, you won't need to configure or tune the system often. Any system configuration that you perform should be the result of careful analysis of your user's needs, application requirements, and operating system performance.

In this lesson, you will learn how to monitor and configure your UnixWare 7 system to provide the best possible performance.

Objectives

This lesson covers the monitoring and configuring of a UnixWare 7 system. By the end of this lesson, you will be able to:

- Describe factors that influence performance
- Set performance goals
- Collect performance data
- Run the **scoadmin(1M) System Monitor**
- Configure the **scoadmin(1M) System Monitor**
- Autotune kernel parameters
- Interpret kernel error messages
- Configure kernel parameters

14.1 Influencing System Performance

A computer system consists of a finite set of hardware and software components. These components constitute the resources of the system. One of the tasks of the operating system is to share these resources among the programs that are running on the system. In addition to the hardware and software components, how the data is organized, and how programs are accessed on your hard disk also affect system performance.

14.1.1 Hardware factors

Your system's hardware has the greatest influence on its performance. It is the ultimate limiting factor on how fast a process will run before it has to start sharing what is available with the operating system and other user processes.

Hardware factors that can limit overall system performance include:

- The speed and width of the system's address and data buses
- The model, clock speed, and the size of the internal memory cache of the system's CPU or CPUs
- The size of the external cache memory
- The amount of memory, the width of its data path, and its access time
- The speed and width of a SCSI bus controlled by a host adapter
- The width of the data path on peripheral controller cards (32-, 16-, or 8-bit)
- Access time for hard disks
- Whether intelligent or dumb serial cards are used
- Network interface cards and network design

On multiprocessor systems, the following considerations also become important:

- Write back external cache (for instructions and data) with cache coherency on each CPU to reduce the number of accesses to main memory
- Support for fully distributed interrupts
- Memory and I/O subsystems must be as fast as possible to keep up with the demands of enhanced CPU performance. Use of intelligent controllers is recommended.

14.1.2 Software factors

The ways in which applications are written usually have a large impact on performance. If your application makes inefficient use of processing power, memory, disk, or other subsystems, it is unlikely that you will improve the situation by tuning the operating system.

The efficiency of the algorithms used by an application, or the way it uses system services, are usually beyond your control unless you have source code.

Some of the factors you can take into consideration when assessing the efficiency of an application's code include:

- Does it use algorithms that make the best use of time and storage?
- The optimization of the compiler being used.
- Are shell programs being used where C programs would be more efficient?
- Is it using a large number of system calls?
- Is the application using memory efficiently?

These factors and more will influence the performance of your system.

14.1.3 System organization

Proper organization of your hard disk can improve its performance. Use the following guidelines for organizing your hard disk:

- Move frequently accessed data to the fastest disk available
- Ensure the user's \$PATH environment variable has the directory containing programs the user most frequently uses first

14.2 Setting Performance Goals

To tune a system efficiently, you need a good understanding of the various system resources and how the system is going to be used. You must then establish what goals you need to set to improve performance. To accomplish this:

- Define your goals
- List your goals
- Understand your goals

14.2.1 Defining your goals

The first step in tuning a system is to define a set of performance goals. This can range from discovering and removing system bottlenecks in order to improve overall performance to tuning the system to run a single application.

14.2.2 Listing your goals

The performance goals should be listed in order of priority. Often goals may conflict.

A system may be running a database that uses a large cache and also may require a large portion of memory to compile programs during software development.

Assigning a priority to your goals might involve deciding if database performance is more important than the speed of compilations.

14.2.3 Understanding your goals

Attempt to understand all goals as well as possible. If possible, note what resources are affected by each goal. Do you have any conflicts?

14.3 Collecting Performance Data

After you have identified your performance goals, your next step is to determine how the system is performing at present. Measuring your system's performance depends on the tasks you expect to carry out. Listed below are some of the typical criteria you might use to judge a system:

- The time taken for an interactive application to perform a task
- The time taken to process a database transaction
- The time taken for an application to perform a set number of calculations

There are a number of tools available with your UnixWare 7 system that assist you in monitoring and collecting performance data. They include tools for gathering static and dynamic data. Static tools include:

- **scoadmin(1M)** ⇒ **System** ⇒ **System Information**
- **sysdef(1M)**

Dynamic data gathering and monitoring tools include:

- **rtpm(1M)**
- **scoadmin(1M)** ⇒ **System** ⇒ **System Monitor**
- **ps(1)**
- **sar(1M)**

You should collect data over a duration that is long enough for you to be able to establish normal patterns of usage.

Three potential performance problems include:

- CPU saturation
- I/O system saturation
- Memory saturation

Poor system performance can result from any combination of problems.

14.3.1 Identifying CPU saturation

CPU saturation occurs if the processor cannot execute fast enough to keep the number of processes on the run queue consistently low.

To monitor CPU usage, you can do the following:

- Turn on process accounting to track CPU usage over a period of time
- Use the **ps(1)** and **sar(1M)** commands to help identify CPU saturation

14.3.1.1 Using the **ps(1)** command

You should run the **ps(1)** command at regular intervals. Check for processes that you suspect are consuming CPU cycles that could be run with **cron(1)**.

14.3.1.2 Using **sar(1M)** to check for CPU saturation

The **sar(1M)** command reports activity on single processor systems. To check for CPU saturation with **sar**, use the **-u** and **-q** options as follows:

```
sar -u 60 10
```

If the %idle field is less than or equal to 30%, could indicate a system lacking in processor power. CPU saturation will result if there are low %idle and high %usr and %sys figures.

```
sar -q 60 10
```

CPU saturation is reflected in %runq - sz figures over 2 and %runocc figures greater than 90%.

While the **-u** option gives the CPU usage in percentages, the **-q** option indicates how many processes in the run queue are currently competing for the CPU.

Exercise: Monitoring CPU activity

The shell script **sartest -cpu** intensifies CPU activity on your classroom system.

To gather information on how your classroom system is currently running:

- Step 1. Run the **sar(1M)** command to check for CPU saturation.
- Step 2. Write down the results from the average fields.
- Step 3. Run the shell script **sartest -cpu** on a second multiscreen.
- Step 4. Return to the first multiscreen and run the **sar** commands again and compare the results.

What was your percentage of increase in CPU saturation? How many processes were currently competing for the CPU? Were they system or user processes?

When finished, use the  key, terminate **sartest -cpu**.

14.3.2 Tuning CPU resources

If your system is CPU-bound, you can do several things to improve CPU resources. Most of these will be from the hardware side; however, you can also tune a limited amount of kernel parameters.

14.3.2.1 Hardware

To improve CPU resources, you can change your hardware by:

- Using smart (intelligent) serial cards
- Choosing a disk controller that is capable of using DMA to transfer data to and from memory
- Adding more memory if a considerable amount of swapping is occurring
- Upgrading to a faster CPU or CPUs

- Adding one or more CPUs to a multiprocessor system. If running a single CPU system, upgrade to a multiprocessor system.
- Purchasing an additional system and divide your processing requirements

14.3.2.2 Kernel parameters

To improve CPU resources, you can change the *Process Limit Parameter*:

MAXUP	Controls the number of processes that a user can run at any given time. The default is 80. Reduce this parameter to limit the number of processes a user can run.
-------	---

14.3.3 Identifying I/O saturation

Input/output (I/O) is the process of transferring data from memory to a device, from a device to memory, or from one device to another.

To make the transfer between memory and disk more efficient, the system maintains a buffer cache of the most recently accessed disk data. This reduces the amount of disk I/O that the system needs to perform.

14.3.4 Describing the buffer cache

To minimize seeking data on the hard disk, recently used data blocks are held in a cache of special memory structures called "buffers". Buffers make the operating system more efficient. Depending on the filesystem type and the setting of kernel parameters, the buffer cache is "flushed" (written to the disk) at set intervals. On a typical system, approximately 85% of disk I/O can be avoided by using the buffer cache, though this depends on the mix of jobs running.

UnixWare 7 uses a unified virtual memory architecture. Buffers are allocated as needed by the system. Buffer cache memory allocation is integrated with paging program schemes. Buffers are not preallocated. They are a dynamic or "paged" resource.

14.3.4.1 Viewing buffer cache activity

To determine the activity in the buffer cache, use the **-b** option with the **sar(1M)** command:

```
sar -b 60 10
```

The buffer cache read hit rate (%rcache) indicates the percentage of attempts to read data from disk (or any block device) where the data was already in the buffer cache.

The buffer cache write hit rate (%wcache) indicates the percentage of attempts to write data to disk (or any block device) where the block in which the data was to be written was already in the buffer cache.

The read and write hit rates should show high values. If these values fall, the system is having to access blocks on the disk (or other block devices) rather than in the buffer cache.

Increase the amount of buffer space (BUFHWM) if `sar -b` consistently indicates low hit rates (%rcache < 90, %wcache < 65) and high bread/s and bwrite/s. The %rcache and %wcache values are low temporarily when you start up an application or bring in new data. However, if these values are always low, your system may be a heavier than average user of filesystem data.

14.3.4.2 Viewing I/O activity

A system is I/O bound if the peripheral devices (hard disk, tape, and so forth) cannot transfer as fast as the system requests. This causes processes to be put to sleep, “waiting for I/O,” and leaves the CPU idle for much of the time.

To determine if a system is I/O bound, run the `-u` option with the `sar` command:

```
sar -u 60 10
```

The %wio value displays the percentage of time that each CPU spends waiting for I/O to complete while there are no runnable processes. If this value is high, it is possible that I/O is not keeping up with the rest of the system.

Do not always assume that a high I/O value indicates a disk problem. This could be high due to a tape drive being in use or a shortage of memory.

To determine if the disk is a problem, run the **-d** option with the **sar(1M)** command:

sar -d 60 10

Check for:

Load balancing	How much activity does each disk get? Loads should be balanced across each disk on the system. Watch for unbalanced loads if you have multiple disks. Individual disks may become overloaded before the entire I/O system approaches capacity. High values of the ratio of avserv also suggest that a disk is saturated with requests.
Slow disks	Once a request hits the disk, how long does it take a disk to service the request? await indicates the length a request must wait to be serviced by the disk. avserv indicates how fast the disk can service the request. Long avserv times can indicate a slow disk.
Disk saturation	If all disks have long await times or long queues and high queue occupation then additional disks could help to reallocate disk activity.

If the values of **%busy** and **avque** are both consistently high for the disk device, the device cannot keep up with the requests to transfer data.

A **%busy** average over 50% can indicate I/O saturation.

Exercise: Monitoring I/O saturation

The shell script **sartest -io** has been created to increase I/O activity on your classroom system.

Run the shell script **sartest -io** on one multiscreen.

On a second multiscreen, run the **sar(1M)** command twice using the **-b** and **-d** option.

Compare the results to the information in “Viewing buffer cache activity” and “Viewing I/O activity.” Can you determine where your bottleneck is?

When finished, use the **DEL** key to terminate the **sartest -io** shell script.

14.3.5 Tuning I/O performance

If your system is disk I/O bound, you can make changes to your hardware configuration or tune several kernel parameters to improve I/O performance.

14.3.5.1 Hardware

To tune I/O performance, you can do the following to your hardware configuration:

- Spread filesystems and swap areas across different disks or buses to help spread the load
- Purchase caching disk controllers or advanced RAID drive arrays

14.3.5.2 Kernel Parameters

To tune I/O performance, you can change the following *File System Parameters*:

DNLCSIZE Defines the size of the directory name lookup cache (DNLC). Generally, larger memory systems both support a heavier workload and have more memory to spare, so the DNLCSIZE tunable is automatically adjusted to be larger when more memory is present. You can override the autotuned value by using `idtune`. Run **sar -t** to check whether inode limits should be changed so that more inodes and pages can be kept in memory and reused.

BUFHWM The amount of memory that can be used by the transfer of filesystem structure data such as inodes, indirect blocks, and cylinder groups. It is not file data, which goes through virtual memory (VM) as pages. NOTE: In some other UNIX systems such as UNIX System V Release 3.2 and earlier, the buffer cache was used for practically all disk I/O, including the contents of files (that is, reads and writes). The use of buffer cache changed with the introduction of virtual memory (VM) and, therefore, changed the appropriate tuning of BUFHWM.

Setting BUFHWM too low causes excess filesystem activity to flush the buffer before it can be re-used. Setting BUFHWM too high reduces the page pool size and can increase paging. In general terms, BUFHWM should grow proportionately with memory size. The value of BUFHWM is autotuned and it should not have to be retuned. However, you may want to retune BUFHWM for a specific condition; for example, to reduce the value of BUFHWM so there is more memory for use by X-windows.

Run **sar -b** before and after the change to ensure that the `%r`cache and `%w`cache have not changed dramatically.

NBUF	Buffer cache I/O uses both buffers and buffer headers. (See BUFHWM.) Whenever a buffer header is needed, but none is available, the system dynamically allocates more buffer headers in chunks of NBUF headers at a time. There is no limit to the number of buffer headers in the system, but the tunable BUFHWM limits the number of kilobytes that can be used by buffers and, therefore, effectively limits the number of buffer headers that are allocated. Once allocated, buffer header space cannot be freed for other uses. Therefore, take care when you raise the value of NBUF. A higher value of NBUF decreases the number of times the Kernel Memory Allocator must be called to allocate space for buffer headers, but this also can result in the allocation of headers that are not used.
NHBUF	Specifies the size of the hash table used to locate a buffer, given a device number and a block number. The value must be a power of “2” from “32” to “1024”, inclusive. This value should be about one quarter of the total buffers available. A value between 1/8 and 1/4 of BUFHWM is typically sufficient.
NPBUF	Specifies the number of physical I/O buffers to allocate; one is needed for each active physical read or write. There is no rule for adjusting NPBUF. However, if you expect a lot of I/O and filesystem activity, improvement may be gained by raising NPBUF.
FDFLSHR	Interval in seconds to check the need to write buffer cache and file pages to disk. The default is 1.
NAUTOUP	Number of seconds between filesystem updates. Increasing this value can improve performance, but increase the risk of data loss in a system crash. The default is 60.

14.3.6 Identifying memory saturation

UnixWare 7 systems use two types of memory. *Physical memory* is divided between the area occupied by the kernel and the area available to user processes. *Virtual memory* is an area usually reserved on a local disk called swap.

Memory saturation occurs when the amount of memory required by running programs plus the memory required by the kernel exceeds the amount of physical memory on the system.

UnixWare 7 is designed to provide programs with as much memory as they require, even if the requirements exceed physical memory. When physical memory is exhausted, UnixWare 7 memory management begins to use swap areas on the disk for paging or “swapping”.

Paging is the process by which the contents of physical memory are moved both to and from the swap areas and filesystems. In addition to paging, the operating system frees assigned memory pages by copying modified process pages to the swap area.

With UnixWare 7 Release 7.1, support is provided for up to 8GB of pageable (general purpose) memory. General purpose memory can be used, for example, for running the kernel itself, running user programs, and providing buffers for devices and filesystems. In previous versions of UnixWare 7, memory above 4GB could only be used as dedicated memory, for shared memory used by database programs.

UnixWare supports Large Physical Memory (LPM). That is, the system can utilize up to 64GB of physical memory. UnixWare uses the Physical Address Extension (PAE) mode of the Pentium Pro processor when operating with more than 4GB of memory.

UnixWare 7 also supports dedicated memory. Dedicated memory is memory that is set aside at system initialization time for use by specific shared memory facilities. This memory cannot be used by the system for any other purpose.

Dedicated memory includes Page Size Extension (PSE) Shared Memory (SHM), Dynamic Shared Memory (DSHM) and Fine-Grained Affinity Shared Memory (FGA-SHM). PSE SHM allows processes that use SHM and request a segment (1MB or greater) to improve performance by using 4MB pages (2MB pages with PAE mode). DSHM allows processes to create shared memory segments that are up to 63GB in length and to circumvent the 3GB constraint of user virtual addresses by allowing an application dynamically to create the mappings between its virtual address space and the shared memory object's pages, as needed.

For a more information on configuring large memory systems refer to **scohelp** ⇒ **System Management** ⇒ **Large physical memory**.

For an overview of how the system works refer to **scohelp** ⇒ **An overview of the system** ⇒ **Memory management**.

14.3.6.1 The swap(1M) command

To quickly determine if memory is being exhausted, run the command:

swap -l

Blocks are displayed in 512-bytes. The information displayed includes a **path**, **dev**, **swaplo**, **blocks**, and **free** column.

To determine the number of megabytes of swap space available, add the figures in the **blocks** column, and divide the total number of blocks available by 2048. If the swap area is being used, look further for excess swapping.

14.3.6.2 Checking for excess swapping with sar(1M)

Paging in and paging out is costly in terms of both disk and CPU overhead.

Run the `sar -pgrwud` report. With this information, you can determine whether more memory is needed.

If the `vflt/s` value shown by `sar -p` is greater than 50, then look at `sar -g`. High values for `vscan/s` and `pfree/s` imply that the page-stealing daemon is working overtime to find free pages because of a memory shortage. A high attach rate and a high `vflt` rate indicate a memory shortage, also.

Other indicators of a memory shortage are the `freemem` value of `sar -r` and the `swpot/s` value of `sar -w`. A value of `swpot/s` that is consistently greater than "0" is also an indicator of a memory shortage.

Exercise: Determining free memory

A shell script **sartest -swap** has been created to increase memory usage and initiate swapping on your classroom system.

Use the **sar(1M)** command to determine the amount of free memory on your classroom system.

On a multiscreen, run the **sartest -swap** script.

On a second multiscreen, run the **sar** command and view the memory usage and swapping activity.

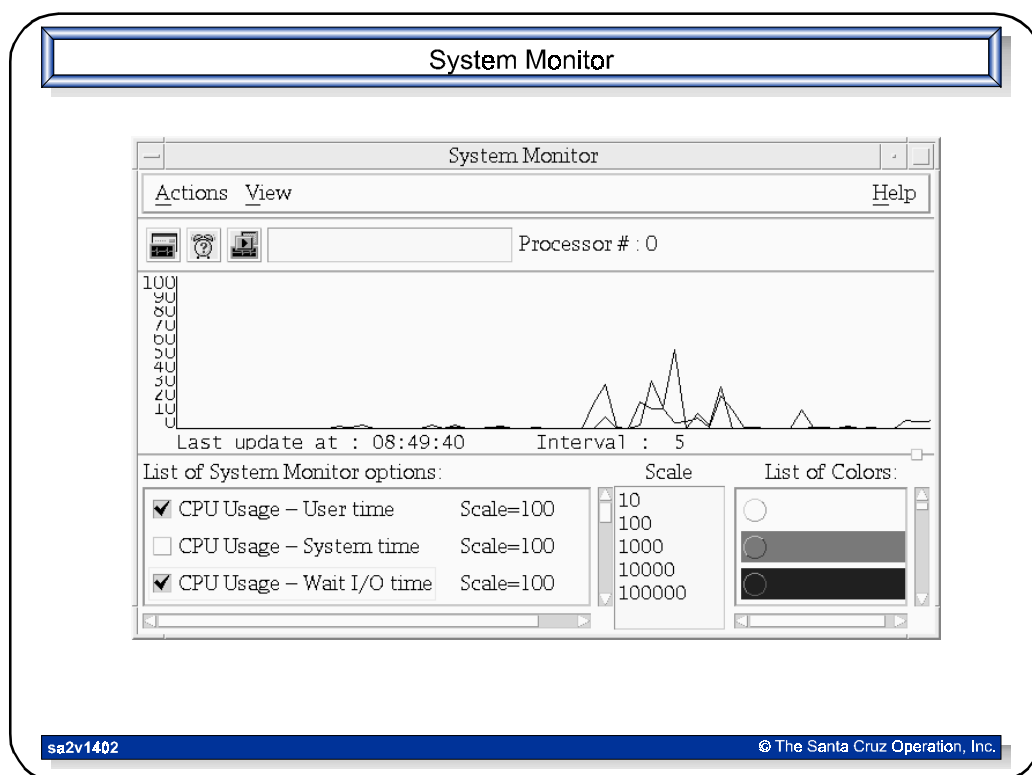
When finished, use the  key to terminate the **sartest -swap** script.

14.3.7 Tuning memory saturation

If memory shortages occur frequently, increase memory in one of two ways:

- Uninstalling optional kernel utilities that are not needed by your applications frees the memory used by the utilities so it can be used by user applications.
- Add extra memory

AV 14-2



14.4 Running the System Monitor

In addition to using the **sar**(1M) and **ps**(1) commands to monitor system performance, you can also use the **scoadmin**(1M) **System Monitor**.

The **System Monitor** includes information about your system's:

- CPU usage
- Free memory
- Free blocks
- Paging
- Swapping

By default, no options are selected and the information updates every five (5) seconds.

To access the **System Monitor**, enter:

scoadmin(1M) ⇒ **System** ⇒ **System Monitor**.

The **System Monitor** is not available in character mode.

Exercise: Accessing the System Monitor

Access the **System Monitor**.

You can monitor these resources with the **System Monitor**. Use each resource explanation as a guide when you are monitoring your system. The resources include:

CPU Usage—User Time

The percentage of CPU time spent running in user mode. This value should slightly be less than system time.

CPU Usage—System Time

If system time is more than double user time, system resources are not being used efficiently.

CPU Usage—Wait I/O Time	The percentage of CPU time spent waiting for I/O. A high wait I/O time usually indicates that the processor is waiting for input from the hard disk. This value should be less than 10 percent.
CPU Usage—Idle Time	The percentage of time the processor is idle. This value should be greater than 10 percent.
Free Memory Pages	The average number of 4096 byte pages available to user processes. This value should be greater than 200.
Free Block for Swapping	The number of 512-byte blocks available on disk for page swapping. This value should be greater than 200.
Page-in requests/per sec	The number of page-in requests received per second. This value should be less than 5.
Page-out requests per sec	The number of page-out requests received per second. This value should be less than 5.
System swap in per sec	The number of transfers per second from the swap area on disk into memory. This value should be less than 1.
System swap out per sec	The number of transfers per second to the swap area on disk from memory. This value should be less than 1.
Block swap in per sec	The number of transfers per second into memory from the disk swap area. This value should be less than 1.
Block swap out per sec	The number of transfers per second from memory to disk swap area. This value should be less than 1.
Process switching per sec	The number of process switches per second in and out of memory. This number should be less than 100 on a 4 to 6 user system.

14.5 Using the System Monitor

The graph at the top of the **System Monitor** window displays a line representing each of the selected resources plotted against a scale of 0 to 100. This range represents a percentage of the option values.

The bottom window is divided into three columns:

- The left column allows you to select the resource to monitor
- The middle column allows you to set the scale. The scale for the four “CPU Usage” options cannot be changed as they can only be shown as a percentage.
- The right column allows you to choose the color in which you want the graph to display. To change the colors in the **List of Colors** column you can edit the `/usr/X/lib/X11/app-defaults/sysmonitor` file.

For more information on changing the **System Monitor** colors, refer to **scohelp** ⇒ **System Management** ⇒ **Performing basic system tuning and monitoring** ⇒ **Setting System Monitor options** ⇒ **Customizing System Monitor colors**.

Exercise: Monitoring CPU Usage

While in the **System Monitor**, follow these steps to select the options that display how much CPU time is spent in user and system time:

- Step 1. Click on ‘CPU Usage - User time.’
- Step 2. Select a color from the “List of Colors” column.
- Step 3. Click on “CPU Usage - System time.”
- Step 4. Select a different color from the “List of Colors” column.

Which resource value should be slightly less?

14.6 Configuring the System Monitor

You can configure the **System Monitor** to:

- Save and play back log data
- Signal an alert if the system goes above or below thresholds for selected resources
- Monitor a specific processor

14.6.1 Managing log data

So that you can view your system resources over a period of time, you can save your data to a file.

Before you begin monitoring, to save your data to a file:

- Step 1. Select **Actions** from the **System Monitor** window.
- Step 2. Select **Log data to file**.
- Step 3. Accept the existing file name or enter a new filename. Click on **OK**.

You can stop the logging of your data by following these steps:

- Step 1. Select **Actions** from the **System Monitor** window.
- Step 2. Select **Stop Logging Data**.

You can playback the log data by following these steps:

- Step 1. Select **Actions** from the **System Monitor** window.
- Step 2. Select **Playback log data**.
- Step 3. Select the file name. Click on **OK**.
- Step 4. Click on **Cancel** to end the playback.

14.6.2 Setting alarms

You can set alarms in the **System Monitor** to alert you when the system goes above or below a certain threshold for selected resources.

From the **System Monitor** window, you can set alarms by following these steps:

- Step 1. Select **View** ⇒ **Alarms**.
- Step 2. Select the resource.
- Step 3. Select from one of the following:
 - **Beep**—Issues a beep if the level of the selected resource goes above or below the entered value
 - **Flash Header**—An alarm icon appears next to the selected resource on the window if it goes above or below the entered value
- Step 4. Enter a value in the “Alarm Above” or “Alarm Below” boxes or both.
- Step 5. Click on **OK**.

Exercise: Setting an alarm

Set the **System Monitor** so that you are signaled with an alarm if CPU Usage - User time goes above 50%.

14.6.3 Setting options

Recall the *interval* that the **System Monitor** reports on your selected resources is five (5) seconds. In addition to setting a different interval, you can also set:

- How the grid displays
- Which processor to monitor if you have a multiprocessor system.

From the **System Monitor** window, select **Actions** ⇒ **Options** to set your options. To save you options, select **Actions** ⇒ **Save Settings**.

AV 14-3

Autotuning Parameters					
	1.	2.	3.	4.	5.
VXFS	NINODE	DEF	LINEAR	64	7000
VXFS	NINODE	MIN	LINEAR	4	150
VXFS	NINODE	MAX	LINEAR	64	28000

1: Name of Tunable
2: The default, min, or max
3: Linearly or as a step function
4: The amount of memory
5: The value of the tunable at that amount of memory

sa2v1403 © The Santa Cruz Operation, Inc.

14.7 Autotuning Parameters

Autotuned parameters are kernel variables whose default values are automatically tuned according to the amount of memory that the system has.

For large memory systems (those with more than 1GB of general purpose memory) a distinction is made between autotuned parameters which consume kernel virtual memory and those that simply manage physical memory.

For these parameters, a table of information relating memory size to the value of the parameter is maintained. This table includes the:

- Current value
- Default value
- Minimum value
- Maximum value

for each autotuned parameter. This table is also passed to the kernel to allow the autotuning of parameters on boot.

These parameters are stored in the `/etc/conf/autotune.d` directory

Before changing any of these parameters, use the **id tune**(1M) command to view the current, default, minimum, and maximum value. For example, to view the values for the VXFSNINODE parameter, enter:

`/etc/conf/bin/id tune -g VXFSNINODE`

Exercise: Running id tune(1M)

Determine the default value for the VXFSNINODE parameter. Run the **id tune** command.

14.8 Interpreting Kernel Error Messages

When system limits are exceeded the kernel will display an error message on the console.

For example, if the number of processes is exceeded on your system or the system is out of swap space, the following error messages display:

```
newproc - no procs
```

```
swapdel - too few free pages
```

Kernel error messages are written to the `/var/adm/log/osmlog` file.

Refer to **scohelp** ⇒ **System Management** ⇒ **Tunable parameters** ⇒ **Kernel messages about exceeding system limits** for more information on kernel error messages.

14.9 Changing Kernel Parameters

You may need to change kernel parameters when:

- You install additional physical memory giving you greater memory resources to allocate
- Kernel error messages are persistent, indicating that certain resources are used up, such as table entries
- The system response time is consistently slow, indicating that other resources are too constrained
- Resource usage needs to be tailored to meet the needs of a particular application

You can use the following configuration tools for examining or changing tunable parameters:

scoadmin(1M) System Tuner

A menu-driven program that allows you to examine and modify the value of tunable kernel parameters.

id tune(1M)

Changes autotuned parameters.

14.9.1 The System Tuner

The **scoadmin(1M) System ⇔ System Tuner** is the preferred method for changing system tunable parameters. The parameters are divided into categories and a description of each parameter is provided.

In addition to changing parameters, you can also use the **System Tuner** to:

- **Reset** values for all parameters changed since the last save
- **Reset to Factory** (resets values to those that were set when your system was installed)

The **System Tuner** cannot be run in character mode.

Before using the **System Tuner**, you should make a backup copy of your existing kernel. This ensures that if a problem occurs during the loading of the new kernel, you can always load the original kernel and change the parameters.

The following steps describe how to use the **System Tuner** to change tunable parameters:

- Step 1. Log in as **root**. Enter the **root** password.
- Step 2. Copy */stand/unix* to */stand/unix.before*
- Step 3. Enter **startx** to start up a graphical screen.
- Step 4. Run **scoadmin(1M)**.
- Step 5. Select **System** ⇒ **System Tuner**.
- Step 6. Click on the “Category” button.
- Step 7. Select the category.
- Step 8. Select the parameter.
- Step 9. Enter the new value. Note that you could **Reset** or **Reset to Factory** as explained earlier.
- Step 10. Click on **OK**.
- Step 11. Rebuild the kernel.
 - If you select **Yes**, the kernel is rebuilt now.
 - If you select **No**, the kernel is not rebuilt until you reboot your system. The current parameters stay in effect.
- Step 12. Reboot your system.
 - Select **Yes** to reboot, the new parameters take effect.
 - If you select **Cancel**, the new parameters will not take effect until you reboot.
- Step 13. Select **OK** to reboot the system now.

14.9.2 The **idtune(1M)** command

In addition to viewing tunable parameters with the **idtune(1M)** command, you can also change tunable parameters.

14.9.2.1 Changing the parameter

To change a tuneable parameter, enter:

```
/etc/conf/bin/idtune param value
```

Answer **y** to confirm the change.

For example, to change the value of MAXUP, enter:

```
/etc/conf/bin/idtune MAXUP 70
```

Answer **y** to confirm the change.

14.9.2.2 Rebuilding the kernel

The kernel must be rebuilt and rebooted for the change to become effective.

To rebuild the kernel, enter:

```
/etc/conf/bin/idbuild -B
```

The kernel is immediately rebuilt. Reboot your system for the changes to take effect.

If you do not use the **-B** option, the kernel is rebuilt on the next system boot.

14.10 Lesson Summary

You now have the tools to monitor your systems for CPU, I/O, and memory saturation using the **scoadmin(1M) System Monitor**. If a performance issue warrants, use the **scoadmin(1M) System Tuner** to tune and rebuild the UnixWare 7 kernel.

Review questions

1. Describe three hardware factors that influence system performance.
2. What are the three steps when setting performance goals?
3. Which three commands can be used to identify CPU saturation?
4. Briefly describe the scheme for allocating memory for disk buffers?
5. Describe the term *paging*.
6. What information is monitored by the **scoadmin(1M) System Monitor**?
7. Which **idtune(1M)** command option is used to view parameters that can be autotuned?

8. Which method is the preferred method to change kernel parameters?

14.11 Lab: Reconfiguring the kernel

You have determined that the number of processes a user can execute simultaneously on your system should be 85. The default value is 80.

Overview

Make a copy of the existing kernel in the */stand* directory before changing a kernel parameter. Run the **scoadmin(1M) System Tuner** and increase the MAXUP parameter to 85. Rebuild and reboot the new kernel.

Procedure

- Step 1. Shut down the system to single-user mode.
- Step 2. Log in as **root**. Enter the **root** password.
- Step 3. Copy */stand/unix* to */stand/unix.before*
- Step 4. Enter **startx** to start up a graphical screen.
- Step 5. Run **scoadmin(1M)**.
- Step 6. Select **System** ⇒ **System Tuner**.
- Step 7. Click on the “Category” button.
- Step 8. Select “Process Limit Parameters.”
- Step 9. Select the MAXUP parameter.
- Step 10. Enter 85 as the new value.
- Step 11. Click on **OK**.
- Step 12. Select **Yes** to rebuild the kernel now.
- Step 13. Select **Yes** to reboot your system.
- Step 14. Select **OK** to reboot the system now.

14.12 Relevant Commands and Files

The following commands and files were presented in this lesson.

Relevant commands

id tune (1M)	Changes kernel parameters
ps (1)	Reports process status
sar (1)	Reports system activity
scoadmin (1M)	Invokes SCOadmin applications (in this lesson, the System Monitor and System Tuner)
swap (1M)	Adds, deletes, and monitors the system swap areas

Relevant files

<i>/stand/unix</i>	The UNIX kernel
<i>/stand/unix.old</i>	The UNIX kernel created before a new kernel is rebuilt

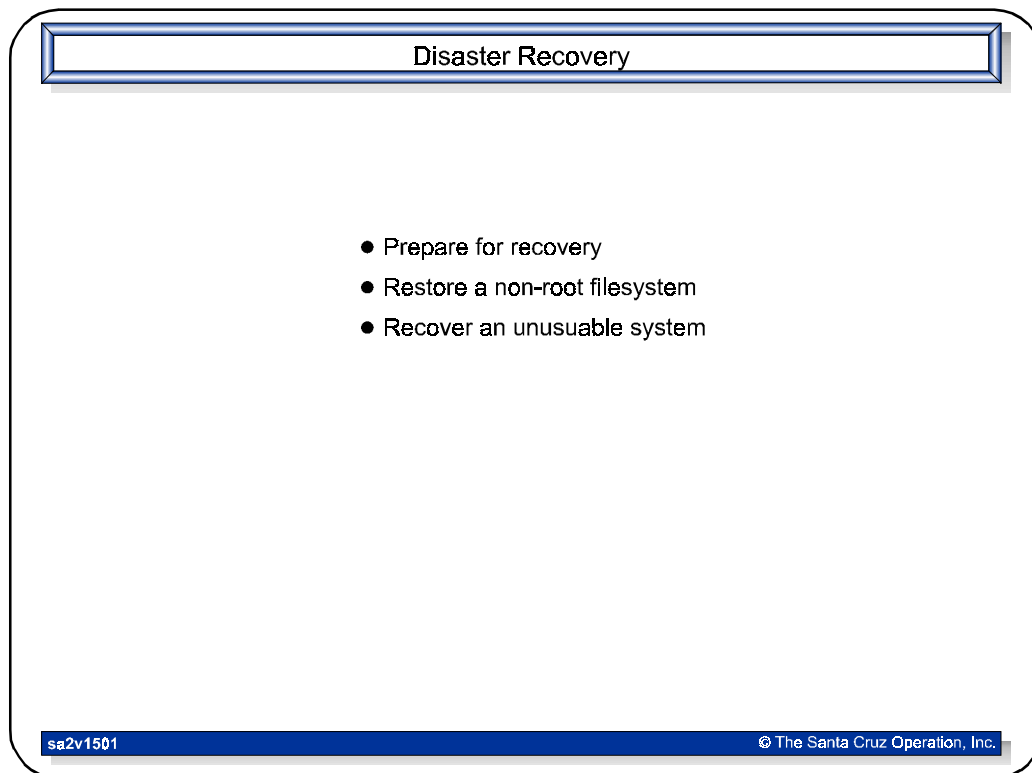
14.13 Answers to Review Questions

1. Hardware factors that influence system performance include the speed and width of the system's address and data bus, the amount of memory, access time for hard disks, and whether smart or dumb serial cards are used.
2. The three steps for setting performance goals are defining your goals, listing your goals, and understanding your goals.
3. The **sar**(1M), **ps**(1), and **scoadmin**(1M) System Monitor commands can be used to identify CPU saturation.
4. Buffer cache memory allocation is integrated with paging program schemes. Buffers are not preallocated. They are a dynamic or "paged" resource.
5. Paging is the process by which the contents of physical memory are moved both to and from the swap areas and filesystems.
6. The CPU usage, free memory, free blocks, paging, and swapping are monitored by the **scoadmin**(1M) System Monitor.
7. The -g option of the **idtune**(1M) command is used to view parameters that can be autotuned
8. Using the **scoadmin**(1M) System Tuner is the preferred method to change kernel parameters.

15

Disaster Recovery

AV 15-1



Lesson 15

Disaster Recovery

Introduction

Now that you are familiar with the files and directories that make up your UnixWare 7 system, you are better prepared to recover from a loss of, or damage to, the data stored on your hard disk. The extent of damage can range from a missing file to a corrupted system. In any event, you want to be able to restore the system to a running state as quickly as possible.

Objectives

This lesson covers the tasks needed to recover your system. By the end of this lesson, you will be able to:

- Prepare for recovery
- Restore a non-root filesystem
- Recover an unusable system

15.1 Preparing for Recovery

Backing up your system regularly is a key component in recovering your system. A good backup allows you to restore files and non-root filesystems, but what happens when you need to restore your root filesystem or replace a primary root disk? In these types of situations, a good backup is only one component of many to ensure a complete system recovery.

15.1.1 Keys to recovering your system

To ensure that you always have a set of emergency media and data backups, SCO recommends that you:

- Create emergency recovery diskettes and tapes immediately after installing and configuring your system

If you install additional system or application software at a later time, create new emergency recovery disks and tapes.

- Back up your system on a regular basis

15.1.2 Recovery components

The following components should be available:

- Original media—Is required if emergency recovery media is not available
- Registration information—If you need to call SCO for technical support, have your license number and registration keys ready. SCO needs this information before they can assist you.
- Emergency recovery diskettes—They are needed to restore your system. You can also restore a limited number of selected files from your emergency recovery diskettes.
- Emergency recovery tapes—Required to restore your system if it is corrupted beyond repair or your hard disk has been reformatted or replaced
- System and backup logs—If these logs are complete (that is, they detail how your system is organized, what backup method you use, and so forth), they provide the necessary information for a full system recovery.
- Backups—Your most recent data backups

If you are receiving errors regarding a non-root filesystem, first run the **fsck(1M)** command to correct any inconsistencies. If running **fsck** does not resolve the problem, you can restore the filesystem.

15.1.3 Running fsck(1M)

Remember when running **fsck**, the filesystem must be unmounted. To run **fsck** from the command line, enter:

```
fsck /dev/rdisk/c0b0t0d0s4
```

Exercise: Running fsck(1M)

Run **fsck(1M)** on your */home* filesystem.

15.1.4 Recreating a non-root filesystem

Before restoring your non-root filesystem, SCO recommends that you recreate the filesystem. **WARNING:** when recreating a filesystem all data is destroyed. Make sure that you have good backups.

The following steps describe how to recreate a non-root filesystem:

- Step 1. Shut down your system to single-user mode.
- Step 2. Unmount the filesystem.
- Step 3. View the options to the **mkfs(1M)** command that were used to originally create your non-root filesystem. Enter:

```
mkfs -m filesystem_name
```

For example, to view the **mkfs** options for the */home* filesystem on disk 1, slice 4, enter:

```
mkfs -m /dev/rdisk/c0b0t0d0s4
```

- Step 4. Recreate your filesystem using the information from Step 3. For example, to recreate the */home* filesystem, enter:

```
mkfs -F vxfs -o ninode=65535,  
bsize=1024,version=4,inosize=256,  
logsize=1024,nolargefiles /dev/rdisk/c0b0t0d0s4  
2056320
```

This **mkfs** example is a one line command.

- Step 5. Mount your filesystem.
- Step 6. Return to the networking state. Enter:

```
init 3
```

15.1.5 Restoring a non-root filesystem

After the filesystem has been recreated with the **mkfs** (1M) program, to restore the filesystem (assuming you use ARCserve /IT v6.6 from Computer Associates to back up your system), do the following:

- Step 1. Log in as **root**.
- Step 2. Mount the filesystem.
- Step 3. Run **scoadmin**(1M).
- Step 4. Select **ARCserveIT Backup Manager**.
- Step 5. Insert the media.
- Step 6. Select **Restore**.
- Step 7. Select the directory in the **Source Area** to be restored.
- Step 8. Select the directory in the **Destination Area** to which the data is restored.
- Step 9. Select **Restore** ⇒ **Run**.
- Step 10. Select **File** ⇒ **Exit** to exit the **Restore** window.
- Step 11. Select **Host** ⇒ **Exit** to exit the **SCO ARCserveIT Backup Manager**.

Exercise: Recreating and restoring a non-root filesystem

Recreate your */home* filesystem using the steps outlined in Section 15.1.4, “Recreating a non-root filesystem,” on page 15-5.

After you recreate your */home* filesystem, use the steps outlined in Section 15.1.5, “Restoring a non-root filesystem,” on page 15-6 to restore your */home* filesystem.

15.2 Recovering an Unusable System

You may need to perform a full recovery of your system if your:

- System will not boot
- System software is corrupted beyond repair
- Hard disk is reformatted or replaced

15.2.1 Recovery components

To ensure a complete recovery of your system, the following components are required:

- Emergency recovery diskettes
- Emergency recovery tapes
- Your SCO licensing and registration information. This is only necessary if you have reformatted or replaced your hard disk.
- Your system log
- Your most recent user filesystem(s) backups. User filesystem data is restored after you have recovered and rebooted your system from your hard disk.

15.2.2 Performing the recovery

After you have verified that your hardware is in working order, use these steps as a guide to perform a full recovery of your system:

- Step 1. Boot your system with the first emergency recovery diskette.
- Step 2. Insert the second emergency recovery diskette and press .
- Step 3. Select “Restore Diskettes” from the emergency recovery menu.
- Step 4. Insert your first tape into the drive and press .

If have more than one tape, you are prompted to insert the additional tapes.

The emergency recovery procedure retensions each tape before restoring the tape to disk.

- Step 5. Press  to continue after the tapes have successfully restored.
- Step 6. Select “Reboot” to reboot your system.
- Step 7. Remove your diskettes and tapes.
- Step 8. Press  to shut down and reboot your system.
- Step 9. Restore your most recent data backups after the system reboots.

15.2.3 Repairing the file privilege database

After the system reboots, you may notice error messages similar to the following:

```
UX:initprivs: WARNING: FILE ' file ' fails validation: entry ignored
UX:initprivs: WARNING: X entries ignored in ' /etc/security/tcb/privs '
```

Recall that restoring system files causes the “last update” time of the file to be modified and does not match the last update time in the file privilege database.

When the system boots to multiuser mode, to synchronize the file privilege database, enter:

```
/etc/security/tools/setpriv -x
```

15.3 Lesson Summary

Restoring filesystems is much easier when all recovery components are available. The key components are:

- Emergency recovery diskettes
- Emergency recovery tapes
- Recent data backups

Review questions

1. List the recovery components.
2. Which option to the **mkfs**(1M) command displays the syntax used to recreate an existing filesystem?
3. Which command would you use to synchronize the dates in the file privilege database?

15.4 Lab: Recovering an unusable system

You will restore your system to the most recent backup in this lab.

Overview

After trying to repair your root filesystem, you have decided to recover your system. You have your emergency disks, tapes, and your most recent backups of your user filesystems.

Your instructor will provide you with the media needed to complete this lab.

Procedure

- Step 1. Boot your system with the first emergency recovery diskette.
- Step 2. Insert the second emergency recovery diskette and press .
- Step 3. Select “Restore Diskettes” from the emergency recovery menu.
- Step 4. Insert your first tape into the drive and press .
- If have more than one tape, you are prompted to insert the additional tapes.
- The emergency recovery procedure retentions each tape before restoring the tape to disk.
- Step 5. Press to continue after the tapes have successfully restored.
- Step 6. Select “Reboot” to reboot your system.
- Step 7. Remove your diskettes and tapes.
- Step 8. Press to shut down and reboot your system.
- Step 9. Restore your most recent data backups after the system reboots.

15.5 Relevant Commands

The following commands were presented in this lesson.

Relevant commands

fsck(1M)	Repairs a filesystem
mkfs(1M)	Creates a filesystem
SCO ARCserveIT	Performs system and data backups
/etc/security/tools/setpriv	Repairs the file privilege database

15.6 Answers to Review Questions

1. The key recovery components include the original media, registration information, emergency recovery diskettes and tapes, system and backup logs, and data backups.
2. The -m option to the **mkfs**(1M) command is used to display the syntax used to recreate an existing filesystem.
3. You would use the `/etc/security/tools/setpriv -x` command to synchronize the dates in the file privilege database.

A

Basic Hardware Configuration

AV A-1

Basic Hardware Configuration

- Identify the major components of a typical system
- Describe the functions of microprocessors and memory
- Describe the bus architectures supported by SCO
- Describe the functions of SCSI host adapters
- Describe supported hard disk configurations
- Identify general compatibility issues
- Maintain a sample administration log book

sa2va01 © The Santa Cruz Operation, Inc.

Appendix A

Basic Hardware Configuration

Introduction

Before, during, and after you install UnixWare 7, you will need to answer questions about your hardware configuration. For example, you will complete an installation checklist as part of your pre-installation analysis, you will configure a hard disk during installation, and you will need to have a basic understanding of your hardware for post-installation tasks (such as installing a printer or CD-ROM drive).

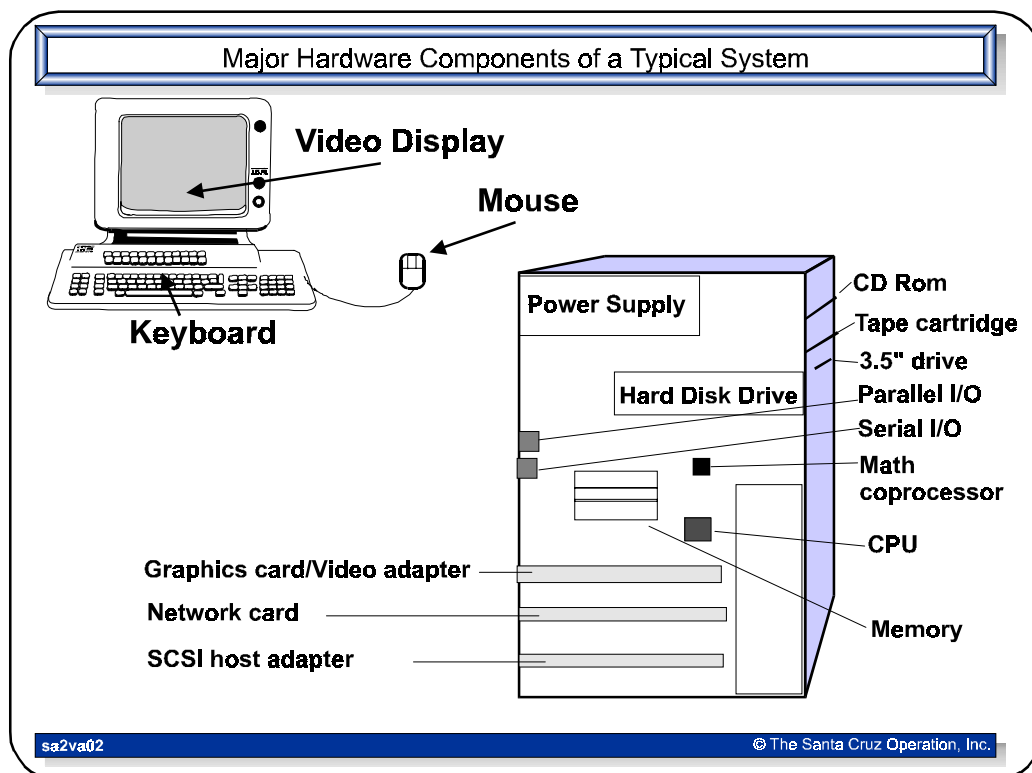
This lesson provides background information on system hardware and discusses hardware and architectures supported by SCO to make your installation and post-installation tasks easier.

Objectives

This lesson describes the basic, or major, hardware components of a typical system configuration. It also describes some specific information you will need to know about your system when installing UnixWare 7. By the end of this lesson, you will be able to:

- Identify the major components of a typical system
- Describe the functions of microprocessors and memory
- Describe the bus architectures supported by SCO
- Describe the functions of SCSI host adapters
- Describe supported hard disk configurations
- Identify general compatibility issues
- Maintain a sample administration log book

AV A-2



A.1 Hardware Overview

Although you and the users you support probably have different types of hardware purchased from a variety of vendors, most systems include some or all of these major hardware components:

- Video display screen (sometimes called a terminal or monitor)
- Mouse
- Keyboard
- One or more hard disk drives
- Power supply
- CD-ROM drive
- Cartridge tape drive
- 3.5" floppy drive (and sometimes a 5.25 inch floppy drive in older systems)
- Parallel I/O
- Serial I/O
- Audio
- Central Processing Unit (CPU)
- Memory
- Graphics card/video adapter
- Network card
- SCSI host adapter

A.1.1 Microprocessors and Memory

These three components are vital to any computer system:

Microprocessor (Central Processing Unit or CPU)

The central unit that executes all data processing (adding, subtracting, logic operations, and so forth)

Memory

Area where the CPU stores the intermediate results of the data that the CPU processes; also reads the data back in and stores programs

Input/Output (I/O) support circuitry

Provides the signals that the CPU needs to operate; includes clocks, controllers, and signal converters

In systems with the *single board* design, all of these vital components are located on one board called the *motherboard*. In modular or bus-oriented systems, the motherboard is divided into a separate processor board and a bus board that contains the slots into which adapter cards are inserted.

In addition to CPU, memory, and I/O support circuitry, the motherboard is comprised of some or all of the following:

Mass storage interfaces

Include connections to CD-ROM drives, hard disk drives, and floppy disk drives

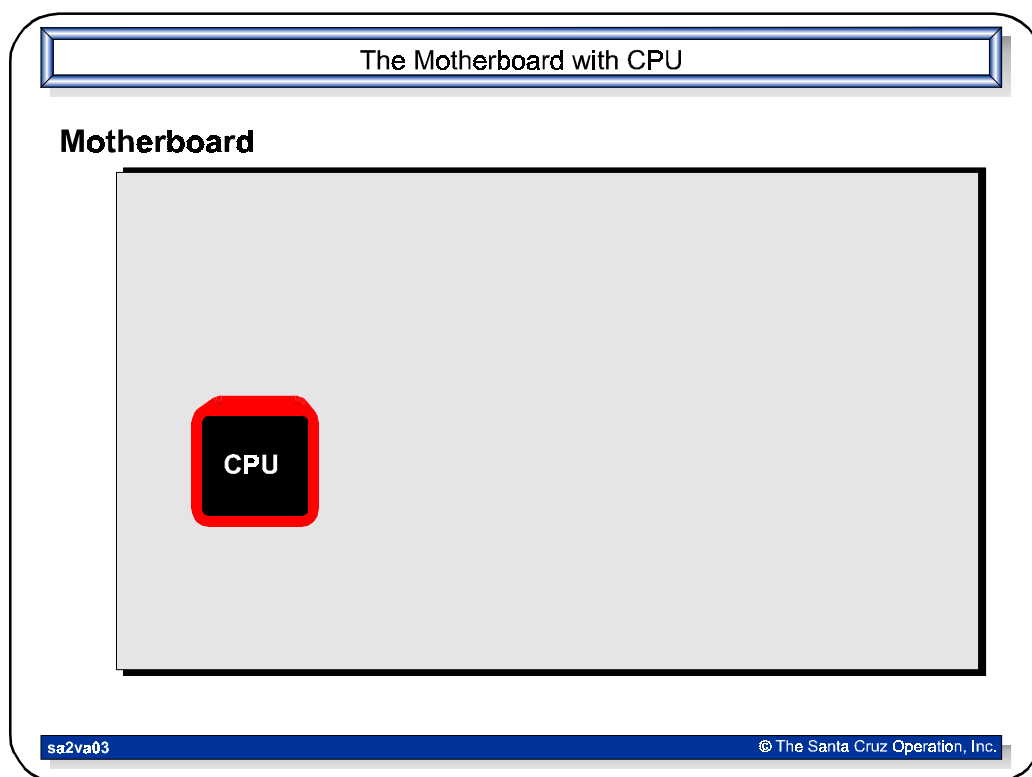
I/O ports

For serial and terminal interfaces, such as terminals, printers, and so forth

BIOS

Provides the system with fundamental operational characteristics and instructions when the system is powered up

AV A-3



Expansion slots	Enable you to enhance your system configuration by adding graphics adapters, network adapters, and so forth. This is also the area where you would install a SCSI interface card. The SCSI interface is discussed later in this lesson in “Small Computer System Interface (SCSI) Host Adapters.”
-----------------	---

A.1.2 Microprocessors

The microprocessor is the central and most essential element of your computer; it performs all logical and arithmetic operations. It reads data from memory, processes that data, and then writes the results back into memory. This is called the *fetch/execute* cycle.

The two main types of microprocessors available in today’s market are:

- The Intel[®] 80x86 family (where x= 3, 4, and so forth)
- The 68000 Motorola[™] series

Intel, credited with the invention of the microprocessor, is currently the largest manufacturer of microprocessors. This section briefly describes these Intel microprocessors:

- i486
- Intel Pentium[®]
- Intel Pentium Pro[™]
- Intel Pentium II[™]
- Intel[®] Pentium[®] III

A.1.2.1 The i486 microprocessor

You need at least an i486 to run a graphical workstation.

Intel introduced the fourth generation of the 80x86 family, the i486, in 1989. It is a powerful microprocessor that includes the following on one chip:

- An improved i386 CPU
- A cache controller including 8KB of cache memory
- An on-board (or on-chip) Floating Point Unit (FPU) with an improved multiplier

A.1.2.2 The Pentium

The Pentium chip, introduced by Intel in 1993, is faster and more powerful than its 80x86 predecessors.

Features of the Pentium include:

- 64-bit data bus
- 32-bit processor
- A superscalar architecture that can execute simple instructions in parallel
- An improved floating-point unit for performance enhancement
- Two separate L1-caches with 8KB each for code and data to support the Modified, Exclusive, Shared or Invalid protocol (MESI)
- 4KB or 4GB page sizes

A.1.3 The Pentium Pro

The Pentium Pro increases speed and boosts the overall performance of the software through a unique combination of processing techniques called *Dynamic Execution*.

In addition to Dynamic Execution, other features of the Pentium Pro include:

- 5.5 million transistor chip
- 256KB L2-cache (or second-level cache)
- An improved superscalar architecture
- Advanced data integrity, reliability, and serviceability
- Compatibility with existing PC applications
- 4KB or 4GB page sizes

A.1.3.1 The Pentium II

The Pentium II processor is the solution of choice for entry level servers. The Pentium II processor provides fast single and dual processor workgroup and web servers. In addition to the features found with the Pentium Pro, the Pentium II processor also incorporates:

MMX™	A technology for advanced server applications. MMX is a set of 57 new instructions introduced by Intel into Pentium and Pentium II processors.
ECC	A feature available on the L2 cache of the Pentium II processor. ECC provides added data integrity and reliability.

A.1.3.2 The Pentium II Xeon

The Pentium II Xeon processor delivers breakthrough levels of performance. It's ideal for any mid-range or higher Intel-based server where performance, scalability, and reliability are essential. Servers based on the Pentium II Xeon processor extend Intel Architecture price/performance to all levels of the enterprise. Xeon features include:

- Large, fast caches
- Thermal protection
- Error checking and correcting
- Functional redundancy checking
- System Management Bus to help ensure maximum reliability and uptime

The UnixWare7 operating system is optimized to exploit the new enterprise capabilities in the Intel Pentium II Xeon processor and usher in a new class of servers for enterprise computing.

UnixWare 7 takes advantage of the Intel Extended Server Memory Architecture - a collection of technologies that allows operating systems and enterprise applications to achieve a fundamental performance benefit and exceed the 4Gbyte memory barrier.

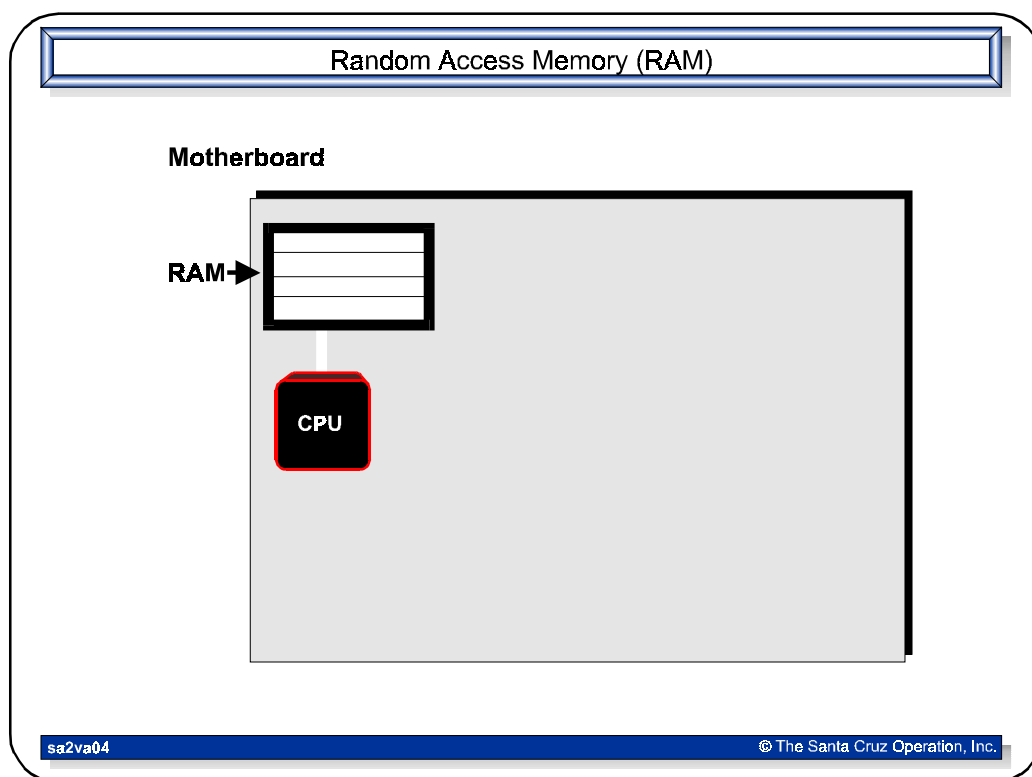
This feature provides for 36-bit addressing which for the first time unleashes the enterprise power of UnixWare 7 with its support for an unprecedented 64Gbytes of memory.

A.1.3.3 Pentium III

The Pentium III processor is designed with capabilities targeted to enhance internet access. It integrates attributes of Intel's latest processor platforms with new technology enabling advanced imaging, 3D graphics, streaming audio and video, and speech recognition applications. The Pentium III processor features 70 new instructions that can enhance the use of Web technologies. The Pentium III features include:

- 450 Mhz and 500 Hhz Clock Speeds
- 70 Internet Instructions
- Intel Processor Serial Number
- P6 Microarchitecture
- 100 Mhz System Bus
- 512k Level 2 Cache
- Intel 440bx Chip Set

AV A-4



A.1.4 Memory

All systems contain two types of memory:

- Random Access Memory (RAM)
- Read-only Memory (ROM)

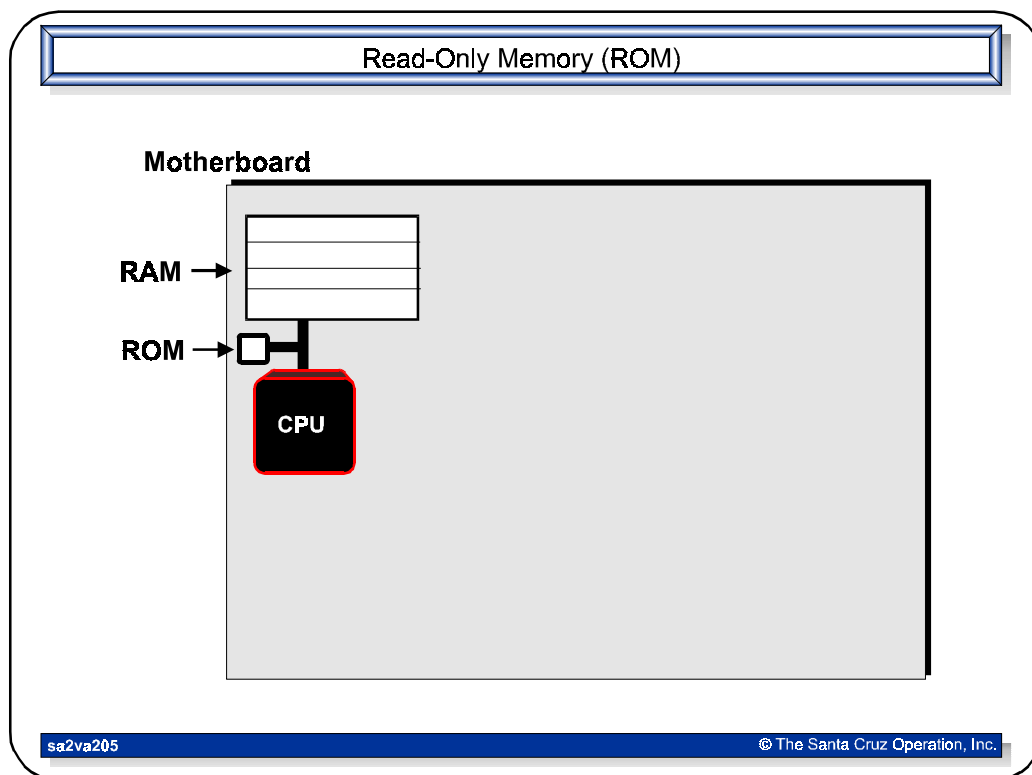
Your system may also have Direct Memory Access (DMA) chips connected to main memory. DMA allows data to transfer between main memory and other devices (such as disk drives) without CPU intervention. DMA is discussed in more detail later in this lesson in “DMA Channels.”

A.1.4.1 Random Access Memory (RAM)

RAM, or main memory, is the computer’s storage area for data. The CPU reads data in from memory, processes that data, and writes the results back into memory.

RAM is volatile—that is, when the machine is powered off, the data in RAM is lost.

AV A-5

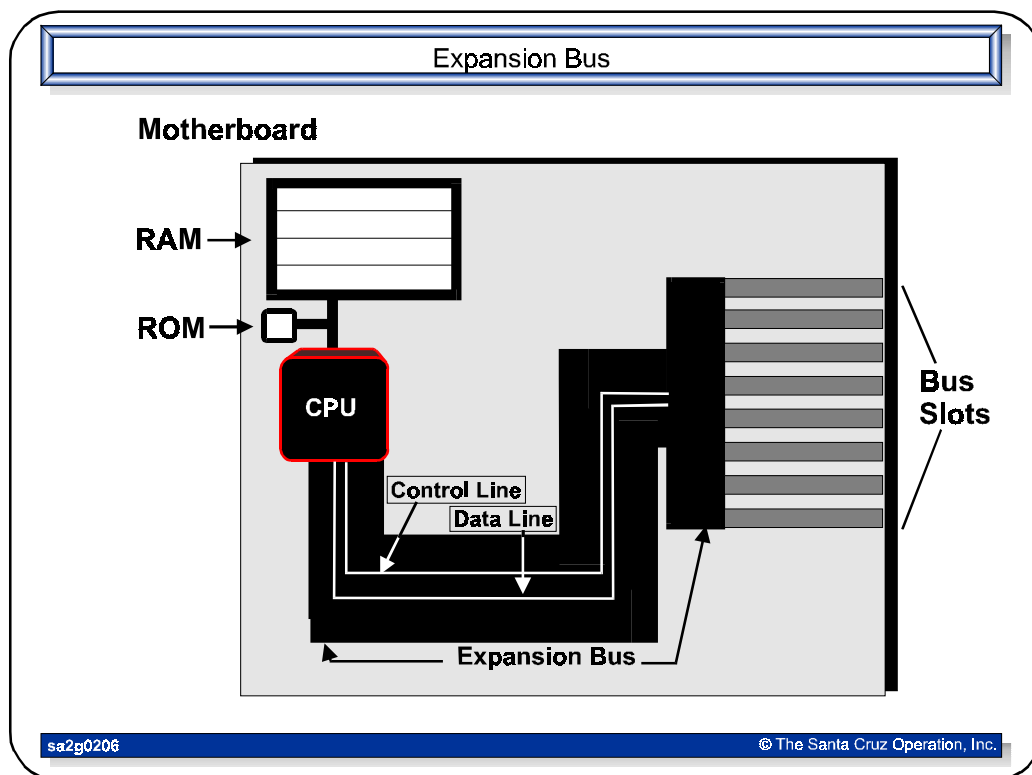


A.1.4.2 Read-Only Memory (ROM)

ROM is a pre-programming memory chip with instructions and information specific to your computer's type.

All computer systems have a ROM-based Basic Input/Output System (BIOS) that contains the boot instructions that are executed when your system powers up. For example, the ROM-BIOS holds the instruction for accessing your keyboard and graphics adapter.

AV A-6



A.2 Bus Basics

A bus, or expansion bus, is a high speed connection of several lines (generally parallel) that transmit control, data, and address signals. The bus is comprised of the following:

Data lines	Carry data between the CPU and the controllers plugged into the bus slots
Control lines	Carry the signals required to control the data transfer

A bus enables you to boost the power of your system and expand its capabilities by adding expansion or adapter cards to its *bus slots*. For example, you might want to add a new graphics adapter card to provide higher resolution on your monitor, or a network adapter card that allows you to network your system to other computers.

Although the main function of the bus is to provide a connection for moving data between add-on cards, the CPU, and the rest of the computer, different types (or architectures) of buses were designed for different purposes and different systems.

A.2.1 Bus architectures

The kind of enhancements (or products) that you can add to your computer are determined by the standards set by your system's expansion bus. This section provides an overview of the following bus architectures supported by SCO:

- Industry Standard Architecture (ISA)
- Micro Channel Architecture (MCA)
- Enhanced ISA (EISA)
- Personal Computer Memory Card International Association (PCMCIA)
- Peripheral Component Interconnect (PCI)
- I²O

A.2.1.1 Industry Standard Architecture (ISA)

The ISA is sometimes called the *classic* bus because it defined the standards that replaced the specifications for the Advanced Technology, or AT[®], bus. The AT bus is the 16-bit successor to older IBM[®] Personal Computer (PC) technology.

The original standards set for the AT bus were not clear. In 1987, the Institute of Electrical Engineers (IEEE) defined a strict set of standards called ISA for the AT bus. ISA added functionality that was needed to move into 16-bit technology, but retained compatibility with earlier computer technologies.

The ISA standard defines the following:

- Bus structure
- Architecture of the CPU and support chips
- Clock frequency

A.2.1.2 Micro Channel Architecture (MCA)

IBM designed the MCA bus in 1987 to match the then powerful i368 microprocessor. It resolved many of the problems seen with the original PC bus.

Although the MCA bus did not really improve system performance, it did give the system more overall potential by adding the following features:

- 32-bit design with 32-bit addressing
- A power supply (or ground) conductor to protect against radio frequency interference
- A redesigned high-speed bus
- Interrupt sharing
- *Bus masters* for managing data across the bus
- A *burst mode* that allows a single device to control the bus for a maximum of 12 milliseconds
- Upgraded clock and data path

A.2.1.3 Enhanced ISA (EISA)

In 1988, nine computer companies joined forces to design the EISA bus. Basically, they borrowed from and improved on the MCA and other bus designs in the industry.

Features of the EISA bus include:

- 32-bit extension for the ISA (or AT) bus
- Bus mastering
- Automated setup
- Interrupt sharing
- New data transfer modes for improved performance
- Backwards compatibility with the ISA (or AT) bus and the older XT bus

A.2.1.4 Personal Computer Memory Card International Association (PCMCIA)

The PCMCIA bus was introduced in 1990 as the standard for expansion boards designed for notebook, laptop, and other compact computer systems. The PCMCIA interface provides a connection between a small, credit card-sized *PC* or *PCMCIA card*, and the host computer.

The original specification, standard 1.0, was intended only as a memory expansion. The latest standard, 2.0, accommodates both storage and I/O devices, including fax modems, SCSI adapters, and Global Positioning System (GPS) receivers.

The PCMCIA standard defines the following:

- File formats
- Data structures
- Device-independent means of accessing card hardware
- Software links independent of operating systems

UnixWare 7's implementation of the PC Card interface is based on technology acquired from Lynxsoft, Inc. This interface is based on the specification for peripheral device support for laptop computers that is published by the PCMCIA.

UnixWare 7 systems support a limited number of PCMCIA adapters but do not currently support the complete PCMCIA subsystem.

A.2.2 Peripheral Component Interconnect (PCI)

Intel introduced the PCI bus in July, 1992, as a standard interface between the CPU and I/O devices. The PCI common standard is independent of the microprocessor family, allowing processors, coprocessors, and support chips to be linked without needing to match the signals between different integrated circuits.

The PCI bus is a *local bus*.

PCI is a high-end solution for powerful personal computers, workstations, and computer systems. It is typically used for video and disk drive interconnections to the CPU. In addition to mandatory design rules and processor independent specifications, its features include the following:

- Expansion to 64 bits
- Multiprocessor support
- Burst transfers with arbitrary length
- ISA, MCA, and EISA support
- High-speed operation
- Automatic detection and configuration of hardware added to PCI slots

A.2.2.1 Intelligent Input/Output (I²O)

I²O is a technology developed by Intel and its partners to define a standard architecture for intelligent I/O that is both device and operating system independent.

I²O was developed to address two key problems relating to I/O:

- Negative impact of I/O interrupts serviced by the CPU on overall system performance
- Costly work of creating separate drivers for an I/O device for the numerous operating system choices

I²O utilizes a special I/O processor that can be incorporated into system motherboards and into I²O peripherals.

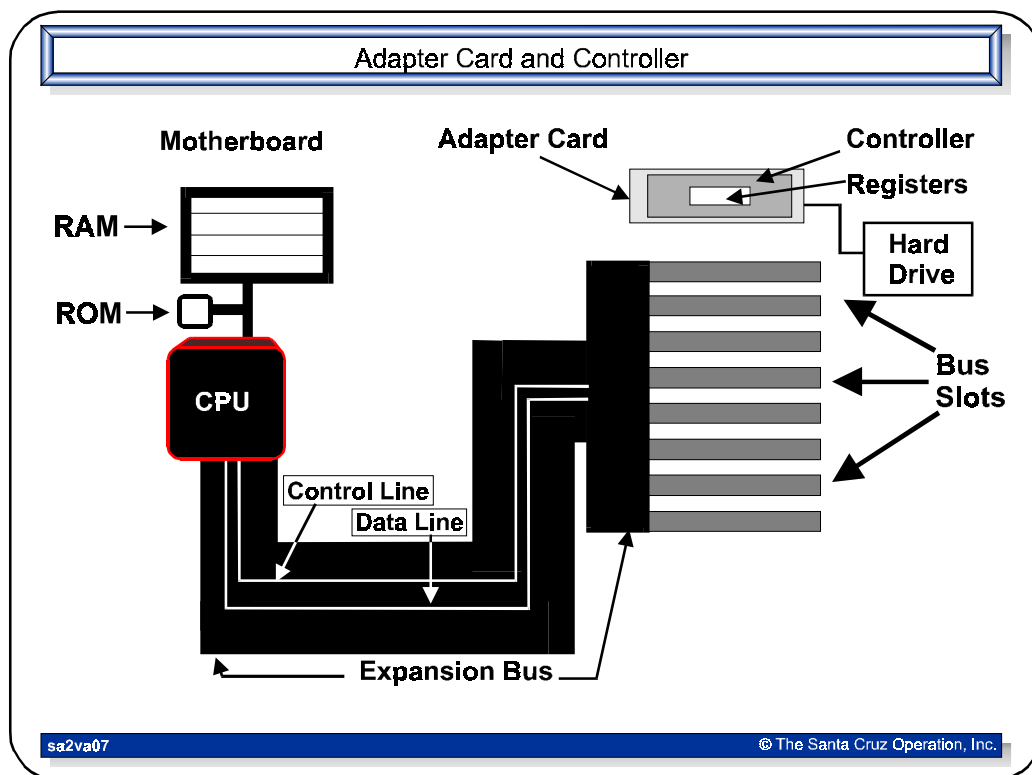
The SCO implementation of the I2O architecture conforms to the specifications of the I2O Special Interest Group (SIG), of which SCO is a member. Further information is available at the I2O SIG web site at: <http://www.i2osig.org/>

A.2.3 Accelerated Graphics Port (AGP)

AGP is port interface pioneered by Intel and specifically designed for graphics I/O. Because it only involves two devices, the processor and video card, and is not expandable, AGP is considered a port, and not a bus. AGP eliminates the need for high cost and limited on board video RAM by allowing the video processor to access the main system memory for doing its calculations.

Check the SCO Compatible Hardware Web Pages at:
http://wdb1.sco.com/chwp/owa/hch_search_form for AGP cards supported with UnixWare 7 video drivers. If there is an accelerated driver available for your AGP card than you will be able to take advantage of the performance benefits of AGP with UnixWare 7.

AV A-7



A.2.4 Adapter cards and controllers

An adapter card is the board that you insert into a bus slot to expand the capabilities of the bus. Examples of adapter cards include interface adapter cards, graphics adapter cards, and network adapter cards.

An adapter card can have one or more *controllers*. A controller is a device, usually a set of connectors and cables, that allows the CPU to communicate with peripheral devices (hard disk drives, floppy disk drives, CD-ROM drives, and so forth).

When you add adapter cards to your system, consider the following:

- For EISA bus systems, you must run the manufacturer's setup program to change the system's configuration before you can use the new hardware
- For MCA and ISA bus systems, you may need to run the manufacturer's setup program to change the system's configuration before you can use the new hardware
- For machines with both PCI and ISA buses, the PCI auto-configuration code cannot read the configuration of cards in the ISA bus slots. You must use the manufacturer's setup program to tell the auto-configuration code which interrupts are in use by ISA bus cards. If you do not mark these interrupts as "used," you may have an interrupt clash between an ISA bus card and a PCI bus card.

A.2.4.1 I/O base addresses

An area of dedicated-purpose RAM called *registers* stores commands that directly control hardware operations and status codes for peripheral devices. To access these registers, the CPU uses a special set of instructions called *IN* and *OUT* instructions.

Controllers map their registers into a start address in a special area of memory called *I/O memory*. This start address is called the *I/O base address*.

Each controller on your system that performs I/O needs a unique memory base address so that the system can locate it. No two controllers can map to the same I/O space. For example, if you are using 0x300, or 300, as the default I/O base address for the 3Com[®] 3c509 network adapter, no other device can use this address.

The I/O base address:

- Is a three-digit hexadecimal number
- Must match the settings on the adapter
- Must not conflict with any other hardware on your system

A.2.4.2 Shared RAM address

The *Shared RAM address* is the address of a unique area of memory allocated to the host machine and network or graphics adapters for read/write operations. This area is sometimes called the *RAM buffer*. Other devices must not use any address in the range from the RAM base address to the end of the allocated memory (that is, the RAM base address + the RAM buffer size).

A.2.4.3 Shared ROM address

Some adapters (for example, the IBM Token-Ring adapters) require that you specify a ROM base address. Like the RAM address, the ROM base address specifies a specific part of memory that is set aside for networking use.

If you have more than one adapter requiring that you specify a ROM address, make sure you do not assign addresses that conflict with the ROM region.

AV A-8

Typical Device Interrupts		
Interrupt	Octal	Device
0*	0	Clock
1*	1	Console
2	2	Networks, tapes and others
3	3	Serial COM2
4	4	Serial COM1
5	5	Alternate parallel port (1p2)
6*	6	Floppy disk
7	7	Main parallel port (1p0 or 1p1)
9*	11	Chain from IRQ2
10	12	
11*	13	SCSI host adapter 1
12	14	SCSI host adapter 1
13*	15	FPU
14*	16	ST506/ESDI/IDE controller 0
15	17	ST506/ESDI/IDE controller 1

*Do not attempt to use these interrupts for any other purpose.

sa2va08 © The Santa Cruz Operation, Inc.

A.2.4.4 IRQs

The CPU knows when an I/O operation is complete in one of three ways:

Busy-wait	Used at boot-time because no other option is available
Polling	Checks the control registers every 1/HZ times per second
Interrupt Requests (IRQs)	Signal the CPU when a controller needs attention

IRQs temporarily suspend normal code execution for the currently running program to execute an *interrupt handler* routine (for example, an input/output operation). When the operation completes, code execution for the program resumes.

One of the most common causes of problems both during and after installation is a device interrupt conflict. Each device on your system must have its own (IRQ) setting, with the exception of devices that support interrupt sharing.

A.2.4.5 DMA channels

Once the CPU knows that the controller has data, the data is transferred to RAM by one of the following methods:

Programmed I/O	Transfers the data byte by byte through the CPU
Dual-ported RAM	Allows simultaneous reading and writing of data
Local bus access	Transfers data via a local bus

As discussed previously in section 2.2, “Memory,” data can also be transferred directly between memory and a device without CPU intervention using Direct Memory Access (DMA). A *DMA chip* or *DMA controller* establishes the path, or *DMA channel*, between a device and main memory.

Each computer has eight DMA channels for conveying data. Devices capable of performing DMA transfers require that you specify the DMA channel you want the adapter to use. Unlike I/O base addresses, IRQs, and memory, DMA channels can, but should not be, shared.

In addition, some channels are reserved or are typically used for certain functions or devices as follows:

DMA channel 4	Reserved for use by the system
DMA channel 0	Typically used for RAM REFRESH
DMA channel 2	Typically used for the floppy disk controller

During configuration, you can choose to use any (or none) of the channels.

A.3 Supported Hard Disk Configurations

UnixWare 7 systems support many hard disks and most hard disk controllers, including the following controllers:

- SCSI
- Integrated Drive Electronics (IDE)
- Enhanced IDE (EIDE)
- Enhanced Small Device Interface (ESDI)
- Compaq Intelligent Disk Array (IDA)

A.3.1 Small Computer System Interface (SCSI) Host Adapters

Small Computer System Interface (SCSI), commonly referred to as *scuzzy*, is an instruction-oriented, high-level interface specification for interconnecting a variety of peripheral devices (such as hard disk drives, CD-ROM drives, and tape drives) to a *SCSI bus*. This SCSI bus exchanges data only among the SCSI devices connected to it. Before you configure a new SCSI host adapter, you may need to run a setup program or configuration utility from the manufacturer.

A SCSI adapter provides the interface between your computer's system (expansion) bus and the SCSI bus. The SCSI host adapter:

- Controls access to SCSI peripherals (such as hard drives)
- Transfers data between peripherals and main memory
- Relays commands, status information, and messages

SCSI standards are defined by American National Standards Institute (ANSI). There are currently three main SCSI standards:

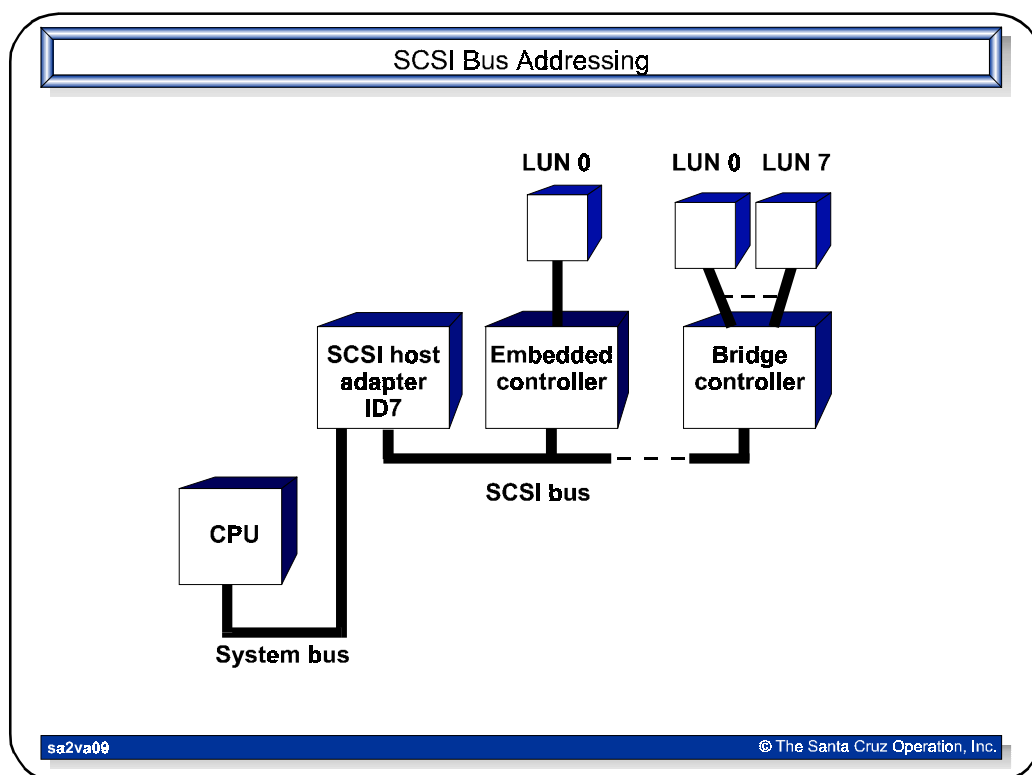
SCSI-1	Approved by ANSI in 1986 as standard X3.131-1986. It defines the basics of the first SCSI buses, including cable length, signaling characteristics, commands and transfer modes. It is limited as compared to functions defined in SCSI-2. It defines only a basic 8-bit narrow bus, and 5 MB/s transfer rate. SCSI-2 replaces the SCSI-1. Devices that adhere to the SCSI-1 standard can in most cases be used with host adapters and other devices that use the higher transfer rates of the more advanced SCSI-2 protocols, but they will still function at their original slow speed.
--------	---

- SCSI -2 Approved by ANSI in 1990. It is an extensive enhancement of the original standard, and defines support for many of the more advanced SCSI features used today. SCSI-2 supports the following new features as enhancements to the original SCSI-1 specification:
- Fast SCSI- This high-speed transfer protocol doubles the speed of the bus to 10 MHz.
 - Wide SCSI—Widening the original 8-bit SCSI bus to 16 bits or even 32 bits permits more data throughput at a given signaling speed.
 - More Devices per Bus—On buses that are running with Wide SCSI, 16 devices are supported (as opposed to 8 with SCSI-1).
 - Improved Cables and Connectors— SCSI-2 defines high-density connections.
 - Active Termination—SCSI-2 defines the use of active termination to provide more reliable bus termination.
 - Command Queuing—Allows multiple outstanding requests between devices on the bus, simultaneously.
 - Additional Command Sets—SCSI-2 added new command sets to support the use of more devices such as CD-ROMs, scanners and removable media.
 - Command Set Enhancements—The command sets for controlling all kinds of devices were enhanced, including diagnostic capabilities and messaging.
- SCSI-3 The SCSI-3 standard is still being defined. Features that it could include are:
- Ultra SCSI— Increase off system bus speed, defining transfer rates up to 20 MHz, meaning 20 MB/s with 8-bit SCSI, or more with Wide SCSI.
 - Improved Cabling—SCSI-3 again improves cabling over the improvements in SCSI-2, for the use of Wide SCSI.
 - Serial SCSI (Firewire)—SCSI-3 contains as one of its different protocol standards the description for the new Serial SCSI, also called Firewire.

For more information on SCSI standards refer to:

- SCSI Trade Association Web Site: <http://www.scsita.org/index.html>
- T10 Committee: <http://www.symbios.com/t10/>

AV A-9



A.3.2 SCSI addresses

Each controller on the SCSI bus, including the host adapter, has an address referred to as its *controller ID* or *target ID*. You can have up to eight controllers on a SCSI 1 bus with IDs from 0 to 7. You can have up to 16 controllers on a 16-bit-wide SCSI II bus with IDs from 0 to 15. The host adapter is usually assigned ID 7.

Although you can install controllers in any order on the bus, they each must have a unique target ID. You usually set the target ID on the SCSI peripheral device by using jumpers, DIP switches, or thumb-wheels. Refer to the documentation that comes with the controller for specific instructions.

A peripheral device connected to a controller is identified by its logical unit number (LUN). Most SCSI devices have a single embedded controller with a LUN of 0 (although bridge controllers can support up to eight devices with LUNs from 0 to 7).

When you configure the first SCSI peripheral connected to an adapter, the SCSI adapters are configured into the kernel.

A.3.3 General compatibility issues

This section provides information on general compatibility issues and configuration parameters. Consider the following compatibility issues when adding hardware to your UnixWare 7 system:

- Some additional hardware (such as host, video, or network adapters) may require new or updated device drivers. These drivers are supplied on a regular basis through the Advanced Hardware Supplement (AHS) program. Check with your reseller or sales representative for details.
- Supported machines are not always supplied with video adapters from the same manufacturer. Check the video adapter and see if it is listed in the ***UnixWare 7 Getting Started Guide and Release Notes***.
- Manufacturers of SCO-supported machines or add-on peripherals may change configuration, functionality, or firmware at any time

To determine whether hardware components are compatible with your machine, you must know the following:

- The microprocessor (for example, i486, Pentium, Pentium Pro, and Pentium II)
- The bus architecture (ISA, EISA, MCA, PCI, and so forth)
- The type of disk controller (IDE, EIDE, and so forth)

A.3.4 General configuration parameters

If your system hangs mysteriously at boot time, during installation, or shortly after installation, check for hardware conflicts between components, including the following:

- DMA channels
- I/O base addresses
- Shared RAM addresses
- Shared ROM addresses
- IRQs

A.4 Plug and Play (PnP)

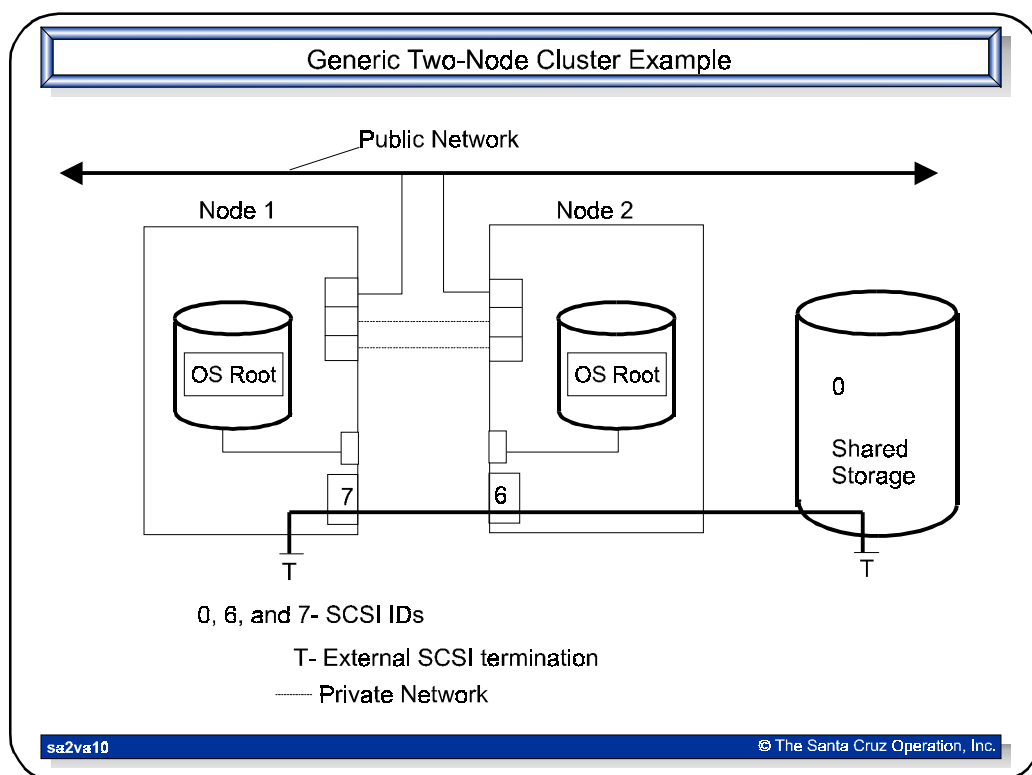
In an attempt to resolve this ongoing problem, the PnP specification was developed by Microsoft, Intel, and many other hardware manufacturers. The goal of Plug and Play is to create a method for hardware and software to work together for a more automatic configuration of devices and assignment of resources. As the name suggests, the goal is to be able to just plug in a new device and immediately be able to use it as quickly and easily as possible. PnP was widely available and used starting in 1995.

PnP requires the cooperation of four main computer components:

System Hardware	The hardware on your system, through the system chipset and system bus controllers, must be capable of handling PnP devices. PCI was designed with PnP in mind. Most PCI-based systems also support PnP on their ISA bus, with special circuitry to link the two together and share resource information. Older PCs with ISA-only or VL-bus system buses generally do not support Plug and Play.
System BIOS	The system BIOS plays a key role in a Plug and Play configuration. Routines built into the BIOS collect information about the PnP devices and determine how resources are allocated. The BIOS also communicates this information to the operating system, which uses it to configure its drivers and other software to make the devices work correctly. In many cases older PCs that have an outdated BIOS but otherwise have support for PnP in hardware (PCI-based Pentiums produced between 1993 and 1995 are the prime candidates) can be made PnP-compliant through a BIOS upgrade.
Peripheral Devices	The devices that you are adding into the system must be designed for PnP compatibility. PnP is now supported for a wide variety of devices including; modem, network, and sound cards.
Operating System	The operating system must also provide support for PnP devices and configuration.

UnixWare 7 supports the configuration of PnP ISA adapters, including sound adapters, SCSI host adapters, multiport serial boards, and others that conform to the Plug and Play ISA Specification Version 1.0A. To configure these devices use the **SCOadmin PnP Configuration Manager** and reference **scohelp** ⇒ **Hardware** ⇒ **Configuring Plug and Play devices**.

AV A-10



A.5 Clustering

The term clustering is applied by the computer industry to mean various things. For example, some specialized systems with many processors in a single cabinet, high-speed interconnects, shared hardware, and possibly shared memory, are referred to as utilizing “clusters” of massively parallel engines. These systems use multiple processors tied together in configurations of 100 or more processors, with some type of message passing and processing capability.

Generally though, a cluster will more often refer to the ability of multiple systems (nodes) to cooperatively work together to share a computing workload or resource. These nodes can exist in a single cabinet or be physically separate, each containing its own processor, memory, disks(s), and be connected via a LAN. Note, that this is not the same as an SMP (Symmetrical Multiprocessor) system, where a single system contains multiple processors that share memory and I/O devices.

A cluster can though be built from a series of connected SMP systems. A LAN-based cluster of systems, each with its own processor, memory and peripherals can appear as a single server to users and applications. It provides a cost effective way to gain features and benefits that to date have only been found on more expensive legacy systems.

UnixWare 7 supports high-availability multi-node system clustering with the SCO Reliant HA 1.1 is an add-on Optional Service for UnixWare 7 systems. For more information on clustering reference <http://www.sco.com/products/clustering/>.

A.6 Maintaining a Sample System Administration Log Book

You should maintain a log book that details the hardware configuration specific to your server, as well as some information about the software installed on that server.

Typically, the log book contains the following information:

- Installed hardware and software, including updates, patches, etc.
- Repair history, both hardware and software
- Details of scheduled and unscheduled maintenance and backups
- Copies of important configuration files
- Locations of media and serial numbers
- Printouts, command output, and so forth, of any other information that you think might be useful. For example:
 - Disk free space (**df -v**)
 - The `/dev` directory (**lr -l /dev | lp**)

Exercise: Using the sample system administration log book

Examine the sample system administration log book that is included with your student kit.

Become familiar with your classroom system. Log in as **root** and enter the following commands:

dfspace

fdisk

sar -r 1 5

sar -u 1 5

scoadmin system information

sdiconfig -l

uname -X

A.7 Lesson Summary

Understanding your system's hardware makes installation, configuration, and maintenance tasks simpler. Most systems are comprised of the same basic components. The motherboard contains the vital components of a system.

The kind of enhancements you can make to your system is determined by its bus architecture.

Before you configure a hard disk, you should have a basic understanding of hard disk device drivers and disk controllers.

For any hardware installation, you should be aware of general compatibility issues and possible hardware conflicts.

You should keep an administration log that details the specifics about your server's hardware and software, including:

- Details of the hardware and software installed
- Repair history
- Details of planned and unplanned maintenance
- Copies of important configuration files

Review questions

1. Name at least seven major components of a "typical" system.
2. What board contains the microprocessor and memory?
3. What are three main functions of a SCSI host adapter?

4. Name at least three types of disk controllers supported by SCO.

5. What is one of the most common causes of problems during installation, and what do you need to do to resolve the problem?

6. List five items that should be listed in a well-maintained system administration log book.

A.8 Relevant Commands

The following command were presented in this lesson.

Relevant commands

df(1)	Displays disk free space; with the -v option, reports the percent of blocks used as well as the number of free blocks
dfspace(1)	Displays disk free space in megabytes
fdisk(1M)	Displays, creates, and deletes partitions
sar(1M)	Reports on system activity
scoadmin(1M)	Invokes SCOadmin applications (in this lesson, the System Information Manager)
sdiconfig(1M)	Determines which SDI disk, tape, and SCSI controllers are present
uname(1)	Prints current system parameters

A.9 Answers to Review Questions

1. Components of a typical system include: the video display screen, mouse, keyboard, hard disk drive, power supply, CD-ROM drive, cartridge tape drive, 3.5" floppy drive, parallel I/O adapter, serial I/O adapter, math coprocessor, CPU, and memory.
2. The motherboard contains the microprocessor and memory.
3. The PCI bus interface is commonly used for video and disk drive interconnections.
4. A SCSI host adapter controls access to SCSI peripherals, transfers data between peripherals and memory, and relays commands (or status information or messages).
5. Disk controllers supported by SCO include Integrated Drive Electronics (IDE), Enhanced IDE (EIDE), Enhanced Small Device Interface (ESDI), Compaq Intelligent Disk Array (IDA), and SCSI disk controllers.
6. Device interrupt conflicts are one of the most common causes of problems during installation; check and make sure each device on your system has its own IRQ setting.
7. Many answers are acceptable. Some more obvious choices include installed hardware and software, server repair history (both hardware and software), details of scheduled and unscheduled maintenance and backups, copies of important configuration files, problems and solutions, and a "to-do" list.

AT74202P001