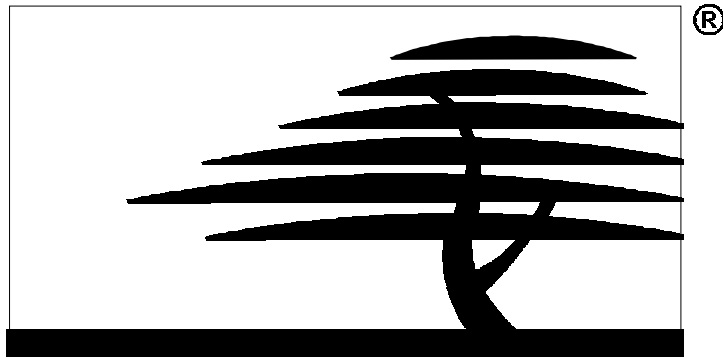




# SCO

**SCO OpenServer™  
Release 5 System  
Administration II:  
System Installation,  
Configuration, and  
Maintenance**

Student Guide

Course Release: 3.0.1

For Software: SCO OpenServer™ Release 5



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





The SCO<sup>®</sup> Authorized Education class you are about to take may form part of the recommended preparation for those seeking the SCO ACE<sup>™</sup> 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<sup>®</sup> systems, including those from other vendors.

### Important!

When registering for any SCO ACE exam at a Sylvan Prometric<sup>™</sup> 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

|                                                                  |            |
|------------------------------------------------------------------|------------|
| <b>Lesson 1: Course Introduction</b>                             | <b>1-1</b> |
| Introduction .....                                               | 1-3        |
| Course goal and purpose .....                                    | 1-3        |
| Target audience .....                                            | 1-4        |
| Student Materials .....                                          | 1-5        |
| The <i>Student Guide</i> .....                                   | 1-5        |
| SCO documentation .....                                          | 1-5        |
| Typographical Conventions .....                                  | 1-6        |
| Using scoadmin(ADM) .....                                        | 1-8        |
| Sample exam items .....                                          | 1-9        |
| <b>Lesson 2: Basic Hardware Configuration</b>                    | <b>2-1</b> |
| Hardware Overview .....                                          | 2-5        |
| Microprocessors and Memory .....                                 | 2-6        |
| Microprocessors .....                                            | 2-9        |
| Memory .....                                                     | 2-13       |
| Bus Basics .....                                                 | 2-17       |
| Bus architectures .....                                          | 2-17       |
| Adapter cards and controllers .....                              | 2-23       |
| Small Computer System Interface (SCSI) Host Adapters .....       | 2-29       |
| SCSI addresses .....                                             | 2-31       |
| Supported Hard Disk Configurations .....                         | 2-32       |
| General Configuration Information .....                          | 2-33       |
| General compatibility issues .....                               | 2-33       |
| General configuration parameters .....                           | 2-34       |
| Maintaining a Sample System Administration Log Book .....        | 2-35       |
| Lesson Summary .....                                             | 2-37       |
| Relevant Commands and Files .....                                | 2-39       |
| Answers to Review Questions .....                                | 2-40       |
| <b>Lesson 3: Files and Filesystems</b>                           | <b>3-1</b> |
| Describing the Directory Hierarchy .....                         | 3-5        |
| The <i>/usr</i> directory structure .....                        | 3-9        |
| The <i>/etc</i> directory structure .....                        | 3-9        |
| The <i>/tcb</i> directory structure .....                        | 3-9        |
| Describing the Role of the Software Storage Objects (SSOs) ..... | 3-10       |
| The SSO architecture .....                                       | 3-10       |
| Benefits of the SSO architecture .....                           | 3-11       |
| Filesystem Types .....                                           | 3-13       |
| HTFS filesystem .....                                            | 3-13       |
| DTFS filesystem .....                                            | 3-13       |

---

---

|                                                 |      |
|-------------------------------------------------|------|
| EAFS filesystem .....                           | 3-14 |
| Other supported filesystems.....                | 3-15 |
| Determining filesystem types .....              | 3-15 |
| Describing the UNIX Disk Structure.....         | 3-17 |
| Physical blocks .....                           | 3-17 |
| Partitions .....                                | 3-17 |
| Divisions .....                                 | 3-19 |
| Utilities for setting up hard disks.....        | 3-21 |
| Lesson Summary.....                             | 3-28 |
| Relevant Commands and Files .....               | 3-29 |
| Answers to Exercises and Review Questions ..... | 3-30 |

---

|                                        |            |
|----------------------------------------|------------|
| <b>Lesson 4: Installation Planning</b> | <b>4-1</b> |
|----------------------------------------|------------|

---

|                                                                 |      |
|-----------------------------------------------------------------|------|
| Identifying OpenServer Products.....                            | 4-4  |
| Identifying OpenServer Hardware Requirements.....               | 4-6  |
| Disk space requirements .....                                   | 4-6  |
| Installation drive requirements .....                           | 4-7  |
| Using SCO documentation to identify hardware requirements ..... | 4-7  |
| Completing an Installation Checklist.....                       | 4-10 |
| Installing Other Partitions.....                                | 4-10 |
| Boot-time loadable drivers (BTLDs).....                         | 4-11 |
| Installation media.....                                         | 4-12 |
| The keyboard .....                                              | 4-13 |
| Licensing your system .....                                     | 4-13 |
| Installation type.....                                          | 4-13 |
| Configuring the basic system.....                               | 4-14 |
| Hard Disk Setup.....                                            | 4-16 |
| Bad track/block .....                                           | 4-17 |
| Software components.....                                        | 4-17 |
| TCP/IP network configuration parameters .....                   | 4-17 |
| IPX/SPX network configuration parameters .....                  | 4-19 |
| Graphics setup.....                                             | 4-19 |
| Mouse.....                                                      | 4-21 |
| E-mail system .....                                             | 4-21 |
| Lesson Summary.....                                             | 4-22 |
| Relevant Commands and Options.....                              | 4-24 |
| Answers to Review Questions .....                               | 4-25 |

---

|                                                |            |
|------------------------------------------------|------------|
| <b>Lesson 5: Operating System Installation</b> | <b>5-1</b> |
|------------------------------------------------|------------|

---

|                                                   |      |
|---------------------------------------------------|------|
| Stopping and Restarting the Installation.....     | 5-4  |
| Stopping and restarting the installation .....    | 5-4  |
| Installing SCO OpenServer Release 5 .....         | 5-5  |
| Troubleshooting Problems During Installation..... | 5-31 |
| System does not boot during installation.....     | 5-31 |
| System cannot find the hard disk .....            | 5-32 |

---



---

|                                                            |      |
|------------------------------------------------------------|------|
| System cannot find CD-ROM.....                             | 5-32 |
| Screen blanks after <b>BooT</b> : prompt .....             | 5-33 |
| System hangs at hardware configuration screen.....         | 5-33 |
| Troubleshooting Problems After Installation.....           | 5-35 |
| System does not boot .....                                 | 5-35 |
| The mouse is not working.....                              | 5-35 |
| Graphical <b>scolgin</b> (XC) on tty02 looks strange ..... | 5-36 |
| Error message: Can't fork.....                             | 5-37 |
| System not licensed or unregistered software.....          | 5-37 |
| Hardware does not work .....                               | 5-37 |
| Problems with broken links .....                           | 5-37 |
| Lesson Summary.....                                        | 5-40 |
| Relevant Commands and Files .....                          | 5-42 |
| Answers to Review Questions .....                          | 5-43 |

---

|                                                   |            |
|---------------------------------------------------|------------|
| <b>Lesson 6: Post-installation Administration</b> | <b>6-1</b> |
|---------------------------------------------------|------------|

---

|                                                         |      |
|---------------------------------------------------------|------|
| Creating an Emergency Boot Floppy Set.....              | 6-5  |
| Requirements for creating a boot/root floppy set .....  | 6-7  |
| Using the Floppy Filesystem Manager .....               | 6-9  |
| Installing Applications .....                           | 6-13 |
| Installing the online documentation.....                | 6-13 |
| Installing SCO Merge .....                              | 6-14 |
| Licensing and Registering Your OpenServer Products..... | 6-17 |
| Licensing the software .....                            | 6-19 |
| Registering the software .....                          | 6-19 |
| Registration reminder messages .....                    | 6-21 |
| Copy Protection Policy .....                            | 6-21 |
| Copy Protection Policy daemons.....                     | 6-22 |
| Lesson Summary.....                                     | 6-23 |
| Relevant Commands and Daemons .....                     | 6-24 |
| Relevant commands.....                                  | 6-24 |
| Relevant daemons .....                                  | 6-24 |
| Answers to Exercises and Review Questions .....         | 6-25 |

---

|                                                       |            |
|-------------------------------------------------------|------------|
| <b>Lesson 7: Managing System Startup and Shutdown</b> | <b>7-1</b> |
|-------------------------------------------------------|------------|

---

|                                                  |      |
|--------------------------------------------------|------|
| Describing the Default Boot Process .....        | 7-5  |
| Standard boot procedure .....                    | 7-5  |
| Responding to the UNIX boot prompt.....          | 7-7  |
| Loading the UNIX kernel .....                    | 7-8  |
| Loading other operating systems .....            | 7-11 |
| Customizing the Default Boot Process .....       | 7-12 |
| Changing the <i>/etc/default/boot</i> file ..... | 7-13 |
| Keywords recognized by <b>init</b> (M) .....     | 7-17 |
| Bootstrings parameters .....                     | 7-19 |
| Hardware Information Displayed at Boot.....      | 7-23 |

---

---

|                                                                     |      |
|---------------------------------------------------------------------|------|
| Sample boot display .....                                           | 7-25 |
| Recognized Devices.....                                             | 7-25 |
| Other pertinent information .....                                   | 7-26 |
| Displaying current hardware configurations.....                     | 7-26 |
| Logging system messages.....                                        | 7-27 |
| Choosing the Mode of Operation.....                                 | 7-29 |
| Following the boot process through related files.....               | 7-29 |
| The <i>/etc/inittab</i> fields .....                                | 7-31 |
| Changing the <i>/etc/inittab</i> file .....                         | 7-37 |
| Tracing Multiuser Initialization .....                              | 7-38 |
| The <i>/etc/rc2</i> file and <i>/etc/rc2.d</i> directory.....       | 7-38 |
| Analyzing <i>/etc/rc2.d</i> scripts .....                           | 7-41 |
| Ensuring backwards compatibility.....                               | 7-42 |
| Customizing Multiuser Initialization .....                          | 7-43 |
| How scripts are executed from <i>/etc/rc2.d</i> .....               | 7-43 |
| Executing commands at system startup .....                          | 7-44 |
| Defining other run-level initialization scripts .....               | 7-44 |
| Deactivating rc scripts .....                                       | 7-45 |
| Locating messages encountered during run-level initialization ..... | 7-45 |
| Troubleshooting System Startup.....                                 | 7-46 |
| Restoring missing or corrupted boot files.....                      | 7-46 |
| Shutting Down the System .....                                      | 7-49 |
| Analyzing shutdown .....                                            | 7-49 |
| Lesson Summary.....                                                 | 7-50 |
| Lab: Creating an rc Script .....                                    | 7-52 |
| Overview.....                                                       | 7-52 |
| Procedure .....                                                     | 7-52 |
| Relevant Commands and Files .....                                   | 7-53 |
| Answers to Exercises and Review Questions .....                     | 7-54 |

---

|                                                               |            |
|---------------------------------------------------------------|------------|
| <b>Lesson 8: Filesystem Configuration and Disk Management</b> | <b>8-1</b> |
|---------------------------------------------------------------|------------|

---

|                                                 |      |
|-------------------------------------------------|------|
| Describing the UNIX Filesystem Structure .....  | 8-5  |
| The Superblock .....                            | 8-5  |
| Inodes.....                                     | 8-7  |
| Data blocks .....                               | 8-9  |
| Managing the <i>/dev</i> Directory .....        | 8-11 |
| Block files .....                               | 8-13 |
| Character files .....                           | 8-13 |
| Raw interfaces.....                             | 8-13 |
| Major and minor device numbers .....            | 8-15 |
| Creating special device files .....             | 8-16 |
| Alternate names for special device files .....  | 8-16 |
| Troubleshooting the <i>/dev</i> directory ..... | 8-17 |
| Describing Important Filesystem Features .....  | 8-18 |
| Configuration issues .....                      | 8-18 |

---

---

|                                                       |      |
|-------------------------------------------------------|------|
| Configuring Filesystems .....                         | 8-23 |
| Dividing the partition.....                           | 8-23 |
| Readying the filesystem for access.....               | 8-24 |
| The <i>/etc/default/filesys</i> file.....             | 8-27 |
| Checking and Repairing Filesystems.....               | 8-29 |
| Filesystem check phases .....                         | 8-31 |
| Checking the DTFS filesystem .....                    | 8-32 |
| Running <b>fsck</b> (ADM).....                        | 8-32 |
| Error messages .....                                  | 8-33 |
| Running <b>ncheck</b> (ADM).....                      | 8-34 |
| Configuring Additional Disks.....                     | 8-35 |
| Possible configurations for secondary hard disks..... | 8-37 |
| Device drivers .....                                  | 8-37 |
| Configuring IDE disks and disk controllers .....      | 8-38 |
| Using <b>mkdev</b> (ADM) to configure disks.....      | 8-38 |
| Commands invoked by <b>mkdev</b> (ADM).....           | 8-41 |
| Completing the installation.....                      | 8-41 |
| Troubleshooting Hard Disks .....                      | 8-42 |
| The <b>badtrk</b> (ADM) command.....                  | 8-42 |
| Lesson Summary.....                                   | 8-44 |
| Lab: Configuring a Hard Disk (Non-SCSI).....          | 8-46 |
| Overview.....                                         | 8-46 |
| Procedure .....                                       | 8-46 |
| Lab: Configuring a Hard Disk (SCSI) .....             | 8-48 |
| Overview.....                                         | 8-48 |
| Procedure .....                                       | 8-48 |
| Relevant Commands and Files .....                     | 8-50 |
| Answers to Exercises and Review Questions .....       | 8-52 |

---

## Lesson 9: Installing and Managing Serial Ports

**9-1**

---

|                                           |      |
|-------------------------------------------|------|
| Standard Serial Ports.....                | 9-4  |
| Using multiscreens.....                   | 9-4  |
| Enabling and disabling serial ports ..... | 9-5  |
| Hardware Communication .....              | 9-6  |
| Line characteristics .....                | 9-6  |
| Escape codes .....                        | 9-7  |
| Examining the UNIX Login Process .....    | 9-9  |
| The <i>/etc/inittab</i> file.....         | 9-9  |
| The <b>getty</b> (M) program.....         | 9-9  |
| The <i>gettydefs</i> file.....            | 9-11 |
| The <b>login</b> (M) process.....         | 9-12 |
| Shell programs .....                      | 9-15 |
| Installing a Serial Adapter Card.....     | 9-17 |
| Serial adapter cards .....                | 9-17 |
| Installing the hardware.....              | 9-18 |

---

---

|                                                   |      |
|---------------------------------------------------|------|
| Configuring the software .....                    | 9-19 |
| Configuring a dumb card .....                     | 9-23 |
| Configuring the serial port .....                 | 9-25 |
| Configuring a smart card .....                    | 9-26 |
| Troubleshooting serial card installations .....   | 9-26 |
| Installing Serial Terminal Hardware .....         | 9-27 |
| About serial terminal hardware .....              | 9-27 |
| Ports and cables .....                            | 9-29 |
| Terminal settings .....                           | 9-31 |
| Managing Serial Ports and Terminals .....         | 9-33 |
| Changing the <i>/etc/inittab</i> file .....       | 9-33 |
| Changing the <i>/etc/gettydefs</i> file .....     | 9-35 |
| The <b>stty</b> (C) command .....                 | 9-35 |
| Setting the Terminal Type .....                   | 9-39 |
| Troubleshooting Terminals .....                   | 9-41 |
| Troubleshooting non-functional terminals .....    | 9-41 |
| Troubleshooting hung or scrambled terminals ..... | 9-45 |
| Lesson Summary .....                              | 9-46 |
| Lab: Rebuilding <i>/etc/inittab</i> .....         | 9-48 |
| Overview .....                                    | 9-48 |
| Procedure .....                                   | 9-49 |
| Relevant Commands and Files .....                 | 9-50 |
| Answers to Exercises and Review Questions .....   | 9-52 |

---

|                                       |             |
|---------------------------------------|-------------|
| <b>Lesson 10: Configuring a Modem</b> | <b>10-1</b> |
|---------------------------------------|-------------|

---

|                                   |       |
|-----------------------------------|-------|
| Adding a Modem .....              | 10-4  |
| Choosing a serial port .....      | 10-5  |
| Configuring a modem .....         | 10-7  |
| Testing the modem .....           | 10-12 |
| Lesson Summary .....              | 10-13 |
| Relevant Commands and Files ..... | 10-14 |
| Answers to Review Questions ..... | 10-15 |

---

|                                                    |             |
|----------------------------------------------------|-------------|
| <b>Lesson 11: Installing and Managing Printers</b> | <b>11-1</b> |
|----------------------------------------------------|-------------|

---

|                                                                           |       |
|---------------------------------------------------------------------------|-------|
| Installing a Parallel Adapter Card .....                                  | 11-4  |
| Installing the hardware .....                                             | 11-4  |
| Configuring the software .....                                            | 11-5  |
| Troubleshooting parallel port installation .....                          | 11-6  |
| Installing Parallel and Serial Printers .....                             | 11-7  |
| Verifying the hardware installation of a parallel or serial printer ..... | 11-8  |
| Setting up RTS/CTS protocol serial printers .....                         | 11-11 |
| Troubleshooting Printer Installations .....                               | 11-12 |
| Adding Printers .....                                                     | 11-14 |
| Adding a local printer .....                                              | 11-14 |
| Adding a remote printer .....                                             | 11-15 |

---

---

|                                                               |       |
|---------------------------------------------------------------|-------|
| Describing the UNIX Print Service .....                       | 11-16 |
| Who can administer the print service .....                    | 11-16 |
| Print scheduler functions .....                               | 11-17 |
| Print request process .....                                   | 11-19 |
| Viewing the print scheduler .....                             | 11-21 |
| Managing the Print Request Directories .....                  | 11-23 |
| Examining the print request log.....                          | 11-25 |
| Clearing the print request directories.....                   | 11-26 |
| Setting Print Request Priorities.....                         | 11-27 |
| Default priority level and limits.....                        | 11-27 |
| Changing the default priority limit for all users .....       | 11-27 |
| Setting a personal priority limit for a user .....            | 11-28 |
| Restoring the default priority limit .....                    | 11-28 |
| Submitting a print job with a different priority.....         | 11-28 |
| Changing the priority of print job residing in the queue..... | 11-29 |
| Bypassing all other awaiting print requests .....             | 11-29 |
| Managing Printer Interface Scripts .....                      | 11-31 |
| Assigning an interface script.....                            | 11-31 |
| Customizing the standard interface script.....                | 11-32 |
| Troubleshooting Common Printer Problems .....                 | 11-34 |
| Printer does not print.....                                   | 11-34 |
| Restarting a printer.....                                     | 11-35 |
| Laser printers and form feeds .....                           | 11-35 |
| Testing printers in PostScript mode .....                     | 11-35 |
| Determining if the print scheduler is active.....             | 11-35 |
| Redirecting output.....                                       | 11-36 |
| Printer output spacing is wrong .....                         | 11-37 |
| Lesson Summary.....                                           | 11-38 |
| Review questions .....                                        | 11-38 |
| Lab: Printer Troubleshooting.....                             | 11-40 |
| Overview.....                                                 | 11-40 |
| Relevant Commands and Files .....                             | 11-41 |
| Answers to Exercises and Review Questions .....               | 11-43 |

---

|                                                            |             |
|------------------------------------------------------------|-------------|
| <b>Lesson 12: Installing Tape Drives and CD-ROM Drives</b> | <b>12-1</b> |
|------------------------------------------------------------|-------------|

---

|                                         |       |
|-----------------------------------------|-------|
| Installing a Tape Drive.....            | 12-4  |
| Hardware settings .....                 | 12-4  |
| Configuring the software .....          | 12-5  |
| Tape device files .....                 | 12-9  |
| Troubleshooting .....                   | 12-9  |
| Installing a CD-ROM Drive .....         | 12-10 |
| Installing the hardware.....            | 12-10 |
| Configuring an Enhanced IDE CD-ROM..... | 12-11 |
| Configuring a SCSI CD-ROM.....          | 12-12 |
| Using the CD-ROM .....                  | 12-13 |

---

---

|                                                             |                 |
|-------------------------------------------------------------|-----------------|
| Troubleshooting .....                                       | 12-13           |
| Lesson Summary .....                                        | 12-14           |
| Relevant Commands and Files .....                           | 12-15           |
| Relevant commands .....                                     | 12-15           |
| Relevant files .....                                        | 12-15           |
| Answers to Exercises and Review Questions .....             | 12-16           |
| <br><b>Lesson 13: Controlling User and System Processes</b> | <br><b>13-1</b> |
| <hr/>                                                       |                 |
| Creating Processes .....                                    | 13-5            |
| Distinguishing Types of Processes.....                      | 13-7            |
| System processes .....                                      | 13-7            |
| User Processes .....                                        | 13-9            |
| Checking Process Activity .....                             | 13-10           |
| The <b>ps</b> (C) command .....                             | 13-10           |
| The <b>sar</b> (ADM) command .....                          | 13-15           |
| The <b>timex</b> (ADM) command.....                         | 13-19           |
| Prioritizing Processes.....                                 | 13-21           |
| Priority ranges .....                                       | 13-23           |
| Calculating process priority .....                          | 13-23           |
| Controlling the <b>nice</b> value.....                      | 13-24           |
| Managing Processes with the Process Manager .....           | 13-25           |
| Managing Process Accounting .....                           | 13-26           |
| Enabling process accounting .....                           | 13-26           |
| Disabling process accounting .....                          | 13-27           |
| Reading process accounting files.....                       | 13-27           |
| Generating process accounting reports .....                 | 13-28           |
| Clearing process accounting files .....                     | 13-28           |
| Automating Processes Using <b>cron</b> (C) .....            | 13-29           |
| System <b>crontabs</b> .....                                | 13-31           |
| Editing system <b>crontabs</b> .....                        | 13-33           |
| Lesson Summary .....                                        | 13-34           |
| Lab: Changing the root <b>crontab</b> .....                 | 13-35           |
| Overview .....                                              | 13-35           |
| Procedure .....                                             | 13-35           |
| Relevant Commands and Files .....                           | 13-36           |
| Relevant commands .....                                     | 13-36           |
| Relevant files .....                                        | 13-36           |
| Answers to Exercises and Review Questions .....             | 13-37           |
| <br><b>Lesson 14: Managing System Security</b>              | <br><b>14-1</b> |
| <hr/>                                                       |                 |
| Describing Trusted System Concepts .....                    | 14-5            |
| Describing a Security Profile .....                         | 14-7            |
| The high security profile.....                              | 14-7            |
| The improved security profile.....                          | 14-7            |
| The traditional security profile .....                      | 14-9            |

---

---

|                                                               |       |
|---------------------------------------------------------------|-------|
| The low security profile .....                                | 14-9  |
| Viewing and Changing your System Security Profile .....       | 14-10 |
| Determining your security profile.....                        | 14-11 |
| Selecting a security profile .....                            | 14-13 |
| Changing a parameter in your security profile .....           | 14-14 |
| Changing a parameter for an individual user.....              | 14-17 |
| Reviewing the Account Database Files .....                    | 14-21 |
| System V files.....                                           | 14-21 |
| Trusted Computing Base Files.....                             | 14-22 |
| Delegating System Administration Responsibility .....         | 14-23 |
| Primary subsystem authorizations .....                        | 14-25 |
| Secondary subsystem authorizations .....                      | 14-29 |
| Using authorizations .....                                    | 14-31 |
| User system privileges .....                                  | 14-31 |
| Subsystem privilege requirements .....                        | 14-33 |
| Allowing Users to Execute Root Commands .....                 | 14-35 |
| Configuring <b>asroot</b> (ADM) .....                         | 14-37 |
| Executing the <b>root</b> command .....                       | 14-38 |
| Advanced Protection Bits .....                                | 14-39 |
| Examining SUID/SGID bits .....                                | 14-41 |
| Setting the sticky bit .....                                  | 14-41 |
| Managing the Trusted Computing Base.....                      | 14-42 |
| Authentication database .....                                 | 14-42 |
| Other database files.....                                     | 14-42 |
| Maintaining the integrity of the Trusted Computing Base ..... | 14-42 |
| Repairing the Trusted Computing Base database .....           | 14-44 |
| Additional Security Measures.....                             | 14-47 |
| Restricting root logins.....                                  | 14-47 |
| Monitoring the use of the <b>su</b> (C) command.....          | 14-49 |
| Additional dial-in security .....                             | 14-50 |
| Lesson Summary .....                                          | 14-51 |
| Lab: Configuring and Using <b>asroot</b> (ADM).....           | 14-53 |
| Overview.....                                                 | 14-53 |
| Procedure .....                                               | 14-53 |
| Relevant Commands and Files .....                             | 14-54 |
| Answers to Exercises and Review Questions .....               | 14-55 |

---

## Lesson 15: Performance Tuning

15-1

---

|                                     |      |
|-------------------------------------|------|
| Influencing System Performance..... | 15-4 |
| Hardware factors .....              | 15-4 |
| Software factors .....              | 15-5 |
| System organization.....            | 15-5 |
| Defining Performance Goals .....    | 15-6 |
| Defining your goals .....           | 15-6 |
| Listing your goals .....            | 15-6 |

---

---

|                                                 |       |
|-------------------------------------------------|-------|
| Understanding your goals .....                  | 15-6  |
| Collecting Performance Data.....                | 15-7  |
| Identifying CPU saturation .....                | 15-9  |
| Tuning CPU resources .....                      | 15-12 |
| Identifying I/O saturation.....                 | 15-13 |
| The buffer cache .....                          | 15-13 |
| Tuning I/O performance.....                     | 15-17 |
| Identifying memory saturation.....              | 15-19 |
| Tuning memory saturation.....                   | 15-23 |
| Identifying Kernel Tables .....                 | 15-24 |
| Setting table limits .....                      | 15-25 |
| Determining table memory size .....             | 15-27 |
| Dynamic kernel tables .....                     | 15-27 |
| Interpreting Kernel Error Messages.....         | 15-29 |
| Configuring Kernel Parameters .....             | 15-31 |
| Configuration tools .....                       | 15-33 |
| Using <b>configure</b> (ADM).....               | 15-35 |
| Lesson Summary.....                             | 15-37 |
| Lab: Reconfiguring the kernel .....             | 15-38 |
| Overview.....                                   | 15-38 |
| Procedure .....                                 | 15-39 |
| Relevant Commands and Files .....               | 15-41 |
| Answers to Exercises and Review Questions ..... | 15-42 |

---

## Lesson 16: Disaster Recovery

**16-1**

---

|                                                             |       |
|-------------------------------------------------------------|-------|
| Preparing for Recovery .....                                | 16-4  |
| Recovering a System File .....                              | 16-5  |
| Product packaging.....                                      | 16-5  |
| Extracting a system file.....                               | 16-5  |
| Restoring a Non-root Filesystem .....                       | 16-7  |
| Running <b>fsck</b> (ADM).....                              | 16-7  |
| Recreating a non-root filesystem .....                      | 16-7  |
| Procedure for restoring a non-root filesystem .....         | 16-8  |
| Restoring the root and boot filesystems.....                | 16-9  |
| Recovery components.....                                    | 16-9  |
| Reconfiguring the root hard disk .....                      | 16-9  |
| Procedure for restoring the root and boot filesystems ..... | 16-10 |
| Replacing the Primary Root Hard Disk .....                  | 16-13 |
| Recovery components.....                                    | 16-13 |
| Partial installation .....                                  | 16-13 |
| Lesson Summary.....                                         | 16-15 |
| Lab: Recovering your root and boot filesystems .....        | 16-16 |
| Overview.....                                               | 16-16 |
| Procedure .....                                             | 16-16 |
| Relevant Commands and Files .....                           | 16-19 |
| Answers to Exercises and Review Questions .....             | 16-20 |

---



## ***Course Introduction***



**Lesson 1**

## **Course Introduction**

### **1.1 Introduction**

---

The ***SCO OpenServer Release 5 Administration II: System Installation, Configuration, and Maintenance*** course is designed for technically-oriented individuals who have a solid understanding of UNIX system administration basics and are required to install and configure SCO OpenServer 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 SCO OpenServer software and troubleshoot the installation
- Perform post-installation tasks, such as licensing their software and creating an emergency boot floppy set
- 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
- Configure modems

- Install tape drives and CD-ROM drives
- Control user and system processes
- Manage system security
- Tune the system for optimum performance
- Prepare for 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 an SCO OpenServer 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 ***SCO OpenServer Release 5 Administration I: User Services*** course.

## 1.2 Student Materials

---

The contents of the box containing your student materials are yours to keep and write notes in. Your instructor might 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 are needed again for later classes.

### 1.2.1 The *Student Guide*

---

The *Student Guide* contains:

- Outline of presentation
- Note-taking space

Each student kit also contains one copy of the *SCO OpenServer Handbook and SCO OpenServer Release Notes*

### 1.2.2 SCO documentation

---

This course refers to the following SCO printed documentation:

|                                     |                                                                                                                                                                                                                                             |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>SCO OpenServer Release Notes</i> | Contains important late-breaking information about installations, hardware requirements, and known limitations.                                                                                                                             |
| <i>SCO OpenServer Handbook</i>      | Provides the information needed to get your SCO OpenServer Release 5 system up and running. It also covers many common tasks, such as managing users, starting and stopping the system, working with filesystems, and configuring printers. |

This course also refers to the online documentation using **scohelp(XC)**.

## 1.3 Typographical Conventions

---

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

|                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>lp</b> or <b>lp(C)</b>                                                                                                                                                                                                                                   | 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)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>root</b>                                                                                                                                                                                                                                                 | System or user names                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <i>arg</i> or <i>arg</i>                                                                                                                                                                                                                                    | Placeholders (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 <b>root</b> 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)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

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

## 1.4 Using scoadmin(ADM)

---

Many procedures in this course require you to start a **SCOadmin** manager such as the **Filesystem Manager**.

SCO OpenServer Release 5 includes both graphical and character-based versions of most administration utilities. When started from the Desktop, the graphical versions automatically run. When started from a character device (such as a terminal), the character versions run.

In this course, the Desktop is used to start the graphical versions of administration utilities. There are several ways to run **scoadmin**(ADM) utilities.

For example, to run the **Filesystem Manager**, a tool that allows you to manage filesystems:

- From the **System Administration** folder, select ⇒ **Filesystems** ⇒ **Filesystem Manager**.
- From a UNIX window, enter **scoadmin**. Select ⇒ **Filesystems** ⇒ **Filesystem Manager**.
- From a UNIX window, enter **scoadmin filesystem**.

Be sure that you are logged in as **root** when invoking these utilities.



## 1.5 Sample exam items

---

The following 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 EAFS filesystem size?
  - A. 2GB
  - B. 256GB
  - B. 512GB
  - C. 1TB
2. Which security profile meets only the minimum requirements for the C2 level of trust?
  - A. High
  - B. Improved
  - C. Traditional
  - D. Low
3. Which file contains a log of device initialization messages:
  - A. */usr/adm/syslog*
  - B. */usr/adm/messages*
  - C. */usr/adm/hwconfig*
  - D. */usr/adm/install.log*

4. Which file contains a list of the filesystems to be checked when you invoke **fsck**(ADM) without arguments?
  - A. `/etc/bcheckrc`
  - B. `/etc/checklist`
  - C. `/etc/ncheck`
  - D. `/etc/default/checklist`
5. During operating system installation, which hard disk setup option is available only if your system has an active UNIX partition with a clean root filesystem and a valid `/etc/default/filesys` file?
  - A. Preserve
  - B. Use whole disk for UNIX
  - C. Customize
  - D. Interactive fdisk/divvy

### Answers to sample exam items

---

1. A
2. B
3. C
4. B
5. A

---

# 2

## *Basic Hardware Configuration*

AV 2-1

**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

sa20201 © The Santa Cruz Operation, Inc.

**Lesson 2**

## **Basic Hardware Configuration**

### **Introduction**

---

Before, during, and after installing SCO OpenServer Release 5, you will answer questions about your hardware configuration. For example, you will complete an installation checklist as part of your pre-installation analysis and configure a hard disk during installation. You will also 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, discusses hardware and architectures supported by SCO, and makes your installation and post-installation tasks easier.

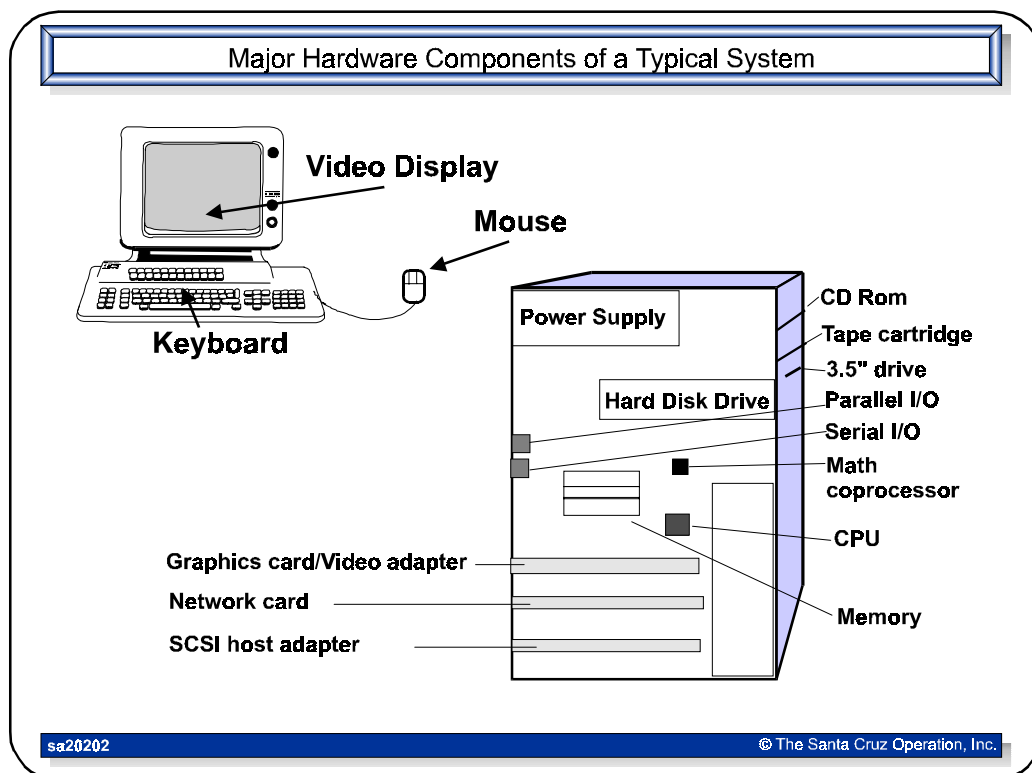
### **Objectives**

---

This lesson describes the basic, or major, hardware components of a typical system configuration. 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 2-2



## **2.1 Hardware Overview**

---

Although you probably support different types of hardware, most systems typically include some or all of the 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
- Parallel I/O
- Serial I/O
- Math coprocessor
- Central Processing Unit (CPU)
- Memory
- Graphics card/video adapter
- Network card
- SCSI host adapter

## 2.2 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

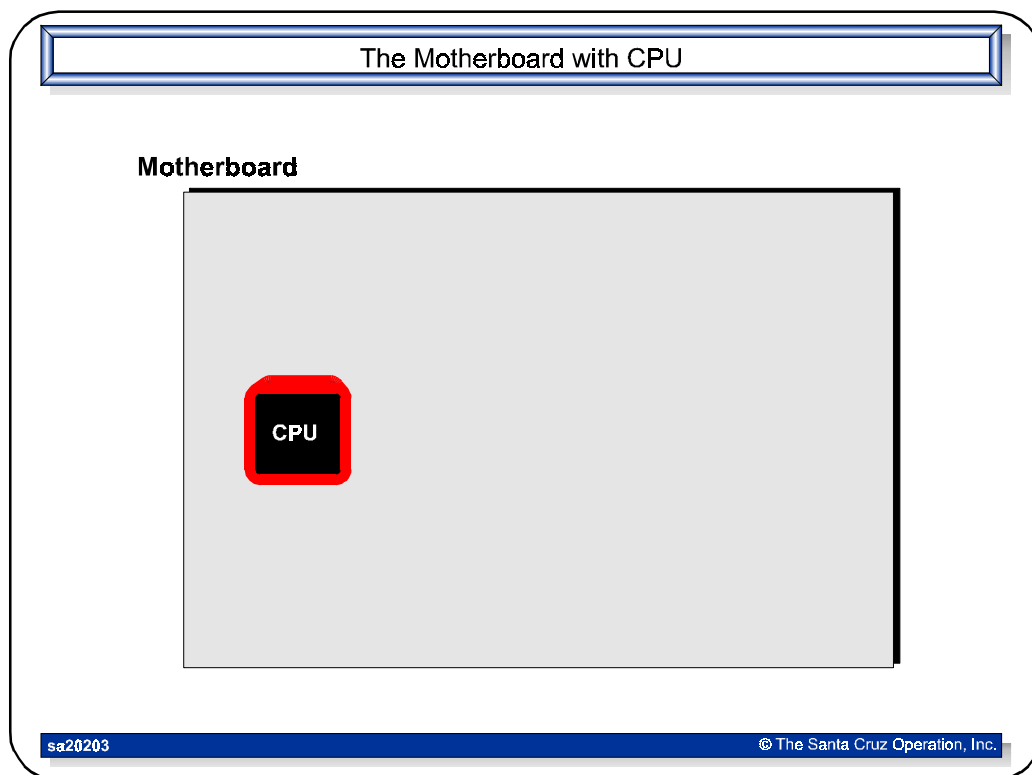
In systems with the *single board* design, all of these vital components are located on one board called the *motherboard*. In modular or bus-oriented systems, the motherboard is divided into a separate processor board and a bus board that contains the slots into which adapter cards are inserted.



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                                                                                                                                                                                                                      |
| Coprocessor             | Performs high speed math calculations with high accuracy                                                                                                                                                                                                                                           |
| BIOS                    | Provides the system with fundamental operational characteristics and instructions when the system is powered up                                                                                                                                                                                    |
| Expansion slots         | Enables 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.” |

AV 2-3



## 2.2.1 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<sup>®</sup> 80x86 family (where x= 3, 4, and so forth)
- The 68000 Motorola<sup>™</sup> series

Intel, credited with the invention of the microprocessor, is currently the largest manufacturer of microprocessors. This section briefly describes the following Intel microprocessors:

- i486
- Pentium<sup>®</sup>
- Pentium Pro<sup>™</sup>
- Pentium<sup>®</sup> II
- Pentium<sup>®</sup> II Xeon<sup>™</sup>

### 2.2.1.1 The i486 microprocessor

---

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

### 2.2.1.2 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

### 2.2.1.3 The Pentium Pro

---

The Pentium Pro processor family is Intel's next generation of processors for high-end desktops, workstations, and servers. 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

#### **2.2.1.4 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.                            |

#### **2.2.1.5 The Pentium II Xeon**

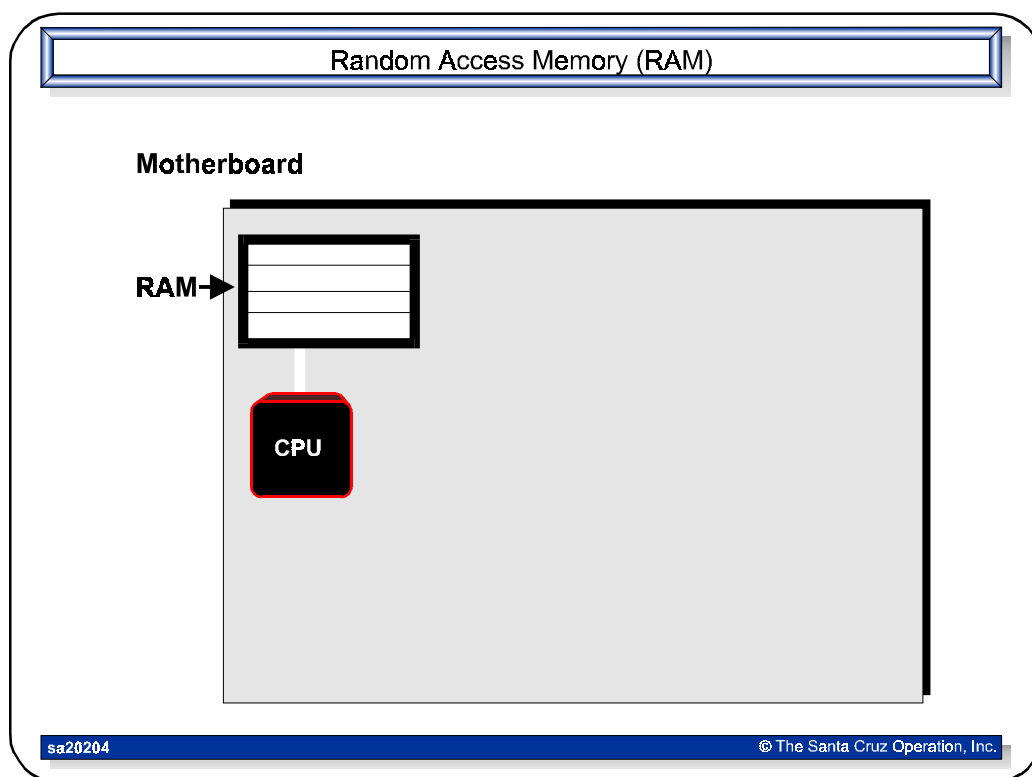
---

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

On June 29, 1998, SCO announced it has tested and certified SCO OpenServer Release 5 to run on the new Intel Pentium II Xeon processor for servers.

AV 2-4



## **2.2.2 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.”

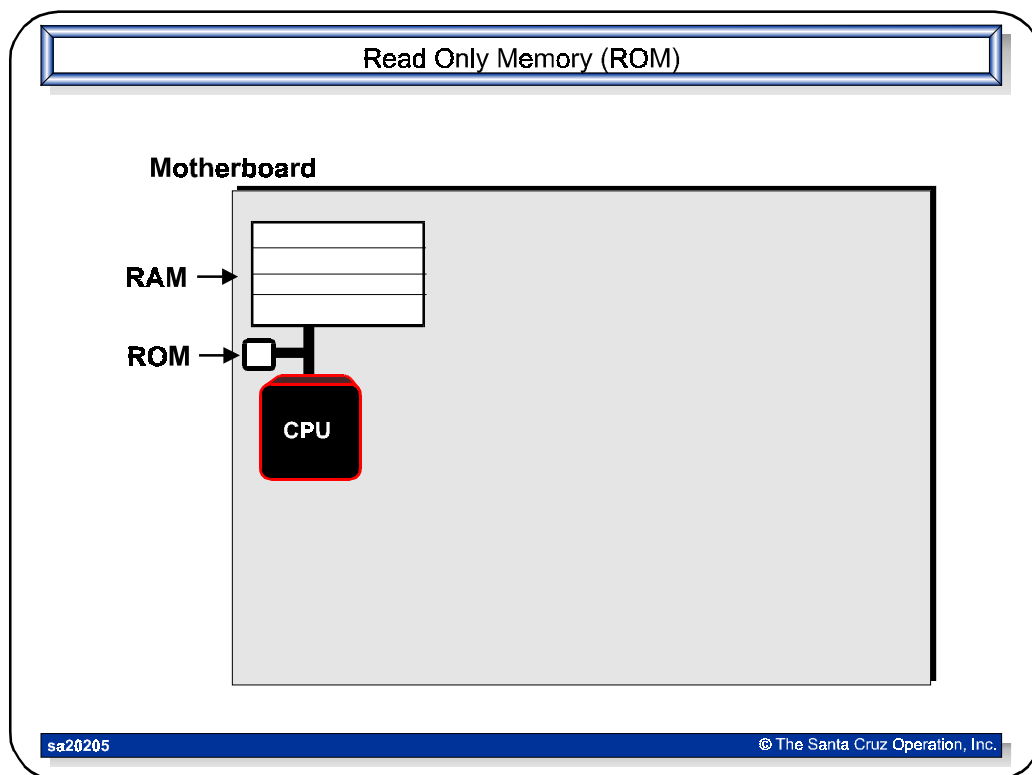
### **2.2.2.1 Random Access Memory (RAM)**

---

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 2-5





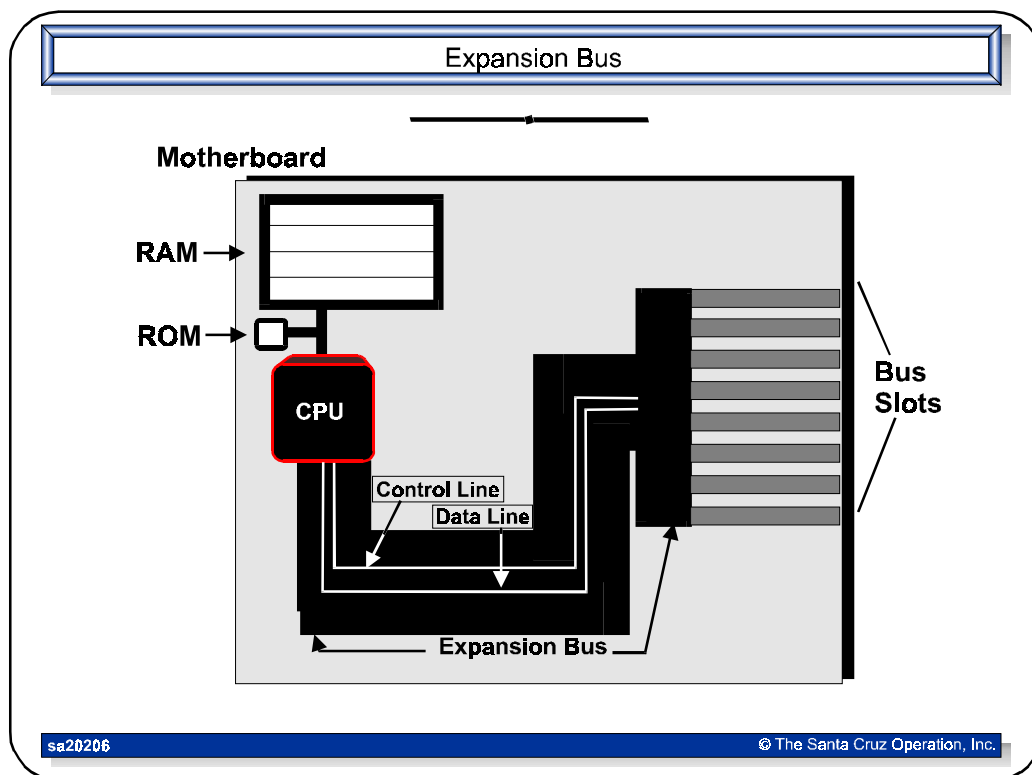
### **2.2.2.2 Read-Only Memory (ROM)**

---

ROM is a preprogrammed 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 2-6



## 2.3 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    | Carries data between the CPU and the controllers plugged into the bus slots |
| Control lines | Carries 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.

### 2.3.1 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)
- Video Electronics Standard Association (VESA) Local Bus
- Peripheral Component Interconnect (PCI)
- Plug and Play (PnP)

### 2.3.1.1 Industry Standard Architecture (ISA)

---

The ISA bus is sometimes called the *classic* bus because it defines standards that replaced the specifications for the Advanced Technology, or AT<sup>®</sup>, bus. The AT bus is the 16-bit successor to older IBM<sup>®</sup> 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

### 2.3.1.2 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

### 2.3.1.3 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

### 2.3.1.4 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:

- File formats
- Data structures
- Device-independent means of accessing card hardware
- Software links independent of operating systems

### 2.3.1.5 Video Electronics Standard Association (VESA) Local Bus

---

The VESA local bus, or VLB, was developed in early 1992 to provide a unified design for *local bus* expansion. A local bus is an alternative to the standard computer bus connections that provides a direct path from peripheral devices, such as video displays and hard disks, to the CPU.

Features of the VLB include the following:

- High-speed operation
- 32-bit design (upgradeable to 64-bit)
- Local bus connector specifications that allow for interchangeable display adapters

### 2.3.1.6 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.

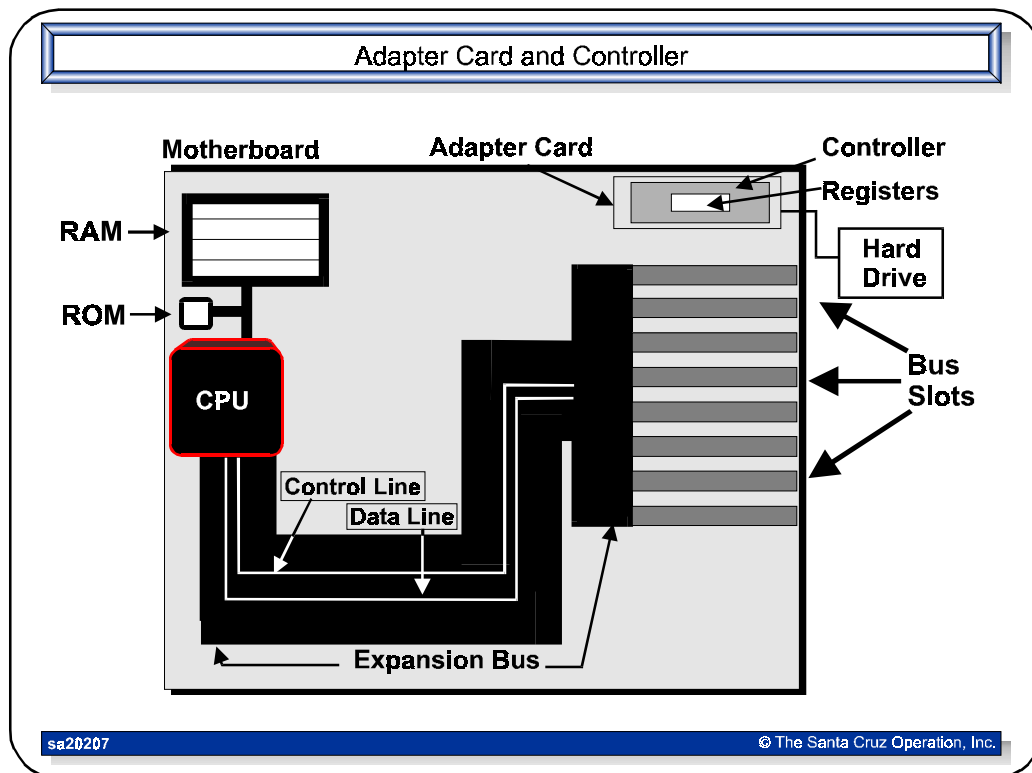
Like the VLB, the PCI bus is a *local bus*.

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, PCI 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

AV 2-7





## 2.3.2 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 adding adapter cards to your system, consider:

- 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 might need to run the manufacturer's setup program to change the system's configuration before you can use the new hardware.
- For Plug and Play ISA, using the features specified, individual expansion boards can tell the system their needs and what substitutions they accept. Then, your system can make the necessary arrangements.
- 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.

### 2.3.2.1 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<sup>®</sup> 3c501 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

### 2.3.2.2 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).

### **2.3.2.3 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 2-8

## 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.

sa20208

© The Santa Cruz Operation, Inc.

### 2.3.2.4 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.

### 2.3.2.5 DMA channels

---

After the CPU knows that the controller has data, the data is transferred to RAM by one of these 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 using a local bus (for example, PCI or VESA) without CPU intervention |

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.

## 2.4 Small Computer System Interface (SCSI) Host Adapters

---

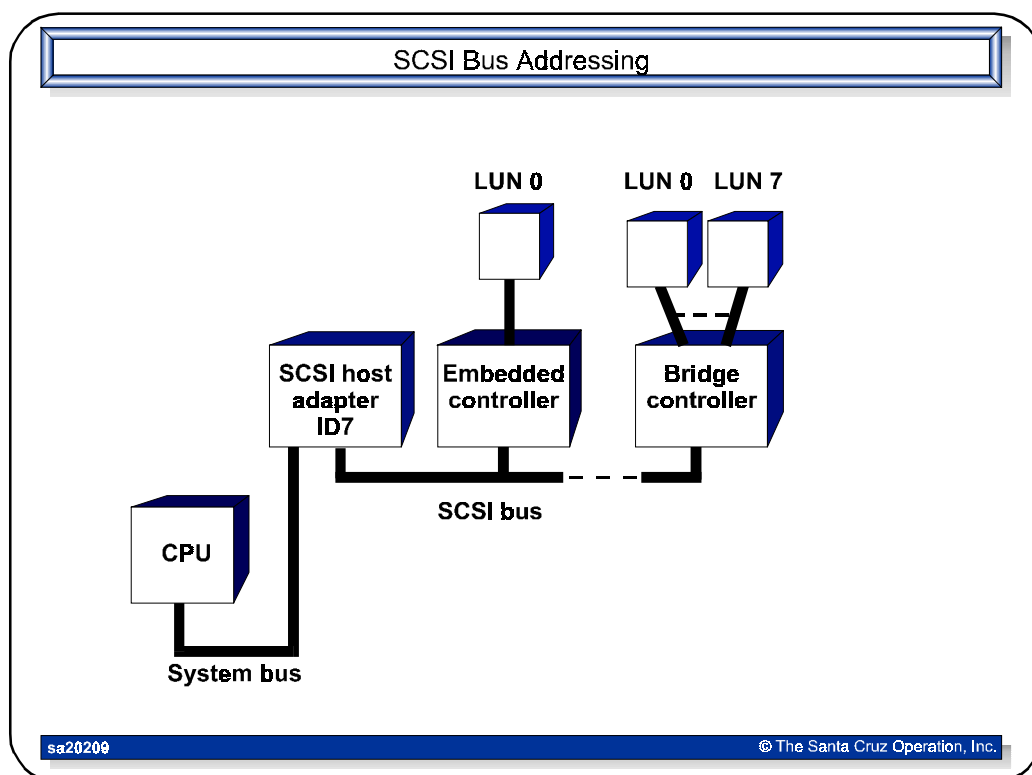
Small Computer System Interface (SCSI), commonly referred to as *scuzzy*, is an instruction-oriented, high-level interface specification for interconnecting a variety of peripheral devices (such as hard disk drives, CD-ROM drives, and tape drives) to a *SCSI bus*. This SCSI bus exchanges data only among the SCSI devices connected to it.

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

Before you configure a new SCSI host adapter, you might need to run a setup program or configuration utility.

AV 2-9





### 2.4.1 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 2 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.

## 2.5 Supported Hard Disk Configurations

---

SCO OpenServer systems support many hard disks and most hard disk controllers, including:

- SCSI
- Integrated Drive Electronics (IDE)
- Enhanced IDE (EIDE)
- Compaq Intelligent Disk Array (IDA)

The following lessons provide more information on hard disk configuration:

- Lesson 3, “Files and Filesystems,” includes a brief overview of the hard disk configuration
- Lesson 4, “Installation Planning,” discusses the choices for setting up your hard disks during initial operating system installation
- Lesson 8, “Filesystem Configuration and Disk Management,” describes how to configure additional disks after initial installation

## 2.6 General Configuration Information

---

This section provides information on general compatibility issues and configuration parameters.

### 2.6.1 General compatibility issues

---

Consider the following compatibility issues when adding hardware to an SCO OpenServer system:

- Some additional hardware (such as host, video, or network adapters) might 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 on the SCO Compatible Hardware Web Pages (CHWP). Using the compatibility data, you can quickly identify the hardware that is appropriate for your particular configuration.
- Manufacturers of SCO-supported machines or add-on peripherals may change configuration, functionality, or firmware at any time

#### 2.6.1.1 Determining if your hardware is compatible

---

To determine whether hardware components are compatible with your machine, you must know:

- The microprocessor (for example, i486, Pentium, Pentium Pro)
- The bus architecture (ISA, EISA, MCA, PCI, and so forth)
- The type of disk controller (IDE, EIDE, and so forth)

#### 2.6.1.2 Hard disk formats

---

Some computers arrive with the hard disk unformatted. If you have such a machine, use the manufacturer's hard disk format procedure before installing SCO OpenServer. This does *not* apply to most SCSI and IDE hard disk drives.

## 2.6.2 General configuration parameters

---

If your system hangs mysteriously at boot time, during installation, or shortly after installation, check for hardware conflicts between components, including:

- DMA channels
- I/O base addresses
- Shared RAM addresses
- Shared ROM addresses
- IRQs

## 2.7 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:

- Installed hardware and software
- Repair history, both hardware and software
- Details of scheduled and unscheduled maintenance and backups
- Copies of important configuration files
- Printouts, command output, and so forth, of any other information that you think might be useful. For example:
  - A copy of the `/etc/default/filesys` file (**cat /etc/default/filesys**)
  - Disk free space (**df -v**)
  - Additional multiport and secondary hard disk parameters (**tail -100 /usr/adm/messages**)

### **Exercise: Using the sample system administration log book**

---

Open the sample system administration log book (included with your student kit) and examine it.

Become familiar with your classroom server by logging in as **root** and entering the following commands. Record the information you discover in the sample log book.

**hwconfig -h**

**sar -r 1 5**

**vmstat -s**

**sar -u 1 5**

**uname -X**

**swconfig**

**divvy**

**fdisk**

**tail -100 /usr/adm/messages**

**df -v**

**cat /etc/default/filesys**

## **2.8 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. Which board contains the microprocessor and memory?
  
  
  
  
  
  
  
  
  
  
3. Which bus interface is commonly used for video and disk drive interconnections?

4. What are three main functions of a SCSI host adapter?
  
5. Name at least three types of disk controllers supported by SCO.
  
6. What is one of the most common causes of problems during installation, and what do you need to do to resolve the problem?
  
7. List five items that should be listed in a well-maintained system administration log book.



## 2.9 Relevant Commands and Files

---

The following commands were presented in this lesson.

### Relevant commands

---

|                      |                                                                                                                                                                                                                                                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>df(C)</b>         | Displays disk free space; with the <b>-v</b> option, reports the percent of blocks used as well as the number of free blocks                                                                                                                                                                                                                    |
| <b>divvy(ADM)</b>    | Divides a disk into partitions                                                                                                                                                                                                                                                                                                                  |
| <b>fdisk(ADM)</b>    | Displays, creates, and deletes partitions                                                                                                                                                                                                                                                                                                       |
| <b>hwconfig(HW)</b>  | Displays hardware configuration information                                                                                                                                                                                                                                                                                                     |
| <b>sar(ADM)</b>      | Reports system activity; with the <b>-r</b> option, reports unused memory pages and swap area disk blocks; with the <b>-u</b> option, reports CPU utilization                                                                                                                                                                                   |
| <b>swconfig(ADM)</b> | Queries the system contents database, lists file status, and lists software modifications                                                                                                                                                                                                                                                       |
| <b>tail(C)</b>       | Displays the last ten lines of a file; you can specify the amount of lines to display with the <b>-n</b> argument, where <i>n</i> is the desired number of lines                                                                                                                                                                                |
| <b>uname(C)</b>      | Prints the name of the operating system. With the <b>-X</b> option, prints additional information that includes the system name, the node name, the operating system release number, the kernel ID, the processor type, the bus type, the serial number, the number of user licences, the OEM number, the origin number, and the number of CPUs |
| <b>vmstat(C)</b>     | Reports paging and system statistics; with the <b>-s</b> option, reports the information in the format “number description”                                                                                                                                                                                                                     |

### Relevant files

---

|                             |                                                                                                                        |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------|
| <i>/etc/default/filesys</i> | Contains default information for mounting filesystems                                                                  |
| <i>/usr/adm/messages</i>    | Contains system log information, including configuration information, error messages, driver information, and so forth |

## 2.10 Answers to Review Questions

---

This section provides suggested answers to the 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.

---

# 3

## *Files and Filesystems*

AV 3-1

Files and Filesystems

- Describe the directory hierarchy
- Describe the role of Software Storage Objects (SSOs)
- Describe the UNIX filesystem types
- Describe the UNIX disk structure

sa20301 © The Santa Cruz Operation, Inc.

**Lesson 3**

## **Files and Filesystems**

### **Introduction**

---

Before you install SCO OpenServer Release 5, you should have a basic understanding of the UNIX filesystem hierarchy, the structure of a UNIX disk, and supported filesystem types. In addition, you should be familiar with the filesystem layout created to accommodate the Software Storage Object (SSO) architecture.

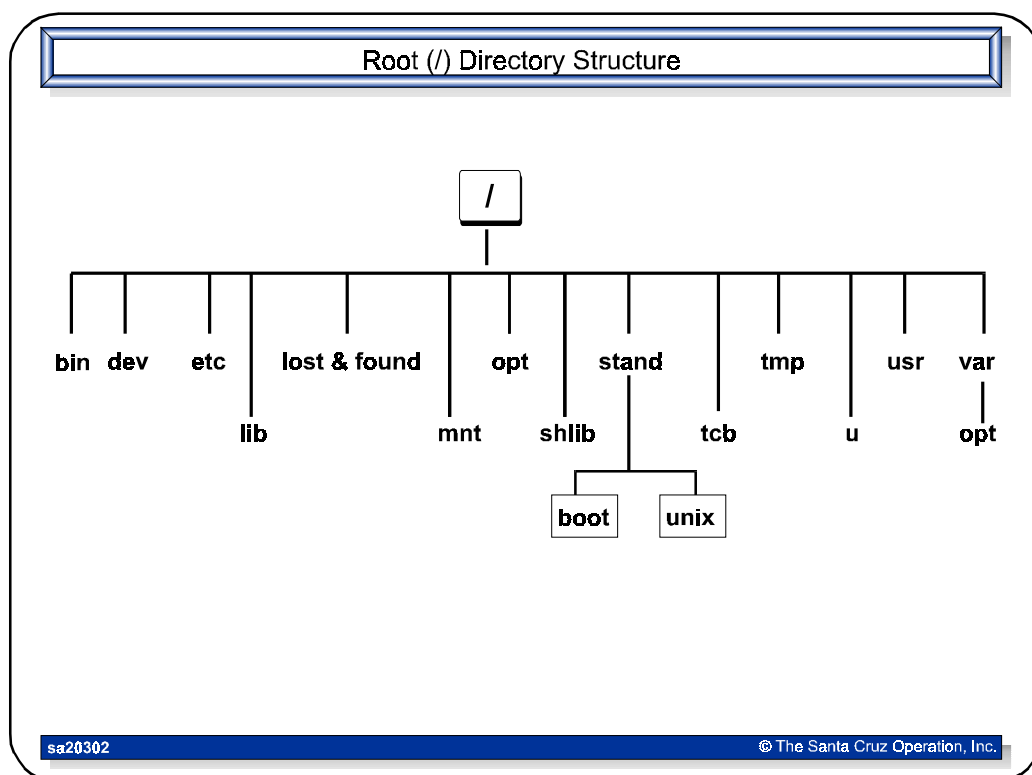
### **Objectives**

---

This lesson covers UNIX disk and filesystem basics. By the end of this lesson, you will be able to:

- Describe the directory hierarchy
- Describe the role of Software Storage Objects (SSOs)
- Describe the UNIX filesystem types
- Describe the UNIX disk structure

AV 3-2



### 3.1 Describing the Directory Hierarchy

---

Knowing about some of the key directories on the SCO OpenServer system can 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 its directories is listed below:

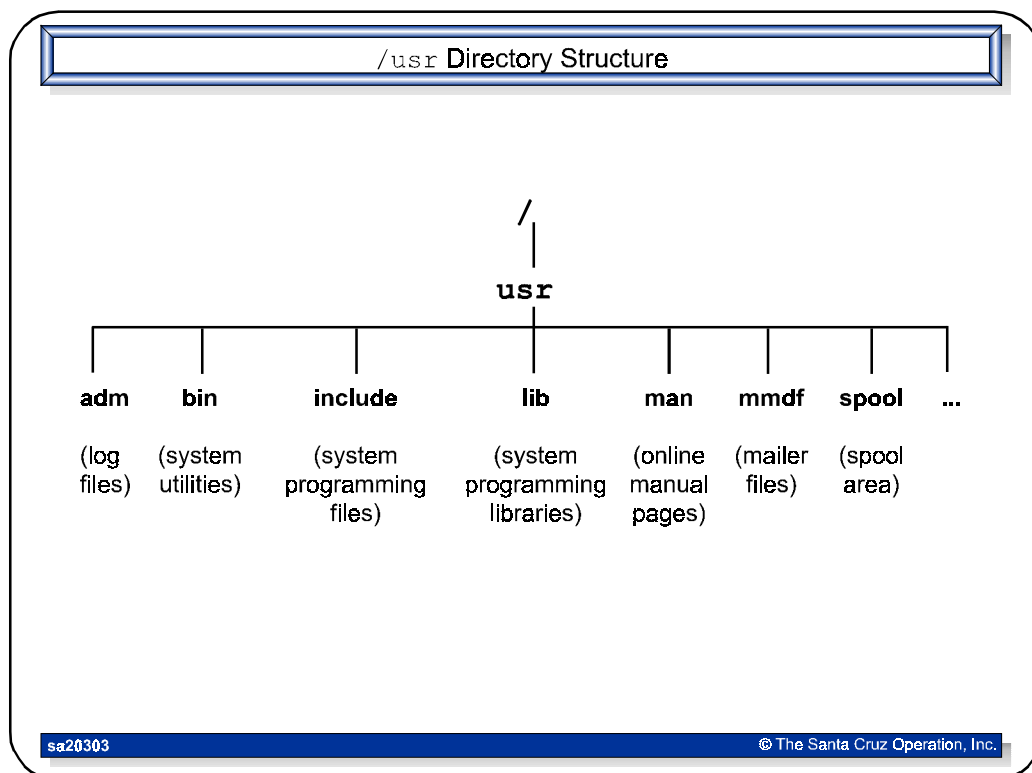
|                    |                                                                                                                                                                                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>/bin</i>        | System binaries                                                                                                                                                                                                                                                                                              |
| <i>/dev</i>        | System device files                                                                                                                                                                                                                                                                                          |
| <i>/etc</i>        | System administration files and tools, including startup and shutdown scripts, kernel generation, and subsystem authorization                                                                                                                                                                                |
| <i>/lib</i>        | C programming libraries                                                                                                                                                                                                                                                                                      |
| <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>        | Contains the actual system software in subdirectories known as <i>software storage objects</i> , or SSOs. All files in <i>/opt</i> are read-only. The <i>/opt</i> directory also contains the shared files of SSOs that clients of a server system can use ( <i>/var/opt</i> contains the non-shared files). |
| <i>/shlib</i>      | Shared libraries                                                                                                                                                                                                                                                                                             |
| <i>/stand</i>      | Directory to access the boot filesystem, which contains the kernel and boot files                                                                                                                                                                                                                            |
| <i>/tcb</i>        | The Trusted Computing Base, used to keep trusted computing data, such as passwords and audit trails                                                                                                                                                                                                          |
| <i>/tmp</i>        | Temporary files created by programs. Files stored in this directory will be erased if <b>cron</b> (C) has been scheduled to remove them.                                                                                                                                                                     |
| <i>/usr</i>        | Contains application and system-specific information; the default location for user accounts                                                                                                                                                                                                                 |
| <i>/var</i>        | Contains the non-shared SSO files that are specific to an individual client or server                                                                                                                                                                                                                        |



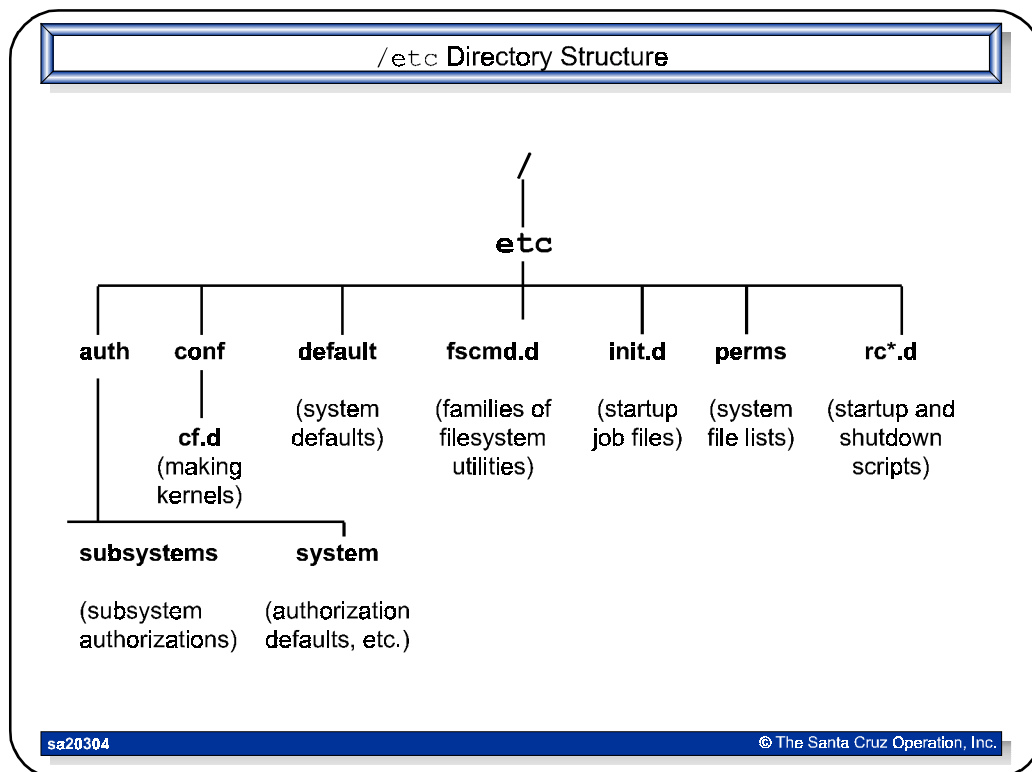


|                   |                                                                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>/var/adm</i>   | Contains data files associated with system administration and accounting. The <i>/usr/adm</i> directory is a symbolic link to <i>/var/adm</i> .                                                         |
| <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 a symbolic link to <i>/var/spool</i> .                             |
| <i>/var/opt</i>   | Contains read/write copies of the configuration files from the <i>/opt</i> directory. The <i>/var/opt</i> directory also contains the non-shared files of SSOs that clients of a server system can use. |

## AV 3-3



## AV 3-4



### 3.1.1 The */usr* directory structure

---

The */usr* directory contains many working files and directories for UNIX utilities. User home directories are placed in this directory unless another user filesystem (for example, */u*) is specified at time of installation. The */usr* directory 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 symbolic links to additional binaries that are UNIX application specific, including <b>uucp(C)</b> , <b>vi(C)</b> , and <b>mail(C)</b> |
| <i>/usr/lib</i>   | Contains symbolic links to configuration files for many UNIX application utilities, including <i>/usr/lib/mail/mailrc</i>                       |
| <i>/usr/man</i>   | Contains symbolic links to online manual pages in <i>/opt</i>                                                                                   |
| <i>/usr/spool</i> | A symbolic link to <i>/var/spool</i> , a storage directory for spooling programs                                                                |

### 3.1.2 The */etc* directory structure

---

These commands and files in the */etc* directory structure relate directly to system administration:

|                     |                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------|
| <i>/etc/auth</i>    | Contains symbolic links to subsystem authorizations and defaults files                                   |
| <i>/etc/conf</i>    | A symbolic link to <i>/var/opt/K/SCO/Unix/5.0.5Eb/etc/conf</i> , the kernel tuning and regeneration area |
| <i>/etc/default</i> | Contains symbolic links to configuration files related to system defaults                                |

### 3.1.3 The */tcb* directory structure

---

In addition to the */etc/auth* directory, these directories and files related to the Trusted Computing Base are also present:

|                                |                                                                                 |
|--------------------------------|---------------------------------------------------------------------------------|
| <i>/tcb/audittmp</i>           | Location of audit records                                                       |
| <i>/tcb/bin</i>                | Audit commands                                                                  |
| <i>/tcb/files/auth/[a-z]/*</i> | Includes user-encrypted passwords, system authorizations, and kernel parameters |
| <i>/tcb/lib</i>                | Audit commands                                                                  |

---

## 3.2 Describing the Role of the Software Storage Objects (SSOs)

---

The filesystem layout accommodates the software storage object (SSO) architecture. Most files in the root filesystem physically reside in two hierarchies:

|                           |                                                                                                                                                                                                                                        |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The <i>/opt</i> hierarchy | Contains read-only versions of <i>all</i> files. For example, the <b>mount</b> (ADM) command and its original version of its configuration file, <i>filesys</i> , reside in the <i>/opt</i> hierarchy. These files are never modified. |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                 |                                                                        |
|-----------------|------------------------------------------------------------------------|
| <i>/var/opt</i> | Contains read/write copies of the configuration files from <i>/opt</i> |
|-----------------|------------------------------------------------------------------------|

A copy of *filesys* also resides in the */var/opt* directory, enabling you to modify a filesystem's mount configuration. If this file becomes corrupt, rather than restoring from a backup, you can copy it from the */opt* hierarchy.

### 3.2.1 The SSO architecture

---

The SSO architecture is the key feature to SCO's software management technology. *Software management* is the packaging and administering of the software on a system and includes the installing, upgrading, licensing, verifying, tracking, patching, and removing of software.

An SSO is the basic unit of software functionality, essentially a software *component*. A component is the smallest unit of software that can stand alone. It can be one of many components in a product, or it can constitute an entire product. For example, the SCO OpenServer Operating System is a software component.

An SSO starts out as a single directory subtree in the */opt* hierarchy. This portion of the SSO remains unchanged, and in the same location, throughout its lifetime on the system. The full path to this portion of each SSO is called the *SSO root*, and is composed of codes for a vendor, a component, and a version as follows:

*/opt/K/VendorCode/ComponentCode/ComponentVersion*

For example, the */etc/mount* program and the *filesys* file are part of the operating system component, version 5.0.5Eb, and are located in the */opt/K/SCO/Unix/5.0.5.Eb* directory.

All system utilities in the root filesystem are actually links to an SSO. For example, all files in the */bin* directory are symbolic links to the files in */var* and */opt*. When using the **l(C)** command to perform a directory listing, note that an "@" appended to the filename indicates a symbolic link. Use the **-Wv** option to the **l(C)** command to view the actual filename.

**Exercise: Listing software components**

---

Use **l -Wv** to list the `/etc/mount` file. What is the version number of the operating system component? Change to the `/etc` directory and use the **l(C)** command to list the files. From the listing, how can you tell that these files are symbolic links?

---

**3.2.2 Benefits of the SSO architecture**

---

The SSO architecture provides these software management features:

|                               |                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Intelligent upgrades          | Allows you to upgrade from an older release of the software to a new release, preserving software configuration information across releases                                                                                                                                                                                                                      |
| Removable patches             | Allows you to easily remove the effect of a patch                                                                                                                                                                                                                                                                                                                |
| Multiple versions             | Provides the ability to keep multiple versions of a product loaded on your machine. Although a single machine can only run one version locally at a time, software servers can make several different versions available to other systems. Loading multiple versions also allows you to test new versions of the software before removing the current version.   |
| Network installation          | <p>Allows you to install the operating system or any other software onto another system over the network, or share it with other machines using NFS connections. It also allows you to manage software on remote machines.</p> <p>For more information, see the <b>SCO OpenServer Networking Guide</b>, “Installing and Managing software over the network.”</p> |
| Diskless and dataless systems | Allows you to configure diskless, dataless, or small-disk machines connected to and sharing software on a central server. Diskless and dataless operations can reduce costs, increase security, and simplify configuration management.                                                                                                                           |

AV 3-5

| Supported Disk Filesystems |          |            |                    |                   |                                                      |
|----------------------------|----------|------------|--------------------|-------------------|------------------------------------------------------|
| Filesystem                 | Bootable | Versioning | Check-<br>pointing | Intent<br>Logging | Advantages                                           |
| AFS                        | X        |            | X                  | X                 | Backwards compatability                              |
| EAFS                       | X        |            | X                  | X                 | Used to create boot filesystem                       |
| DTFS                       |          | X          | X                  | X                 | Uses data compression to save disk space             |
| HTFS                       |          | X          | X                  | X                 | Defaults use caching techniques for high performance |
| S51K                       | X        |            | X                  |                   | Backwards compatability                              |
| XENIX                      | X        |            |                    |                   | Backwards compatability                              |
| DOS                        |          |            |                    |                   | Eliminate need for DOS utilities                     |

sa20305 © The Santa Cruz Operation, Inc.

## 3.3 Filesystem Types

---

Filesystem types differ in capabilities that can affect how they are used. Listed below are the most often used filesystem types and a brief discussion of each.

### 3.3.1 HTFS filesystem

---

The High Throughput Filesystem (HTFS) filesystem transparently provides significant performance improvement by using caching to increase data throughput. This is the default filesystem for the SCO OpenServer Host and Enterprise products.

An HTFS filesystem:

- Uses checkpointing. The filesystem is marked clean at regular intervals. If a filesystem is marked clean when the system halts, it might not be necessary for it to be checked by **fsck**(ADM).
- 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. Intent logging works in tandem with checkpointing.
- Uses versioning to allow users to undelete a file
- Supports long filenames and symbolic links
- Has approximately 134,000,000 inodes per filesystem
- Has a maximum file size of 2GB and a maximum filesystem size of 1TB
- Has a logical data block size of 1KB
- Is not a bootable filesystem

Configuring an HTFS filesystem, checkpointing, and intent logging are covered in detail in Lesson 8, “Filesystem Configuration and Disk Management.”

### 3.3.2 DTFS filesystem

---

The Desktop Filesystem (DTFS) is designed for use on laptop, desktop, and notebook computers. This is the default filesystem for the SCO OpenServer Desktop product.

A DTFS filesystem:

- Uses transparent data compression and a more efficient disk media format to increase the storage capacity of your disks
- Uses checkpointing, intent logging, and versioning
- Supports long names and symbolic links
- Can contain either user data or an inode on any block on the disk. The number of inodes is not fixed and is limited to the size of the partition.
- Has a maximum file size of 2GB and a maximum filesystem size of 1TB
- Uses a variable block size that is a multiple of 512 bytes up to 4KB. This helps minimize wasted space at the end of incomplete blocks.
- Is not a bootable filesystem

### 3.3.3 EAFS filesystem

---

The Extended Acer Fast Filesystem (EAFS) is the default filesystem used by SCO UNIX Systems Versions 2.0 and 3.0, and is also the default filesystem for the boot filesystem (*/stand*) on SCO OpenServer products.

An EAFS filesystem:

- Uses checkpointing and intent logging
- Supports long filenames and symbolic links
- Has 65535 inodes per filesystem
- Has a 2GB maximum for file and filesystem size
- Uses 1KB logical data blocks
- Is a bootable filesystem and is used when creating the boot filesystem



### 3.3.4 Other supported filesystems

---

Other filesystems supported by SCO include:

- Acer Fast Filesystem (AFS)
- S51K Filesystem
- Rockridge CD-ROM Filesystem
- High Sierra/ISO9660 CD-ROM Filesystem
- DOS Filesystem
- Network Filesystem (NFS)
- NetWare® for UNIX Client Filesystem (NUCFS)
- LAN Manager Client Filesystems (LMCFS)
- XENIX

Refer to the *SCO OpenServer System Administration Guide*, “About filesystems,” for more information on supported filesystem types.

### 3.3.5 Determining filesystem types

---

The **fstyp(ADM)** command displays filesystem types.

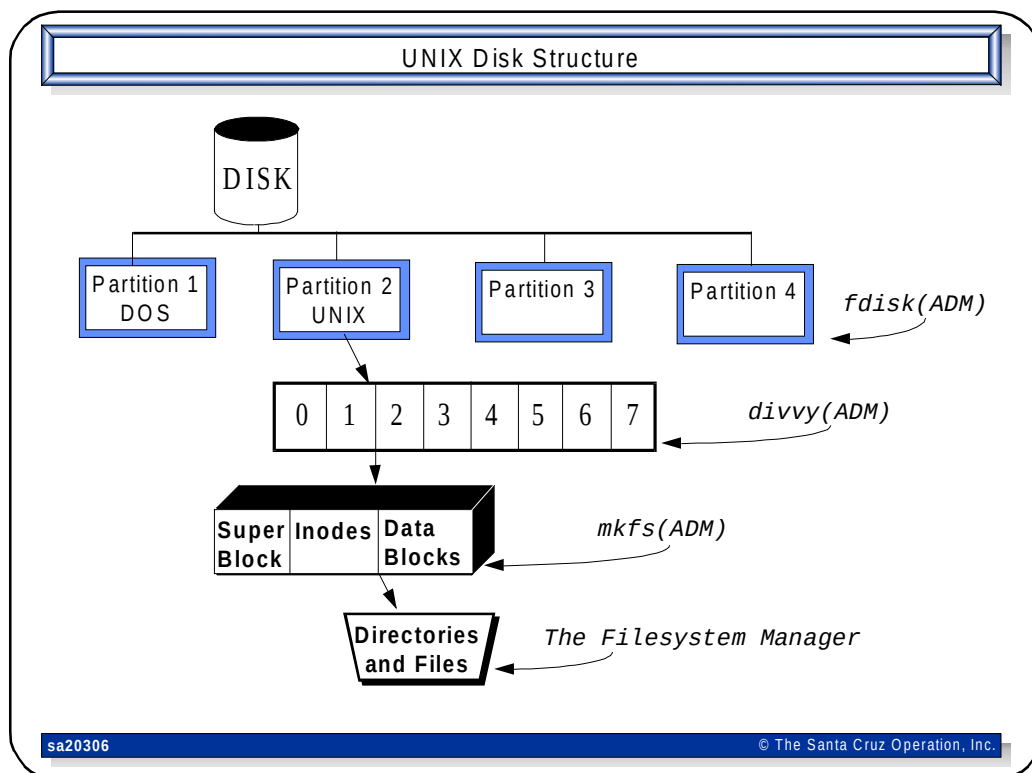
**Exercise: Determining the boot filesystem type**

---

---

```
fstyp /dev/boot
```

## AV 3-6



## 3.4 Describing the UNIX Disk Structure

---

Before creating files and directories, you must have a basic understanding of the disk structure. A UNIX disk is comprised of:

- Physical blocks
- Partitions
- Divisions

### 3.4.1 Physical blocks

---

All disks contain a low-level format consisting of 512-byte records called *physical blocks*. This low-level format is originated at the disk manufacturer. Utilities exist to refresh this structure.

### 3.4.2 Partitions

---

You can divide the hard disk into one to four contiguous data areas for the installation of operating systems such as UNIX, OS/2, and DOS.

### 3.4.2.1 The UNIX partition

---

The UNIX partition contains two sections:

- Section one, the system, contains the boot codes and partition layout tables.
- Section two, the division area, contains up to 7 divisions numbered 0-6.

One additional division, division 7, is used to describe the entire partition.

During the installation of SCO OpenServer, four divisions are created by default:

- The boot filesystem, */dev/boot*, mounted to */stand*, contains the **boot(F)** program and the UNIX kernel.
- The root filesystem, */dev/root*, mounted to *(/)*, contains all other system files.
- A non-filesystem division called swap, */dev/swap*, is used for virtual memory paging.
- A non-filesystem division called recover, */dev/recover*, is used as a scratch area for the **fsck(ADM)** program when repairing filesystems.

### **3.4.3 Divisions**

---

UNIX filesystem divisions are used to create a system of files based on a hierarchical directory structure. These divisions contain operating system, application, and user data.

Non-filesystem divisions are typically used for virtual memory (swap), system maintenance, and some database applications.

AV 3-7

| The dkinit(ADM) Command |        |
|-------------------------|--------|
| Disk Parameters         | Values |
| 1. Cylinders            | 1000   |
| 2. Heads                | 64     |
| 3. Write Reduce         | 0      |
| 4. Write Precomp        | 0      |
| 5. Ecc                  | 0      |
| 6. Control              | 0      |
| 7. Landing Zone         | 0      |
| 8. Sectors/track        | 32     |

sa20307

© The Santa Cruz Operation, Inc.

### 3.4.4 Utilities for setting up hard disks

---

When you set up your hard disk and configure filesystems during installation, these utilities or commands, are invoked:

- **dkinit**(ADM)
- **fdisk**(ADM)
- **badtrk**(ADM)
- **divvy**(ADM)

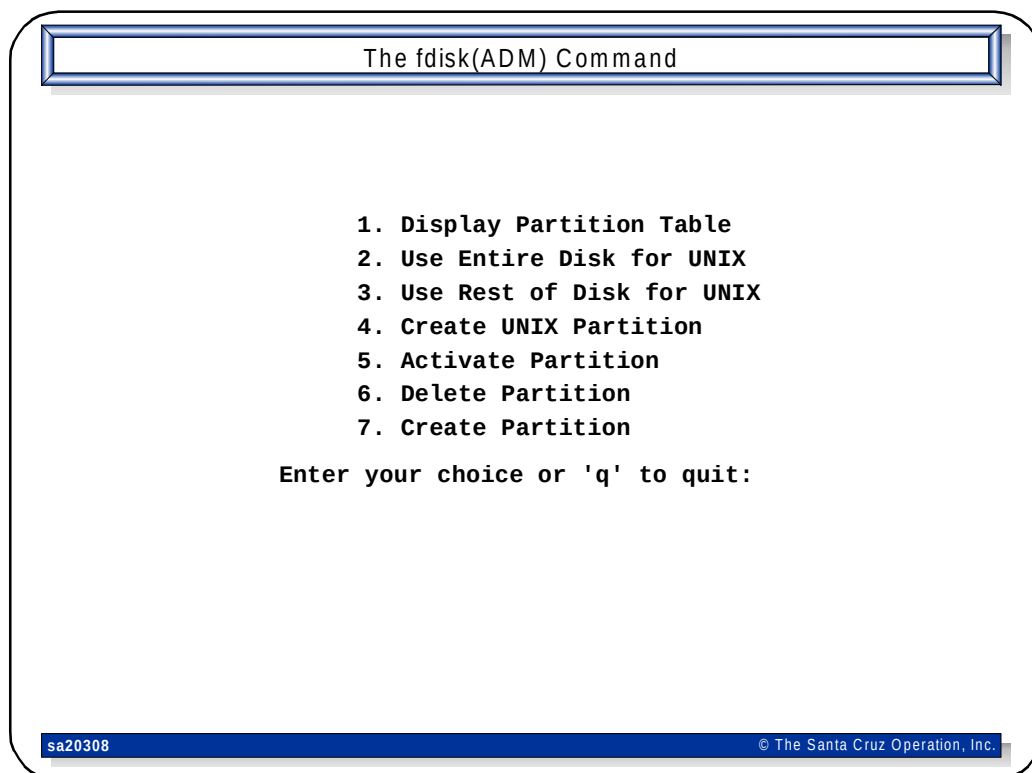
#### 3.4.4.1 The **dkinit**(ADM) command

---

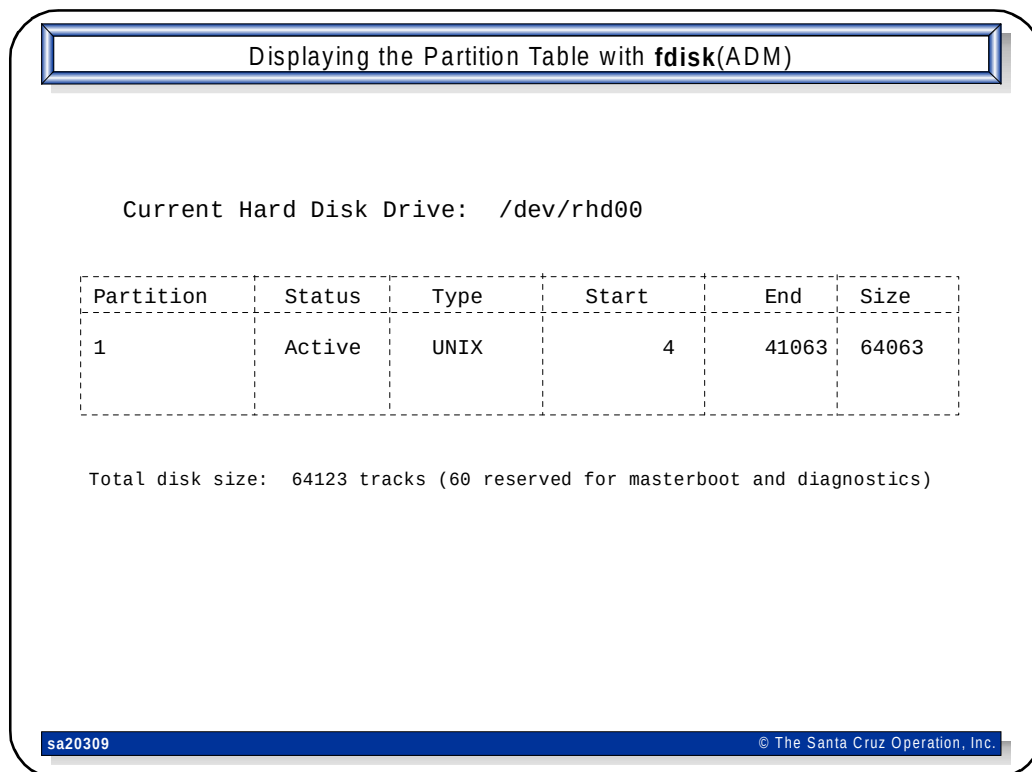
The **dkinit**(ADM) command is not invoked when adding a SCSI disk. For non-SCSI disks, **dkinit** enables you to:

- Display default disk parameters, such as the number of heads, cylinders and sectors per track
- Modify existing parameters for non-standard disks
- Select the default for standard disks

## AV 3-8



## AV 3-9





### 3.4.4.2 The **fdisk**(ADM) command

---

The **fdisk**(ADM) command enables you to create a UNIX partition. Several options are provided when creating partitions:

- **Use Entire Disk for UNIX**

If you select this option, any existing partitions are destroyed. The UNIX partition is automatically marked Active.

- **Use Rest of the Disk for UNIX**

If you select this option, any existing partitions are preserved.

- **Create UNIX Partition**

Use this option to specify the starting and ending tracks.

- **Create a Partition**

Select this option to create a non-UNIX partition.

- **Activate Partition**

If you have not selected **Use Entire Disk for UNIX**, select this option to activate the partition.

The **fdisk**(ADM) command reports disk sizes in tracks. The number of tracks is equal to the number of heads multiplied by the number of cylinders. The last cylinder is reserved for disk diagnostics.

---

**Exercise: Displaying the partition table**

---

Run **fdisk**(ADM) to display the partition table for the primary disk.

AV 3-10



### 3.4.4.3 The **badtrk(ADM)** command

---

The **badtrk(ADM)** command is intended for use with ST506, IDE, EIDE ESDI, and SCSI disk controllers. It will not function with an IDA controller, as it manages its own internal defect list.

When configuring a disk during installation, **badtrk(ADM)**:

- Scans a hard disk partition for flaws
- Creates a new bad track table for non-SCSI disks
- Creates a new bad block table for SCSI disks
- Prints the current bad block table
- Adds and deletes entries in the bad block table

The **badtrk(ADM)** command is covered in more detail in Lesson 8, “Filesystem Configuration and Disk Management.”

Refer to the **badtrk(ADM)**, **badblk(ADM)**, and **scsidisk(HW)** manual pages for more information on bad blocking a SCSI hard disk.

## AV 3-11

The **divvy**(ADM) Command

| Name     | Type       | New FS | # | First Block | Last Block |
|----------|------------|--------|---|-------------|------------|
| boot     | EAFS       | no     | 0 | 0           | 15359      |
| swap     | NON FS     | no     | 1 | 15360       | 80895      |
| root     | HTFS       | no     | 2 | 80896       | 1023988    |
|          | NOT USED   | no     | 3 | -           | -          |
|          | NOT USED   | no     | 4 | -           | -          |
|          | NOT USED   | no     | 5 | -           | -          |
| recover: | NONFS      | no     | 6 | 1023989     | 1023998    |
| hd0a     | WHOLE DISK | no     | 7 | 0           | 1025007    |

1023999 1K blocks for divisions, 1008 1K blocks reserved for the system

### 3.4.4.4 The **divvy**(ADM) command

---

The **divvy**(ADM) command divides a disk partition into a number of separate areas called divisions. When the **divvy**(ADM) command is invoked, it presents these choices:

- **n**[name] Name or rename a division.

Enter **n** to specify the division you want to name or rename. The **divvy**(ADM) command creates two device files for the name you enter. For example, if you enter *sales*, **divvy**(ADM) creates */dev/sales* and */dev/rsales*. The block device name is used later when adding the filesystem to the filesystem table.

- **c**[create] Create a new filesystem on this division.

Enter **c** to bring the division into use.

- **t**[type] Select or change filesystem type on new filesystems.

Enter **t** to select a filesystem other than the default, HTFS. Note that if you are installing the Desktop version of the filesystem of an SCO OpenServer System, DTFS is the default filesystem type.

- **s**[tart] Start a division on a different block.

Enter **s** to specify where you want the division to start. If it is the first filesystem created, enter **0**.

- **e**[nd] End a division on a different block.

Enter **e** to specify the ending block number.

- **r**[estore] Restore the original division table.

Enter **r** if you make an error or do not want any of the changes you made to take effect. The original division table is restored.

Continue to answer these questions if multiple filesystems are needed for this partition. After you complete filesystem specification, select **q** to quit and then select **i** to install the **divvy** table. The **divvy**(ADM) command then invokes **mkfs**(ADM) to create the filesystem(s) on the disk.

## 3.5 Lesson Summary

---

The UNIX disk structure consists of physical blocks, partitions, and divisions.

Filesystems are a division of the UNIX partition. They are comprised of directories and files. The root (/) directory is the hierarchical starting point of all files and directories.

SCO OpenServer supports many filesystem types, including HTFS, DTFS, and EAFS. Filesystem layout in SCO OpenServer systems accommodate the new SSO architecture.

The commands used to set up hard disks include **dkinit**(ADM), **fdisk**(ADM), **badtrk**(ADM), and **divvy**(ADM).

## Review questions

---

1. Which directory contains the majority of the files and tools to administer startup and shutdown?
2. What is “software management”?
3. Which two filesystems allow you to undelete a file?
4. Name the three components that make up the structure of a UNIX disk.

## 3.6 Relevant Commands and Files

---

The following command and files were presented in this lesson.

### Relevant Commands

---

|                     |                                                                                                                                               |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>badtrk</b> (ADM) | Scans the hard disk for flaws and creates a badtrack/bad block table                                                                          |
| <b>cron</b> (C)     | Executes commands at specified dates and times                                                                                                |
| <b>divvy</b> (ADM)  | Divides a UNIX partition                                                                                                                      |
| <b>dkinit</b> (ADM) | Displays or changes hard disk parameters                                                                                                      |
| <b>fdisk</b> (ADM)  | Displays, creates, or deletes partitions                                                                                                      |
| <b>fstyp</b> (ADM)  | Determines a filesystem type                                                                                                                  |
| <b>l</b> (C)        | Lists the contents of a directory; with the <b>-Wv</b> options, displays the actual files that symbolic links point to in a directory listing |
| <b>mount</b> (ADM)  | Mounts filesystems and remote resources                                                                                                       |

### Relevant files

---

|                     |                                                                                                              |
|---------------------|--------------------------------------------------------------------------------------------------------------|
| <i>/dev/boot</i>    | The boot filesystem                                                                                          |
| <i>/dev/recover</i> | A non-filesystem division used at system startup                                                             |
| <i>/dev/root</i>    | The root filesystem                                                                                          |
| <i>/dev/swap</i>    | A non-filesystem division used when memory is low and data and programs are swapped to disk <i>/dev/root</i> |

## 3.7 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Listing software component

---

The version number as of this release is 5.0.0CI; an @ sign is appended to the filename of any symbolic link.

### Exercise: Determining the boot filesystem type

---

The boot filesystem is an EAFS filesystem type.

### Review questions

---

This section provides the answers to the review questions.

1. The `/etc` directory contains the majority of files and tools used to administer system startup and shutdown.
2. *Software management* is the packaging and administering of the software on a system and includes installing, upgrading, licensing, verifying, tracking, patching, and removing software.
3. The DTFS and HTFS filesystems allow you to undelete a file.
4. A UNIX disk is comprised of physical blocks, partitions, and divisions.



---

# 4

## *Installation Planning*

AV 4-1

The slide is titled "Installation Planning" and contains a bulleted list of three tasks. The footer includes the identifier "sa20401" and the copyright notice "© The Santa Cruz Operation, Inc.".

**Installation Planning**

- Identify SCO OpenServer products
- Identify SCO OpenServer hardware requirements
- Complete an installation checklist

sa20401 © The Santa Cruz Operation, Inc.

**Lesson 4**

## **Installation Planning**

### **Introduction**

---

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

After completing a pre-installation analysis, you will have an overall picture that describes the amount and kind of work that the server is expected to do. An installation checklist is provided to make the actual installation faster and easier. This checklist includes the questions you will need to answer during installation. The answers are determined during the analysis phase allowing you to complete the installation without interruption.

### **Objectives**

---

This lesson describes the tasks you need to complete before installing the operating system software. By the end of this lesson, you will be able to:

- Identify SCO OpenServer products
- Identify the hardware that SCO OpenServer systems require
- Complete an installation checklist

## 4.1 Identifying OpenServer Products

---

An important part of an installation analysis is determining the software you are installing. The SCO OpenServer family includes three operating systems, a general-purpose Internet server, and a single, combined development system. For more information on the latter, see the documentation provided with the SCO OpenServer Development System.

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Host System       | This product is designed for low-cost, turnkey solutions and mission-critical applications that require a central application host and do not require networking. In addition to many features that deliver improved performance and reliability, it contains full graphical support, including SCOadmin, an OSF/Motif <sup>®</sup> system administration tool that also runs on character terminals. The Host System also contains the SCO Advanced File and Print Server, which provides NetBEUI networking for native Microsoft Windows file and print connections. |
| Enterprise System | This product includes all the features of the Host System and complete networking tools, including installation over a network, TCP/IP (with SCO Advanced File and Print Server running over TCP/IP), NFS, IPX/SPX, SCO Gateway for NetWare, LAN Manager Client, Netscape Communicator <sup>™</sup> and the Netscape FastTrack server. As such, it is the most powerful and versatile server in the product line.                                                                                                                                                      |
| Desktop System    | This product is designed as a single-user workstation and contains full graphical support as well as full networking connectivity. The Desktop System also includes Netscape Communicator.                                                                                                                                                                                                                                                                                                                                                                             |

Additional products and components included on the SCO OpenServer CD-ROM:

- Netscape Communicator<sup>™</sup> 4.0.5
- SCO Advanced File and Print Server Release 3.5.2
- SCO<sup>®</sup> ArcServe<sup>®</sup> /Open from Cheyenne<sup>®</sup> Lite 2.2
- SCO Doctor<sup>™</sup> Lite
- SCO Intel Optimizing C Compiler (icc) 2.1.4

On the Optional Services CD-ROM:

- Netscape<sup>®</sup> Mail Server<sup>™</sup> 2.02
- RealNetworks<sup>™</sup> products
- SCO ArcServe/Open from Cheyenne (Full product set)
- SCO Doctor (Full product set)
- SCO Internet Products, including Netscape<sup>®</sup> browsers and servers
- SCO Merge<sup>™</sup> Releases 3.2.2t and 4.0.1
- SCO TermLite<sup>™</sup> 1.0.7
- SCO VisionFS<sup>™</sup> 2.0.1
- SCO VisionFS

For a complete description of these products, see the */info/readme.htm* file on the Optional Services CD-ROM.

## 4.2 Identifying OpenServer Hardware Requirements

---

To run SCO OpenServer software, a server must meet certain minimal requirements that are listed in the **SCO OpenServer Release Notes**. The **Release Notes** also list suggested configurations.

The following list shows recommended installation hardware requirements that are representative of a typical installation, but might not be appropriate for every site:

|               |                                                  |
|---------------|--------------------------------------------------|
| CPU           | 80486DX minimum<br>Pentium or better recommended |
| System memory | 16MB minimum<br>32MB recommended                 |
| Disk space    | 400MB minimum<br>1.2GB or better recommended     |

Refer to the **SCO OpenServer Release Notes**, “Installation hardware requirements,” “Disk space,” for more detailed information on various scenarios.

|                          |                        |
|--------------------------|------------------------|
| root (/) filesystem size | 400MB recommended      |
| /stand filesystem size   | 15MB                   |
| Swap device size         | 24MB minimum           |
| Monitor                  | VGA, 640x480 or better |

### 4.2.1 Disk space requirements

---

Disk space requirements vary depending on the software you choose to install. If you are installing SCO products or other application software, you must increase the root filesystem size by the required amount, plus any resulting increases in swap. SCO products include:

**Add:**

|                                    |      |
|------------------------------------|------|
| SCO Virtual Disk Manager           | 1MB  |
| SCO Symmetrical Multiprocessing    | 1MB  |
| SCO OpenServer Development System  | 51MB |
| SCO Advanced File and Print Server | 80MB |

|                                 |      |
|---------------------------------|------|
| SCO ArcServe/Open from Cheyenne | 24MB |
|---------------------------------|------|

|                 |     |
|-----------------|-----|
| SCO Doctor Lite | 5MB |
|-----------------|-----|

The swap area is a specific section of the hard drive 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`.

Swap area ranges from 150% times system memory to 200% times system memory (for servers with 32MB of RAM or more).

## 4.2.2 Installation drive requirements

---

To install an SCO OpenServer system, you must have:

- One 3.5-inch floppy drive, for use with the boot installation floppy disk
- A SCSI CD-ROM or EIDE (ATAPI) CD-ROM

or

- One bootable CD-ROM that uses the "El Torito" standard of the Rock Ridge filesystem. The El Torito filesystem identifies a boot file on the CD-ROM, then emulates the A: drive on your computer to load the file and boot the system.

## 4.2.3 Using SCO documentation to identify hardware requirements

---

You can find the hardware requirements for OpenServer products in:

|                                            |                                         |
|--------------------------------------------|-----------------------------------------|
| <b><i>SCO OpenServer Release Notes</i></b> | Included with the software distribution |
|--------------------------------------------|-----------------------------------------|

|                                       |                                                                                      |
|---------------------------------------|--------------------------------------------------------------------------------------|
| <b><i>SCO OpenServer Handbook</i></b> | Included with the software distribution; also included with the online documentation |
|---------------------------------------|--------------------------------------------------------------------------------------|

***SCO Hardware Compatibility Pages (HCWP)***

Available on the SCO WWW site at:  
[http://wdb1.sco.com/chwp/owa/hch\\_search\\_form](http://wdb1.sco.com/chwp/owa/hch_search_form)

### 4.2.3.1 The SCO OpenServer Release Notes

---

The ***Release Notes*** contain:

- All system requirements for the product you are about to install
- General information about the version of the product

You should read the **Release Notes** thoroughly before beginning any installation.

#### 4.2.3.2 The **SCO OpenServer Hardware Compatibility Web Pages (HCWP)**

---

To determine server hardware compatibility with a system, refer to the manufacturer's specifications for the hardware and the HCWP.

The HCWP defines hardware that:

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

It 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 that vendor

The HCWP list hundreds 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
- Terminal/Printer servers



- Data/Fax modems
- Multiconsole systems
- X Terminals
- Uninterruptable power supplies (UPS)
- Terminals

To search for supported hardware, access the *HCWP* at:

[http://wdb1.sco.com/chwp/owa/hch\\_search\\_form](http://wdb1.sco.com/chwp/owa/hch_search_form)

#### 4.2.3.3 The *SCO OpenServer Handbook*

---

The *SCO OpenServer Handbook* describes how to install, configure, and start using SCO OpenServer software.

##### **Exercise: Examining the *SCO OpenServer Handbook***

---

Examine the *SCO OpenServer Handbook* that came with the software you are installing in class. Which chapter discusses configuring video adapters and monitors?

In which chapter will you find the information needed to use Boot Time Loadable Drivers?

## 4.3 Completing an Installation Checklist

---

You must answer many questions when planning for the installation of the operating system on a server, including:

- Is there a requirement for other operating systems on the hard disk?
- What level of security is needed?
- How many users will access this server?
- Will a graphical environment be used?
- Which applications are to be used?

An installation and upgrade checklist is included with the **SCO OpenServer Handbook** and is designed to help you determine the answers to all relevant questions before you begin a server installation. A copy of the **SCO OpenServer Handbook** is included with your student guide.

As you complete the installation analysis for your classroom server, fill out the installation and upgrade checklist in the **SCO OpenServer Handbook**. Your instructor will help you determine the correct choices for your classroom.

During installation, you must answer a sequence of questions about customizing the server. Use the completed installation checklist to answer the questions.

### 4.3.1 Installing Other Partitions

---

If you plan to install Windows NT<sup>™</sup>, OS/2, or DOS partitions on your server, SCO recommends that you install these partitions *before* you install SCO OpenServer.

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

- Use the NT or OS/2 tools
- Use **fdisk**(ADM) to switch to or from OS/2 and NT; **fdisk**(ADM) displays OS/2 and NT partitions as OS/2

If you plan to install DOS 6.0 or higher partitions, note the following:

- A DOS partition must be the primary partition on a disk.
- A DOS partition should have a *minimum* size of 7MB.
- A DOS partition can be any size greater than 7MB.

DOS partitions can be created before or after the SCO OpenServer installation. If you create the partition before you install SCO OpenServer, you must choose the *Preserve installation* option. This preserves the existing DOS partition during hard disk initialization.

Refer to the **SCO OpenServer Handbook**, “Using other operating systems with your SCO system,” for more information about DOS and other partition restrictions.

### **4.3.2 Boot-time loadable drivers (BTLDs)**

---

BTLDs are drivers that provide access to hardware devices that are not included in the install kernel (*unix.install*). Using a BTLD, you can install onto new hardware. You can use a vendor’s device driver and hardware element to install the operating system.

If your hardware requires BTLDs, you must have a floppy disk that contains those drivers before you begin installation.

Typically, BTLDs are installed in units called *packages*. You can add BTLDs:

- By typing **link** at the **Boot :** prompt and pressing , then entering the package names when prompted
- By using the **link =“pkg1 pkg2...”** bootstring argument, where **pkg1**, **pkg2**, and so on are the names of BTLD packages to be linked into the loaded UNIX system kernel
- By using **link** in conjunction with the restart option at the **Boot :** prompt by typing **restart link=“pkg1 pkg2...”**

### 4.3.3 Installation media

---

SCO OpenServer requires either of these boot devices:

- One 3 1/2 inch floppy disk drive
- One CD-ROM drive that is configurable in your system's CMOS as being a boot device, and that does not use a proprietary boot format.

The SCO OpenServer media kit provides both a primary installation floppy disk and the installation CD-ROM. In addition, images of all floppy disks you might need exist in the */images/boot* directory on the CD-ROM. These include:

*install.img*      An image of the primary installation disk that you can use to replace a lost or damaged copy.

*notebook.img*    An image of a small kernel installation disk, for use on systems with small memory (16 or 24MB) configurations. This kernel has all of the disk drivers removed except for IDE/ATAPI (wd) and Adaptec 1520 (smad). Kernel buffer allocations are also reduced and might affect installation speed.

*btld.img*        An image of the Boot-Time Loadable Drivers disk containing drivers for SCSI host adapters not supported by the primary installation diskette. There are also additional BTLDs provided for convenience.

The installation media device must be installed in the server before you begin the software installation.

If you use your primary tape or CD-ROM drive, the installation autodetects your hardware and fills in the correct entry for you. If your hardware is not set at the factory defaults, you will input the correct values for the following:

- SCSI tape or CD-ROM:
  - Host adapter 0 (required)
  - ID number

Refer to the SCO HCWP for a list of supported CD-ROM drives.

### 4.3.4 The keyboard

---

Select the language for which your keyboard is configured. For example, US English.

### 4.3.5 Licensing your system

---

An End User License Agreement (EULA) is no longer shipped with the SCO OpenServer media. Obtain your license agreement from your SCO Authorized Reseller.

References in the *SCO OpenServer Handbook* to enclosed, hard-copy licensing agreements are incorrect.

#### 4.3.5.1 License number

---

When you install, you must provide a license number. By license agreement, it must be unique; that is, two servers on a network cannot be installed using the same license number.

#### 4.3.5.2 License code

---

When you install, you must provide a license code.

#### 4.3.5.3 License data

---

Not all products have license data. If you do not receive a license number, it is not required.

### 4.3.6 Installation type

---

There are two types of installations:

|                      |                                                                                                                             |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------|
| Upgrade installation | Preserves your existing partition, filesystems, and data                                                                    |
| Fresh installation   | Does not automatically preserve existing filesystems; it overwrites the previous installation unless you indicate otherwise |

### 4.3.7 Configuring the basic system

---

The basic system parameters include specifying the system name, the domain name, the level of security, and the time zone.

#### 4.3.7.1 System name

---

The system name is the name of your server. Programs on the server such as **mail(C)** use the system name to identify the correct data destination.

The name must:

- Be unique on a network
- Begin with a letter
- Contain only lowercase letters and numbers
- Have a maximum of seven characters

Although you can use a maximum of eight characters in the name, SCO recommends that you select a name that is seven characters or less; some communications programs (such as UUCP) only recognize server names of seven characters or less.

If you are responsible for administering many servers, you can construct a common server-naming convention for each server. Sample valid server names include:

- *srv100e* (describes the building the server is in, along with its location in that building)
- *sales1* (identifies the set of applications on the server, or the users who use them)
- *rome15* (describes the general geographical location of the server)

Refer to the **SCO OpenServer Handbook**, “TCP/IP configuration parameters,” “System Name.”

#### 4.3.7.2 Domain name

---

The domain name is an abbreviation for the server site and a code for the site’s organization or country. For example, *sco.com* is the domain name used by SCO.

If your server is networked, use your registered 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 *name.UUCP*, where *name* is the name of your company or organization.

### 4.3.7.3 Security Profile

---

Select one of the following four security profiles:

|                 |                                                                                                                                                                                   |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| High (above C2) | Approximates the C2 class of trust defined by the Department of Defense Trusted Computer System Evaluation Criteria; recommended for systems containing confidential information. |
| Improved (C2)   | Meets the minimum requirements of the C2 level of trust; recommended for systems accessed by groups of users who share information.                                               |
| Traditional     | Provides compatibility with other UNIX systems; recommended for systems that use traditional UNIX security and want to incorporate some of the C2 security features.              |
| Low             | Provides limited security; recommended only for systems not publicly accessible, with a small number of cooperating users.                                                        |

Refer to the *SCO OpenServer System Administration Guide*, “Security profiles.”

### 4.3.7.4 Time zone

---

You must select a time zone when you install.

Select “other” if your time zone is not represented in the choices for Geographical Area. Choose “Manually select” and you are prompted for:

- Time zone name
- West or East of GMT
- Hours from GMT

### 4.3.8 Hard Disk Setup

---

You can use the whole disk to install SCO OpenServer Release 5 or partition the disk with up to four partitions. If additional hard disks are detected, you can also install partitions and filesystems on that disk.

Hard disk configurations include:

**Preserve existing layout** This option is available if your system has an active UNIX partition with a clean root filesystem and a valid `/etc/default/filesys`. This option retains the filesystem layouts from the previous installation, except for the root filesystem. It removes everything from the root filesystem and installs the new system, leaving all other data intact.

This option is not available when configuring a second disk.

**Use whole disk for UNIX** Clears any existing partitions and divisions from your hard disk and creates a default UNIX partition and division.

**Customize** Clears any existing partitions, creates the same layout and allows you to create or modify divisions.

**Interactive fdisk/divvy** Defers disk setup until the end of the installation prompts. The installation then runs the interactive **fdisk**(ADM) and **divvy**(ADM) commands to create partitions and hard disks.

This option is not available if you are configuring a second disk.

**Defer** Defers the installation of the second hard disk. You can configure additional disks with the **mkdev**(ADM) command after installation.

For more information on setting up hard disks, refer to the *SCO OpenServer Handbook*, “Installing or upgrading an SCO system.”



### 4.3.9 Bad track/block

---

A bad track is an area of the hard disk that is not reliable for data storage. You can choose one of the following options for checking bad tracks and bad blocks.

- None (the system will not check for bad tracks or blocks)
- Thorough/destructive (ensures highest disk integrity, but is slowest)
- Thorough/non-destructive
- Quick/destructive
- Quick/non-destructive

Refer to the *SCO OpenServer Handbook*, “Fixing bad tracks and bad blocks on hard disks,” for more information. Refer to the **scsidisk**(HW) and **badtrk**(ADM) manual pages for more information about SCSI hard drive bad tracking.

### 4.3.10 Software components

---

You can install whole and partial components in any combination. After installation, you can add or remove software as described in the *SCO OpenServer Handbook*, “Administering system software.” Refer to the *SCO OpenServer Release Notes* for more information concerning software component hard drive requirements.

### 4.3.11 TCP/IP network configuration parameters

---

You can choose to defer some or all of the TCP/IP network configuration parameters.

#### 4.3.11.1 Network card

---

You do not have to enter this information if you want to defer network configuration, or if your system can autodetect your network card.

If you do choose to configure networking during installation, you might be asked for this information:

- Interrupt vector (IRQ)
- Base I/O address (if applicable)
- ROM address
- DMA (if applicable)
- Base RAM address (if applicable)
- Local Ring Broadcast (Token-ring only)
- Slot number (if applicable)
- Cable type (if applicable)

Refer to the ***Configuring Network Connections***, “Configuring parameters” and “Configuring network hardware,” for more information on configuring your network adapter.

#### 4.3.11.2 IP address

---

The Internet Protocol (IP) address identifies and differentiates your machine from all others on the network. The IP address consists of two parts: a network address and a host address. The format for this address is four decimal numbers separated by periods. For example, 132.145.80.9.

Refer to ***Configuring Network Connections***, “TCP/IP configuration parameters,” “IP address,” for more information on IP addresses.

#### 4.3.11.3 Network mask

---

The network mask determines, based on your network class, which part of your network address is used as a network ID, and which part is used as the system, or host, ID.

If you are not configuring a subnet, use the default value.

Refer to ***Configuring Network Connections***, “TCP/IP configuration parameters,” “Netmask setting.”

### 4.3.12 IPX/SPX network configuration parameters

---

You can choose to defer some or all of the TCP/IP network configuration parameters.

#### 4.3.12.1 IPX/SPX network number

---

The IPX/SPX network number links DOS workstations clients and SCO system software over the NetWare network.

Refer to the *SCO OpenServer Handbook*, “IPX/SPX configuration parameters,” “Network number.”

#### 4.3.12.2 IPX/SPX internal network number

---

The IPX/SPX internal network number must be a unique number for every machine on all networks.

Refer to the *SCO OpenServer Handbook*, “IPX/SPX configuration parameters,” “Internal network number.”

#### 4.3.12.3 IPX/SPX framing type

---

IPX/SPX allows you to set the type of framing performed by the network adapter driver.

Refer to the *SCO OpenServer Handbook*, “IPX/SPX configuration parameters,” “Framing type.”

### 4.3.13 Graphics setup

---

The graphics setup parameters configure your graphics card and allow you to specify whether the **Scolopin** is on or off.

#### 4.3.13.1 Video adapter

---

SCO OpenServer provides a VESA graphics driver that conforms to the VESA Video BIOS Extension (VBE) 2.0 and 1.2. Because of the generic nature, the VESA driver works with most graphic adapters.

VESA VBE 2.0 graphic adapters includes support for:

- 256 or 64K colors at resolutions of 640x480 to 1600x1200
- Uses a generic frame buffer to display graphics

You can also check the documentation that came with your video card to determine the type of video card you have. During installation, you can select a video card from a list of supported video cards.

Refer to the ***SCO OpenServer Handbook***, “Configuring video adapters,” ***SCO OpenServer Hardware Compatibility Web Pages (HCWP)***, and ***SCO OpenServer Features and Limitations***.

### 4.3.13.2 Video mode

---

During installation, the default graphics mode is set to VESA 800x600 with 256 colors.

Check the documentation that came with your video card to select the appropriate video mode. The video mode determines the resolution (for example, VGA 640x480), the amount of colors your video display can use (for example, 256), and the speed (for example, 60Hz).

### 4.3.13.3 Video monitor

---

Check the documentation that came with your monitor to determine the monitor type, for example, ViewSonic® 17G or NEC MutliSync XP21. This information is often available on the monitor itself.

### 4.3.13.4 Graphical login

---

During installation, you must decide if you will use the graphical environment. The selections are:

- On
- Off

Leave the graphical login choice on if you want a graphical Desktop multiscreen.

Refer to the ***SCO OpenServer Handbook***, “Customizing startup of the Graphical Environment.”

### 4.3.14 Mouse

---

If you are using the graphical environment, you should specify a mouse during installation. The available mouse options include:

- Serial mouse:
  - COM1
  - COM2
- Bus mouse:
  - IRQ2 (INTR2)
  - IRQ3 (INTR3)
  - IRQ5 (INTR5)
- Keyboard mouse:
  - low resolution
  - high resolution

Refer to the *SCO OpenServer Handbook*, “Adding mice and other graphic input devices.”

### 4.3.15 E-mail system

---

The choices for an e-mail system are:

- MMDF
- Sendmail

If you are not sure which to choose, ask the network administrator for your site, or choose MMDF.

Refer to the *SCO OpenServer Mail and Messaging Guide*, “Comparison of sendmail with MMDF.”

## 4.4 Lesson Summary

---

---

The SCO OpenServer family of products consists of:

- Desktop System
- Host System
- Enterprise System

While some variation is possible, the recommended hardware configuration for SCO OpenServer Release 5.0.5 products consists of:

- A Pentium processor (or better)
- 32MB RAM
- 1.2GB of free disk space
- A supported CD-ROM drive

You can use the **SCO** HCWP to investigate whether specific server hardware is compatible with SCO OpenServer software. The **SCO OpenServer Release Notes** and the **SCO OpenServer Handbook** contain useful information pertinent to installation. All server installations begin with an analysis of the server. You must consider many points, including:

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

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

## Review questions

---

1. What are the names of the three SCO OpenServer Release 5 operating system products?
2. What drives are required to install the SCO OpenServer operating system?
3. Identify the parts of the installation checklist that pertain to the server hard drive and list them below.

## 4.5 Relevant Commands and Options

---

The following commands and options were presented in this lesson.

### Relevant commands

---

|                     |                                                                                                                  |
|---------------------|------------------------------------------------------------------------------------------------------------------|
| <b>badtrk</b> (ADM) | Finds and keeps track of bad tracks on your hard disk (not used for IDA disks)                                   |
| <b>divvy</b> (ADM)  | Divides a UNIX partition into UNIX divisions and other required devices such as swap and recover                 |
| <b>dkinit</b> (ADM) | Allows you to display, modify, or select default disk parameters while installing (not used for SCSI disks)      |
| <b>fdisk</b> (ADM)  | Creates partitions on a hard drive; also used to activate one partition as the one that boots on system power up |
| <b>mkfs</b> (ADM)   | Creates UNIX filesystems (superblock, inodes, data blocks)                                                       |
| <b>mkdir</b> (C)    | Creates directory files in a UNIX filesystem                                                                     |

### Relevant options

---

|             |                                                              |
|-------------|--------------------------------------------------------------|
| <b>link</b> | Allows BTLDS to be incorporated into the kernel at boot time |
|-------------|--------------------------------------------------------------|



## 4.6 Answers to Review Questions

---

---

This section provides suggested answers review questions.

### Review questions

---

1. The three SCO OpenServer Release 5 operating system products are Desktop System, Host System, and Enterprise System.
2. To install the SCO OpenServer operating system, you must have one 3.5-inch floppy drive and a supported CD-ROM drive or a supported bootable CD ROM.
3. The following items in the installation checklist that pertain to the server hard drive are installing non-UNIX partitions (before installation), installation type (fresh or upgrade), hard disk setup (customized, interactive, or use entire disk), and second hard disk setup.



---

# 5

## *Operating System Installation*

AV 5-1

Operating System Installation

- Stop and restart the installation
- Install the operating system software
- Troubleshoot problems during the installation
- Troubleshoot problems after the installation

sa20501 © The Santa Cruz Operation, Inc.

**Lesson 5**

## **Operating System Installation**

### **Introduction**

---

After you have completed the installation analysis and checklist for your server, you can install the operating system with confidence. This lesson provides a detailed discussion of the installation procedure.

When installation is complete, examine the server to determine if the installation was successful. If it was not successful, you must determine exactly where the software or hardware problem is before attempting a repair or reinstallation.

### **Objectives**

---

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

- Stop and restart the installation
- Install the operating system software
- Troubleshoot problems during the installation
- Troubleshoot problems after installation

## 5.1 Stopping and Restarting the Installation

---

The installation of the operating system can be divided into two stages:

|                  |                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Setup            | The installation prompts you with questions and records your answers. Your replies determine which software components are installed, how server hardware is configured, and the disk division layout. |
| Software loading | The installation program installs and configures your server using the parameters defined during setup. This stage is not interactive.                                                                 |

### 5.1.1 Stopping and restarting the installation

---

You can stop the installation during both the setup and software loading stages of installation.

#### 5.1.1.1 After setup begins

---

To stop and restart the installation process after setup begins:

- Step 1. Press **F2**. You are prompted to confirm.
- Step 2. Enter **y** to stop the installation. The system displays an **Installation aborted** message.

Warning: Do not use the **DEL** or **CTRL\** keys to cancel the installation. This causes the system to reboot; you will have to restart the installation to continue, and any configuration information that you entered is lost.

- Step 3. Verify that the Boot Disk is in the drive.

- Step 4. Press **ENTER**. The **BOOT :** prompt appears.

- Step 5. Restart the installation as follows:

- To restart the installation from where you left off, press **ENTER**
- To restart the installation from the beginning, enter **restart** or **restart link=<pkg>** (if you are loading BTLDS)

### 5.1.1.2 After software loading begins

---

To stop and restart the installation process after the software loading begins:

- Step 1.        Verify that the Bootable CD-ROM or Boot Disk is in the appropriate drive.
- Step 2.        Press the computer's Reset button or power the system off and on.
- Step 3.        Enter one of the following at the **BooT :** prompt:
  - **restart**
  - **restart link=<pkg>** (if you are loading BTLDS)

### 5.1.2 Installing SCO OpenServer Release 5

---

During the installation, use the checklist you completed in the previous lesson, "Installation Planning," to respond to the installation prompts.

These steps describe how to install the SCO OpenServer operating system software.

To boot from floppy:

- Step 1.        Insert the Boot Disk into the drive.
- Step 2.        Turn on your computer. The **BooT :** prompt appears. Go to Step 3.

To boot from CD-ROM:

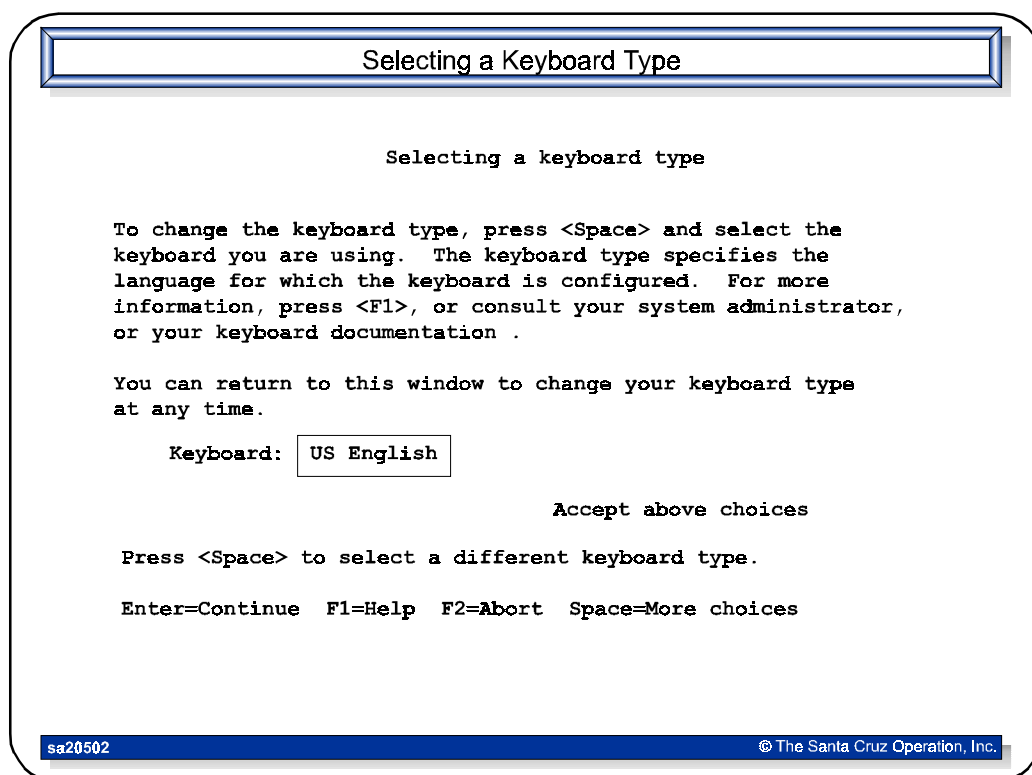
- Step 1.        Turn on your computer. Verify the CD-ROM drive is the bootable device.
- Step 2.        Insert the CD-ROM into CD-ROM drive.
- Step 3.        Press  to continue at the **BooT :** prompt.
- Step 4.        Press  to begin the installation.
- Step 5.        Press  to accept the Restricted Rights Legend.
- Step 6.        Press  to accept the terms and conditions of the End User License Agreement (EULA).





- Step 7. Change or accept the displayed parameters for your CD-ROM drive:
- For a SCSI CD-ROM, enter the SCSI host adapter type, number, LUN number, and bus.
  - For an IDE CD-ROM, enter which controller (primary or secondary) is being used for the CD-ROM and if it is the primary or slave device.
- Step 8. Press  to accept your choices.
- Step 9. Insert your installation CD-ROM (if you did not boot from the CD-ROM).
- Step 10. Accept or change the parameters of the CD-ROM drive:
- Select **OK** to continue the installation, or
- Select **Cancel** and re-enter the CD-ROM parameters.
- Press  to accept the CD-ROM parameters.

AV 5-2



Step 11.       Select the appropriate keyboard type (for example, US English).

Note that the bottom two lines of the screen give brief help for completing each field. For more detailed help, press **[F1]**. Often, you can also press **[SPACE]** to see a list of options from which you can select or bring up a secondary set of prompts.

AV 5-3

Licensing Your System

Licensing your system

Type in the license number and license code exactly as they appear on your Certificate of License and Authenticity.

License number: Type in  
License code: Type in

Accept above choices

Esc=Previous F1=Help

sa20503 © The Santa Cruz Operation, Inc.

AV 5-4

Suitability for Automatic Upgrade

It will take a few minutes to check whether you can run an automatic Upgrade on your system (retaining all user and configuration files.

To verify whether your system is suitable for an automatic Upgrade, highlight Upgrade below and press <Enter>.

To begin a Fresh installation (which overwrites the current root filesystem with the new product), highlight Fresh below and press <Enter>.

Upgrade Fresh

Press <Enter> to begin a Fresh installation.

sa20504 © The Santa Cruz Operation, Inc.

Step 12. Enter the License Number and License Code. If this screen contains a field for License Data, enter it).

Press  to accept.

Step 13. Select **Fresh** for a new installation.

Refer to the *SCO OpenServer Handbook*, “Installing or upgrading an SCO system,” “Choosing an upgrade path,” “About the upgrade installation type” for information on performing an upgrade installation.

Step 14. Select **OK** to continue.

AV 5-5

Configuring the Basic System

If the information below does not match your requirements, follow the instructions at the bottom of the screen to change the values:

|                   |                                 |
|-------------------|---------------------------------|
| System name:      | roseyl                          |
| Domain name:      | sco.com                         |
| Security Profile: | Traditional                     |
| Time zone:        | EasternStandard/<br>Summer Time |
| Language:         | Standard C (English)            |

Accept above choices

Type in a unique system name for this machine.

sa20505© The Santa Cruz Operation, Inc.

AV 5-6

Initial System Profile

The profile you choose pre-selects the software packages most useful for your system type. Press <F1> for details on each profile.

|                                               |     |
|-----------------------------------------------|-----|
| Standard Enterprise system configuration:     | Yes |
| Lightweight character terminal configuration: | No  |

If you run large database products, choose the following:

|                    |    |
|--------------------|----|
| Database services: | No |
|--------------------|----|

Accept above choices

Press <Space> to toggle between Yes and No.

sa20506© The Santa Cruz Operation, Inc.

- Step 15. Enter your system name and domain name, and select appropriate values for the security profile and time zone. Do not change the language.

Changing the language field does not change the language used during installation; the installation continues in English. It does set the default locale for the system. The locale defines your environment in terms of language and cultural conventions.

Press  to accept your choices.

The “Initial system profile” screen appears.

- Step 16. Select **Yes** to install the “Standard Enterprise system configuration.” This selection excludes the Lan Manager Client, SCO Gateway for NetWare, and PC Interface Server. You can install these products later when the optional products screen displays.

If you select **No**, you must choose which software to install when the optional product screen displays.

- Step 17. Select **No** to “Lightweight character terminal configuration.”

If you select **Yes**, graphical tools are not installed and your system is set up to use character terminal configuration.

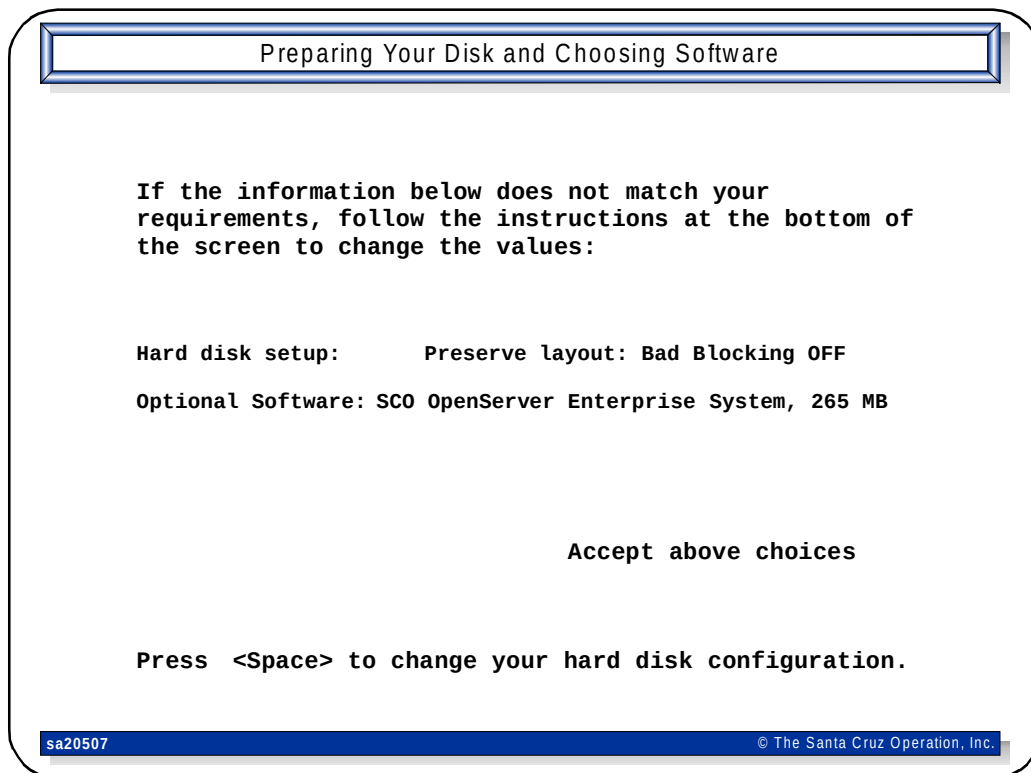
- Step 18. Select **No** to “Database services.” The Database services are not installed.

If you select **Yes**, the **suds** driver is installed. The **suds** driver is used with these databases:

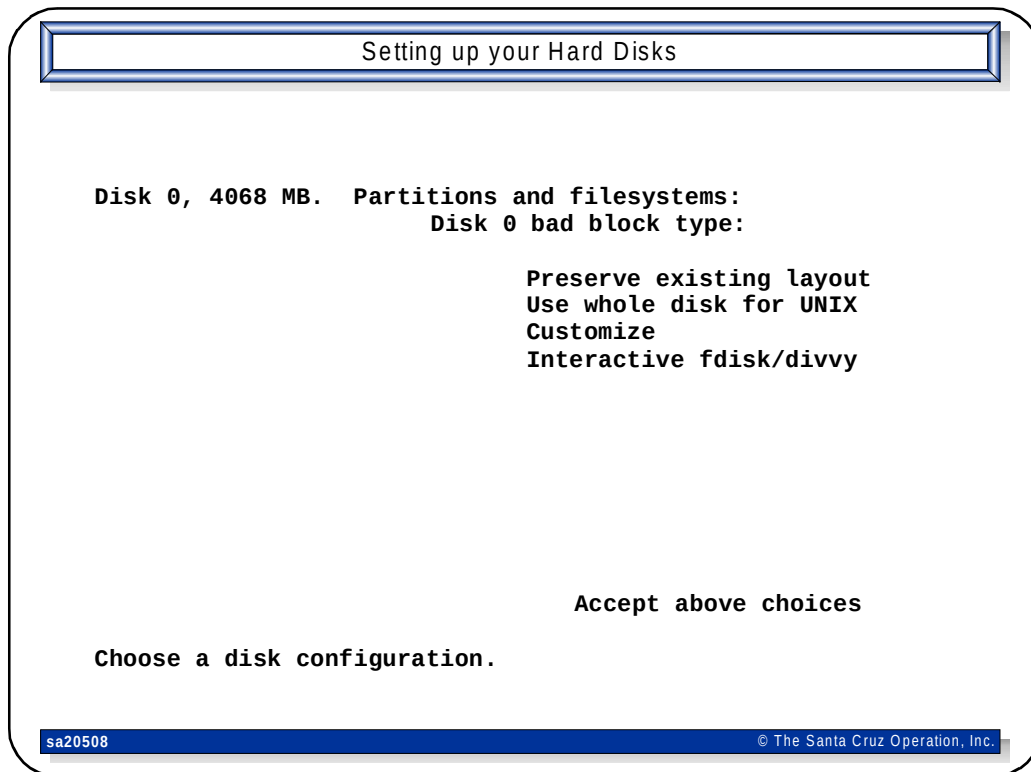
IBM DB2 v2 and succeeding versions  
Informix ODS v7.12 and succeeding versions  
Sybase SQL Server 11.0.2 and succeeding versions

Press  to accept your choices.

AV 5-7



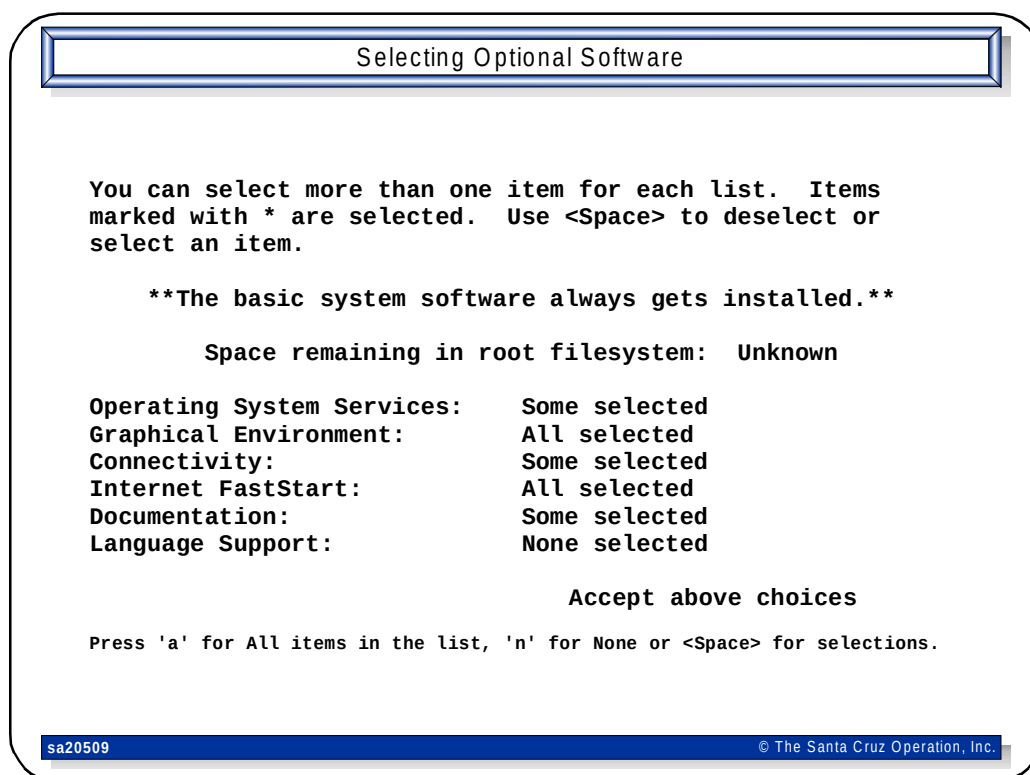
AV 5-8





- Step 19. Press the space bar twice for hard disk setup choices. The “Setting up your hard disks” screen appears. If you select:
- Step 20. Select **Interactive fdisk/divvy**. Press .
- Step 21. Press  until you return to the “Preparing your disk and choosing software” screen.
- Step 22. Select the “Optional software” field and press .

AV 5-9



- Step 23.      Select the optional software to install:
- Press ☐ SPACE for a list of optional software.
  - Press ☐ SPACE to deselect Documentation. An asterisk (\*) denotes a selected item.
- Step 24.      Do not install any additional software. Press ☐ ENTER until the “Configuring optional software” screen appears.

AV 5-10

**Configuring Optional Software**

If the information below does not match your requirements, follow the instructions at the bottom of the screen to change the values.

Please refer to the Installation Checklist in the OpenServer Handbook for assistance when selecting the Email system.

|                     |                            |
|---------------------|----------------------------|
| Network Card:       | Autodetect                 |
| Network Address:    | No network card configured |
| Video and Graphics: | VESA SVGA                  |
| Mouse:              | No mouse configured        |
| Email system:       | MMDF                       |

Accept above choices

Press <Space> for network card selection options.

sa20510
© The Santa Cruz Operation, Inc.

AV 5-11

**Configuring Your Network**

Setting network addresses for scosysv.sco.com

TCP/IP networking parameters:

|                    |                 |
|--------------------|-----------------|
| IP address:        | 132.147.103.197 |
| Net mask:          | 255.255.255.0   |
| Video and graphics | 132.147.103.255 |

IPX/SPX networking parameters

|                             |          |
|-----------------------------|----------|
| IP internal network number: | Deferred |
| IPX network number:         | Deferred |
| IPX framing type:           | Deferred |

Accept above choices

Type in your IPX internal network number. Press <F5> to defer configuration.

sa20511
© The Santa Cruz Operation, Inc.

Step 25. Configure optional software as follows:

- For the “Network card,” select **Autodetect** for the system to try and automatically identify the card, or **Manually select** to choose from a list of network cards and I/O addresses. You can also choose **Deferred** to defer the configuration of your network card.

Select **Autodetect** unless your instructor tells you otherwise.

- For the “Network address” (if you are not deferring configuration), press  and the “Configuring your network” screen appears.

Enter the networking parameters from your installation checklist. Press  to accept your choices.

If you plan to install the “IPX/SPX DOS Component” or “SCO Gateway for NetWare,” you might also want to configure the IPX/SPX™ networking parameters at this time.

For “Video and graphics” (if you are not deferring configuration), press  and the “Graphics setup” screen appears. Press  for a list of options for the video card, video mode, monitor, and graphical login.

- For the “Mouse,” press  for a list of options. Select the appropriate mouse. Press .
- For the “Email system,” press  and choose either **MMDF** or **sendmail**.

Press  to accept your choices.

AV 5-12

**Setting Your Root Password**

Setting your root password

Enter a root password that is at least 5 characters long and a combination of lowercase letters, uppercase letters, numbers, or symbols. To defer setting a root password until after the installation is complete, press <F5>

Root password:

Please type again to confirm:

Accept above choice

Esc=Previous   Enter=Continue   F1=Help   F2=Abort   F5=Defer

sa20512© The Santa Cruz Operation, Inc.

When you complete configuring optional software, the “Setting your root password” screen appears.

Step 26. Enter the **root** password (twice) for your server (use the password indicated by your instructor). Press  to accept.

Step 27. Select **OK** to proceed after you have provided all the information necessary to install your software.

You can also select **Cancel** if you need to change any of your answers.

AV 5-13

### System V Hard Disk Initialization

Starting hard disk initialization

System V Hard Disk Initialization

The primary hard disk in the system will now be initialized.

The hard disk initialization requires you to set the disk parameters, specify the size of the UNIX partition, and control the layout of the filesystems and swap area. System default values are available at each selection.

Initialization Selection:

1. Continue with hard disk initialization.
2. Return to the Installation Query Manager (IQM).  
This will return you to the serial/activation screen. All previous answers will be lost.
3. Exit and abort installation.

Enter your choice:

sa20513

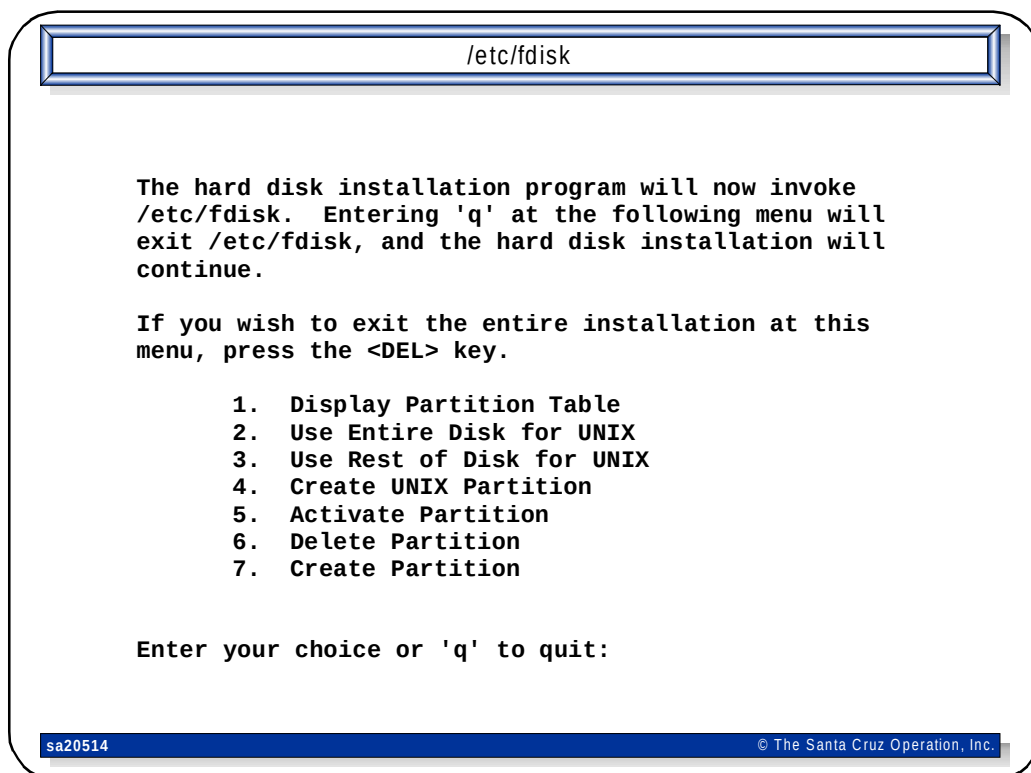
© The Santa Cruz Operation, Inc.



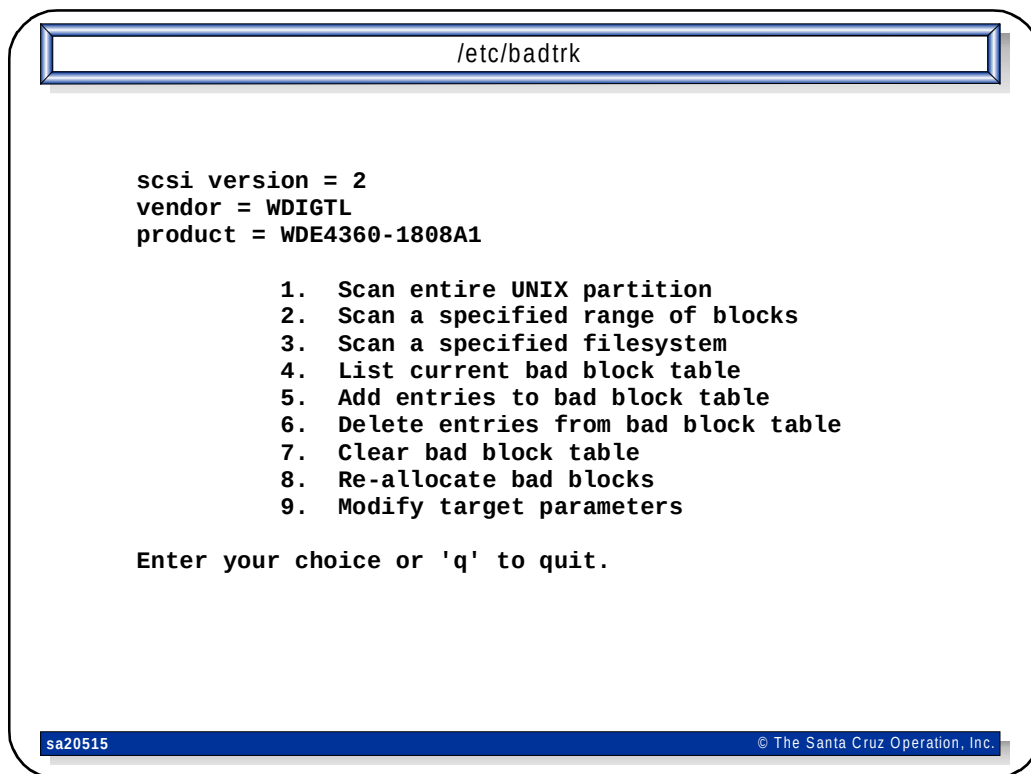
Step 28. Enter the option(s) indicated by your instructor.

The system runs **fdisk**(ADM).

AV 5-14



AV 5-15



- Step 29. Enter **1**, to “Continue with hard disk initialization.” The system runs **fdisk**(ADM).
- Step 30. Examine and change disk partitions as per your installation checklist.
- Step 31. Select a bad track or bad block strategy as indicated by your instructor.

AV 5-16

**Swap and Boot Space Allocation**

There are 4176900 1K blocks in the UNIX area. Between 16000 and 512000 1K blocks should be reserved for the swap area.

Please enter the swap-space allocation, or press <Return> to get the default allocation of 98000 1K blocks:

There are 4176900 1K blocks in the UNIX area.  
Between 15000 and 64000 1K blocks should be reserved for the boot filesystem.

Please enter the boot-space allocation, or press <Return> to get the default allocation of 20000 1K blocks:

sa20516© The Santa Cruz Operation, Inc.

AV 5-17

**Filesystem Creation**

Do you want a separate /u filesystem? (y/n)  
Enter allocation of 1K blocks for the /u filesystem (80000 to 3745546) 80000

The layout of the filesystems and swap area is now prepared.

Do you wish to make any manual adjustments to the sizes or names of the filesystems or swap area before they are created on the hard disk? (y/n) y

sa20517© The Santa Cruz Operation, Inc.

- Step 32. Enter the swap space allocation or accept the default as indicated by your instructor.
- Step 33. Enter the boot space allocation or accept the default as indicated by your instructor.
- Step 34. Enter **y** to create a separate `/u` filesystem.
- Step 35. Enter the `/u` filesystem allocation as indicated by your instructor.
- Step 36. Enter **y** to make manual adjustments to the divisions of the filesystem. The system runs **divvy**(ADM).

## AV 5-18

| divvy Table (continued below) |            |        |   |             |            |
|-------------------------------|------------|--------|---|-------------|------------|
| Name                          | Type       | New FS | # | First block | Last block |
| boot                          | EAFS       | yes    | 0 | 0           | 14999      |
| swap                          | NON FS     | no     | 1 | 15000       | 29999      |
| root                          | HTFS       | yes    | 2 | 30000       | 1023988    |
|                               | NOT USED   |        | 3 | -           | -          |
|                               | NOT USED   |        | 4 | -           | -          |
|                               | NOT USED   |        | 5 | -           | -          |
| recover                       | NON FS     | no     | 6 | 1023989     | 1023998    |
| d1057all                      | WHOLE DISK | no     | 7 | 0           | 1025006    |

sa20518 © The Santa Cruz Operation, Inc.

## AV 5-19

| divvy Table (continued)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <p>1023999 1K blocks for divisions, 1008 1K blocks reserved for the system.</p> <p>n[ame]            Name or rename a division<br/> c[reate]          Create a new filesystem on this division<br/> t[ype]            Select or change filesystem type on new filesystem<br/> p[revent]        Prevent a new filesystem from being created on this division<br/> s[start]          Start a division on a different block<br/> r[estore]        Restore the original division table</p> <p>Enter your choice or q to quit</p> |  |

sa20519 © The Santa Cruz Operation, Inc.

Step 37. Examine or alter the **divvy** table as per your installation worksheet.

Refer to the **SCO OpenServer Handbook**, “Adding hard disks,” for more information, specifically, the sections on “Partitioning a hard disk using fdisk” and “Dividing a disk partition into divisions using divvy.”

Step 38. Enter **i** when prompted to install the division setup as shown.

Step 39. Reply to any prompts for mount points.

When complete, the software loading stage of installation begins.

As the installation loads each selected software component onto the hard disk, messages are logged in the:

*/var/opt/K/SCO/SoftMgr/1.0.2a/custom/custom.log* file.

After installation, you can use the **custom(ADM)** command to view the software products and components. For example, to display the details of all tasks in the custom logfile, enter:

**/etc/custom -G all | more**

The software loading stage of the installation requires from one to three hours, depending on your server speed. The rest of the software can load unattended.

At this point, the following happens:

- The kernel is rebuilt according to the configuration of your system and software components. If you used a BTLD, the driver is built into this new kernel.
- The new kernel is named *unix*, and a copy of this new kernel is created and is stored in the */stand* filesystem as *unix.safe*. The install kernel is also stored on the */stand* filesystem as *unix.install*.
- The */etc/fixmog* utility runs and “cleans up” the system.
- Messages display that remind you to create your emergency root and boot floppy set and register your software.

Step 40. Remove the your installation media. Press .

Step 41. Press  to reboot your system.

- Step 42. Press  at the **Boot :** prompt. When you see **INIT: SINGLE USER MODE**, enter the **root** password to bring the system up in system maintenance mode (single-user mode).

Messages might appear reminding you to register your SCO OpenServer software.

Some SCO Authorized Educations Centers (AECs) use specialized SCO OpenServer software that does not display registration reminders, so you may or may not see them in class. The SCO OpenServer software you install in your workplace will definitely generate reminders until you register the software.

- Step 43. Bring the system up to multiuser mode. You can now add user accounts, local printers, other hard disks, and so forth.

Additionally, if you installed Connectivity components and deferred the network configuration, you can configure the network anytime. If you plan to use network mail, be sure to configure the mail system as well.

If you left space for additional divisions on your hard disk, you can create them now.



## 5.2 Troubleshooting Problems During Installation

---

This section discusses 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 listed as supported on the SCO HCWP
- Make certain that your server devices are recognized at boot time:
- Watch the boot display
- Check for hardware conflicts between components:
  - DMA
  - Interrupt vectors
  - Memory addresses
- Check for documented incompatibilities in the appropriate documentation

Refer to the ***SCO OpenServer Handbook***, “Troubleshooting the installation,” for more information on how to solve installation problems.

### 5.2.1 System does not boot during installation

---

If the boot prompt does not display, verify that:

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

You can run the manufacturer’s floppy drive diagnostic tests to determine that the floppy drive hardware is functioning correctly.

Use these guidelines when troubleshooting boot disks and boot floppy drive problems:

- If the floppy drive works, you might have a damaged boot disk. Try booting the boot disk on another machine.

- If an error occurs while reading a 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, recreate a new boot disk from your installation CD-ROM. Follow the steps outlined in your **SCO OpenServer Release Notes**, Chapter 1, “Creating installation media using **ezcp**,” “Creating installation media - UNIX,” and “Creating installation media - DOS.”

### 5.2.2 System cannot find the hard disk

---

If the installation program is unable to locate your primary hard disk, it may be because the drivers for the hard disk are not in the software distribution, and you require BTLDS for the hard disk. Check the manufacturer’s documentation for the required drivers.

Use these guidelines when troubleshooting this problem:

- Restart the installation with the appropriate **link** statement and package name and reinstall the drivers.
- Check and confirm the target ID if a primary SCSI hard disk is installed. Refer to the manufacturer’s documentation for specifics.
- Restart the installation. Enter the non-default values for your hard disk.

### 5.2.3 System cannot find CD-ROM

---

If the installation program is unable to locate your SCSI CD-ROM, verify the correct target ID:

- Refer to the manufacturer’s documentation and determine the target ID setting of the CD-ROM.
- Refer to the manufacturer’s documentation and determine that the CD-ROM drive is terminated properly in relation to any other SCSI devices that are on the SCSI bus.

### 5.2.4 Screen blanks after Boot : prompt

---

If you can see a display while the server goes through its self-test, but the screen blanks after the kernel is loaded, you might be using an:

- Improperly configured EGA or VGA adapter
- Unsupported EGA or VGA adapter

Confirm that the jumpers on the video card are set correctly, and that the card is not set in emulation mode. Try experimenting with VGA, EGA, CGA, and MONOCHROME switches.

### 5.2.5 System hangs at hardware configuration screen

---

When you start the installation from the boot disk, the system first displays information about your hardware configuration, then a series of diagnostic messages. If the process stops and does not continue, this usually indicates a problem with the initialization of the corresponding device driver.

Refer to the *SCO OpenServer Handbook*, Appendix G, “Kernel initialization check letters,” for more information about startup messages.

#### **Exercise: Defining kernel initialization check letters**

---

---

Turn to Appendix G, “Kernel initialization check letters,” in the *SCO OpenServer Handbook*, read the introductory paragraph, and examine the contents of the chapter before continuing with the rest of this lesson.

Common startup diagnostic messages include:

E            print information about the numeric coprocessor

If the system hangs here, the numeric coprocessor is either missing or not located.

F6           floppy disk driver

If the system hangs here, the floppy disk driver is not found.

F25          Adaptec SCSI host adapter driver

If the system hangs here, the SCSI adapter target ID 0 is not found.

H7           mount the root filesystem

If the system hangs here, the problem is likely to be a damaged */dev/root* filesystem.

## 5.3 Troubleshooting Problems After Installation

---

This lesson discusses some of the common problems that might occur after you install the operating system.

### 5.3.1 System does not boot

---

If the following message displays:

```
PANIC: init
```

during the initial hard disk boot, it can mean that the boot block is written on a bad track or block. Regardless of the reason, reinstall your SCO OpenServer system software. If supported by your hard drive or SCSI adapter, be sure to scan for bad blocks during the “Preparing your disk and choosing software” screen of the installation setup stage.

Refer to the **scsi**(HW) and **badtrk**(ADM) manual pages for more information about hard drive preparation.

### 5.3.2 The mouse is not working

---

If you deferred mouse configuration during installation, or did not configure your mouse properly, follow these steps to configure your mouse:

- Step 1. Log in as **root**, shut down your system, and enter system maintenance mode.
- Step 2. Enter **mkdev mouse**.
- Step 3. Select “Install a mouse” from the menu.
- Step 4. Select the appropriate mouse type from the options listed. When prompted:
  - For a serial mouse, select the port on which the mouse is installed.
  - For a bus mouse, select the type of mouse you are installing and its parameters. If you want to change the default parameters, be careful when changing the interrupt vector for the mouse.
  - For a keyboard mouse, select the required resolution.

Refer to the *SCO OpenServer Handbook*, “Changing the keyboard mouse resolution setting” and “Solving slow or no response from keyboard mice” for more information about high resolution mice.

- Step 5. Specify the terminals and multiscreens that you want to allow to accept input from the mouse. Do not specify any *tty* ports where mice are actually connected. You can choose to allow any or all other terminals and console multiscreens to use the mouse. By default, the mouse is associated with all console multiscreens.

If you install more than one mouse, make sure the terminal specification for each mouse does not overlap. You can allow only one mouse for input on a given *tty*.

- Step 6. Relink the kernel.

- Step 7. Reboot your system:

**/etc/shutdown -g0 -y**

You can use the **Hardware/Kernel Manager** or **mkdev mouse** at any time to allow or prevent mouse input on different terminals, to remove mice, or to check your current mouse configuration.

### 5.3.3 Graphical **scolgin(XC)** on **tty02** looks strange

---

If the graphical **scolgin(XC)** screen that appears on **tty02** on the console looks unusual—for example, the login window does not fit in the display—you may have incorrectly configured your video hardware during the installation.

The following steps describe how to correct the problem:

- Step 1. Log in as **root**.
- Step 2. Switch to **/dev/tty01** on the console by pressing   .
- Step 3. Run **scoadmin(ADM)**.
- Step 4. Select **Video Configuration Manager**.
- Step 5. Make the corrections to the previous selections.

### 5.3.4 Error message: Can't fork

---

If your system swap space is insufficient, you might see this error message:

Can't fork

Increase the amount of swap space using the **swap -a** command.

Refer to the **swap(ADM)** manual page for more information.

### 5.3.5 System not licensed or unregistered software

---

The next lesson, “Post-installation Administration,” describes how to license and register your software.

### 5.3.6 Hardware does not work

---

If some hardware on the server (for example, a CD-ROM or DAT tape drive) does not work once the SCO OpenServer product has been installed, but works under another operating system, the hardware might not be:

- Configured correctly
- Supported by the product

### 5.3.7 Problems with broken links

---

The filesystem layout accommodates the Software Storage Object (SSO) architecture and files in the root filesystem reside in two hierarchies:

|                       |                                                                                 |
|-----------------------|---------------------------------------------------------------------------------|
| <code>/opt</code>     | Contains read-only versions of all files                                        |
| <code>/var/opt</code> | Contains read/write copies of the configuration files from<br><code>/opt</code> |

The use of the `/opt` and `/var/opt` directories is in compliance with the System V Interface Definition, Issue 3 (SVID 3). For example, the `/etc/mount` file is physically located in `/opt/K/SCO/Unix/5.0.5Eb/etc/mount` and a symbolic link refers to its location in the `/etc` directory.

The standard names for an SSO are:

- `/opt/K/vendorCode/componentCode/version`
- `/var/opt/K/vendorCode/componentCode/version`

You need to be able to identify and fix the links between the new hierarchies and the traditional UNIX structure.

### 5.3.7.1 Identifying broken symbolic links

---

These steps describe how to find broken links:

Step 1. Log in as **root** and change into the `/etc` directory:

```
cd /etc
```

Step 2. List `motd` (**l motd**) and note that it is a symbolic link:

```
-rw-r--r-- root sys 36 Jan 31 20:22 motd@
```

Step 3. For contrast, view the symbolic link of `motd`. Enter:  
**l -wv** command:

```
-rw-r--r- 1 root sys 36 Jan 31 20:22  
motd-> /var/opt/K/SCO/Unix/5.0.5Eb/etc/motd
```

Step 4. Rename `motd` as `old_motd`:

```
mv motd old_motd
```

Step 5. Use **vi** and create a new `motd`.

Step 6. When done, list `motd` and note that the symbolic link is now broken:

```
l motd  
  
-rw-r--r-- 1 root sys 36 Apr 4 09:30  
motd
```



### 5.3.7.2 Fixing broken symbolic links

Use the **Software Manager** to find and correct single or multiple broken links:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(ADM)**.
- Step 3. Select **System Administration** ⇨ **Software Manager**.
- Step 4. Select **Software** ⇨ **Verify Software**.

The **Verify Software** option checks the symbolic links for a specific software package. The **Verify System** option checks the symbolic links for the entire system.

- Step 5. Select to check for **Broken/missing symbolic links**.
- Step 6. Select **Fix Discrepancies**. The following message appears:

This process will restore all broken or missing links found.

- Step 7. Select **Continue**.
- Step 8. Click on **OK** when complete.
- Step 9. Select **Done**.
- Step 10. Select **Host** ⇨ **Exit** to exit the **Software Manager**.

#### Exercise: Fixing broken symbolic links

Break the symbolic link to `/etc/motd` and use the **Software Manager** to restore it.

## 5.4 Lesson Summary

---

You can stop the installation while in the setup stage by pressing **F2**. When you are in the software loading stage, you stop the installation by powering the system off and on.

To return to the point of installation when you stopped, you can reboot with the boot media and press **ENTER** at the **Boot :** prompt. However, SCO recommends that if you must restart the installation, you should start over completely by typing **restart** (and the **link** argument, if appropriate) at the **Boot :** prompt.

You can use Boot Time Loadable Drivers (BTLDs) to install hardware drivers that are not currently part of the install kernel.

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 server 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 installation

If you suspect you have defective installation media, try the media on another, similar server. Replace it if necessary.

If you have an installation problem, follow these guidelines:

- Determine whether the problem is with the hardware or software. If the problem is software-related, the installation configuration itself may be incorrect.
- Verify that your server peripherals are set to the values (IRQ, Base RAM, DMA) that you used during their installation.
- Verify the video configuration. An incorrect monitor setup can create the appearance of complete failure on the server.

## Review questions

---

1. What are the two stages of installation? What occurs during each stage?
2. How can you stop the installation during the setup stage?
3. How do you restart your installation?
4. List two problems that can occur during installation and outline the steps you would take to correct them.
5. List two problems that can occur immediately after installation and outline the steps you would take to correct them.

## 5.5 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                       |                                                                                                                                   |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>hwconfig</b> (HW)  | Displays information about your system's hardware                                                                                 |
| <b>link</b> (HW)      | Links and edits a new driver into the kernel after it has been loaded, but before it runs                                         |
| <b>scoadmin</b> (ADM) | Invokes SCOadmin applications (in this lesson, the <b>Software Manager</b> and the <b>Video Configuration Manager</b> )           |
| <b>swap</b> (ADM)     | Adds, deletes, and monitors the system swap area used by memory; with the <b>-a</b> option, increases the size of the swap device |

### Relevant files

---

|                     |                                                                          |
|---------------------|--------------------------------------------------------------------------|
| <i>boot</i>         | The UNIX boot program                                                    |
| <i>custom.log</i>   | The log file where installation messages are stored for future reference |
| <i>unix</i>         | The UNIX kernel that is created during the installation                  |
| <i>unix.install</i> | The copy of the install kernel which is saved on the root filesystem     |
| <i>unix.safe</i>    | A copy of the previous kernel file                                       |

## 5.6 Answers to Review Questions

---

This section provides suggested answers to the review questions.

1. The two stages of installation are setup and software loading. During setup, the installation program prompts you with questions about how the server hardware is configured, which software components are to be installed, and the disk division layout.

During software loading, the installation program installs and configures your server using the parameters that you defined in the setup stage.

2. You can stop the installation during the setup stage by pressing <F2> and answering to stop the installation.
3. You can restart your installation by inserting the Boot Disk in the drive and entering “restart” or pressing <Enter> at the **Boot :** prompt.
4. Problems that can occur during installation include: system does not boot—verify the correct floppy disk is being used; system does not boot—verify that the floppy disk is good by trying it on another server; system can not find hard disk—restart, use correct BTLID, use correct SCSI parameters; system can not find CD-ROM—check target ID, termination; screen blanks after **Boot :** prompt—unsupported video adapter, and so forth.
5. Problems that can occur immediately after installation include: graphical scologin on tty02 looks strange—correct video adapter installation using CHARM (scoadmin hardware); can’t fork: insufficient swap space—increase the size of swap; symbolic link are broken—use the **Software Manager** to find and correct single or multiple broken links; Select **System Administration =>Software Manager**; select either **Software =>Verify Software** or **Software =>Verify System**; select to check for **Broken/missing symbolic links**, select **Fix Discrepancies**.

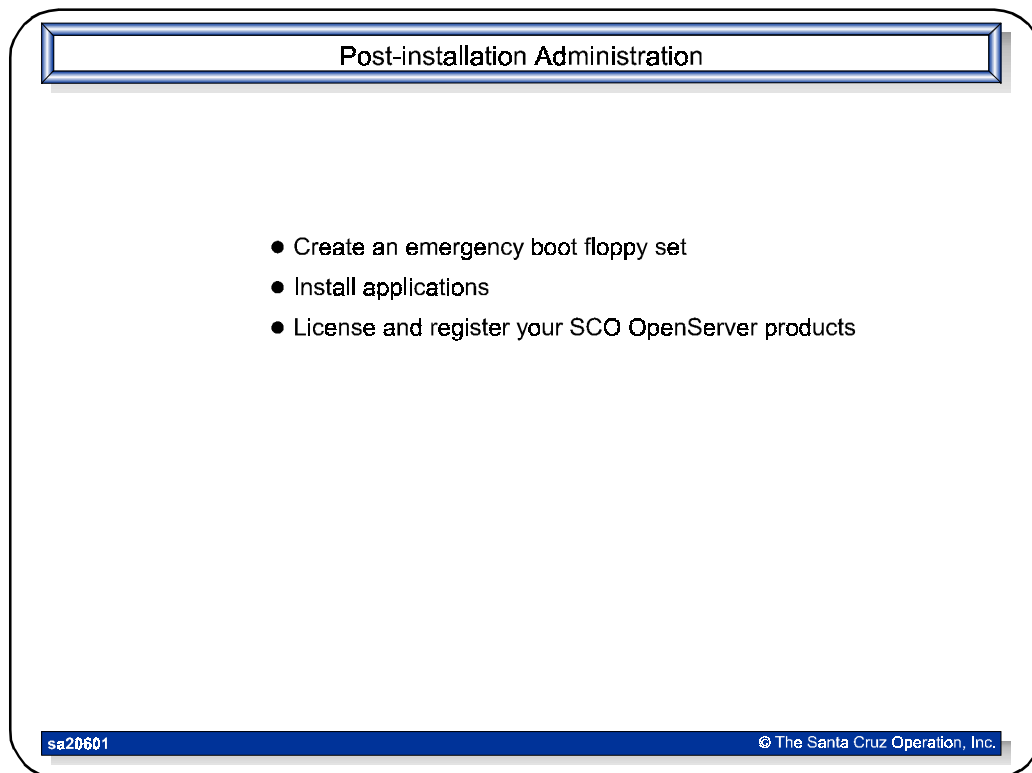


---

# 6

## *Post-installation Administration*

AV 6-1





**Lesson 6**

## **Post-installation Administration**

### **Introduction**

---

Some of the activities that you perform immediately after installation are among the most critical of all.

Having an emergency boot floppy set can later help you solve system problems without taking the time and trouble to reinstall the complete product.

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

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

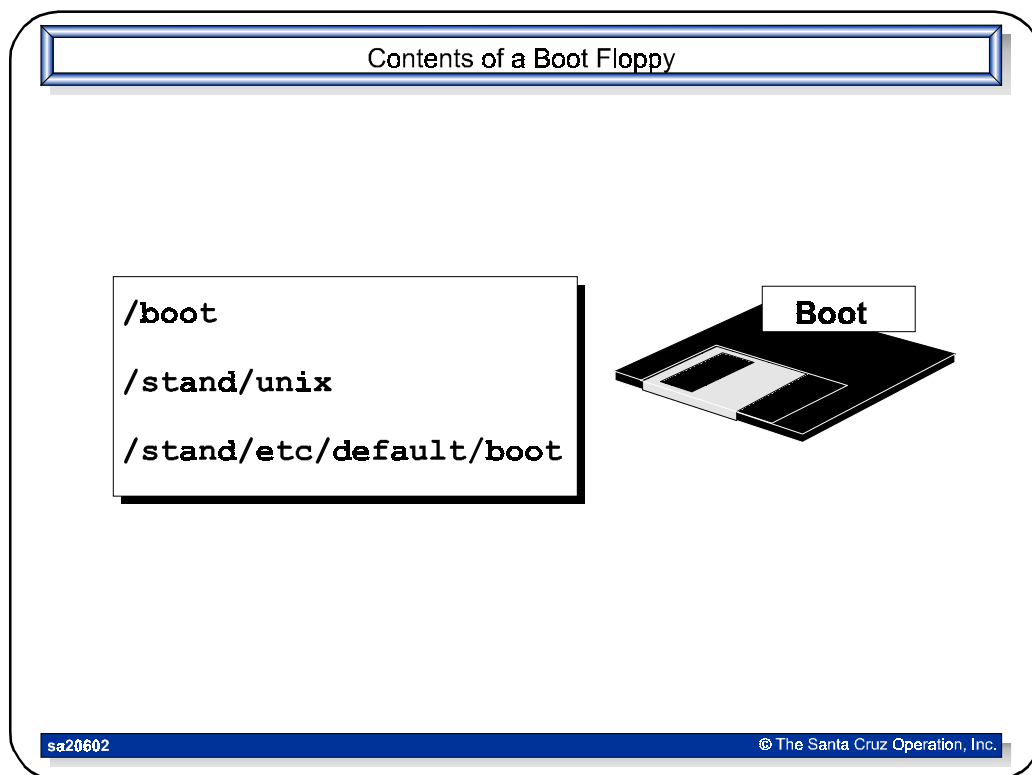
### **Objectives**

---

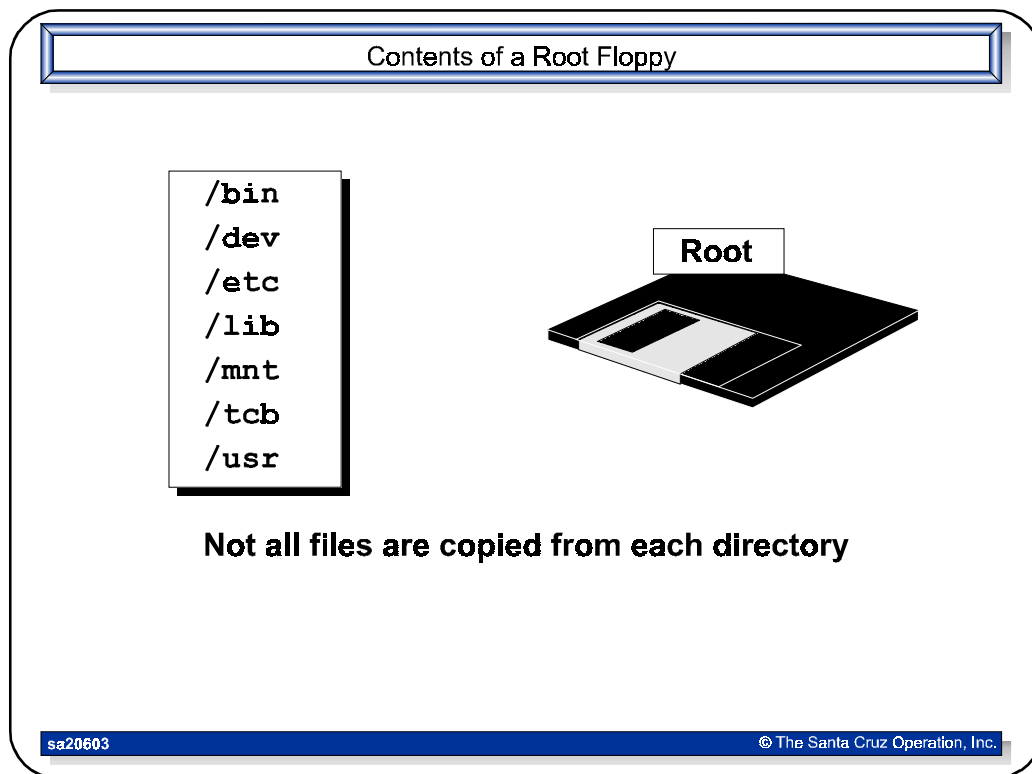
The purpose of this lesson is to identify and perform critical post-installation activities. By the end of this lesson, you will be able to:

- Install applications
- License and register your SCO OpenServer products

AV 6-2



AV 6-3



## **6.1 Creating an Emergency Boot Floppy Set**

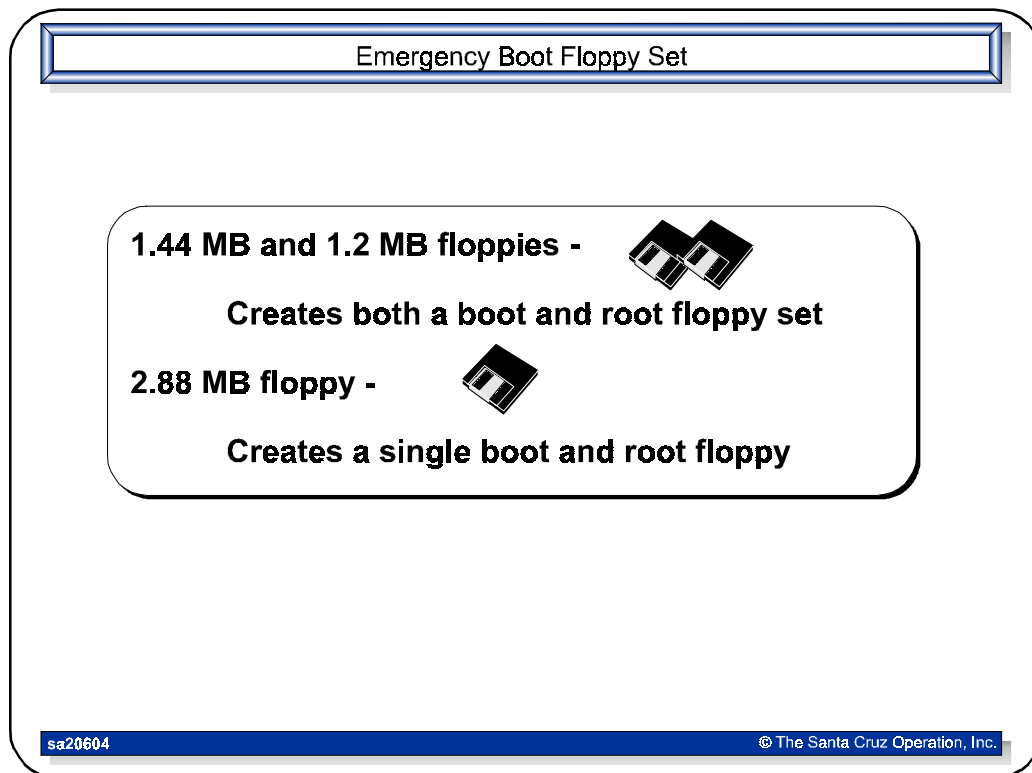
---

The emergency boot floppy set is often referred to as “boot and root floppy disks.” These floppy disks allow you to recover your system in the event of a catastrophic system failure, for example, when the server does not respond to your attempts to start it. You can use the floppy disks to restore a corrupted root filesystem without reinstalling.

After installing the SCO OpenServer software on your system, you should:

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

AV 6-4



### 6.1.1 Requirements for creating a boot/root floppy set

---

You can create the emergency boot floppy set on the following media:

- Two 1.2MB 5.25-inch (96tpi, 15 sectors per track) floppy disks
- Two 1.44MB 3.5-inch (135tpi, 18 sectors per track) floppy disks
- One 2.88MB 3.5-inch (135tpi, 36 sectors per track) floppy disk

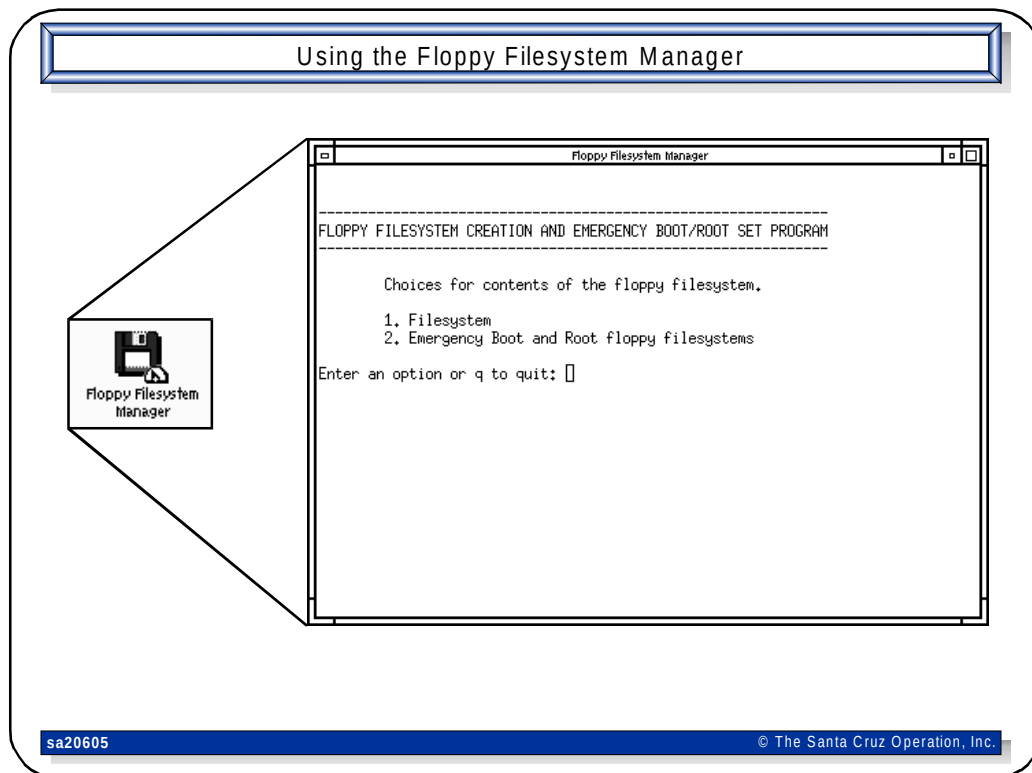
The root floppy disk creates a RAM disk that requires 2MB of contiguous memory. When there is insufficient contiguous memory, you see the message:

```
/dev/ramfs: cannot open
```

If you receive this error message, shut down the system to single-user mode and create the emergency boot floppy set. If this does not resolve the error message, add more memory to your system.

For additional information, refer to the *SCO OpenServer Handbook*, “Creating an emergency boot floppy set.”

AV 6-5



## 6.1.2 Using the Floppy Filesystem Manager

---

These steps describe how to create an emergency boot/root floppy set using the **Floppy Filesystem Manager** (note that you can also use **mkdev fd** from the command line):

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(ADM)**.
- Step 3. Select **Filesystems** ⇨ **Floppy Filesystem Manager**.
- Step 4. Select option **2**, Emergency boot and root floppy filesystems.
- Step 5. Select the correct density of the floppy disk you are using:
- 1** for 96tpi (1.2MB)
  - 2** for 135tpi (1.44MB)
  - 3** for 135tpi (2.88MB)
- Step 6. Select **1** to create the root filesystem disk; select **3** to create a single boot/root disk.
- Step 7. Insert a blank floppy disk when prompted. If the floppy disk is already formatted, enter **n**; if not, enter **y**.
- If you are using an unformatted floppy disk, the track and head numbers being counted display as the floppy disk formats. (If the `/etc/default/format` file contains the string `VERIFY=Y`, the format is also verified.)
- The following message appears:
- Do you want a shell escape to place extra files on the root filesystem? (y/n)
- Step 8. Enter **y** to place additional files on the disk, such as `/usr/bin/vi` and `/usr/lib/terminfo/a/ansi`. The `<bootroot>` prompt appears.
- Step 9. Copy the necessary files to the disk. For example, to copy `/usr/bin/vi` to the disk, enter:
- ```
cp /usr/bin/vi /mnt/usr/bin

mkdir -p /mount/usr/lib/terminfo/a

cp /usr/lib/terminfo/a/ansi /mnt/usr/lib/terminfo/a
```

- Step 10. Enter **exit 0** when you are done copying additional files to the disk. The following message appears:

Compressing ramdisk image ...

When complete, the following message appears:

Root filesystem floppy creation complete.

The system then checks the filesystem using **fsck**(ADM).

- Step 11. Press  when prompted (after the floppy is successfully checked).

- Step 12. Remove the floppy disk from the drive and label it with “root” and the machine name. Write-protect it so that you do not accidentally erase information.

- Step 13. Press  to continue.

- Step 14. Enter **2** to create a boot disk.

You have two options for creating the boot disk:

- Create a system-specific disk that contains the same kernel that is currently on your system; that is, you can use this floppy disk to boot the machine it was created on.
- Create a “generic” kernel; that is, you can use this floppy disk to boot any SCO OpenServer machine.

Note that the generic kernel might not contain certain device drivers, and this can cause problems. For example, if your tape drive requires you to install a special driver, it will not work with the generic kernel.

- Step 15. Insert the floppy disk and press  to continue.

- Step 16. Respond to the prompts as you did for creating the root disk.

The boot disk is created, and you see messages as files are copied to the disk. Then, as before, the filesystem is checked using **fsck**(ADM).



### **Exercise: Creating an emergency boot floppy set**

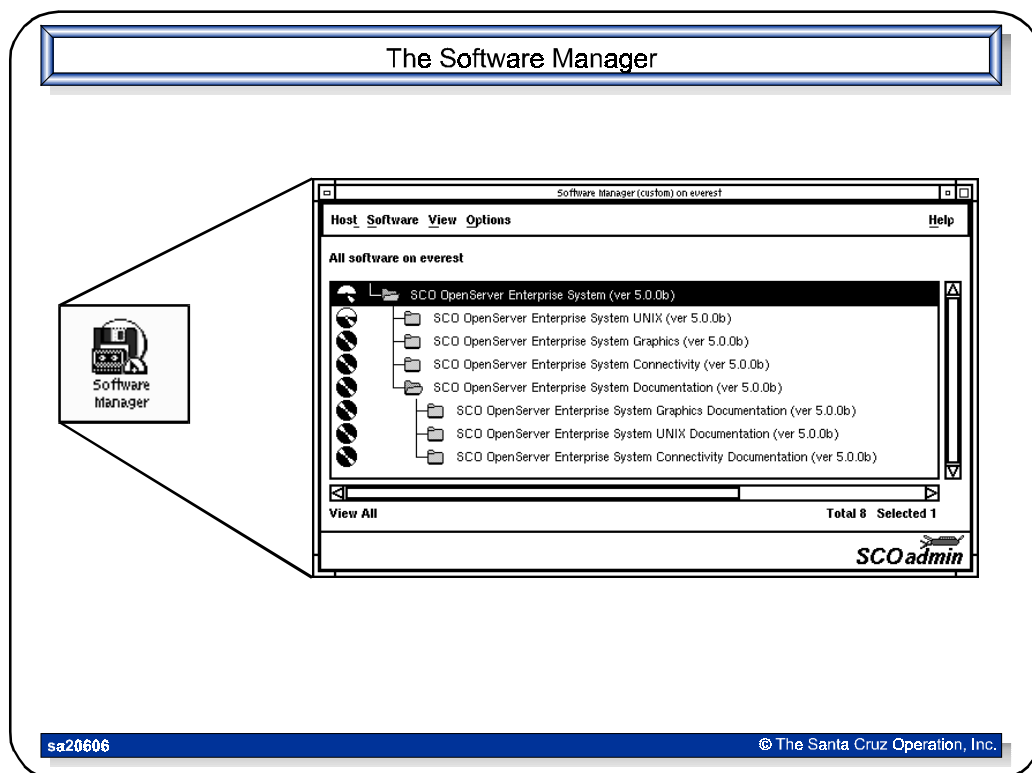
---

Use the steps outlined in “Using the **Floppy Filesystem Manager**” to create the emergency boot floppy disks, either system-specific or generic as indicated by your instructor.

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

Remember to update your system log book.

AV 6-6



## 6.2 Installing Applications

---

Use the **Software Manager** to install applications. Some software requires licensing as you install it. For other software, such as SCO Merge™, you can defer licensing but must complete it later using the **License Manager**.

Product licensing and registration are covered later in this lesson.

### 6.2.1 Installing the online documentation

---

When you installed the operating system in the previous lesson, you did not install the online documentation. These steps describe how to install the online documentation:

Step 1. Login as **root** and verify that you are in single-user mode:

```
who -r
. run-level S June2 14:31 S 1 1
```

If you are not in single-user mode, enter **init 1**, then enter the **root** password when prompted to enter System Maintenance Mode.

Step 2. Run **scoadmin(ADM)**.

Step 3. Select **Software Manager** ⇒ **Software**.

Step 4. Select **Install New**.

Step 5. Choose to install from your own server (do not select the option “another host”). Click on **Continue**.

Step 6. Insert the CD-ROM.

Step 7. Select “CD-ROM Drive 0.” Click on **Continue**. The system examines the media and displays a list of products that you can install.

Step 8. Select “OpenServer Enterprise System (ver 5.0.5m).”

Step 9. Select **OpenServer Enterprise System Documentation**. If desired, you can click on **Expand** or **Expand Fully** to view the documentation components.

Step 10. Select all documentation components, then click on **Install**. The **Installation Progress** window displays the filenames as they are copied from the installation media.

- Step 11. Select **Host** ⇒ **Exit** when installation is complete to exit the **Software Manager**.

### **Exercise: Installing the online documentation**

---

Use the steps outlined in “Installing the online documentation” to install the online documentation on your classroom server.

## **6.2.2 Installing SCO Merge**

---

These steps describe the installation and licensing of SCO Merge from the SCO OpenServer Optional Services CD-ROM.

- Step 1. Log in as **root** and verify that you are in single-user mode:

```
who -r  
. run-level S June2 14:31 S 1 1
```

If you are not in single-user mode, enter **init 1**, then enter the **root** password when prompted to enter System Maintenance Mode.

- Step 2. Run **scoadmin(ADM)**.
- Step 3. Select **Software Manager** ⇒ **Software** ⇒ **Install New**.
- Step 4. Choose to install from your own server (do not select the option “another host”).
- Step 5. Insert the Optional Services CD-ROM.
- Step 6. Select the media device, then select **Continue**. The system examines the disk and displays a list of products that you can install.
- Step 7. Select **SCO Merge**.
- Step 8. Click on **Install**. After several files are copied, the **License Manager** appears.

- Step 9. Enter the license number, license code, and license data from the End User License Agreement (EULA). Note that if you do not correctly enter the information, the product is not usable.

When licensing is complete, the system begins copying files from the installation media. The **Installation Progress** window displays the filenames as they are copied.

- Step 10. Click on **OK**. Remember to reboot your system to make the new kernel effective.

- Step 11. Select **Host** ⇒ **Exit** when installation is complete to exit the **Software Manager**.

- Step 12. Select **File** ⇒ **Exit** to exit SCAdmin.

### **Exercise: Installing SCO Merge software**

---

Follow steps outlined in “Installing SCO Merge” to install SCO Merge software on the classroom server.

AV 6-7

| Licensing and Registration Summary |                                                                   |                                   |                             |
|------------------------------------|-------------------------------------------------------------------|-----------------------------------|-----------------------------|
| Item                               | Where to find                                                     | When you need                     | Where you enter the item    |
| License Number                     | Certificate of License and Authenticity (with software)           | For licensing during installation | Installation setup phase    |
| License Code                       | Certificate of License and Authenticity (with software)           | For licensing during installation | Installation setup phase    |
| License Data                       | Certificate of License and Authenticity (with software; optional) | For licensing during installation | Installation setup phase    |
| System ID                          | License Manager (generates with install)                          | For registering                   | License Manager, "Register" |
| Registration Key                   | SCO Registration Center (given when you call in)                  | For registering                   | License Manager, "Register" |

sa20607 © The Santa Cruz Operation, Inc.

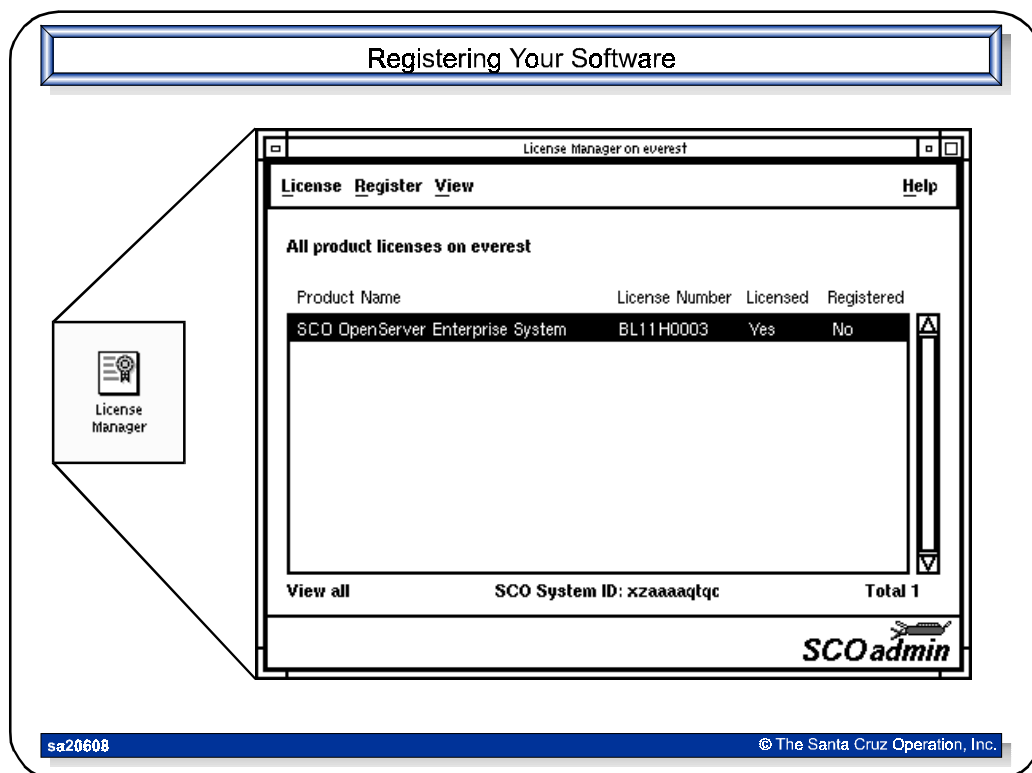
## **6.3 Licensing and Registering Your OpenServer Products**

---

Licensing and registering your operating system software consists of these activities:

- Licensing the software
- Registering the software, which consists of:
  - Locating the System ID and obtaining the Registration Key
  - Using the **License Manager** to register the software

AV 6-8





### 6.3.1 Licensing the software

---

Some software, such as the base operating system, must be licensed before installation can occur. Other software that is installed but has not yet been licensed (such as additional user licenses for the SCO OpenServer Enterprise distribution) must be licensed using the **License Manager**.

The three terms on the End User License Agreement 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, but not all. It is a descriptor that further defines custom configurations. This field is not always present, and in those cases, is not required. |

You can view licensing information for an SCO OpenServer product using the **License Manager**.

These steps describe how to license software that is not licensed during installation:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(ADM)**.
- Step 3. Select **License Manager** ⇨ **License Product**.
- Step 4. Enter the license number, license code, and license data (if applicable).
- Step 5. Select **License**. You can now add additional licenses.
- Step 6. Select **Close** to exit the **License Manager**.

### 6.3.2 Registering the software

---

After you have installed and licensed your SCO OpenServer product, you should register it. In addition to keeping you in compliance with your license agreement, this allows you to 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 SCO OpenServer installation. It is generated each time your SCO software is installed. Therefore, if you reinstall your SCO system, a new System ID is generated. The System ID is displayed when you run the <b>License Manager</b> . |
| 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 you register (identified by its license number).                                                                |

### 6.3.2.1 Locating the System ID and obtaining the Registration Key

---

The SCO System ID is listed in the **License Manager**, and is generated automatically when you install.

You must obtain a Registration Key from the SCO Registration Center. Use the instructions in the **SCO Software Registration** booklet that came with your software.

When you contact the SCO Registration Center, you are asked to provide the License Number, License Code, License Data, and System ID for your installation.

### 6.3.2.2 Using the License Manager to register

---

After you receive your Registration Key from the SCO Registration Center, follow these instructions to complete your software registration:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(ADM)**.
- Step 3. Select **License Manager** ⇒ **Register** ⇒ **Register Product**.
- Step 4. Enter the Registration Key at the “**Registration Key**” prompt, then select **Register**.  
  
This registers your software license, and registration reminders no longer display.
- Step 5. Complete the SCO Product Registration Form.

### 6.3.3 Registration reminder messages

---

Until you register your software, registration reminders appear regularly on the console urging you to do so. Over time, these messages become longer and more insistent. After the software is registered, the reminders no longer appear.

These registration messages do not appear in the classroom if the SCO Authorized Education Center is using customized software.

Refer to the **SCO OpenServer Handbook**, “Licensing and registering SCO products,” for more information about managing your software licenses.

#### **Exercise: Registering your software**

---

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

System ID:

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

Registration Key:

### 6.3.4 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.
- After boot, 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 is detected.
- The SCO system, detecting the duplicate, displays a message to `/dev/console` and the `/usr/adm/syslog` file upon receipt of the broadcast.

### 6.3.4.1 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: SCO Merge Release 5.0
Serial Number: SC0123456
Host name: ahab.sco.com
IP address: 132.147.18.45
```

```
Product: SCO Open Server Enterprise System Release 5.0
Serial Number: SC0123456
Host name: ahab.sco.com
IP address: 132.147.18.45
```

### 6.3.5 Copy Protection Policy daemons

---

The Copy Protection Daemon (**cpd**) is launched during boot and runs continuously. If the policy manager daemon (**pmd**—which administers this policy) detects that the **cpd** has been killed or tampered with, it does not grant any new licenses and prints a message to */dev/console* and the *syslog* file. In this case, reboot the system.

The warning message displays if **cpd** is not running:

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

#### 6.3.5.1 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.

## 6.4 Lesson Summary

---

Create and test at least two emergency boot/root floppy sets for each system after you have installed the operating system. Use the **Floppy Filesystem Manager**.

Use the **Software Manager** to install software on your server. Some software is licensed during the installation; other software does not require licensing.

Typically, SCO OpenServer software that must be licensed during the installation must also be registered with the SCO Registration Center. Use the **License Manager** to do so.

The Copy Protection daemon (**cpd**) and the Policy Manager daemon (**pmd**) monitor the network for duplicate server serial numbers.

## Review questions

---

1. Outline the procedure for creating an emergency boot floppy set.
2. Briefly describe the process for installing applications with the **Software Manager**.
3. What steps are required for licensing and registering OpenServer products such as SCO Merge and the operating system?

## 6.5 Relevant Commands and Daemons

---

The following commands and daemons were presented in this lesson.

### 6.5.1 Relevant commands

---

|                       |                                                                                                                                                      |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>mkdev</b> (ADM)    | Calls scripts to add peripheral devices; with the <b>fd</b> argument, creates boot and root and filesystem floppy disks                              |
| <b>scoadmin</b> (ADM) | Invokes SCOadmin applications (in this lesson, the <b>Software Manager</b> , the <b>Floppy Filesystem Manager</b> , and the <b>License Manager</b> ) |

### 6.5.2 Relevant daemons

---

|            |                            |
|------------|----------------------------|
| <b>cpd</b> | The copy protection daemon |
| <b>pmd</b> | The policy manager daemon  |

## 6.6 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Creating an emergency boot floppy set

---

To view the contents of the root disk, enter:

```
mount /dev/fd0135ds18 /mnt
```

```
cd /mnt
```

```
l
```

You will see a compressed file called *ramdisk.2*.

### Review questions

---

This section provides suggested answers to the review questions.

1. To create a an emergency boot floppy set, go into single-user mode, invoke the **Floppy Filesystem Manager**, or run **mkdev fd**, select the “create the emergency boot and root floppy filesystem” option, select the correct density of the floppy drive you are using, choose “boot,” then “root” (if two floppy disks are required), copy additional files to the disk if desired, and test the disks.
2. To install applications using the **Software Manager**, invoke the **Software Manager**, select **Software => Install New**, choose the appropriate host, insert the installation media, and select the software to install. Once selected, choose “install” to begin. If necessary, license during installation and register afterwards.
3. To license an SCO OpenServer product, invoke the **License Manager**, select **License => License Product**. Enter the appropriate license number, license code, and license data (if applicable).

To register a SCO OpenServer product, invoke the **License Manager** and select **Register => Register Product**. Enter the appropriate Registration Key at the Registration Key prompt, then select **Register**.





---

# 7

## *Managing System Startup and Shutdown*

AV 7-1

**Managing System Startup and Shutdown**

- Describe the default UNIX boot process
- Customize the UNIX boot process
- Use the hardware information displayed at boot time
- Choose the mode of operation
- Trace and customize multiuser initialization
- Troubleshoot system startup
- Analyze system shutdown

sa20701 © The Santa Cruz Operation, Inc.

**Lesson 7**

## **Managing System Startup and Shutdown**

### **Introduction**

---

Starting your SCO OpenServer 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 SCO OpenServer 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:

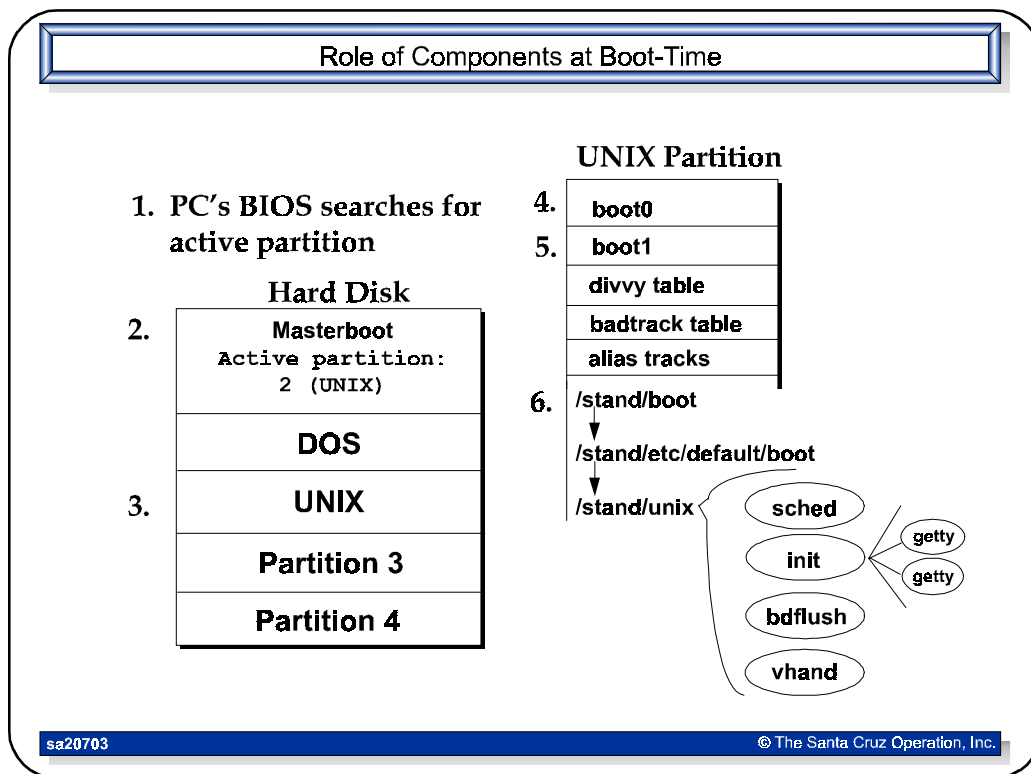
- Describe the default UNIX boot process
- Customize the UNIX boot process
- Use the hardware information displayed at boot time
- Choose the mode of operation
- Trace and customize multiuser initialization
- Troubleshoot system startup
- Analyze system shutdown

AV 7-2

| Bootable UNIX Filesystems - SCO OpenServer |     |    |
|--------------------------------------------|-----|----|
|                                            | Yes | No |
| AFS                                        | X   |    |
| EAFS                                       | X   |    |
| S51K                                       | X   |    |
| XENIX                                      | X   |    |
| HTFS                                       |     | X  |
| DTFS                                       |     | X  |

sa20702 © The Santa Cruz Operation, Inc.

AV 7-3



## 7.1 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**(HW).

The **boot** program is interactive and is used to load and execute the UNIX kernel. It can load and execute any program linked for stand-alone execution. When installed, stand-alone programs such as DOS and Windows NT can also be started from the UNIX boot prompt.

Before booting the UNIX kernel, ensure proper execution of the boot process by following these guidelines:

- The **boot** program must be run from the boot filesystem. The *boot* filesystem, */dev/boot*, is mounted to the */stand* directory.
- The boot filesystem is an EAFS filesystem. The **boot** program can also load files from XENIX, S51K, and AFS filesystems. It does not recognize other filesystem types.

### 7.1.1 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 the hard disk, the boot procedure has six stages:

1. The system BIOS loads the master boot block into memory from track 0, sector 0 of the hard disk.
2. The masterboot block then loads the partition boot block (*boot0*) from sector 0 of the active partition.
3. Assuming the UNIX partition is active, *boot1* is loaded.
4. *boot1* loads */stand/boot* from the boot filesystem.
5. */stand/boot* executes any commands found in *.bootrc* and locates the */stand/etc/default/boot* file.
6. */stand/boot* displays the UNIX boot prompt on the console.

AV 7-4

| UNIX Kernels Residing in <i>/stand</i> |                                                     |
|----------------------------------------|-----------------------------------------------------|
| <b>unix</b>                            | default kernel                                      |
| <b>unix.safe</b>                       | kernel created after first kernel rebuild           |
| <b>unix.old</b>                        | kernel created after any subsequent kernel rebuilds |
| <b>unix.install</b>                    | original kernel from installation diskette          |

sa20704 © The Santa Cruz Operation, Inc.

## 7.1.2 Responding to the UNIX boot prompt

---

During the last stage, the UNIX boot prompt displays:

SCO OpenServer(TM) Release 5

Boot  
:

You can enter other UNIX kernel names at the boot prompt. Three kernels are created during the installation process and reside on the boot filesystem in the */stand* directory:

|                     |                                                                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>unix</i>         | Loaded by default at boot time                                                                                                                                              |
| <i>unix.safe</i>    | Created during the first rebuild of the kernel after installation. It is identical to <i>/unix</i> except that all non-SCO device drivers have been omitted from it.        |
| <i>unix.install</i> | The original kernel from your Boot installation disk. It is used to build your emergency boot floppy set, should <i>/stand/unix</i> be too large to fit on a single floppy. |

A fourth kernel, *unix.old*, exists if any subsequent rebuilds of the kernel take place.

If a problem occurs loading the default UNIX kernel during the boot process, you can boot *unix.safe*, or if it exists, *unix.old*.

**Exercise: Listing the UNIX kernels in */stand*.**

---

---

How many copies of the UNIX kernel are located in the */stand* directory?

### 7.1.3 Loading the UNIX kernel

---

When the system displays the UNIX **Boot :** prompt, do one of the following:

- Press the  key to load the default UNIX kernel
- Enter the name of another UNIX kernel, for example *unix.safe*, *unix.old*, *unix.install*, and so forth
- Enter **unix auto** to boot the */stand/unix* kernel without operator or administrator intervention (that is, without being prompted to enter  d to boot to multiuser or to enter the **root** password to enter system maintenance mode)

If you do not give a response at the UNIX boot prompt, the **boot(HW)** program autoboots */stand/unix*. Instructions are taken from the */stand/etc/default/boot* file.

#### **Exercise: Shut down the system and boot *unix.safe***

---

Shut down your system.

At the UNIX boot prompt, enter **unix.safe** and press . When the system reaches system maintenance mode, enter the **root** password, shut down the system again and boot the default UNIX kernel to multiuser mode.





AV 7-5

| Operating Systems Loaded with <b>bootos</b> (ADM) |                            |
|---------------------------------------------------|----------------------------|
| <b>ccpm</b>                                       | Concurrent CP/M            |
| <b>dos</b>                                        | Matches any DOS filesystem |
| <b>dos_12</b>                                     | DOS 12-bit FAT             |
| <b>dos_16</b>                                     | DOS 16-bit FAT             |
| <b>dos_32</b>                                     | DOS 32-bit FAT             |
| <b>dos_ext</b>                                    | DOS extended partition     |
| <b>nt</b>                                         | NT(TM)                     |
| <b>OS2</b>                                        | OS/2                       |
| <b>OS2_hpfs</b>                                   | OS/2 HPFS                  |
| <b>UNIX</b>                                       | SCO OpenServer System      |
| <b>XENIX</b>                                      | XENIX Operating System     |

sa20705 © The Santa Cruz Operation, Inc.

### 7.1.4 Loading other operating systems

---

The **bootos**(HW) program is a stand-alone program that can be called from **boot**(HW). It loads and runs an operating system from a root disk partition other than the current active partition.

To boot DOS, enter **bootos dos** or **dos** at the UNIX boot prompt.

To determine which operating systems are present, enter **bootos ?** at the UNIX boot prompt.

Refer to **bootos** for more information on booting different operating systems.

#### **Exercise: Determining if other operating systems exist**

---

Shut down your system.

At the UNIX boot prompt, enter:

**bootos ?**

After the partition information displays:

Press  to boot the default UNIX kernel.

Use the **bootos**(HW) online manual pages to determine the syntax to boot the NT operating system.

## 7.2 Customizing the Default Boot Process

---

The **boot(F)** file is used by several programs involved in the startup process, including **boot(HW)** and **init(M)**. Two distinct copies of *boot* exist:

- The “main” copy resides in the root filesystem in */etc/default*.
- The other resides on the boot filesystem in */stand/etc/default*.

This allows programs that run from the *boot* filesystem to be read by */etc/default/boot* before the kernel is loaded. If you want to customize your system’s startup process, edit the */etc/default/boot* file. Do not edit the */stand/etc/default/boot* file. The contents of */etc/default/boot* are written to */stand/etc/default/boot* during a proper shutdown of the system. The changes will be in effect on the next restart of your system.

**Exercise: Comparing the */etc/default/boot* and */stand/etc/default/boot* files**

---

Compare the contents of the */etc/default/boot* file and */stand/etc/default/boot*; are there any differences between these two files?

### 7.2.1 Changing the */etc/default/boot* file

---

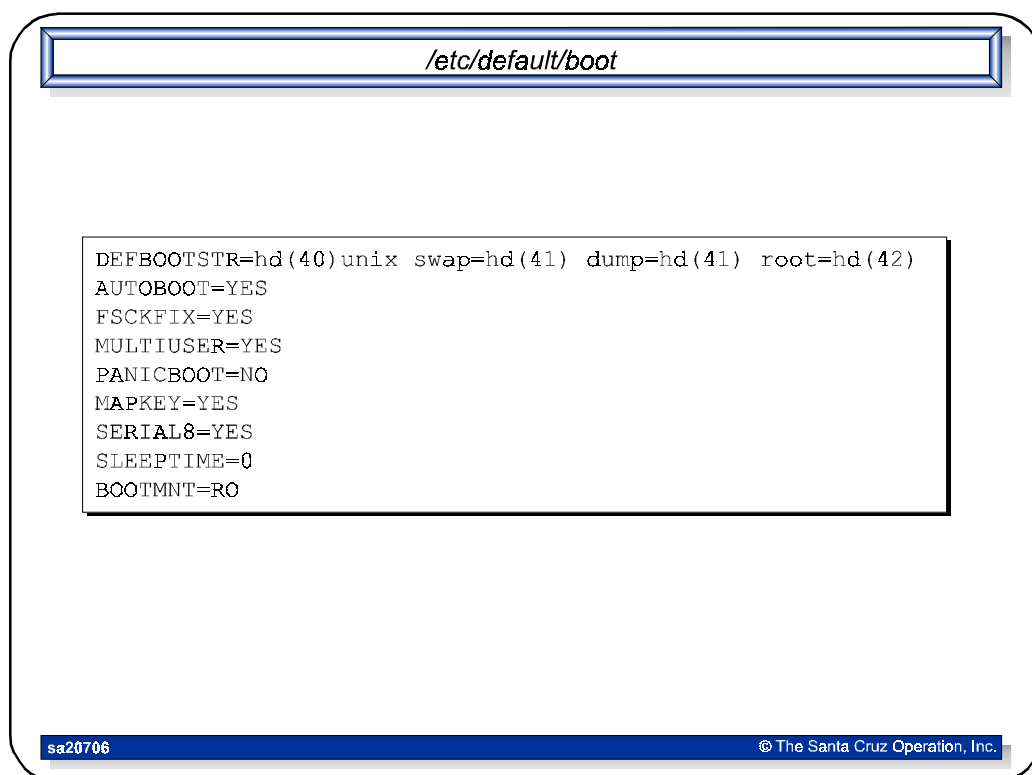
To customize your system startup, you can change the default boot and **init** settings in the */etc/default/boot* file. The **boot**(HW) program recognizes keywords and aliases in the following format:

name=value

For example:

TIMEOUT=10

AV 7-6



### 7.2.1.1 Keywords recognized by boot(HW)

---

The following includes some of the keywords recognized by **boot**(HW):

- **defbootst=***standalone arguments...*

Specifies the program (*standalone*) to run (with optional *arguments*) when either:

- You press only 
- **AUTOBOOT=YES** and you do not enter anything within the time specified by **TIMEOUT**

Bootstrings are discussed later in this lesson in “Bootstring parameters.”

- **AUTOBOOT=YES** or **NO**

If set to YES, The **boot** program automatically loads the UNIX kernel as defined in **DEFBOOTSTR**. If set to NO, you must respond to the **Boot :** prompt by pressing . The default is NO.

- **BOOTMNT=RO** or **RW** or **NO**

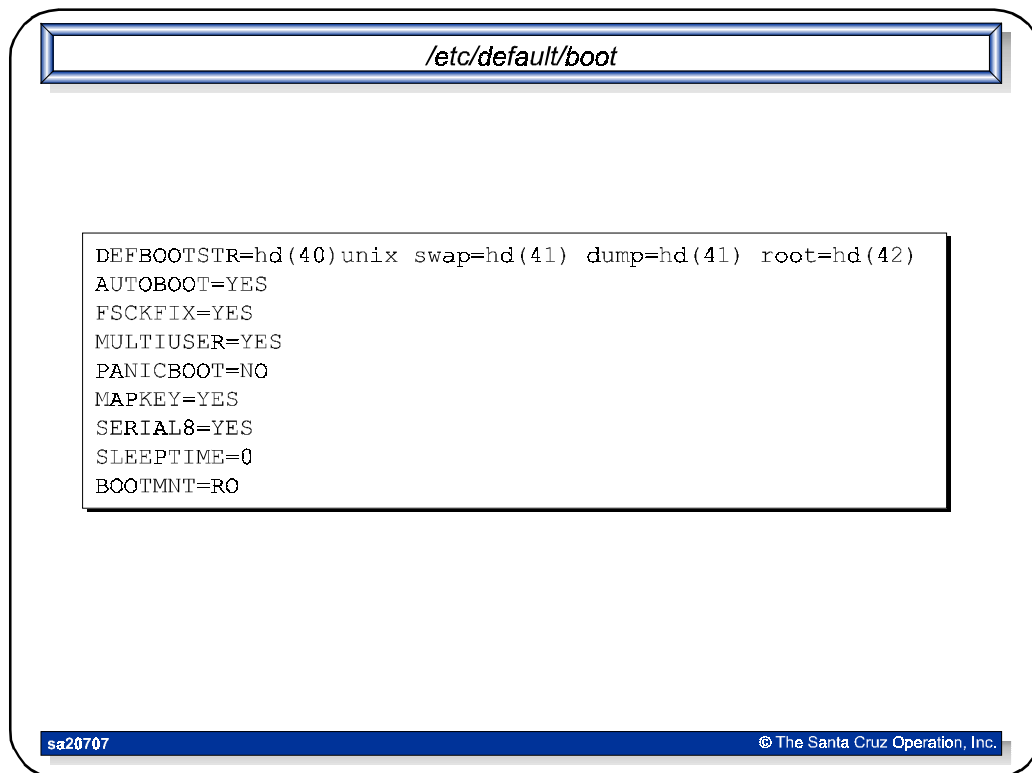
Determines how or if the boot file system is mounted:

- **RO**—Mounts *boot* as read-only
- **RW**—Mounts *boot* as read-write
- **NO**—Does not mount *boot*

- **TIMEOUT=***seconds*

Specifies the number of *seconds* the **boot** program waits before loading the kernel specified by **DEFBOOTSTR**. The default is 60 seconds. If you want to change the default, you must add this parameter to */etc/default/boot*.

AV 7-7





---

## 7.2.2 Keywords recognized by **init(M)**

---

The following includes some keywords recognized by **init(M)**:

- **FSCKFIX=YES** or **NO**

Determines how **fsck(ADM)** repairs root filesystem problems when autobooting is turned on (that is, **AUTOBOOT=YES**). If set to **YES**, **fsck(ADM)** automatically makes repairs (if necessary) to the root filesystem. If set to **NO**, you control the repairs made by **fsck(ADM)**.

- **MAPKEY=YES** or **NO**

If set to **YES**, the console device is set to 8 bits, no parity. When set to **NO**, **init(M)** invokes the **mapkey(M)** program for you to configure the console keyboard mapping.

- **MULTIUSER=YES** or **NO**

When set to **YES**, if you do not intervene during the boot process, the system proceeds to multiuser mode. If set to **NO**, **init(M)** runs **/bin/sulogin** and waits for a password to enter system maintenance mode or you can type **CTRL d** to enter multiuser mode.

- **PANICBOOT=YES** or **NO**

Determines whether or not the system reboots after a panic. By default, this parameter is set to **NO**.

- **SERIAL8=YES** or **NO**

Determines whether or not (**YES** or **NO**) **init(M)** allows the use of 8-bit characters (no parity) on a console configured over a serial line.

- **SLEEPTIME=seconds**

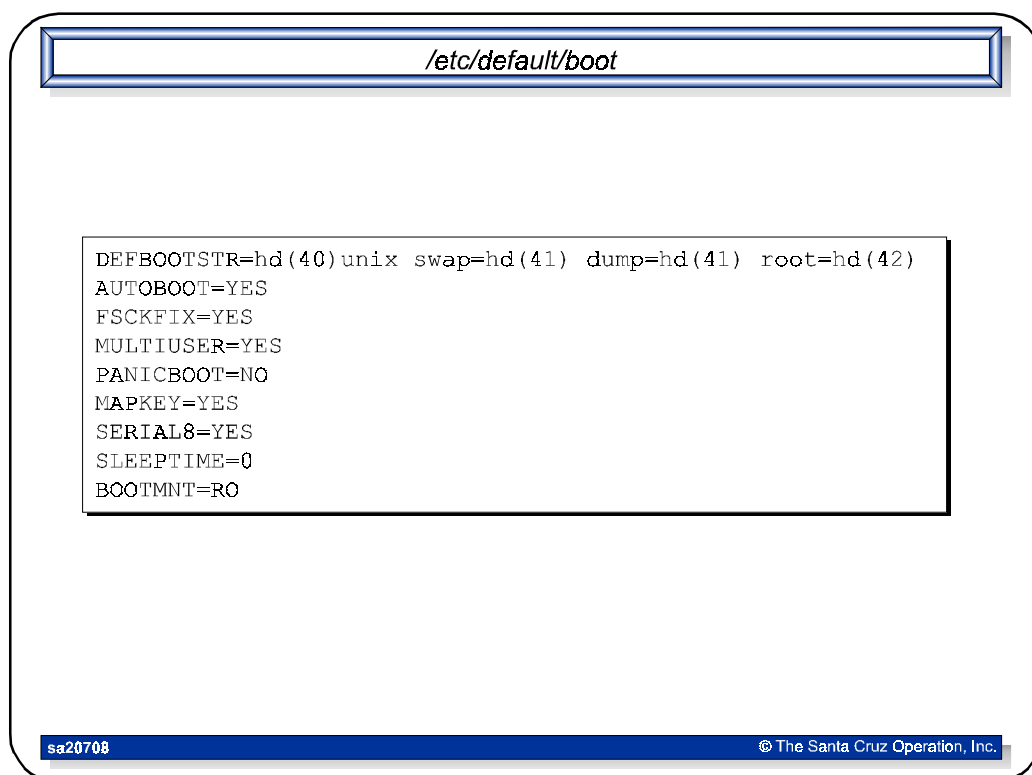
Sets the number of seconds that **init(M)** checks **inittab(F)** for changes. The default is 0, which suppresses the periodic examination.

### Exercise: Changing the TIMEOUT parameter

---

Edit **/etc/default/boot** so that the system will autoboot in 10 seconds.

AV 7-8



## 7.2.3 Bootstrings parameters

---

A *bootstring* is an argument specified on the **boot**(HW) command line. The **boot** program executes some bootstrings internally, then loads a standalone program and passes the bootstring parameters on to the kernel and device drivers.

### 7.2.3.1 The default bootstring

---

The **boot** program uses the following bootstring by default:

```
DEFB00TSTR=hd(40)unix swap=hd(41) dump=hd(41) root=hd(42)
```

The parameters in this bootstring specify the following:

|                          |                                                                                                  |
|--------------------------|--------------------------------------------------------------------------------------------------|
| <code>hd(40)unix</code>  | Load the UNIX kernel from the boot filesystem residing on the first hard disk                    |
| <code>swap=hd(41)</code> | Use the swap space residing on the first hard disk                                               |
| <code>dump=hd(41)</code> | Write an image of memory in the swap space residing on the first hard disk if the system crashes |
| <code>root=hd(42)</code> | Use the root filesystem residing on the first hard disk                                          |

### 7.2.3.2 Recognizing a non-default tape configuration at boot

---

The default configuration recognized for a SCSI cartridge tape is `ha=0`, `id=2`, and `lun=0`. If you have recently added a second tape drive and want the second tape drive to be used as the default, at the `UNIX Boot :` prompt you can append the default bootstring by entering:

**defbootstr Stp=wdha(1,1,0)**

`defbootstr`                      Defined in `/stand/etc/default/boot`

`Stp`                                SCSI tape device

`wdha`                              FUTURE DOMAIN 7000 host adapter as defined in  
`/etc/default/scsihas`

`(1, 1, 0)`                        Where 1= second host adaptor, 1= id, 0 = lun

### 7.2.3.3 Change the swap and dump device

---

To permanently use a swap and dump device from a different hard disk, edit the following DEFB00TSTR entry in */etc/default/boot*:

```
swap=hd(xx) dump=hd(xx)
```

During the current boot, at the UNIX Boot : prompt, enter:

```
defbootstr swap=hd(xx) dump=hd(xx)
```

where *xx* is the minor number for the new swap and dump device.

AV 7-9

#### Kernel Initialization Check Letters (Partial List)

- D - Checks direct memory access**
- E - Initializes numeric processor and multiprocessor memory**
- F - Initializes pseudo and I/O devices (F0-F34)**
  - F1 - audit driver**
  - F4 - serial driver**
  - F6 - floppy disk driver**
  - F34 - Compaq IDA disk driver**
- G - Initializes Programmable Interrupt Controller and multiprocessor**
- H - Initializes various system resources (H0-H13)**
  - H2 - inodes**
  - H4 - file table**
  - H13 - print error messages**
- I - Prints total kernel and user memory**
- J - Initializes floating point emulator**
- K - Opens swap device**
- M - Initializes vendor-specific memory ECC support**

sa20700

© The Santa Cruz Operation, Inc.

## 7.3 Hardware Information Displayed at Boot

---

Each time the kernel starts, it checks the hardware that it expects to find in the system, initializes various kernel tables, mounts the root filesystem, opens the swap device, and prints configuration information.

At each stage of checking, if an error is encountered, the kernel displays a letter (plus an optional number if the stage is subdivided into more checks) for each subsystem, table, and device driver. If the process stops and does not continue at a particular letter, this usually indicates a problem with the initialization of the corresponding device driver.

Refer to the **SCO OpenServer Handbook**, “Appendix G,” for an explanation of letters and optional numbers.

AV 7-10

| Kernel ID                                                  |               |        |     |                                      |
|------------------------------------------------------------|---------------|--------|-----|--------------------------------------|
| device                                                     | address       | vector | dma | comment                              |
| %fpu                                                       | -             | 13     | -   | type=80387                           |
| %serial                                                    | 0x03F8-0x03FF | 4      | -   | unit=0 type=Standard nports=1        |
| %serial                                                    | 0x02F8-0x02FF | 3      | -   | unit=1 type=Standard nports=1        |
| %floppy                                                    | 0x03F2-0x03F7 | 6      | 2   | unit=0 type=135ds18                  |
| %console                                                   | -             | -      | -   | unit=vga type=0 12 screens=68k       |
| %parallel                                                  | 0x0378-0x037A | 7      | -   | unit=0                               |
| %tape                                                      | 0x0300-0x0304 | 5      | 1   | type=wangtek                         |
| %adapter                                                   | 0x0340-0x035F | 11     | -   | type=spad ha=0 id=7                  |
| %smc/wdn                                                   | 0x0240-0x025F | 9      | -   | type=8003EP addr=00:00:c0:a0:92:26   |
| %tape                                                      | -             | -      | -   | type=S ha=0 id=2 lun=0 bus=0 ht=spad |
| %disk                                                      | -             | -      | -   | type=S ha=0 id=0 lun=0 bus=0 ht=spad |
| %Sdsk                                                      | -             | -      | -   | cyls=1002 hds=64 secs=32 fts=b       |
| mem: total = 15996k, kernel = 5104k, user = 10892k         |               |        |     |                                      |
| swapdev = 1/41, swplo = 0, nswap = 30000, swapmem = 15000k |               |        |     |                                      |
| rootdev = 1/42, pipedev = 1/42, dumpdev = 1/41             |               |        |     |                                      |
| kernel: Hz = 100, i/o bufs = 1420k                         |               |        |     |                                      |

sa20710

© The Santa Cruz Operation, Inc.



### 7.3.1 Sample boot display

---

The column headings in a boot display include:

|                          |                                                              |
|--------------------------|--------------------------------------------------------------|
| The <i>device</i> field  | Represents the name of the hardware                          |
| The <i>address</i> field | Displays the device address in hexadecimal                   |
| The <i>vector</i> field  | Represents the interrupt vector used by the device           |
| The <i>dma</i> field     | Displays the direct memory access channel used by the device |
| The <i>comment</i> field | Describes other details about the hardware                   |

### 7.3.2 Recognized Devices

---

The following describes the screen output on the opposite page:

|                       |                                                                                                                                                             |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>fpu</code>      | Floating-point unit present, specifically the Intel 80387 coprocessor or Pentium CPU                                                                        |
| <code>floppy</code>   | High-density 3.5-inch floppy disk drive ( <code>type=135ds18</code> )                                                                                       |
| <code>serial</code>   | COM1 with one port ( <code>nports=1</code> ), no multiport is installed                                                                                     |
| <code>parallel</code> | Parallel port ( <code>lp0</code> )                                                                                                                          |
| <code>console</code>  | Console has a VGA video adapter ( <code>unit=vga</code> ) compatible with (type 0) the IBM VGA design                                                       |
| <code>disk</code>     | Western Digital st506 controller, number 0 ( <code>W0</code> ), hard drive 0 ( <code>unit 0</code> ) as well as the number of cylinders, heads, and sectors |

A disk attached to a SCSI host adapter appears as:

```
name=disk vec=- dma=- type=S ha=0 id=0 lun=0 bus=0 ht=spad
name=Sdisk vec=- dma=- cyls=1002 hds=64 secs=32 fts=b
```

### 7.3.3 Other pertinent information

---

The system displays the following information, including how much memory is installed and the size of the swap space:

|                          |                                                                                                |
|--------------------------|------------------------------------------------------------------------------------------------|
| <code>mem: total</code>  | The total amount of memory available in 1KB blocks                                             |
| <code>kernel</code>      | The amount of memory used by the kernel                                                        |
| <code>user</code>        | The amount of memory remaining for user programs                                               |
| <code>swapdev</code>     | The disk and area of disk being used for swap space                                            |
| <code>swp lo</code>      | Where the swap area begins (default is 0)                                                      |
| <code>nswap</code>       | The size of the swap area in 512-byte blocks                                                   |
| <code>swapmem</code>     | The size of swap in 1KB blocks                                                                 |
| <code>rootdev</code>     | The disk and area of disk used for the root filesystem                                         |
| <code>pipe dev</code>    | The disk and area of disk in which pipes are created                                           |
| <code>dumpdev</code>     | The disk and area of disk to which an image of memory is written during a system crash         |
| <code>Hz</code>          | The frequency of interrupts generated by the clock timer. This value is 100 cycles per second. |
| <code>i/o buffers</code> | The size of the buffer cache in 1KB blocks of RAM                                              |

### 7.3.4 Displaying current hardware configurations

---

The **hwconfig**(C) command displays the hardware configuration as determined at boot time. In addition, you can use the **eisa**(ADM) command to list the cards installed in an EISA machine, and the **slot**(ADM) command to display the cards installed in an MCA machine.

**Exercise: Displaying the current hardware configuration**

---

Execute the following command and use the online manual pages to determine the output:

**hwconfig -h**

The **h** option displays the output in columnar format.

---

### 7.3.5 Logging system messages

---

The **syslogd**(ADM) system daemon is called at system startup to record messages to the following files:

|                          |                                                     |
|--------------------------|-----------------------------------------------------|
| <i>/usr/adm/messages</i> | A copy of the boot screen and kernel error messages |
| <i>/usr/adm/hwconfig</i> | Device initialization messages                      |
| <i>/usr/adm/syslog</i>   | A copy of the boot screen and debugging messages    |

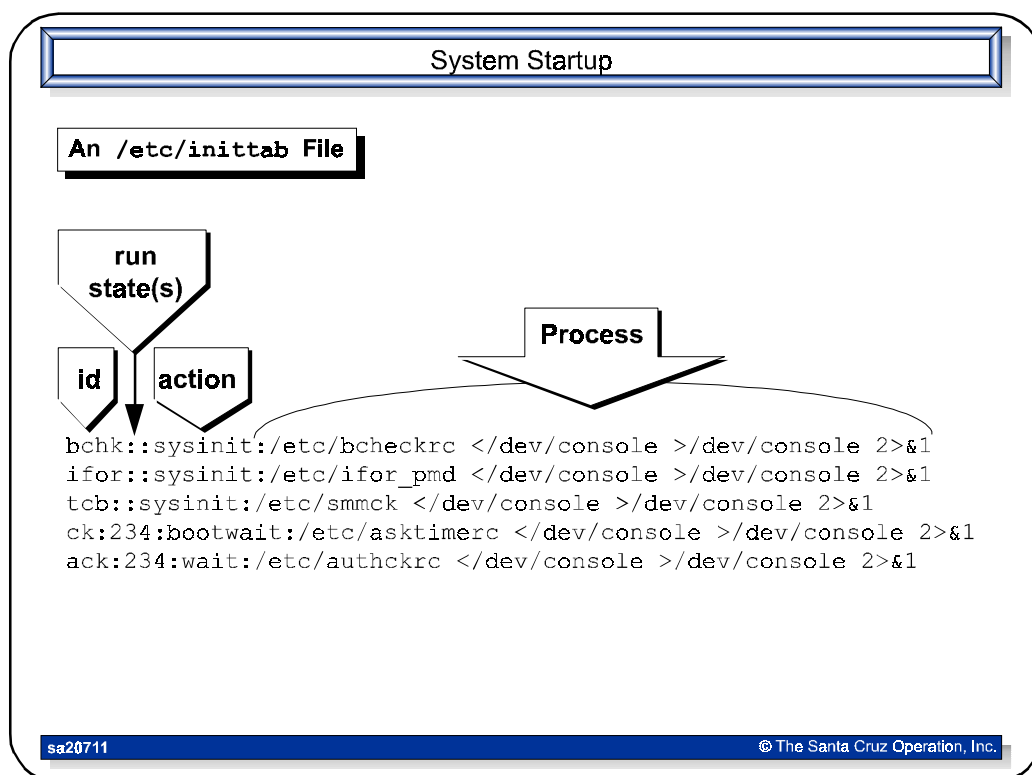
These files are updated when **init**(M) prompts you to choose the mode of operation during system startup.

**Exercise: Viewing the */usr/adm/messages* file**

---

View the contents of */usr/adm/messages* and determine the date and time your system was last booted.

AV 7-11



## 7.4 Choosing the Mode of Operation

---

After **boot**(HW) completes its portion of kernel initialization, the **init**(M) program, which is also responsible for determining the mode of operation and readying terminals for users to log in, continues the initialization.

### 7.4.1 Following the boot process through related files

---

After **init** completes its instructions from */etc/default/boot*, further initialization instructions are taken from */etc/inittab*.

The following section shows the order in which **init** invokes the startup programs with a brief discussion of each element.

Before choosing the mode of operation from the */etc/inittab* file, **init** spawns a Bourne shell to execute the following:

|                      |                                                                                                                                                                                                                   |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>/etc/bcheckrc</i> | Responsible for checking the root filesystem for corruption. If corruption is found, <b>fsck</b> (ADM) is called to repair the filesystem. If the root filesystem is clean, the initialization process continues. |
| <i>/etc/ifor_pmd</i> | Implements the license policy manager. Any SCO application that requires a license checks this daemon to see if it is licensed to run.                                                                            |
| <i>/etc/smmck</i>    | Checks the Trusted Computing Base database in the directories, <i>/etc/auth</i> and <i>/tcb</i> for any missing or corrupted files, and prompts for restoration if necessary                                      |

When prompted to choose the mode of operation, if multiuser mode is selected, **init**(M) spawns a Bourne shell to execute the following:

|                       |                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------|
| <i>/etc/asktimerc</i> | Prompts for, or sets, the system time and date                                                                          |
| <i>/etc/authchkrc</i> | Checks the Trusted Computing Base for any files that may have been restored during the first check by <i>/etc/smmck</i> |

AV 7-12

### Initialization of Run-Levels

#### Extract from `/etc/inittab`

```
r0:056:wait:/etc/rc0 1> /dev/console 2>&1 </dev/console
r1:1:wait:/etc/rc1 1> /dev/console 2>&1 </dev/console
r2:2:wait:/etc/rc2 1> /dev/console 2>&1 </dev/console
r3:3:wait:/etc/rc3 1> /dev/console 2>&1 </dev/console
sd:0:wait:/etc/uadmin 2 0 >/dev/console 2>&1 </dev/console
fw:5:wait:/etc/uadmin 2 2 >/dev/console 2>&1 </dev/console
rb:6:wait:/etc/uadmin 2 1 >/dev/console 2>&1 </dev/console
```

sa20712

© The Santa Cruz Operation, Inc.

## 7.4.2 The */etc/inittab* fields

---

The */etc/inittab* file specifies the programs that **init**(M) 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 consists of four fields:

|                  |                                                                             |
|------------------|-----------------------------------------------------------------------------|
| <i>label</i>     | A unique identification of up to four characters                            |
| <i>run-level</i> | The run-level at which <b>init</b> executes an entry                        |
| <i>action</i>    | A keyword indicating the action to be taken                                 |
| <i>process</i>   | The program that <b>init</b> executes upon entering the specified run-level |

### 7.4.2.1 Run-levels

---

A run-level 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.

The run-levels are:

|         |                                                                                                                                                                                                          |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0       | State of shutdown                                                                                                                                                                                        |
| 1       | Single-user mode; unmounts all filesystems except the <i>boot</i> and <i>root</i> filesystems. All processes spawned by <b>init</b> are terminated.                                                      |
| 2       | Multiuser mode; all filesystems are mounted and all processes are running                                                                                                                                |
| 3 and 4 | Extended multiuser mode. These run-levels are not normally supported under SCO OpenServer systems.                                                                                                       |
| 5 and 6 | Stop the UNIX system, reboot, and display the UNIX boot prompt. When the TIMEOUT parameter in <i>/stand/etc/default/boot</i> is reached, the system auto boots.                                          |
| s and S | Single-user mode; all filesystems remain mounted. All processes spawned by <b>init</b> are terminated. When entered from other than <i>/dev/tty01</i> , run-level s,S also reassigns the system console. |

Refer to the **init**(M) manual page for more information on reassigning the console device.

### 7.4.2.2 Other **init**(M) states

---

The **init**(M) 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.

### 7.4.2.3 Entering run-levels

---

When a 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).

For example, to change to single-user mode, enter:

**init 1**

The **init** program terminates any processes that were started by */etc/inittab* and were running at run-level 2.

If you are using the **init** command to go from multiuser to single-user mode, or to the shutdown state, warnings are not issued to the users.

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

```
wall   
System is shutting down in 15 minutes  
 d
```

#### Exercise: Using the **wall**(ADM) command

---

Send the message, “System is shutting down in 15 minutes” using the **wall**(ADM) command.



You can also use the **shutdown**(ADM) command and specify the new run-level. For example, to enter single-user mode:

**shutdown -g5 -y -i1**

To determine your current run-level, use the **-r** option to the **who**(C) command.

### **Exercise: Determining your current run-level**

---

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

AV 7-13

| inittab(F) Action Definitions                                  |                                                                                                                                     |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>respawn</b>                                                 | If the process does not exist, start the process, and when it dies restart the process.                                             |
| <b>wait</b>                                                    | Upon entering the matching run-level, start the process and wait for its termination.                                               |
| <b>once</b>                                                    | Upon entering the matching run-level, start the process; do not wait for its termination. When it dies, do not restart the process. |
| <b>boot</b>                                                    | The entry is to be processed only at boot-time read of the file.                                                                    |
| <b>bootwait</b>                                                | The entry is to be processed the first time when the system goes from single-user to multiuser state after the system is booted.    |
| <div>sa20713</div> <div>© The Santa Cruz Operation, Inc.</div> |                                                                                                                                     |

AV 7-14

| More inittab(F) Action Definitions                             |                                                                                                                                          |
|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>powerfail</b>                                               | Execute the process associated with this entry only when it receives a power fail signal.                                                |
| <b>powerwait</b>                                               | Execute the process associated with this entry only when it receives a power fail signal and wait until it terminates before continuing. |
| <b>off</b>                                                     | Ignore this process.                                                                                                                     |
| <b>ondemand</b>                                                | This is the same as respawn, used with run-levels a, b, c and e.                                                                         |
| <b>initdefault</b>                                             | This is used to determine the run-level.                                                                                                 |
| <b>sysinit</b>                                                 | Entries are executed before <b>init(M)</b> accesses the console.                                                                         |
| <div>sa20714</div> <div>© The Santa Cruz Operation, Inc.</div> |                                                                                                                                          |

#### 7.4.2.4 Defining the action and process fields

---

The action field in */etc/inittab* further tells **init**(M) how the program executes. The process field tells **init** what program is executed.

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-level 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 mode, 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.

For example, the **getty**(C) program is executed by **init**, resulting in a login prompt appearing on the terminal. When the user logs out, **init** **respawns** another **getty**, resulting in another login prompt on the terminal.

A complete description of the login process is covered in Lesson 9, “Installing and Managing Serial Ports.”

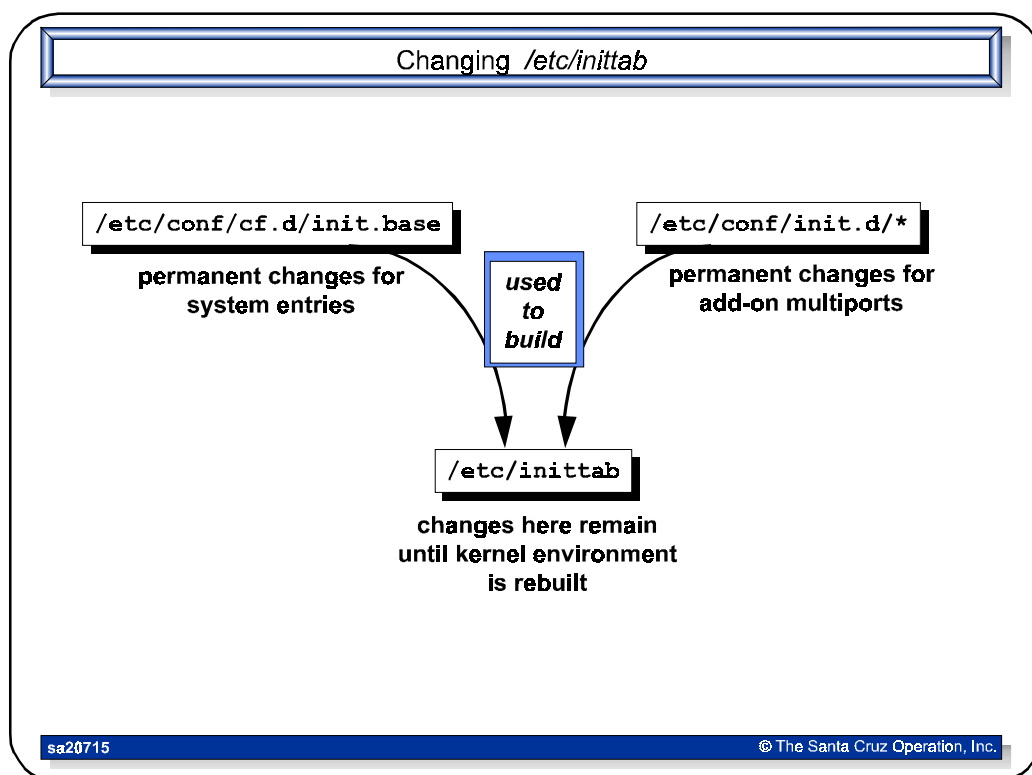
Refer to the **inittab**(F) manual page for a detailed description of action definitions.

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

---

View */etc/inittab* and determine at what run-level the **getty**(C) program is executed on */dev/tty1a*.

AV 7-15



### 7.4.3 Changing the */etc/inittab* file

---

There are times when you will need to make changes to the */etc/inittab* file. For example, if your terminals were initially installed at 9600 and you increase the baud rate to 19200, the pointers in */etc/inittab* are changed to reflect the new baud rate.

When */etc/inittab* is modified, the entries are only temporary, as */etc/inittab* is recreated each time the kernel environment is rebuilt.

The */etc/inittab* file is made up of several files. The two primary files are:

- The */etc/conf/cf.d/init.base* file, which contains entries associated with system startup and the console
- The */etc/conf/init.d/sio* file, which contains entries associated with standard serial ports

Some multiport card manufactures do not append the *sio* file. Instead, they create their own file(s) in the */etc/conf/init.d* directory.

The kernel will use the */etc/conf/cf.d/init.base* file and any files found in the */etc/conf/init.d* directory to recreate */etc/inittab*.

The steps to recreate the */etc/inittab* file are covered in Lesson 9, “Installing and Managing Serial Ports.”

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

---

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

## 7.5 Tracing Multiuser Initialization

---

When entering run-level 2 (multiuser mode), **init**(M) executes the scripts that mount additional filesystems, then starts multiuser system processes and the user login process. The */etc/rc2* file and the files in */etc/rc2.d* control multiuser initialization.

### 7.5.1 The */etc/rc2* file and */etc/rc2.d* directory

---

The following entry in */etc/inittab* instructs **init**(M) to execute the */etc/rc2* script:

```
r2:2:wait:/etc/rc2 1> /dev/console ...
```

The */etc/rc2* file contains instructions to execute the scripts stored in the */etc/rc2.d* directory.

**Exercise: Searching the */etc/rc2* file**

---

View the file */etc/rc2* and locate the lines below that indicate the search of the files in the */etc/rc2.d* directory:

```
echo 'Changing to state 2'
```

```
if [ -d /etc/rc2.d ]
```

Change to the */etc/rc2.d* directory and list the files.

AV 7-16

| /etc/rc2.d    |              |
|---------------|--------------|
| P00SYSINIT    | P90RESERVED  |
| P01MOUNTFSYS  | P93scohttp   |
| P03RECOVERY   | P95calserver |
| P04CLEAN      | S00MDAC      |
| P05RMTMPFILES | S25pm        |
| P15HWDNLOAD   | S35dlpi      |
| P16KERNINIT   | S60sync      |
| P20syssetup   | S80lp        |
| P21perf       | S84rpcinit   |
| P70uucp       | S85tcp       |
| P75cron       | S89nfs       |
| P86mmdf       | s99apcssd    |
| P86scologin   | messages     |
| P87USRDAEMON  |              |
| P88USRDEFINE  |              |

sa20716 © The Santa Cruz Operation, Inc.



---

## 7.5.2 Analyzing */etc/rc2.d* scripts

---

The */etc/rc2.d* directory contains scripts for several programs, including the **lpsched** and **cron** daemons. In addition to the default startup scripts, scripts are included that clean up temporary files, log files, and lock files for application programs.

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

|                     |                                                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------|
| <i>I01MOUNTFSYS</i> | Mounts filesystems specified in <i>/etc/default/filesys</i> and starts the audit subsystem if auditing is enabled |
| <i>P20syssetup</i>  | Generates the system ID                                                                                           |
| <i>P70uucp</i>      | Cleans up <b>uucp</b> locks, status, and temporary files                                                          |
| <i>P75cron</i>      | Starts the <b>cron</b> daemon                                                                                     |
| <i>S80lp</i>        | Starts up the <b>lpsched</b> daemon                                                                               |
| <i>S85tcp</i>       | Starts TCP/IP services                                                                                            |
| <i>S89nfs</i>       | Starts NFS services and mount networked filesystems                                                               |
| <i>P93scohttpd</i>  | Starts the SCOhelp daemon                                                                                         |
| <i>P95calserver</i> | Starts the SCOfalendar daemon                                                                                     |

Refer to the **SCO OpenServer System Administration Guide**, “Appendix A,” “Customizing UNIX system startup,” for a complete listing and definition of */etc/rc2.d* scripts.

### **Exercise: Reading */etc/rc2.d* scripts**

---

View the */etc/rc2.d/I01MOUNTFSYS* script. Which program is **init(M)** executing to mount the filesystems?

### 7.5.3 Ensuring backwards compatibility

---

Several of the scripts found in the */etc/rc2.d* directory are front-end scripts for other scripts found in the */etc/rc.d* directory.

#### **Exercise: Locating scripts in */etc/rc.d***

---

---

View */etc/rc2.d/P16KERNINIT*.

To which directory is this file pointing?

## 7.6 Customizing Multiuser Initialization

---

It may be necessary to add your own initialization procedures in the `/etc/rc2.d` directory or to 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 *I*, *P*, *S*, or *K*.

### 7.6.1 How scripts are executed from `/etc/rc2.d`

---

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

- Files that begin with *I* for Interactive and *S* for Serial are executed serially. The script completes before the next one is run.
- Files that begin with *P* for Parallel are executed in parallel and run concurrently.
- When going from a higher or lower **init** state to **init** state 2, files that begin with an *I*, *S*, or *P* are executed by the **prc\_sync**(ADM) command with the *start* option.
- When changing from a higher **init** state to **init** state 0, scripts that begin with *K* for Kill are executed by the **prc\_sync**(ADM) command with the *stop* option. *K* files are found in the `/etc/rc0.d` directory.
- Files that begin with characters other than *I*, *P*, *S*, or *K* are ignored.
- Files are executed in order of their file names sorted according to the ASCII collation sequence. For example, *S80lp* handles print service startup. It will be executed after any script that begins with *S79*, and before any script that begins with *S81*.

Refer to the **rc2**(ADM) and **prc\_sync**(ADM) manual pages for more information on the executing of **rc** scripts.

## 7.6.2 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, edit the file */etc/rc2.d/P88USRDEFINE*.

Follow these steps to add the command:

- Step 1. Edit the */etc/rc2.d/P88USRDEFINE* 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.

## 7.6.3 Defining other run-level initialization scripts

---

When you switch to single-user mode, **init**(M) 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:/etc/rc1 1 > /dev/console ...
```

The */etc/rc1* file contains instructions to execute the scripts stored in the */etc/rc0.d* directory.

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

```
r0:056:wait:/etc/rc0 1> /dev/console ...
```

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

## 7.6.4 Deactivating rc scripts

---

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

## 7.6.5 Locating messages encountered during run-level initialization

---

Startup, shutdown, and error messages are written to the appropriate directories. Note that these files are overwritten each time the system changes run-levels.

|                   |                              |
|-------------------|------------------------------|
| Single-user mode  | <i>/etc/rc0.d/messages/*</i> |
| Multiuser mode    | <i>/etc/rc2.d/messages/*</i> |
| State of shutdown | <i>/etc/rc0.d/messages/*</i> |

## 7.7 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, first check the following:

- Are the electrical outlets working properly?
- 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?

Many of the problems discussed in this section relate to missing or corrupted files.

### 7.7.1 Restoring missing or corrupted boot files

---

On rare occasions, one or more of the critical system programs is accidentally modified or removed, preventing the system from booting or operating correctly. In cases where your system does not boot, you must boot from your emergency boot floppy set to access the system and restore critical files from backup or from your emergency floppy set.

Refer to the **SCO OpenServer Handbook**, “Starting and stopping the system,” for step by step instructions to restore missing or corrupted boot files.

The most common error messages are:

- `boot not found`

This message means that the *boot* file is missing from the */stand* directory on the hard disk. To correct this error, use your emergency boot floppy set and restore the */boot* file from your boot floppy.

- `unix not found`

This message means that the `/stand/unix` file is missing from your hard disk. To correct this error, either:

- Boot from another kernel, such as *unix.old*, and restore `/stand/unix` from your backup. After restoring, reboot your system.
- Use your emergency boot floppy set and restore the `/stand/unix` from your boot floppy

### **Exercise: Correcting boot errors**

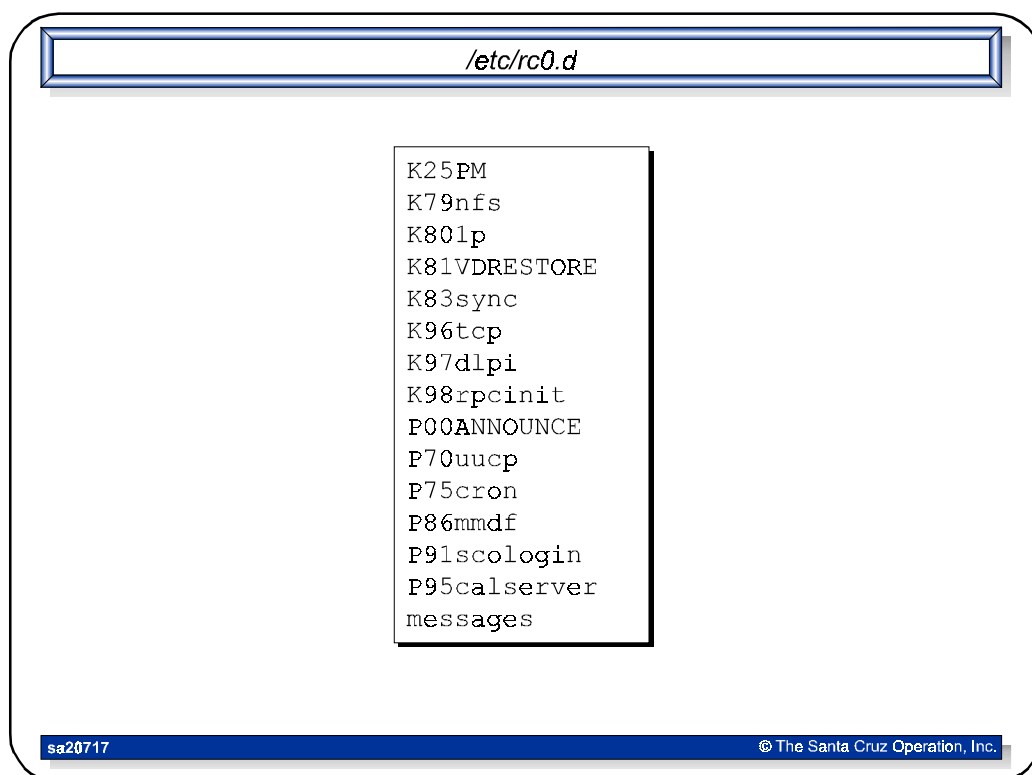
---



---

Use your *SCO OpenServer Handbook* and review the steps to correct the error: `boot not found`.

AV 7-17





## 7.8 Shutting Down the System

---

The primary function of the **shutdown**(ADM) 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 the superuser or a user with the *shutdown* subsystem authorization to execute **shutdown**(ADM). The */etc/shutdown* script checks to see if root or an authorized user is performing the command. If not, the shutdown does not proceed.
- Runs **init**(M)

### 7.8.1 Analyzing shutdown

---

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

```
r0:056:wait:/etc/rc0 1> /dev/console ...
```

The */etc/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

## 7.9 Lesson Summary

---

---

You can customize the default UNIX boot process by editing the parameters in */etc/default/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 SCO OpenServer system.

## Review Questions

---

1. Name the command that allows you to boot other operating systems at the **Boot :** prompt.
2. Which default boot file is accessed by the **boot(HW)** program during system startup?
3. Name the file that you would edit to change the boot parameter **TIMEOUT**.
4. When is the hardware information that is displayed at boot time written to the */usr/adm/messages* file?
5. What is a major difference between run-level 1 and run-level S?
6. Which directory contains the startup scripts used to initialize multiuser mode?

7. Which script file contains the information to start the SCO help daemon when the system enters multiuser mode?
  
  
  
  
  
  
  
  
  
  
8. In which SCO manual can you find the steps to recover from the “boot not found” error?
  
  
  
  
  
  
  
  
  
  
9. When performing a shutdown, what does the */etc/rc0* file accomplish?

## 7.10 Lab: Creating an rc Script

---

When the system enters multiuser mode, each console multiscreen (*/dev/tty01*, */dev/tty03*, and */dev/tty04*) should reflect a different color.

### 7.10.1 Overview

---

Create an executable shell script in your home directory named *S30colors*. Use the **setcolor(C)** command to set different colors for the above console multiscreens when the system next enters multiuser mode.

### 7.10.2 Procedure

---

- Step 1. Log in as **root**.
- Step 2. Create a file named *S30color* in your home directory.
- Step 3. Add the following information:  
  

```
# Set monitor colors for console multiscreens  
  
setcolor -o red hi_white black > /dev/tty01  
setcolor -o blue hi_white black> /dev/tty03  
setcolor -o green hi_white black > /dev/tty04
```
- Step 4. Save your file and quit.
- Step 5. Change permissions so that the file is read and executable by **root**.
- Step 6. Copy the file to the */etc/rc2.d* directory.
- Step 7. Test the results by shutting down and restarting your system.

## 7.11 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                       |                                                                                                                                              |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>boot</b> (HW)      | Loads the UNIX system                                                                                                                        |
| <b>hwconfig</b> (HW)  | Displays hardware configuration information                                                                                                  |
| <b>getty</b> (M)      | Allows logins over bidirectional lines                                                                                                       |
| <b>init</b> (HW)      | Initializes process control                                                                                                                  |
| <b>bootos</b> (HW)    | Loads and runs a non-UNIX operating system                                                                                                   |
| <b>shutdown</b> (ADM) | Terminates all processes gradually                                                                                                           |
| <b>syslogd</b> (ADM)  | Logs system messages                                                                                                                         |
| <b>wall</b> (ADM)     | Sends a message to all users currently logged in                                                                                             |
| <b>who</b> (C)        | Lists the users currently logged in to the system; with the <b>-r</b> option, indicates the current run-level of the <b>init</b> (M) process |

### Relevant files

---

|                                |                                                                       |
|--------------------------------|-----------------------------------------------------------------------|
| <i>/stand/boot</i>             | SCO Open Server boot program                                          |
| <i>/stand/etc/default/boot</i> | Contains default boot parameters                                      |
| <i>/etc/inittab</i>            | Contains instructions for <b>init</b> (M)                             |
| <i>/etc/rc0.d</i>              | Scripts executed when going to single-user mode and state of shutdown |
| <i>/etc/rc2.d</i>              | Scripts executed when going to multiuser mode                         |
| <i>/etc/shutdown</i>           | Script that shuts down the system in an orderly fashion               |

## 7.12 Answers to Exercises and Review Questions

---

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Listing the kernels in */stand*

---

Use the **l(C)** command to list the kernels in */stand*.

The kernels listed in */stand* are */unix*, */unix.safe*, */unix.install*, and (possibly) */unix.old*.

### Exercise: Determining if other operating systems exist

---

The syntax to boot the NT operating system is **bootos nt**

### Exercise: Comparing */etc/default/boot* and */stand/etc/default/boot* files

---

To compare */etc/default/boot* and */stand/default/boot*, enter:

```
more /etc/default/boot
more /stand/etc/default/boot
```

Both files should be identical.

### Exercise: Changing the TIMEOUT parameter

---

Use **vi(C)** to edit the TIMEOUT parameter the */etc/default/boot* file:

```
TIMEOUT=10
```

### Exercise: Viewing the */usr/adm/messages* file

---

The time and date stamp appears on the first line of the */usr/adm/message* file.

## Exercise: Using the wall(ADM) command

---

To send a message that the system will be shutting down in 15 minutes, do the following:

```
wall <Enter>
System will be shutdown in 15 minutes
<Ctrl> d
```

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

---

By default, */etc/getty* is started at run-level 2.

A **getty**(C) process can be executed at run-levels 2, 3, and 4. SCO OpenServer systems do not usually use run-levels 3 and 4.

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

---

The */etc/conf/init.d* directory is made up of the following three files by default: *scohttp*, *scologin*, and *sio*. Other files may exist if an intelligent multiport card has been installed.

## Exercise: Reading */etc/rc2.d* scripts

---

The **init**(M) program executes */etc/mountall* to mount the filesystems.

## Exercise: Locating scripts in */etc/rc.d*

---

The */etc/rc2.d/P16KERNINIT* script points to */etc/rc.d/6*.

## Exercise: Correcting boot errors

---

Use the steps outlined in the *SCO OpenServer Handbook*, “Starting and Stopping the System,” “boot not found.”

## Exercise: Viewing the */etc/rc0.d* scripts

---

The */etc/rc0.d/K96tcp* script terminates TCP/IP. It is executed after the */etc/rc0.d/K79nfs* script.

## Review questions

---

This section provides suggested answers to the review questions.

1. The **bootos**(HW) command allows you to boot other operating systems at the **BooT :** prompt.
2. The **boot**(HW) program accesses the `/stand/etc/default/boot` file during system startup.
3. You edit the `/etc/default/boot` file to change the **TIMEOUT** boot parameter.
4. The hardware information displayed at boot time is written to `/usr/adm/messages` when **init**(M) prompts you to enter the **root** password to enter single-user mode or to press <Enter> to enter multiuser mode.
5. All filesystems remain mounted when entering run-level S and the system console device is reassigned.
6. The `/etc/rc2.d` directory contains the startup scripts used to initialize multiuser mode.
7. The `/etc/rc2.d/P93scohttd` file contains the information to start the **SCOhelp** daemon when the system enters multiuser mode.
8. The **SCO OpenServer Handbook** contains the steps to recover from the “boot not found” error.
9. During a shutdown, the `/etc/rc0` file echoes that the system is coming down, runs scripts stored in the `/etc/rc0.d` directory, kills off all processes, and unmounts all filesystems.



---

# 8

## *Filesystem Configuration and Disk Management*

AV 8-1

Filesystem Configuration and Disk Management

- Describe the structure of a UNIX filesystem
- Manage the `/dev` directory
- Configure filesystems
- Describe important filesystem features
- Check and repair filesystems
- Configure additional hard disks
- Troubleshoot hard disks

sa20801 © The Santa Cruz Operation, Inc.

**Lesson 8**

## **Filesystem Configuration and Disk Management**

### **Introduction**

---

You learned some basic concepts about UNIX filesystems and disks structures in Lesson 3, “Files and Filesystems.” As the system administrator, you are responsible for the integrity and security of user data, and need a greater understanding of the structure of the UNIX filesystem to configure, maintain, and troubleshoot filesystems and hard disks.

### **Objectives**

---

This lesson provides an in-depth discussion of the UNIX 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 structure of a UNIX filesystem
- Manage the */dev* directory
- Describe important filesystem features
- Configure filesystems
- Check and repair filesystems
- Configure additional hard disks
- Troubleshoot hard disks



## 8.1 Describing the UNIX Filesystem Structure

---

SCO OpenServer systems take advantage of several filesystem structures. Each structure provides file access and control mechanisms. Filesystems are comprised of a superblock, inodes, and data blocks.

### 8.1.1 The Superblock

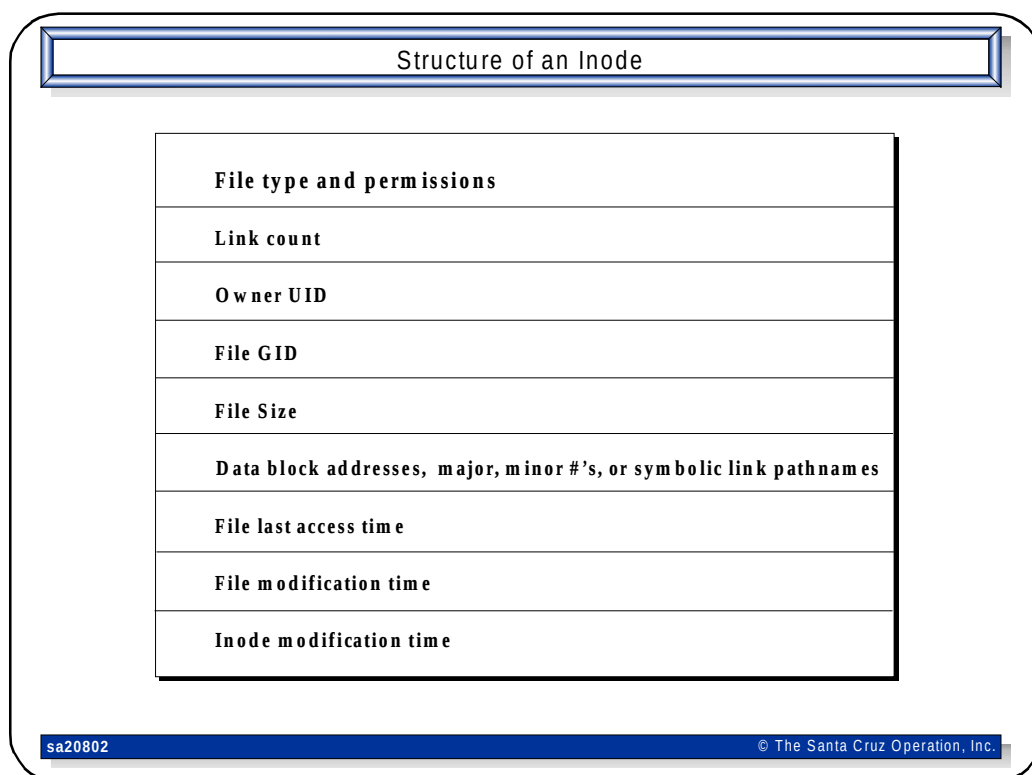
---

The superblock contains information critical for filesystem access and maintenance. It includes the type of filesystem, its structure, size, and the location of the inode table and data blocks.

It also includes the condition of the filesystem. For example, when a system is shut down improperly, it is marked *dirty*. Upon rebooting the system, **fsck**(ADM) runs to clean the filesystem. Likewise, when the filesystem is shut down properly, the filesystem is marked *clean*.

Whenever a filesystem is in use, this component is cached in memory and maintained by the kernel.

AV 8-2



## 8.1.2 Inodes

---

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)
- User ID (UID)
- 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 the associated inode number.

### 8.1.2.1 Reserved inodes

---

The system uses inodes 0, 1, and 2 as follows:

- Inode 0 indicates deleted files and directories
- Inode 1 contains system creation times and is not used for file access
- Inode 2 (the root directory) is the top level of each mounted filesystem's hierarchical structure

### 8.1.2.2 Viewing inode numbers

---

To display inode numbers associated with filenames, use the **-i** option with the **ls(C)** command.

To display inode numbers associated with directory files, use the **-ld** options with the **ls(ADM)** command.

#### **Exercise: Associating file names with inode numbers**

---

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



### **8.1.3 Data blocks**

---

The data block area of the filesystem contains both allocated and non-allocated blocks. Block sizes depend on the filesystem structure being used and range from multiples of 512-bytes to 4KB.

Although a file can use many data blocks, a file is accessed by a single inode.

## AV 8-3

| Special Device Files |   |         |         |    |     |        |       |                  |  |
|----------------------|---|---------|---------|----|-----|--------|-------|------------------|--|
| brw-----             | 2 | sysinfo | sysinfo | 1, | 0   | Apr 02 | 15:35 | /dev/hd0         |  |
| brw-----             | 2 | sysinfo | sysinfo | 1, | 0   | Apr 02 | 15:35 | /dev/dsk/0s0     |  |
| brw-rw-rw-           | 7 | bin     | bin     | 2, | 64  | Apr 02 | 15:35 | /dev/fd0         |  |
| brw-rw-rw-           | 7 | bin     | bin     | 2, | 60  | Apr 02 | 15:35 | /dev/fd0135ds18  |  |
| brw-rw-rw-           | 7 | bin     | bin     | 2, | 60  | Apr 02 | 15:35 | /dev/dsk/f03ht   |  |
| brw-rw-rw-           | 7 | bin     | bin     | 2, | 53  | Apr 02 | 15:35 | /dev/dsk/f1t     |  |
| brw-rw-rw-           | 7 | bin     | bin     | 2, | 53  | Apr 02 | 15:35 | /dev/dsk/f15ht   |  |
| brw-rw-rw-           | 7 | bin     | bin     | 2, | 53  | Apr 02 | 15:35 | /dev/dsk/f1q15dt |  |
| crw-rw-rw-           | 1 | bin     | bin     | 5, | 128 | Apr 02 | 15:35 | /dev/tty1A       |  |
| crw-rw-rw-           | 1 | bin     | bin     | 5, | 0   | Apr 02 | 15:35 | /dev/tty1a       |  |
| crw-----             | 1 | sysinfo | sysinfo | 1, | 0   | Apr 02 | 15:35 | /dev/rhd00       |  |
| crw-----             | 1 | sysinfo | sysinfo | 1, | 15  | Apr 02 | 23:45 | /dev/rhd01       |  |

sa20803

© The Santa Cruz Operation, Inc.

## 8.2 Managing the /dev Directory

---

The UNIX kernel does not directly access server hardware devices such as hard disks, terminals, and printers. Instead, the kernel uses files that represent system hardware.

These 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. Character device files are also known as raw devices.

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

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

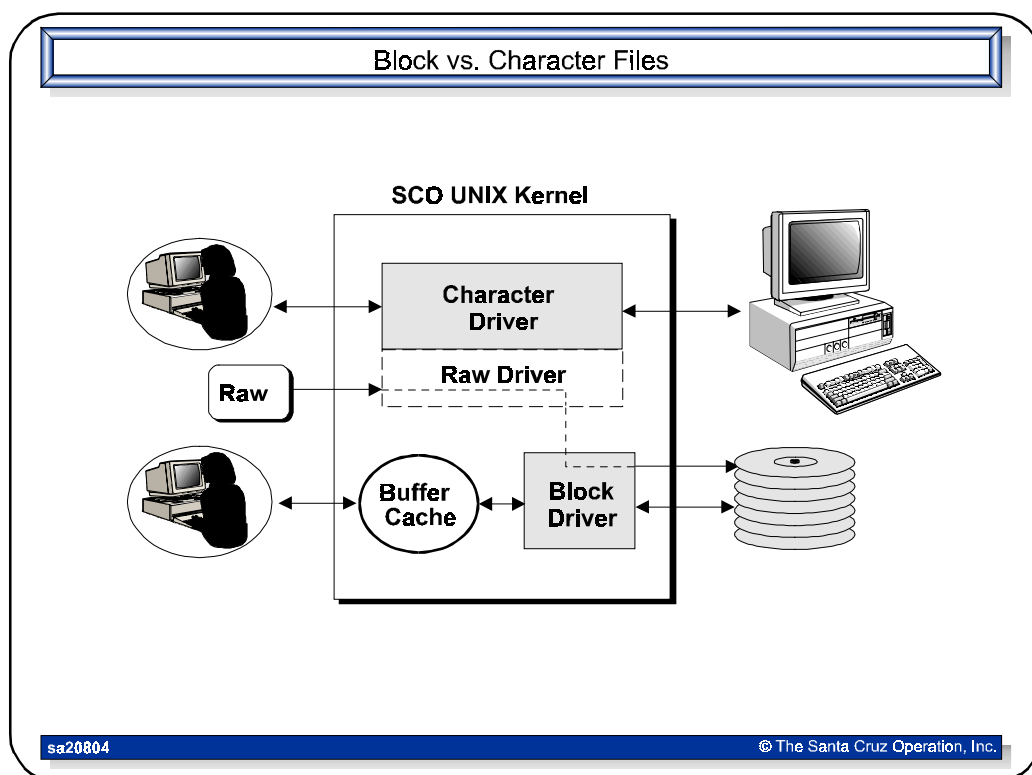
- /dev/fd\*      Represents floppy devices
- /dev/tty\*    Represents serial devices
- /dev/lp\*      Represents parallel devices

### **Exercise: Listing device files**

---

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

AV 8-4



### 8.2.1 Block files

---

You can identify a block file if you run the **l(C)** command and see a *b* in the first column before the file permissions.

Devices that use block special files read and write 1KB of data at a time. The data is buffered by the operating system.

Devices that use the block interface receive requests from the kernel to transfer 1KB of data to and from a device. Because most requests for data are not exactly 1KB, the kernel packages them into one or more 1KB requests and, in doing so, uses a buffer cache.

Filesystems are examples of block special devices.

### 8.2.2 Character files

---

You can identify a character file if you run the **l(C)** command and see a *c* in the first column before the file permissions.

Character files manage varying amounts of data. Examples of character special devices include serial ports, parallel ports, disk drives, and tape drives.

### 8.2.3 Raw interfaces

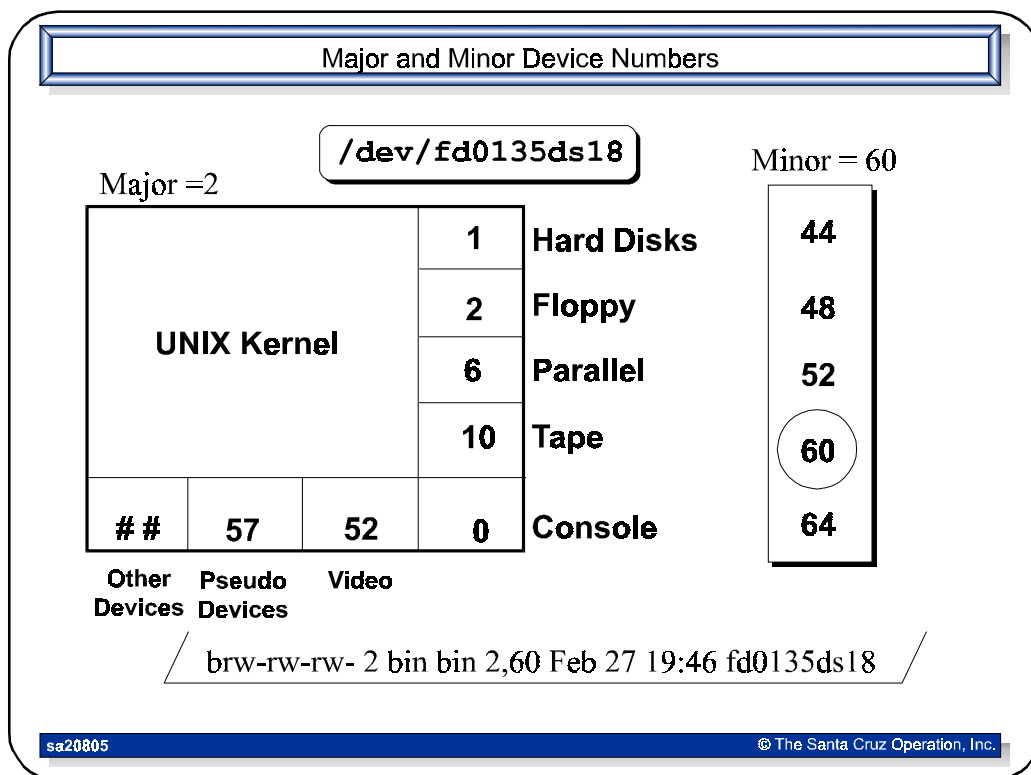
---

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 **l(C)** 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 8-5



## 8.2.4 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 further describe the device driver.

### 8.2.4.1 Major device numbers

---

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

- Hard drives have a major number of 1.
- Floppy drives have a major number of 2.
- Non SCSI cartridge tapes have a major number of 10.

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

## 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, `/dev/rct0`, the first non-SCSI cartridge tape device, has a major number of 10. However, two additional device files exist which access the same device but perform a different function.

Minor numbers representing non-SCSI cartridge tape devices include:

- `10, 0` represents the first device (`/dev/rct0`)
- `10, 128` represents the first device when using the **tape(C)** command (`/dev/xct0`)
- `10, 8` represents the first device in no-rewind mode (`/dev/nrct0`)

A minor number can also represent a second device when multiple types of the same device are installed. For example, `10, 16` represents the second non-SCSI cartridge tape device (`/dev/rct1`).

### **Exercise: Comparing major and minor device numbers.**

---

Perform a long listing of the floppy disk device `/dev/dsk/f0q*`. How many major numbers do you see? How many different minor numbers?

## **8.2.5 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 can recreate the missing special device file.

It is important that you print the contents of your `/dev` directory and place it in your log book. Most of the information needed to create a device file is not stored on your system!

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

**mknod name [c | b] major minor**

For example, to recreate `/dev/fd048`, enter:

**mknod /dev/fd048 b 2 4**

You might have to change the permissions, ownership, and group to that of the original file. A few entries for device files are placed in the `/usr/lib/mkdev/perms/*` directories. To locate an entry:

**grep 'fd048' /usr/lib/mkdev/perms/\***

If the entry is not found in the `/usr/lib/mkdev/perms` directory, check your *System Log Book*.

## **8.2.6 Alternate names for special device files**

---

Use links to create alternative names for frequently used devices. For example, to link the name `tape1` to `/dev/rct0`:

**ln /dev/rct0 /dev/tape1**

You can now access the tape drive by either device name.



## 8.2.7 Troubleshooting the */dev* directory

---

Potential sources of problems in the */dev* directory include:

- The removal of special device files, either by accident or by program malfunction
- Misspelled commands 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:

```
find /dev -type f -exec l {} \;
```

or

```
l /dev | grep "^-"
```

## 8.3 Describing Important 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 SCO OpenServer Release 5 system has at least the following two filesystems on the primary hard disk:

|      |                                                                                                                       |
|------|-----------------------------------------------------------------------------------------------------------------------|
| root | Contains the programs and directories that comprise the operating system                                              |
| boot | Contains the information needed to boot your system, the <b>boot</b> (HW) program, and the kernel, <i>/stand/unix</i> |

During installation, if the root filesystem is larger than 146MB, and more than 78MB of space remains, you are prompted for a third filesystem, */u* (user).

If your root filesystem is greater than 780MB and adequate space remains, then */dev/scratch*, used by the **fsck**(ADM) program when repairing filesystems, is created.

### 8.3.1 Configuration issues

---

Filesystem configuration is part of the installation process when adding hard disks. Before creating a filesystem, you need to consider the following:

- Checkpointing
- Intent logging
- Versioning

#### 8.3.1.1 Checkpointing

---

Checkpointing is the process of transitioning filesystems to a clean (consistent) state when the filesystem is inactive. Filesystem data consist of user file data and “metadata” (data that refers to accounting information used internally by the filesystem to represent a file).

By default, HTFS, EAFS, AFS, and S51K filesystems periodically *checkpoint* (write) cached metadata back to the disk. This increases the probability that the filesystem metadata is in a consistent state if the system is halted unexpectedly.

For the root filesystem, the kernel parameter **ROOTCHKPT** controls this feature. If overhead is an issue, disable checkpointing by setting **ROOTCHKPT** to 0; then, rebuild the kernel (required).

For non-root filesystems, to disable checkpointing, refer to Section 8.4.3, “The `/etc/default/filesys` file,” on page 8-27.

---

### 8.3.1.2 Intent logging

---

Intent logging minimizes system downtime. It does not increase the reliability of the filesystem. When the system is halted unexpectedly, instead of the **fsck**(ADM) command checking entire filesystems for inconsistencies, it only checks logged transactions.

A log file is created as a special file in the root of each mounted filesystem. This file is hidden. If logging is disabled, the log file appears as `.ilog0000`.

Intent logging is enabled by default for the HTFS, DTFS, AFS, and EAFS filesystems.

For the root filesystem, the kernel parameter **ROOTLOG** controls this feature. If overhead is an issue, disable intent logging by setting **ROOTLOG** to 0; then, rebuild the kernel (required).

For non-root filesystems, to disable intent logging, refer to Section 8.4.3, “The `/etc/default/filesys` file,” on page 8-27.

---

### 8.3.1.3 Versioning

---

Versioning allows you to recover deleted files with the **undelete**(C) command. It applies only to DTFS and HTFS filesystems and is not configured by default.

The following steps describe how to enable versioning:

- Step 1.        Log in as **root**.
- Step 2.        Run **scoadmin**(ADM).
- Step 3.        Select **Hardware/Kernel Manager**.
- Step 4.        Select **Tune Parameters**.
- Step 5.        Enter **10** for Filesystem configuration.

- Step 6. Set the kernel parameters as follows:
- For non-root filesystems, set the **MAXVDEPTH** kernel parameter to the desired number of undeletable (versioned) files allowed. The maximum value is 65535. A value of 0 disables versioning.
- Set the **MINVTIME** to the desired number of seconds (*n*) before a file is versioned. The default value is 0.
- For the root filesystem, set the **ROOTMAXVDEPTH** kernel parameter to the desired maximum number of undeletable (versioned) files on a root filesystem allowed. The maximum value is 65535. A value of 0 disables versioning.
- Set the **ROOTMINVTIME** to the desired number of seconds (*n*) before a file is versioned. The default value is 0.
- Step 7. Select **q** when complete. The following message appears:
- Would you like to update the system configuration files with your change? (y/n)
- Step 8. Enter **y**.
- The system updates the configuration files.
- Step 9. Select **Relink Kernel**.
- The system prompts you to confirm you want to relink the kernel.
- Step 10. Select **Relink**.
- After the kernel is rebuilt, the system prompts you with the following message:
- Do you want this kernel to boot by default? (y/n)
- Step 11. Select **y**.
- The system then prompts you with:
- Do you want the kernel environment rebuilt? (y/n)

- Step 12. Enter **y**.
- Wait for the message that the kernel has been successfully linked and installed.
- Step 13. Press  to return to the **Hardware/Kernel Manager** window.
- Step 14. Select **Kernel** ⇒ **Exit** to exit the **Hardware/Kernel Manager**.
- Step 15. Shut down and reboot your system to activate the new kernel:
- /etc/shutdown -i6 -g0 -y**

## AV 8-6

**divvy(ADM)**

| Name    | Type       | New FS | # | First Block | Last Block |
|---------|------------|--------|---|-------------|------------|
| boot    | EAFS       | no     | 0 | 0           | 15359      |
| swap    | NON FS     | no     | 1 | 15360       | 80895      |
| root    | HTFS       | no     | 2 | 80896       | 1023988    |
|         | NOT USED   | no     | 3 | -           | -          |
|         | NOT USED   | no     | 4 | -           | -          |
|         | NOT USED   | no     | 5 | -           | -          |
| recover | NONFS      | no     | 6 | 1023989     | 1023998    |
| hd0a    | WHOLE DISK | no     | 7 | 0           | 1025007    |

1023999 1K blocks for divisions, 1008 1K blocks reserved for the system

sa20806
© The Santa Cruz Operation, Inc.

## AV 8-7

**divvy(ADM) Instructions**

```

n[ame]          Name or rename a division.
c[reate]        Create a new file system on this division.
t[ype]          Select or change filesystem type on new filesystems.
p[revent]       Prevent a new file system from being created on this division.
s[tart]         Start a division on a different block.
e[nd]           End a division on a different block.
r[estore]       Restore the original division table.

Enter your choice or q to quit
i[nstall]       Install the division set-up shown
r[eturn]        Return to the previous menu
e[xit]          Exit without installing a division table

Please enter your choice:

```

sa20807
© The Santa Cruz Operation, Inc.

## 8.4 Configuring Filesystems

---

The **divvy**(ADM) command is an interactive command that lets you divide a UNIX partition into a maximum of seven filesystems (0-6). Division 7 is reserved and refers to the entire partition. You can use the entire partition as one filesystem or divide a partition into multiple filesystems.

The size of the filesystem is calculated in 1KB blocks. For example, if you have determined that a 100MB filesystem is needed, starting at block 0, you would enter the size as 0-99999.

The maximum filesystem sizes created by **divvy** include:

- 1TB for DTFS
- 1TB for HTFS
- 2GB for other filesystems

### 8.4.1 Dividing the partition

---

The **divvy** command is called automatically during the operating system installation and when configuring additional disks. The **divvy** command:

- Enables you to divide a UNIX partition into a maximum of 7 filesystems (0-6); division 7 is reserved and refers to the entire partition
- Calculates the size of the filesystems in 1KB blocks
- Invokes **mkfs**(ADM) to create the filesystem(s)

#### **Exercise: Determining the number of filesystems**

---

To display the division table:

**divvy**

How many filesystems are present?

## 8.4.2 Ready the filesystem for access

---

When filesystem creation is complete, your next step is to invoke the **Filesystem Manager** to add the new filesystem to your system and specify the following:

- The device file
- 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 filesystem type
- The access mode (read-write or read-only)
- If users can mount the filesystem
- When to mount the filesystem (now or at startup)
- Check and repair options

You can also use the **Filesystem Manager** to specify filesystem-specific advanced options, for example, checkpointing, versioning, intent logging, and many other options. The available advanced options depend on the type of filesystem you are adding (HTFS, DTFS, EAfs, and so forth).

The changes you make with the **Filesystem Manager** are written to the */etc/checklist* and */etc/default/filesys* files.

Refer to the *SCO OpenServer System Administration Guide*, “Modifying filesystem mount configuration.”

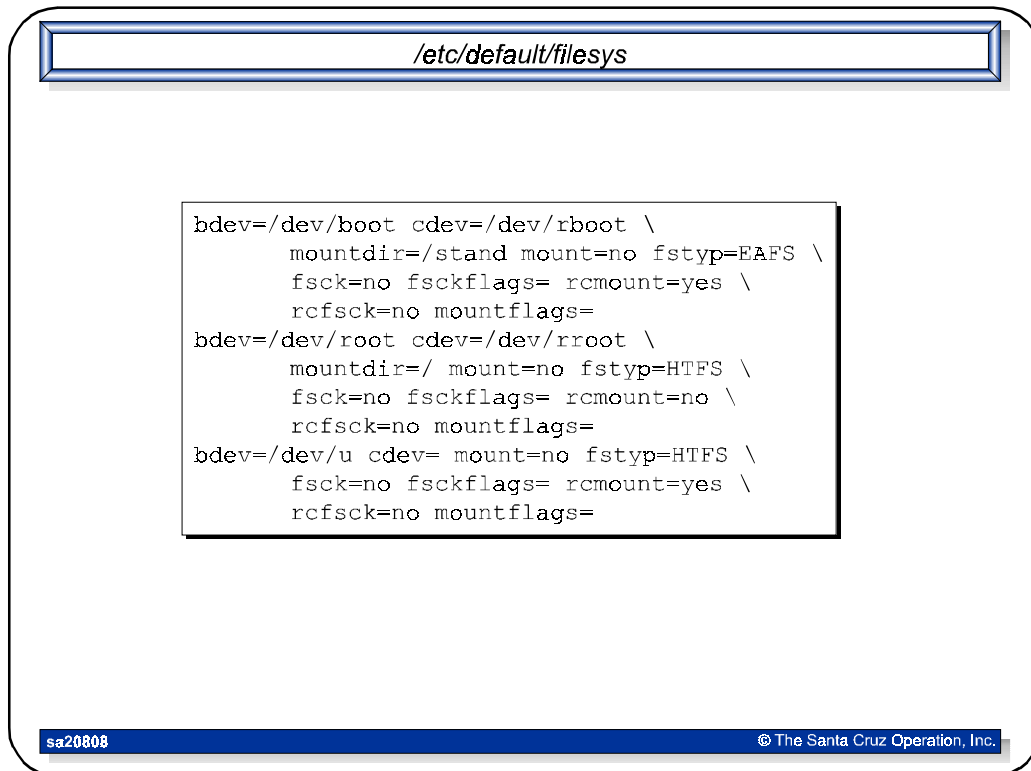


### 8.4.2.1 The */etc/checklist* file

---

The */etc/checklist* file contains a list of the filesystems to be checked when **fsck**(ADM) is invoked without arguments. The **fsck** command is discussed in detail in “Checking and Repairing Filesystems” later in this lesson.

AV 8-8



### 8.4.3 The `/etc/default/filesys` file

---

The `/etc/default/filesys` file contains default information when mounting filesystems. Default options include:

|                        |                                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <code>bdev</code>      | Represents the block device name of the filesystem                                                                                           |
| <code>cdev</code>      | Represents the raw device name of the filesystem                                                                                             |
| <code>mountdir</code>  | The absolute pathname of the directory to which the filesystem is mounted                                                                    |
| <code>mount</code>     | Specifies if a filesystem can be mounted by an ordinary user; if set to <code>no</code> , only the <b>root</b> user can mount the filesystem |
| <code>rcmount</code>   | Determines if the filesystem is mounted when going to multiuser mode, is never mounted, or the administrator is prompted for confirmation    |
| <code>fsckflags</code> | Defines any flags to be passed to <b>fsck</b> if the filesystem is marked dirty when entering multiuser mode                                 |

You can add other filesystems by editing `/etc/default/filesys` or using the **Filesystem Manager** in `scoadmin`(ADM). These options can include:

|                        |                                                                                                                                                                                                                                                                                                           |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>maxvdepth</code> | The maximum number of previous versions that will be maintained for version files in a filesystem. The filesystem must be mounted with a <code>maxvdepth</code> value greater than 0; a value of 0 disables versioning.                                                                                   |
| <code>minvtime</code>  | <p>The minimum time (in seconds) that a file must exist before it can be versioned. The filesystem must be mounted with a <code>minvtime</code> greater than 0.</p> <p>Note that default versioning is configured as a kernel parameter. These defaults can be overridden for individual filesystems.</p> |
| <code>nochkpt</code>   | Mounts the filesystem with checkpointing disabled                                                                                                                                                                                                                                                         |
| <code>no log</code>    | Mounts the filesystem with intent logging disabled                                                                                                                                                                                                                                                        |

Refer to the **mount**(ADM) manual page or a complete definition of mount options.

**Exercise: Viewing */etc/default/filesys***

---

View the */etc/default/filesys* file. Can an ordinary user mount and unmount the */u* filesystem?

## 8.5 Checking and Repairing Filesystems

---

The **fsck**(ADM) command checks and repairs the structure of a filesystem. The checking is performed in a series of phases, and each phase checks for different inconsistencies within the filesystem.

In most instances of filesystem corruption, **fsck**(ADM) automatically repairs a filesystem without interaction. If you need to interact with **fsck**, you need to know the definition of phases, prompts, and error messages.

Use the following guidelines for running **fsck**:

- You can use the **Filesystem Manager** or the command-line interface to check and repair a filesystem..
- 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 filesystem when running **fsck** from the command line, all filesystems listed in */etc/checklist* are examined.

AV 8-9

**fsck(ADM) Check Phases**

**fsck(ADM) Check Phases**

```
** Phase 0 - Replay Log
** Phase 1 - Check Blocks and Sizes
** Phase 1b - Rescan for more DUPS
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Free List Bitmap
** Phase 6 - Salvage Free List Bitmap
```

sa20800

© The Santa Cruz Operation, Inc.

### 8.5.1 Filesystem check phases

---

When you check and repair an HTFS, EAFS, AFS or S51K filesystem, the **fsck**(ADM) utility scans and examines the filesystem structures and reports on its progress. Each phase compares components and checks that these components agree with each other. These phases include:

- |                                  |                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Phase 0 - Replay Log             | Performs a fast check if intent logging is enabled on the filesystem and you did not request a full check                                                                                                                                                                                                                                         |
| Phase 1 - Check Blocks and Sizes | 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.                                                                                                                                                                 |
| Phase 1b                         | Scans the filesystem a second time if duplicate blocks were found                                                                                                                                                                                                                                                                                 |
| Phase 2 - Check Pathnames        | Cleans up error conditions caused by improper inode status, out-of-range inode pointers, and directories that point to bad inodes                                                                                                                                                                                                                 |
| Phase 3 - Check Connectivity     | <p>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 <i>lost+found</i> directory.</p> <p>When the directory is severed, the name of the file is lost, so <b>fsck</b> assigns the inode number as the new name the file</p> |
| Phase 4 - Check Reference Counts | Checks the link count of each entry that survived Phase 2 and 3                                                                                                                                                                                                                                                                                   |
| Phase 5 - Check Free List Bitmap | Examines the free-block list maintained by the filesystem to resolve any missing or unallocated blocks. If changes are made, <b>fsck</b> (ADM) rebuilds the free block list.                                                                                                                                                                      |

#### Phase 6 - Salvage Free List Bitmap

Reconstructs a free-block list from the altered filesystem specified in Phase 5

### 8.5.2 Checking the DTFS filesystem

---

When you check and repair a DTFS filesystem, **fsck** works non-interactively to repair inconsistencies and reports on its progress with the following messages:

Phase 1 - Check Super Block

Phase 2 - Check File Structure

Phase 3 - Check Directory Entries

Phase 4 - Check Block Bitmap

Refer to the *SCO OpenServer System Administration Guide*, “Administering Filesystems,” “Filesystem check phases (DTFS),” for a detailed description of the phases performed on a DTFS filesystem.

### 8.5.3 Running fsck(ADM)

---

When you boot the system after an abnormal shutdown, the **fsck(ADM)** command runs automatically on the root filesystem. Other filesystems are checked when the system enters multiuser mode.

To check and repair the `/u` filesystem, do one of the following:

- Run **fsck** and assume a yes response to all questions asked by **fsck(ADM)**:

```
fsck -y /dev/u
```

- Run the fast check option:

```
fsck -f /dev/u
```

- Run a full check:

```
fsck -ofull /dev/u
```



If **fsck**(ADM) finds that the root filesystem has been modified, it will either remount the root filesystem or reboot the system automatically. The root filesystem is only remounted if the damage to it is minor.

When running **fsck** on the root filesystem, you should first unmount all other filesystems.

### **Exercise: Repairing the /u filesystem**

---

Unmount the /u filesystem and run **fsck**(ADM) using the full option.

## **8.5.4 Error messages**

---

Error messages may appear in any phase. For example, if an inode was not connected to a directory in Phase 4, **fsck**(ADM) would reconnect it to the *lost+found* directory. If you are running **fsck** in interactive mode, the following message, prompting you to answer y(es) to reconnect the file to the *lost+found* directory, appears:

```
UNREF FILE I=782 OWNER=230 MODE=0777 SIZE=5046 MTIME=March 8
1995 (RECONNECT?)
```

The following list describes the fields in this message:

|                    |                                      |
|--------------------|--------------------------------------|
| I=782              | The inode number of the 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 1995 | The last time the file was modified  |

Refer to **fsck** manual page for a complete definition of all error messages reported.

### 8.5.5 Running **ncheck**(ADM)

---

If **fsck**(ADM) reports file errors, it is best to exit the program and run the **ncheck**(ADM) 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 */u* filesystem, enter:

```
ncheck -i <inode number> /dev/u
```

If you do not specify a filesystem, **ncheck** checks all filesystems that are listed in the */etc/checklist* file.

#### **Exercise: Associating inode numbers with filenames**

---

Use the **-i** option to the **ls**(C) command to determine the inode number for the */etc/default/tar* file. Search the root filesystem with **ncheck**(ADM) and display the results.

## **8.6 Configuring Additional Disks**

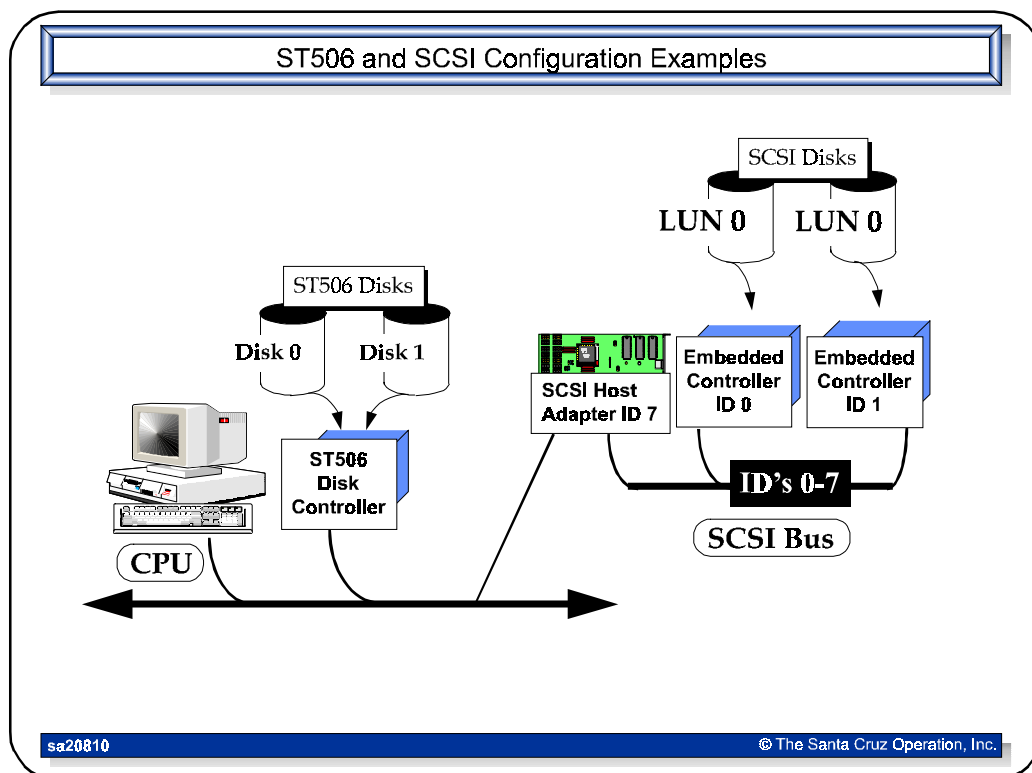
---

You installed the primary root disk during installation (and possibly a second hard disk). You can configure additional hard disks using **mkdev hd** or the **Hardware/Kernel Manager** in **scoadmin**(ADM).

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
- Ensure that the hard disk has been formatted and passes the vendor's diagnostic tests

AV 8-10



### 8.6.1 Possible configurations for secondary hard disks

---

You can add many types of hard disks to your system, provided the disk controller supports the disk. The following 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. Each host adapter, including the primary, is capable of supporting up to seven disk drives with embedded controllers on a SCSI 1 bus, and up to 15 on a 16-bit-wide SCSI 2 bus.

- Boot disk on an IDE, EIDE, or ESDI controller
- ISA, EISA, and MCA bus machines can support up to two IDE or EIDE controllers. Each controller can support two disk drives. 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.

Refer to the following sections in the *SCO OpenServer Handbook* for additional information on configuring hard disks: “Installing a hard disk” and “Configuring a SCSI hard disk.”

### 8.6.2 Device drivers

---

The following is a list of device drivers for hard disks:

|              |                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------|
| <b>wd</b>    | ISA, EISA, MCA, and PCI bus machines that present a WD1010 or ST506 interface, including IDE and EIDE drives |
| <b>Sdsk</b>  | All SCSI disks                                                                                               |
| <b>esdi</b>  | ESDI disks on MCA bus machines                                                                               |
| <b>st506</b> | ESDI disks configured to present an ST506 interface                                                          |

### 8.6.3 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.

To configure IDE disks and disk controllers, you must do the following:

- Set 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.
- Select 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. Refer to “BIOS support for disks larger than 1024 cylinders” in the *SCO Open Server Handbook*.

### 8.6.4 Using mkdev(ADM) to configure disks

---

To configure disks with **mkdev(ADM)**, use the **hd** argument and supply **mkdev** with information about your disk controller and hard disk:

For IDE controllers, specify:

- The controller number
- The disk number

For SCSI host adapter, specify:

- The host adapter type; supported host adapter types can be found in the */etc/default/scsihas* file
- The host adapter number
- The number of the bus on the host adapter; needed only if you are installing a controller on a host adapter that controls two SCSI buses. Specify to which bus the controller is connected.
- The target ID
- The logical unit (LUN) number of the disk

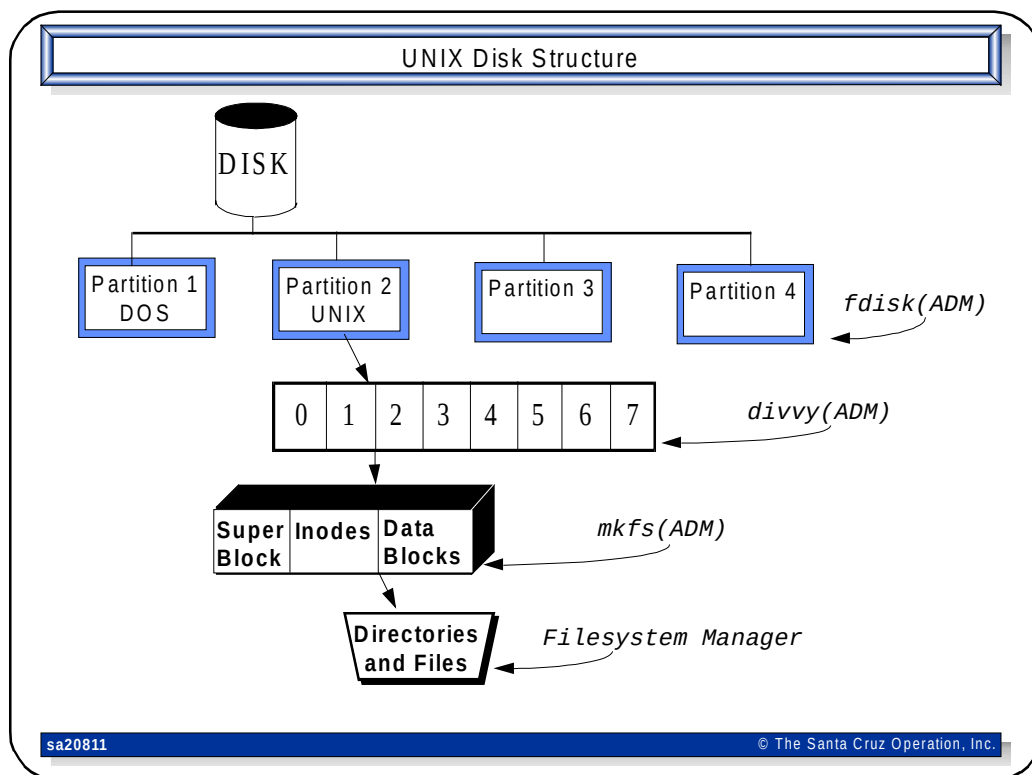
If the disk is the first device added to a SCSI bus, supply the following information about the host adapter:

- Interrupt vector
- Start I/O hexadecimal address
- End I/O hexadecimal address

When configuring SCSI, **mkdev** is run twice:

- First, to set up the SCSI configuration files and link the device drivers into the kernel; a kernel rebuild is required
- Second, to set up partitions and filesystems

AV 8-11





### 8.6.5 Commands invoked by mkdev(ADM)

---

When **mkdev hd** runs, and you supply information about your controller and disk, several commands are automatically invoked. These programs set disk parameters, scan the disk for bad tracks, if necessary, and create partitions and filesystems.

The commands invoked by **mkdev hd** are:

- **dkinit**(ADM)
- **fdisk**(ADM)
- **badtrk**(ADM)
- **divvy**(ADM)

### 8.6.6 Completing the installation

---

Once **mkdev**(ADM) completes, do the following in the order listed:

1. Use the **Filesystem Manager** to add the new filesystem(s).
2. Mount the filesystem(s) with the **Filesystem Manager** or the **mount**(ADM) command.

SCO recommends that you shut down your system and reboot to automatically mount your new filesystem. This ensures that the */etc/default/filesys* file is properly configured.

## 8.7 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.

### 8.7.1 The **badtrk(ADM)** command

---

Although **badtrk(ADM)** is primarily used during installation and when configuring additional disks, it can be used to scan the disk and add any new bad tracks or bad blocks to its tables. When bad track or bad block addresses are added, the operating system avoids the areas of the disk that cannot be written to.

For IDE, EIDE, and ESDI disks, **badtrk** maps any flawed tracks to good ones elsewhere on the disk.

For SCSI disks, if supported by the disk controller, **badtrk** tries to use spare disk blocks that are maintained by the disk controller as replacements for bad blocks.

You can run a quick or thorough scan. Typical scan rates are 18MB per minute for quick, and 6MB per minute for thorough.

When running **badtrk** on a disk that has data, select the non-destructive scan. Selecting the destructive scan destroys the data on the disk.

When running **badtrk** you can scan:

- An entire UNIX partition
- A specified range of blocks
- An entire specified filesystem

#### 8.7.1.1 Bad tracking a disk

---

If a bad track develops on a non-SCSI disk, an error message displays similar to the one listed.

```
wd: Error on fixed disk ctrl=0 dev=1/43 block=31434
cmd=00000020 status=000005180, sector=62899,
cylinder/head=483/4
```

The portion of the error message “dev=1/43” denotes the area of the disk where the error has occurred. In this case, instead of selecting “Scan the Entire Partition,” select “Scan an Entire Filesystem.”

The major and minor number indicates the `/u` filesystem.

### **8.7.1.2 Steps to running badtrk(ADM)**

---

The following steps describe how to scan for bad tracks on a disk:

- Step 1. Shut down the system and reboot to single-user mode.
- Step 2. Perform a full system backup.
- Step 3. Ensure that the non-root filesystems are unmounted.
- Step 4. Execute **badtrk(ADM)**.
- Step 5. Select **Scan Disk**, then select the area to scan.
- Step 6. Select a **Quick scan** or **Thorough scan**.
- Step 7. Select **Non-Destructive Scan**.
- Step 8. Run **fsck(ADM)** on all filesystems when the scan completes.
- Step 9. Mount all non-root filesystems.
- Step 10. Check the *lost+found* directory for all filesystems to locate any unreferenced files.
- Step 11. Return to multiuser mode:

**init 2**

Refer to the **badtrk(ADM)**, **badblock(ADM)**, and **scsidisk(HW)** manual pages for more information on bad tracking disks.

## 8.8 Lesson Summary

---

SCO OpenServer systems includes filesystem types that incorporate the latest in filesystem technology while ensuring compatibility with earlier releases of the operating system. Filesystems are comprised of a superblock, inodes, and data blocks.

The UNIX kernel does not directly access server hardware devices such as hard disks, terminals, and printers. Instead, the kernel uses 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`.

When you configure filesystems, you need to consider checkpointing, intent logging, and versioning. Checkpointing is the process of transitioning filesystems to a clean (consistent) state when the filesystem is inactive. Intent logging minimizes system downtime. Versioning allows you to recover deleted files with the **undelete(C)** command.

SCO OpenServer 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.

You can use the **badtrk(ADM)** command to troubleshoot hard disk problems.

## Review questions

---

1. Which inode represents the root directory of each mounted filesystem?
2. What command do you use to create special device files?
3. Which filesystems support intent logging?
4. Which file contains the information used when mounting a filesystem?

5. When configuring a SCSI disk, what information do you need to supply about the SCSI host adapter?
  
6. List in order the programs invoked when you run **mkdev hd** to configure a hard disk attached to an IDE controller.
  
7. When bad tracking a disk that has data, what type of scan should be performed?

## 8.9 Lab: Configuring a Hard Disk (Non-SCSI)

---

Configure a hard disk attached to a non-SCSI controller. Your instructor will provide you with the controller/host adapter and disk number. Partition and filesystem parameters are also provided.

### 8.9.1 Overview

---

Run **mkdev hd** to configure a second hard disk. Create a single filesystem. Name the filesystem *sales*.

Invoke the **Filesystem Manager**. The filesystem will be mounted to */u/sales*. This filesystem should mount when going to multiuser mode, and all users should be able to mount and unmount the *sales* filesystem.

### 8.9.2 Procedure

---

The following steps describe how to configure a non-SCSI hard disk on your system:

- Step 1. Log in as **root**.
- Step 2. Enter **mkdev hd**.
- Step 3. Select the type of controller provided by your instructor.
- Step 4. Enter the information provided by your instructor about your controller and disk.
- Step 5. Select the default disk parameters when **dkinit**(ADM) is invoked (unless your instructor advises you otherwise).
- Step 6. Choose to overwrite all of the present contents of your hard disk if prompted. Wait for **fdisk**(ADM) to start.
- Step 7. Create a single partition that uses the entire disk. Wait for **badtrk**(ADM) to start.
- Step 8. Run a quick and destructive scan (unless your instructor advises you otherwise). Wait for **divvy**(ADM) to start.
- Step 9. Create the *sales* filesystem. Wait for **mkdev**(ADM) to exit.
- Step 10. Add the filesystem using the **Filesystem Manager**.

- Step 11. Enter */dev/sales* as the name of the filesystem.
- Step 12. Enter */sales* as the name of the directory to which the filesystem is mounted.
- Step 13. Select the option that allows users to mount this filesystem.
- Step 14. Reboot your system to ensure your mount configuration is correct.
- Step 15. Copy the file */usr/pub/ascii* to the */sales* filesystem.
- Step 16. Access the filesystem and list the contents.

## 8.10 Lab: Configuring a Hard Disk (SCSI)

---

Configure a second hard disk attached to a SCSI host adapter. Your instructor will provide the host adapter type, the host adapter number, the bus number of the host adapter, the target ID, and the logical unit number (LUN).

### 8.10.1 Overview

---

Run the **mkdev hd** command line to configure a second hard disk. Create a single filesystem. Name the filesystem *sales*.

Invoke the **Filesystem Manager**. The filesystem will be mounted to */u/sales*. This filesystem should mount when going to multiuser mode, and all users should be able to mount and unmount the *sales* filesystem.

### 8.10.2 Procedure

---

The following steps describe how to configure a SCSI hard disk on your system:

- Step 1. Log in as **root**.
- Step 2. Enter **mkdev hd**.
- Step 3. Select **2** to add a hard disk to a SCSI controller.
- Step 4. Enter the information provided by your instructor about the host adapter and disk.
- Step 5. Enter **y** to update the SCSI configuration and follow the prompts to rebuild a new kernel. Wait for **mkdev(ADM)** to exit.
- Step 6. Load the new kernel and reboot your system.
- Step 7. Enter **mkdev hd** again and enter the information that was previously entered in Step 3. If prompted to overwrite the present contents of your hard disk, answer **yes**.  
  
Wait for **fdisk(ADM)** to start.
- Step 8. Create a single partition that uses the entire disk. Wait for **badtrk(ADM)** to start.



- Step 9. Run a quick and destructive scan (unless your instructor advises you otherwise). Wait for **divvy**(ADM) to start.
- Step 10. Create the *sales* filesystem. Wait for **mkdev** to exit.
- Step 11. Add the filesystem using the **Filesystem Manager**.
- Step 12. Enter */dev/sales* as the name of the filesystem.
- Step 13. Enter */sales* as the name of the directory to which the filesystem is mounted.
- Step 14. Select the option that allows users to mount this filesystem.
- Step 15. Reboot your system to ensure your mount configuration is correct.
- Step 16. Copy the file */usr/pub/ascii* to the */sales* filesystem.
- Step 17. Access the filesystem and list the contents.

## 8.11 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                       |                                                                                                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>badblk</b> (ADM)   | Reports inodes that contain bad disk blocks specified by their logical block numbers (blkno)                                                                           |
| <b>badtrk</b> (ADM)   | Scans the hard disk for flaws and creates a badtrack/bad block table                                                                                                   |
| <b>dkinit</b> (ADM)   | Displays or changes hard disk parameters                                                                                                                               |
| <b>divvy</b> (ADM)    | Divides a UNIX partition                                                                                                                                               |
| <b>fdisk</b> (ADM)    | Displays, creates, or deletes partitions                                                                                                                               |
| <b>fsck</b> (ADM)     | Checks and repairs filesystems                                                                                                                                         |
| <b>grep</b> (C)       | Searches for a pattern                                                                                                                                                 |
| <b>mkdev</b> (ADM)    | Calls scripts to add peripheral devices; when used with the <b>hd</b> argument, configures a hard disk; when used with the <b>fs</b> argument, configures a filesystem |
| <b>mkfs</b> (ADM)     | Constructs a filesystem on a special device                                                                                                                            |
| <b>mknod</b> (C)      | Creates a directory entry and a corresponding inode for a special file                                                                                                 |
| <b>mount</b> (ADM)    | Mounts a filesystem                                                                                                                                                    |
| <b>ncheck</b> (ADM)   | Generates a pathname and inode number list for all files in the specified filesystem                                                                                   |
| <b>scoadmin</b> (ADM) | Invokes SCOadmin applications (in this lesson, the <b>Filesystem Manager</b> )                                                                                         |
| <b>umount</b> (ADM)   | Unmounts a filesystem                                                                                                                                                  |
| <b>undelete</b> (C)   | Allows you to recover (or undelete) a previously unlinked (or deleted) file; available only for versioned files                                                        |

## Relevant files

---

|                             |                                                                             |
|-----------------------------|-----------------------------------------------------------------------------|
| <i>/etc/checklist</i>       | Contains filesystem names used by <b>fsck</b> (ADM) and <b>ncheck</b> (ADM) |
| <i>/etc/default/filesys</i> | Contains default information for mounting filesystems                       |
| <i>/etc/default/scsihas</i> | Contains supported SCSI host adapters                                       |

## 8.12 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Associating file names with inode numbers

---

To determine the inode numbers of the root (/) and */stand* directories, enter:

```
l -id /
```

```
l -id /stand
```

### Exercise: Listing device files

---

To perform a long listing and display the contents of the */dev* directory, enter:

```
l dev | more
```

### Exercise: Comparing major and minor numbers

---

The floppy disk device */dev/dsk/f0q\** has one major number (2) and five minor numbers.

### Exercise: Determining the number of filesystem

---

By default, the root hard disk has a root and boot filesystem. A */u* filesystem can also exist along with any other filesystem divisions.

### Exercise: Viewing */etc/default/filesys*

---

Normally users cannot mount the */u* filesystem. If the mount parameter is equal to yes, ordinary users can mount a filesystem.

## Exercise: Repairing the /u filesystem

---

To unmount and repair the /u filesystem, enter:

```
umount /dev/u
```

```
fsck -ofull /dev/u
```

## Exercise: Associating inode numbers with filenames

---

To determine the inode number for the /etc/default/tar file and search the root filesystem with **ncheck**(ADM) and display the results, enter:

```
l -i /etc/default/tar
```

```
ncheck -i inode_number /dev/root
```

## Review questions

---

This section provides suggested answers to the review questions.

1. Inode 2 represents the root directory of each mounted filesystem.
2. You can create special device files with the **mknod**(C) command.
3. The HTFS, DTFS, AFS, and EAFS filesystems support intent logging.
4. The /etc/default/filesys file contains the information used when mounting a filesystem.
5. When configuring a SCSI disk, you need to supply the following information about the SCSI host adapter: the host adapter number, the number of the bus on the host adapter (if you are installing a controller on a host adapter that controls two SCSI buses), the target ID, and the logical unit (LUN) number of the disk.
6. When configuring a hard disk attached to an IDE controller, the following programs are invoked, in this order, by **mkdev**(ADM): **dkinit**(ADM), **fdisk**(ADM), **badtrk**(ADM), and **divvy**(ADM). The **divvy**(ADM) command invokes **mkfs**(ADM) to create the filesystem.
7. When bad tracking a disk that has data, you should perform a non-destructive scan.



---

# 9

## *Installing and Managing Serial Ports*

AV 9-1

Installing and Managing Serial Ports

- Identify standard serial ports
- Describe the levels of hardware communication
- Describe the login sequence
- Install a serial adapter card
- Install serial terminal hardware
- Manage terminal configuration files and utilities
- Troubleshoot common serial terminal problems

sa20001© The Santa Cruz Operation, Inc.



**Lesson 9**

## **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 step-by-step procedures for installing and configuring adapter cards and terminal hardware, as well as 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
- Describe the levels of hardware communication
- Describe the login sequence
- Install a serial adapter card
- Install serial terminal hardware
- Manage terminal configuration files and utilities
- Troubleshoot common serial terminal problems

## 9.1 Standard Serial Ports

---

By default, SCO OpenServer systems include support for the console and standard serial ports.

All serial device names are stored in the `/dev` directory and begin with the prefix `tty`. Each serial port has two device files associated with each port. One is normally used to attach a terminal or printer; the other is used to attach to a modem.

Standard serial ports include the following:

|                         |                                                                       |
|-------------------------|-----------------------------------------------------------------------|
| <code>/dev/tty1a</code> | Represents the first standard serial port, known as COM1              |
| <code>/dev/tty1A</code> | Represents the first standard serial port and provides modem control  |
| <code>/dev/tty2a</code> | Represents the second standard serial port, known as COM2             |
| <code>/dev/tty2A</code> | Represents the second standard serial port and provides modem control |

### Exercise: Listing serial ports

---

List the serial ports in the `/dev` directory. How can you distinguish modem control serial ports from non-modem control serial ports?

### 9.1.1 Using multiscreens

---

With the **multiscreen**(M) feature, you can access up to twelve different screens, each corresponding to a separate device file. Each can be accessed one at a time through the console. To access a multiscreen, enter:

  <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/tty01` through `/dev/tty12`.

## 9.1.2 Enabling and disabling serial ports

---

To enable a serial port for login purposes, run the **enable(C)** command. For example, to enable */dev/tty1a*:

**enable tty1a**

These files are updated:

- */etc/inittab*
- */etc/conf/init.d/sio*

To disable a serial port, run the **disable(C)** command. For example, to disable */dev/tty1a*:

**disable tty1a**

These files are updated:

- */etc/inittab*
- */etc/conf/init.d/sio*

## 9.2 Hardware Communication

---

---

At a basic level, hardware devices use two levels of communication when they connect to UNIX servers:

- Line characteristics
- Escape codes

### 9.2.1 Line characteristics

---

The line characteristics of a hardware device (such as a serial terminal) are the qualities or attributes of the communications connection between the device itself and the server. Line characteristics include:

- Baud rate
- Parity
- Stop bits
- Data bits

The **stty**(C) command sets these terminal I/O options. The **stty**(C) command is discussed in detail later in this section.

Refer to the **getty**(M) manual for more information about the default UNIX line characteristics.

## 9.2.2 Escape codes

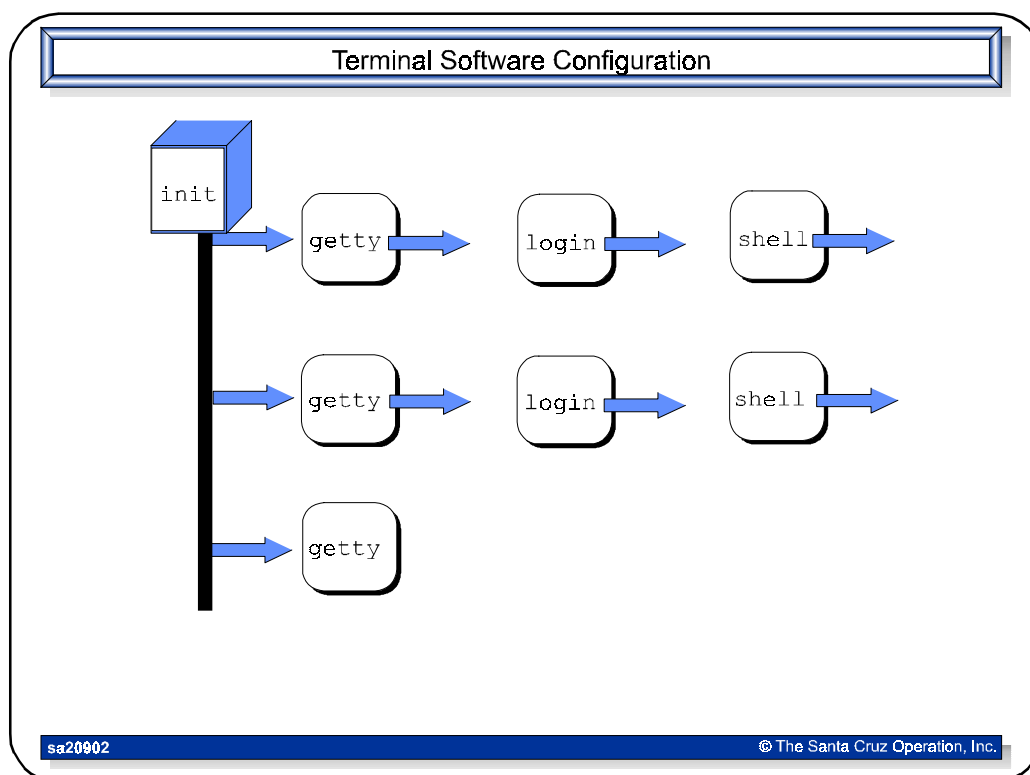
---

Escape codes determine the terminal display characteristics. The escape codes that serial terminal devices use vary from terminal type to terminal type.

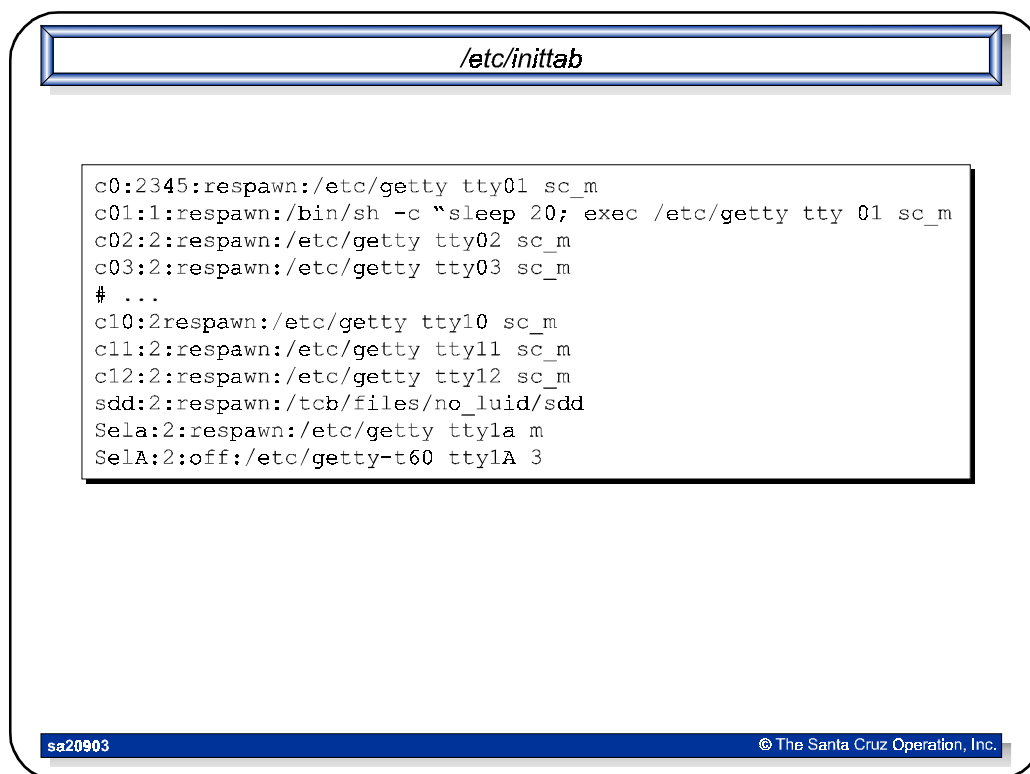
The UNIX operating system uses a series of files as databases to correctly send these escape codes. These database files are stored in the `/usr/lib/terminfo/?/*` directories and files.

The terminfo terminal capability database is discussed later in this section.

## AV 9-2



## AV 9-3



## 9.3 Examining the UNIX Login Process

---

To begin the UNIX login process, **init**(M) reads certain files for instructions and executes a number of programs as the system enters multiuser mode. In order to troubleshoot problems associated with terminals, you need to first understand the login process.

The programs and files associated with each step are discussed in this section.

### 9.3.1 The */etc/inittab* file

---

When entering run-level 2, the **init**(M) program examines */etc/inittab* for entries associated with terminals. This is determined by the third field of each line. If a terminal was configured and enabled for */dev/tty1a*, **init** would scan the following entry:

```
Se1a:234:respawn:/etc/getty tty1a m
```

The *respawn* action causes **init** to spawn the **getty**(M) program for */dev/tty1a*.

One of the functions of **init** is to pass to **getty** the last character and the name of the tty as its arguments. In this example, the name of the tty is *tty1a*, and the character argument is *m*.

#### **Exercise: Identify the getty(M) process**

---

Use the **ps**(C) command to determine if a **getty**(M) is running on */dev/tty05*.

### 9.3.2 The **getty**(M) program

---

The **getty**(M) program is responsible for initializing the terminal communications line. It uses the character argument from */etc/inittab* as a pointer to the */etc/gettydefs* file to determine the line characteristics of the serial port. The **getty** program also displays the login prompt.

## AV 9-4

**/etc/gettydefs**

**Sample from /etc/gettydefs**

```
m # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # m
sc_m # B9600 HUPCL SCANCODE # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY SCANCODE
      #\r\n!login: # sc_m

# 4-5-6:  2400 - 4800 - 9600 baud cycle (direct connect via patchboard)
#
4 # B2400 HUPCL # B2400 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # 6
5 # B4800 HUPCL # B4800 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # 4
6 # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # 5
```

sa20904 © The Santa Cruz Operation, Inc.

## AV 9-5

**Baud Rate Settings in /etc/gettydefs**

```
.
.
.
0 # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # 0
#
1 # B300 HUPCL_OPOST CR1 NR1 #
   B300 CS8 SANE HUPCL TAB3 IXANY #\r\n!login: # 2
2 # B1200 HUPCL_OPOST CR1 ECHOE NL1 #
   B1200 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # 3
3 # B2400 HUPCL_OPOST CR1 ECHOE NL1 #
   B2400 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # 1
.
.
.
m # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # m
n # B19200 HUPCL # B19200 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # n
o # B38400 HUPCL # B38400 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n!login: # o
```

sa20905 © The Santa Cruz Operation, Inc.



### 9.3.3 The *gettydefs* file

The */etc/gettydefs* file contains information used by **getty**(M) to set up the speed and terminal settings for a serial line. It also supplies information on what the login prompt should look like.

In keeping with the example for */dev/tty1a*, **getty** looks for the line beginning with the character argument *m*. Each field is separated by a *#* sign and contains several fields. For example, the line:

```
m # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY
#r\n\n!login: # m
```

contains these fields:

|                                         |                                                                                                                                   |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| m                                       | The <i>label</i> field contains the argument by which <b>getty</b> finds the entry to use.                                        |
| B9600 HUPCL                             | The <i>initial flags</i> are the baud rate and certain operational parameters to set before running the <b>login</b> (M) program. |
| # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY | The <i>final flags</i> are the parameters to set after running the <b>login</b> program.                                          |
| # \r\n\n!login:                         | The <i>login prompt</i> that displays on the terminal.                                                                            |
| # m                                     | The <i>next label</i> field tells <b>getty</b> to repeat the entry if it is unsuccessful.                                         |

#### Exercise: Determining baud rates

Use **grep**(C) to determine the label field in the */etc/gettydefs* file that represents a baud rate of 38400.

### 9.3.4 The login(M) process

---

If a login name is typed in at the login prompt, **getty**(M) execs the **login**(M) program using the login name as its argument. The **login** command:

- Displays the password prompt and turns off echoing so the password is not displayed
- Validates and encrypts the user's password according to the information found in the password files
- Executes the shell found in the */etc/passwd* file for the particular user if the login and password information are correct
- Continues to run after the user logs in, so that the **License Manager** daemon is kept current as to the number of user licenses in use

Refer to the **login**(M) manual page for more information on its capabilities.

**Exercise: Determining when the login(M) program starts**

---

Start a login process on `/dev/tty05`. Do not type in your password. Return to `/dev/tty01` and run **ps(C)** on `/dev/tty05`. Is the **login(M)** program running? If so, why?

AV 9-6

| Terminal Administration |                                                       |                                                              |
|-------------------------|-------------------------------------------------------|--------------------------------------------------------------|
| Program                 | Files                                                 | Display                                                      |
| init                    | /etc/inittab                                          |                                                              |
| getty                   | /etc/systemid<br>/etc/default/issue<br>/etc/gettydefs | hostname<br>Welcome to SCO OpenServer<br>Release 5<br>login: |
| login                   | /etc/passwd                                           | Password:                                                    |
| ksh                     | /etc/profile<br>\$HOME/.profile                       | \$                                                           |

sa20006

© The Santa Cruz Operation, Inc.

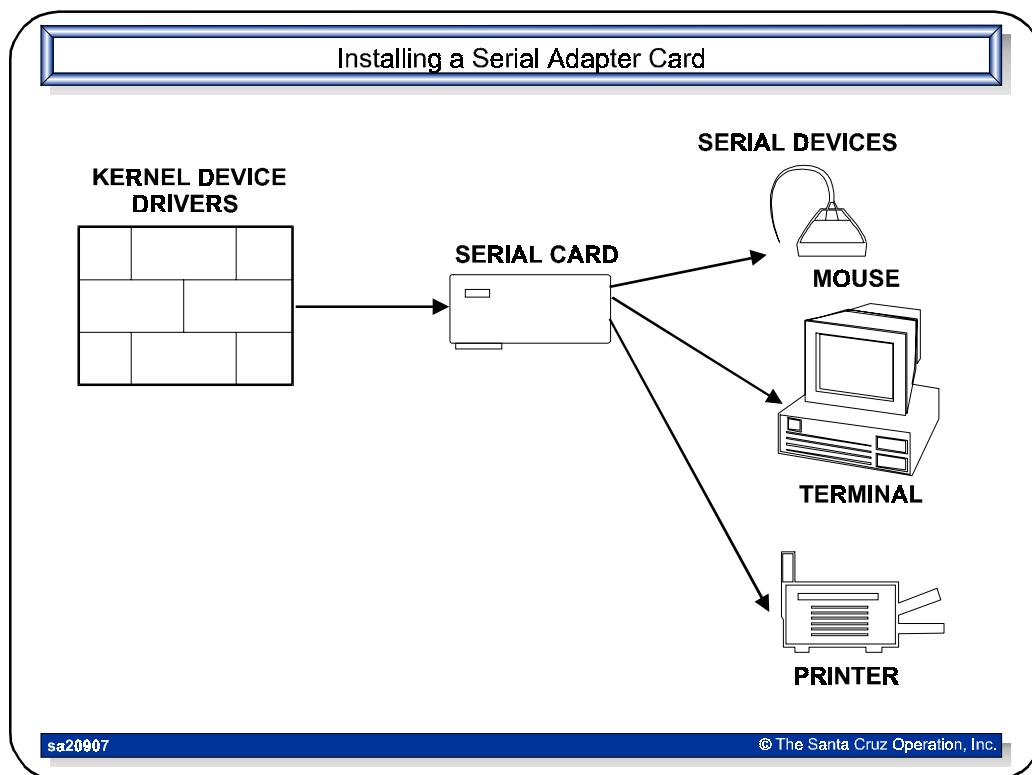
### 9.3.5 Shell programs

---

The shell program sets up the user's environment and displays the UNIX command prompt.

Refer to the **cs**h(C), **ks**h(C), and **sh**(C) manual pages for a review of the functions of a shell program when a user logs in.

AV 9-7



## 9.4 Installing a Serial Adapter Card

---

Most servers 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 serial mouse, serial printer, or serial terminal to these ports.

To connect more than two serial devices to your server, 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 32 ports. Some cards can add up to 256 ports.

### 9.4.1 Serial adapter cards

---

The two basic types of serial adapter cards are *dumb* cards and *smart* cards.

Dumb (standard or non-intelligent) 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 by setting DIP switches so that the kernel recognizes the card as COM1 or COM2 (typically, you can only install two dumb cards on a server)

Smart (intelligent) cards:

- Have on-board processors and memory and use these resources to manage serial I/O through the server
- Include a device driver and configuration software that are installed either with vendor-specific software, or with the **custom**(ADM), **install**(ADM), **xinstall**(ADM), or **installpkg**(ADM) commands. The method of installation is manufacturer-dependent.
- May or may not use interrupts, depending on the manufacturer

Refer to the **SCO OpenServer Handbook**, “Adding serial and parallel ports,” for a complete list of supported adapter cards. Note that each serial adapter card is unique—that is, the SCO OpenServer software has special driver code for each adapter listed.

## 9.4.2 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)

When possible, select the default device interrupts and settings. When you have installed the card, record the installation in your system administration log book.

Refer to the **SCO OpenServer Handbook**, “Adding serial and parallel ports,” for a complete discussion of hardware installation.

### **Exercise: Installing a serial port adapter card**

---

Using the instructions provided by the instructor, install a serial adapter card in your server. Be sure to verify all switches and options on the card before closing up the server.

Update your system administrator log book when you are done.



### 9.4.3 Configuring the software

---

Before the UNIX kernel can recognize your smart or dumb serial adapter card, you must create the special device file that will represent the card and its switch settings.

You can create these new device files for the additional ports as follows:

- For dumb cards, run the **Hardware/Kernel Manager** ⇨ **Serial Port** or **mkdev serial**.
- For smart cards, run the **Software Manager**, or the manufacturer's suggested program.

When you add serial adapter cards, you are creating serial ports that are *terminal ready*, meaning:

- You can easily connect a serial terminal to the port—which may or may not be appropriate for all serial ports on your server.
- Appropriate terminal configuration files are updated at the time you install the card.
- When these files are updated, no login prompt is sent to the port, because many serial devices such as a serial mouse or a serial printer do not require logins.
- When you do connect a supported terminal with the correct cable, you can configure the port for a login prompt using the **enable(C)** command.

When you add a serial adapter, the following changes are made on your server:

- Device special files in the */dev* directory are created to access the new ports.
- Entries are added to the */etc/inittab*, */etc/ttytype*, */etc/conf/init.d/sio*, and */etc/auth/system/ttys* files.

### **9.4.3.1 Dumb card naming conventions**

---

Serial port device special files that are created for dumb cards include:

- `/dev/tty1a`, `/dev/tty1b`, `/dev/tty1A`, `/dev/tty1B`, and so on, for COM1
- `/dev/tty2a`, `/dev/tty2b`, `/dev/tty2A`, `/dev/tty2B`, and so on, for COM2

Device special files that contain lowercase letters (such as `/dev/tty1a`) are terminal control device files, by convention.

Device special files that contain uppercase letters (such as `/dev/tty1A`) are modem control device files, by convention.

### **9.4.3.2 Smart card naming conventions**

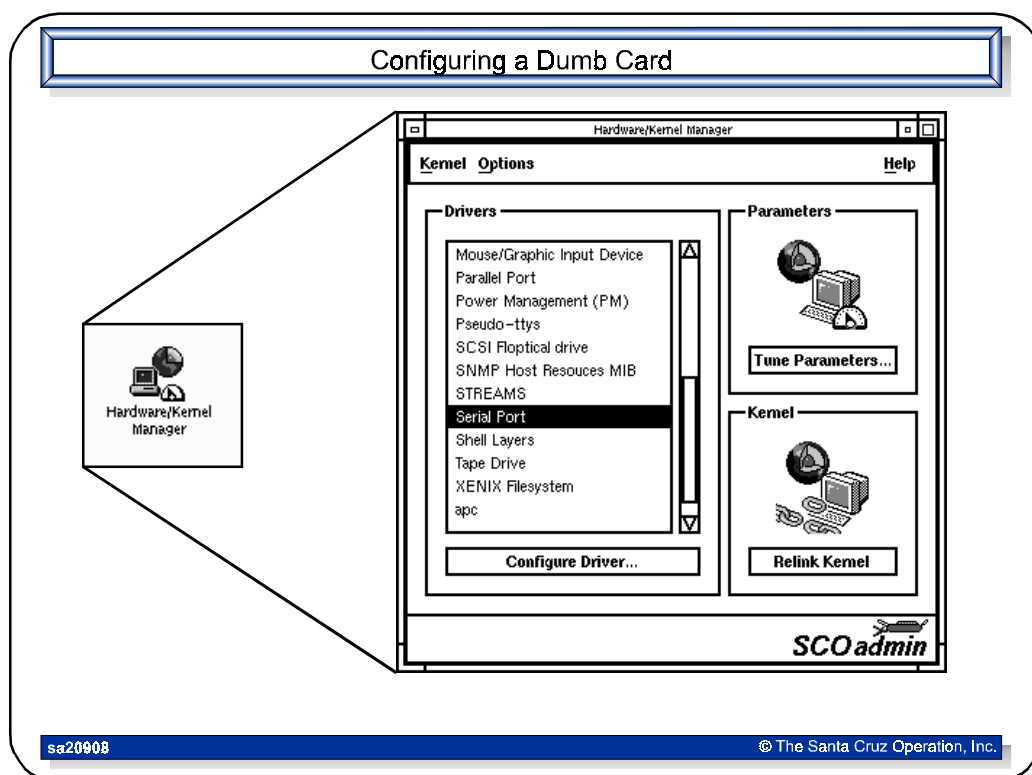
---

Some sample serial port device files that are created for smart cards by the vendor-specific installation script include:

- `/dev/ttya1`, `/dev/ttya2`, and so on
- `/dev/ttyaa`, `/dev/ttyab`, and so on

The `tty` identifiers in these examples (*a1*, *a2*, *aa*, *ab*) are examples only. The filenames are vendor-specific and will vary.

AV 9-8



---

### 9.4.4 Configuring a dumb card

---

These steps describe how to configure the dumb multiport card using the **Hardware/Kernel Manager** ⇒ **Serial Port** (note that you could also use **mkdev serial** from the command line).

Step 1. Boot the system, log in as **root**, and enter system maintenance mode.

Step 2. Start the **Hardware/Kernel Manager. Select Serial Port.**

The system determines which serial cards (if any) are configured and displays them.

Step 3. Click on **Add** and select the:

- Description of the serial card
- IRQ
- I/O Addr

Step 4. Click on **OK** to install a new serial card.

Step 5. Click on **YES** to relink the kernel. Answer ‘**y**’ to the prompts.

Step 6. Click on **OK** for a successful relink of your system. Press .

Step 7. Select **Kernel** ⇒ **Exit** to exit the **Hardware/Kernel Manager**.

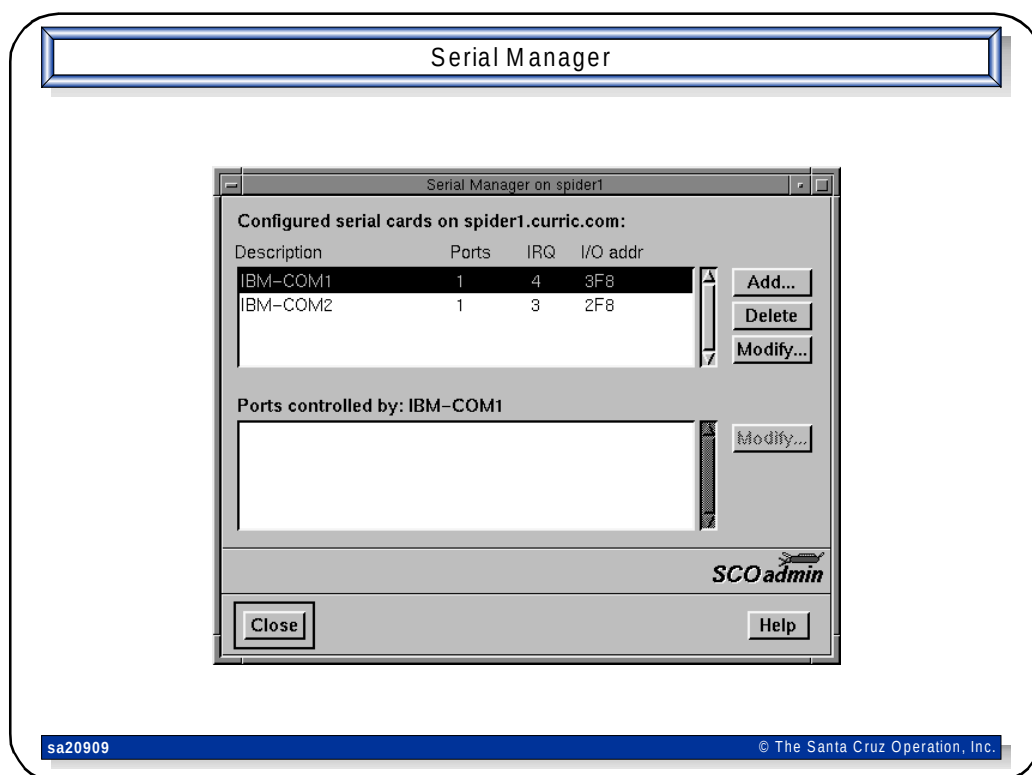
Step 8. Shut down and reboot your system.

Step 9. Verify that the SCO OpenServer bootup screen recognizes the new card. The screen display should include an entry similar to:

```
%serial 0x03f8-0x03ff 4 - UNIT=0 TYPE=Standard
nports=32
```

Refer to the **SCO OpenServer Handbook**, “Adding a serial card at a non-standard address or interrupt,” if you need to install a serial card differently from the displayed configurations.

AV 9-9



## 9.4.5 Configuring the serial port

After adding the serial card, follow these steps to configure the serial ports:

Step 1. Start the **Hardware/Kernel Manager**. Select **Serial Port**.

Step 2. Select the serial port used for the terminal, and click on **Modify**.

|            |                                                                                                                                                                                                                                          |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Speed(bps) | Try the highest speed port speed supported by your hardware. Match this speed setting with the terminal configuration . For example, if you set the terminal speed to 9600, use that setting here. The setting depends on your hardware: |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

8250 UART    Select no higher than 9600 bps

16450 UART    Select no higher than 19200 bps

16550 UART    Set up to 115200 bps

|                 |                                                                    |
|-----------------|--------------------------------------------------------------------|
| Line-mode label | Leave as is unless debugging problems with an incoming connection. |
|-----------------|--------------------------------------------------------------------|

|              |                                               |
|--------------|-----------------------------------------------|
| Login/answer | Select “On.” This enables the port for login. |
|--------------|-----------------------------------------------|

|                |                                                                                       |
|----------------|---------------------------------------------------------------------------------------|
| Receive buffer | Defer modifying this control until you need to debug problems or improve performance. |
|----------------|---------------------------------------------------------------------------------------|

Step 3. Click on **Close and Press**  .

Step 4. Select **Kernel** ⇒ **Exit** to exit the **Hardware/Kernel Manager**.

### Exercise: Adding a dumb serial card

If you installed a dumb multiport serial adapter in your server, follow the steps outlined in “Configuring a dumb card” and configure it now.

Be sure to mark the configuration specifics in your system administration log book.

### **9.4.6 Configuring a smart card**

---

- Step 1.            Boot the system, log in as **root**, and enter system maintenance mode.
- Step 2.            Enter the installation program suggested by the smart card manufacturer.
- Step 3.            Supply answers based on the card specifics and the manufacturer's suggestions.

### **9.4.7 Troubleshooting serial card installations**

---

If the boot process does not accurately report your adapter, then:

- The switches on the card may not be set properly. Recheck the documentation that comes with the adapter for the proper switch settings.
- Try bringing the server into multiuser mode. Some smart cards are not detected until the server is in multiuser mode.



## 9.5 Installing Serial Terminal Hardware

---

You installed at least one available serial port in the previous section. You may recall that such serial ports are terminal ready, although the precise, specific kind of terminal you are connecting to is not known to the operating system yet.

This section describes terminal configuration and cabling.

### 9.5.1 About serial terminal hardware

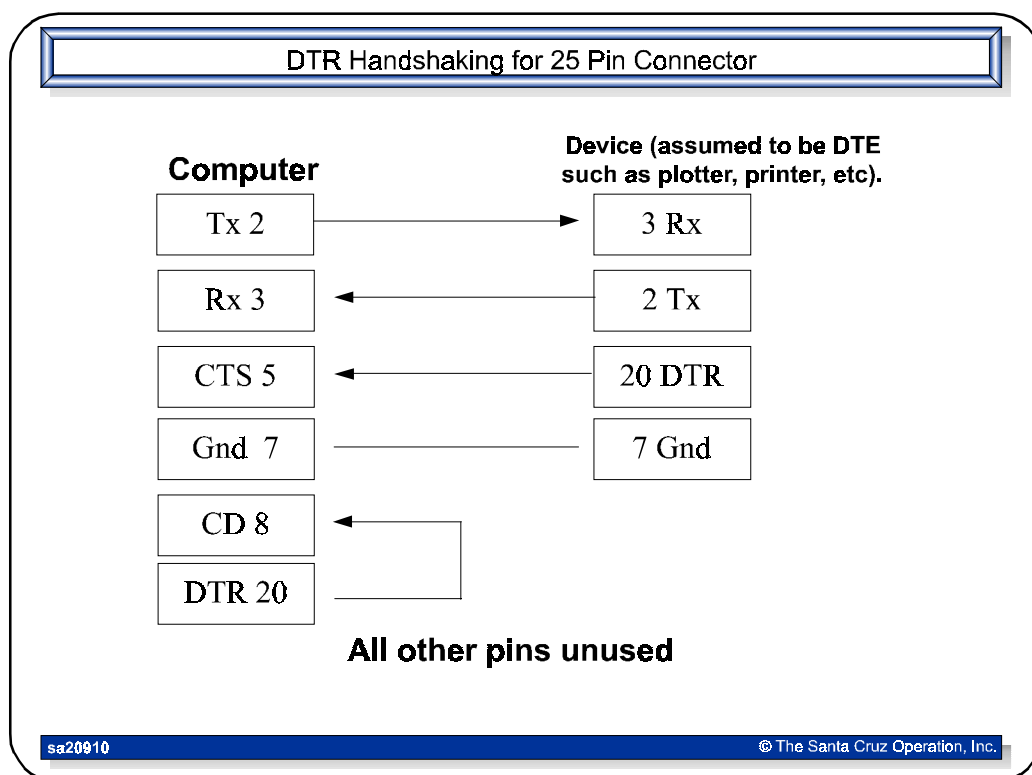
---

Before installing a terminal, do the following:

- Examine the hardware manual for instructions on connecting the terminal to a serial line.
- Refer to the **terminals**(M) manual page for a comprehensive list of supported terminals.
- Refer to the **terminfo**(M) and **terminfo**(F) manual pages for more information about the **terminfo** database.

Support for terminals is provided through the **terminfo** database, which contains the definitions and classifications of keystrokes and control sequences that vary from terminal to terminal.

AV 9-10



## 9.5.2 Ports and cables

---

The two types of serial (RS-232) ports used for the terminal are:

- DTE (data terminal equipment—for example, terminals serial mice)
- DCE (data communications equipment—for example, modems)

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 phonejack
- RJ-45
- Other vendor-specific cables

Typically, the physical connection between a terminal and a server requires only 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 many terminals, the cable must be set up so that 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 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.

Refer to the **SCO OpenServer Handbook**, “Adding serial terminals,” for more information about ports, cables, and null modems. This section includes information on installing scancode compatible terminals and running applications on scancode terminals.

AV 9-11

**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<br>DISP            | F2<br>GENERL           | F3<br>KEYBRD        |
| F4<br>COMM            | F5<br>MISC             | F6<br>TABS          |
| F7<br>A/BACK          | F8<br>F/KEYS           | F9<br>LABELS        |
| F10<br>EXIT           |                        |                     |

sa20011
© The Santa Cruz Operation, Inc.

### 9.5.3 Terminal settings

---

Use the terminal setup screens to select specific settings for the following:

- 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

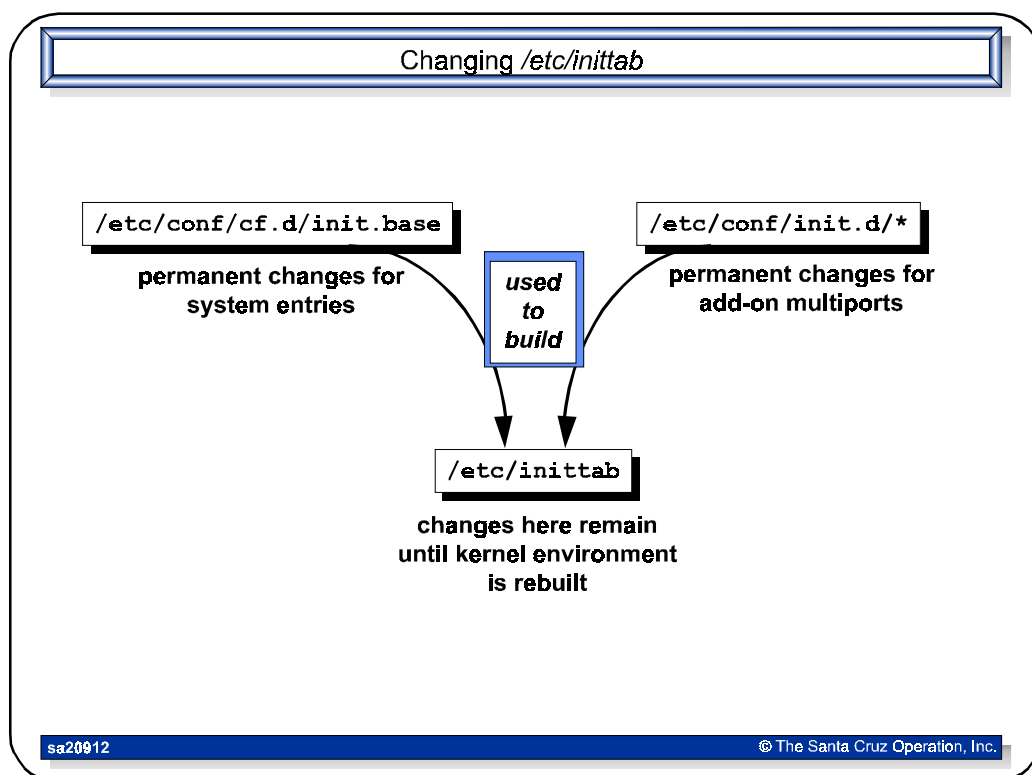
#### **Exercise: Connecting your terminal**

---

---

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 9-12



## 9.6 Managing Serial Ports and Terminals

---

Now that you understand the login process and know how to install and configure serial ports, this section provides some additional information about the files and commands associated with managing serial ports and terminals.

### 9.6.1 Changing the */etc/inittab* file

---

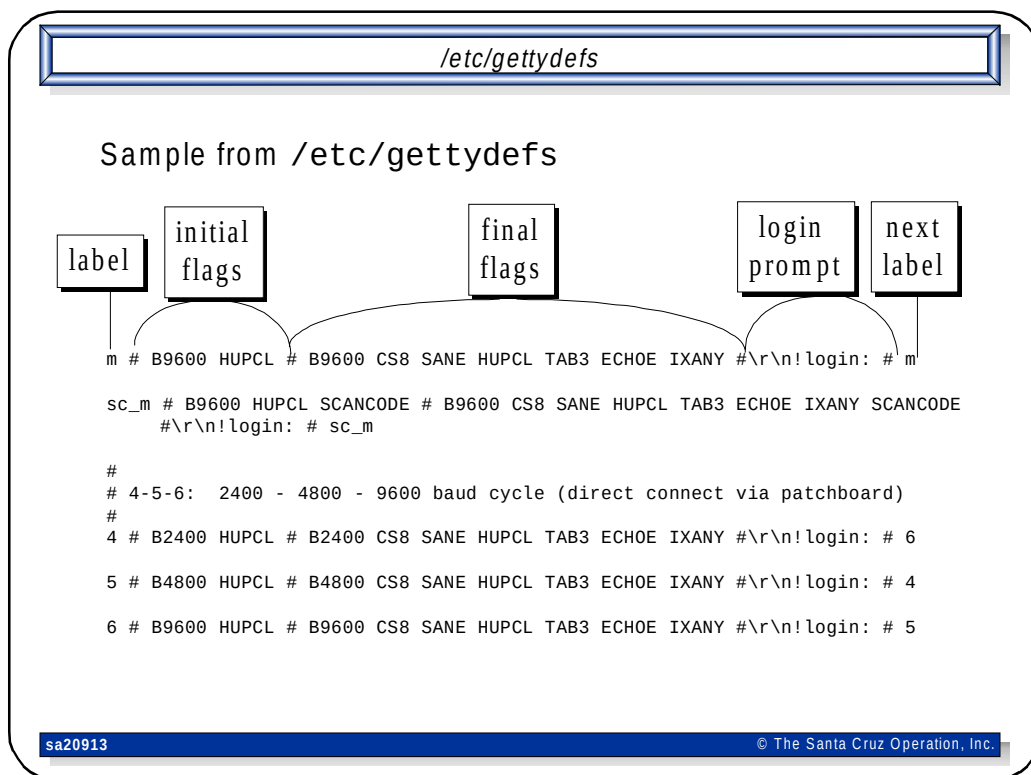
If you want to make changes to the */etc/inittab* file, make sure you are editing the appropriate files. Do not edit */etc/inittab*. The */etc/inittab* file is recreated each time the kernel environment is rebuilt.

For example, to change a serial port entry, do the following:

- Step 1.        Disable the port.
- Step 2.        Edit the applicable file in the */etc/conf/cf.d* or */etc/conf/init.d* directory.
- Step 3.        Run the command **/etc/conf/bin/idmkinitt** to create a new *inittab* file in the */etc/conf/cf.d* directory.
- Step 4.        Copy */etc/conf/cf.d/inittab* to */etc/inittab*.
- Step 5.        Enable the serial port.

Refer to the **idmkinitt**(ADM) manual page for more information on recreating the */etc/inittab* file.

## AV 9-13





## 9.6.2 Changing the */etc/gettydefs* file

---

You can edit *gettydefs* to add new terminal settings or change existing ones. In addition to baud rates, the file contains *initial* and *final flag* settings. These are some of the common flags that make up the line characteristics:

|       |                                                                                      |
|-------|--------------------------------------------------------------------------------------|
| HUPCL | Hang up line on final close (usually when a user logs out)                           |
| SANE  | A composite flag that sets a number of terminal characteristics to reasonable values |
| TAB3  | Expands tabs with spaces                                                             |
| IXANY | Enables any character to restart output                                              |

Before editing the */etc/gettydefs* file, you should first make a copy of this file.

When editing */etc/gettydefs*, to ensure that your settings are valid, execute the following command:

```
getty -c /etc/gettydefs | more
```

The system scans the file and displays the results.

Refer to **stty(C)** manual page for more information on initial and final flag settings in */etc/gettydefs*.

## 9.6.3 The **stty(C)** command

---

Instead of making changes to the */etc/inittab* and */etc/gettydefs* files, it may be necessary to check the status of a serial line or issue changes from the command prompt. You can do this using the **stty** command.

### 9.6.3.1 Displaying line characteristics

---

To display the baud rate, the parity scheme, and other information about the serial line */dev/tty1a* from the current port, enter:

```
stty -a < /dev/tty1a
```

To display settings *for* the current port, enter:

```
stty -a
```

To send settings to another serial port, enter:

```
stty -echoe < /dev/tty1a
```

### **9.6.3.2 Changing line characteristics**

---

On some terminals the backspace key does not erase the deleted character. To correct this, enter:

```
stty echoe
```

### **9.6.3.3 Restoring terminal characteristics**

---

If a terminal display becomes scrambled, send the SANE flag to restore a normal set of terminal characteristics by entering:

```
␣stty sane␣
```

### **Exercise: Using the SANE flag**

---

This exercise simulates a non-functional terminal. At the prompt, enter:

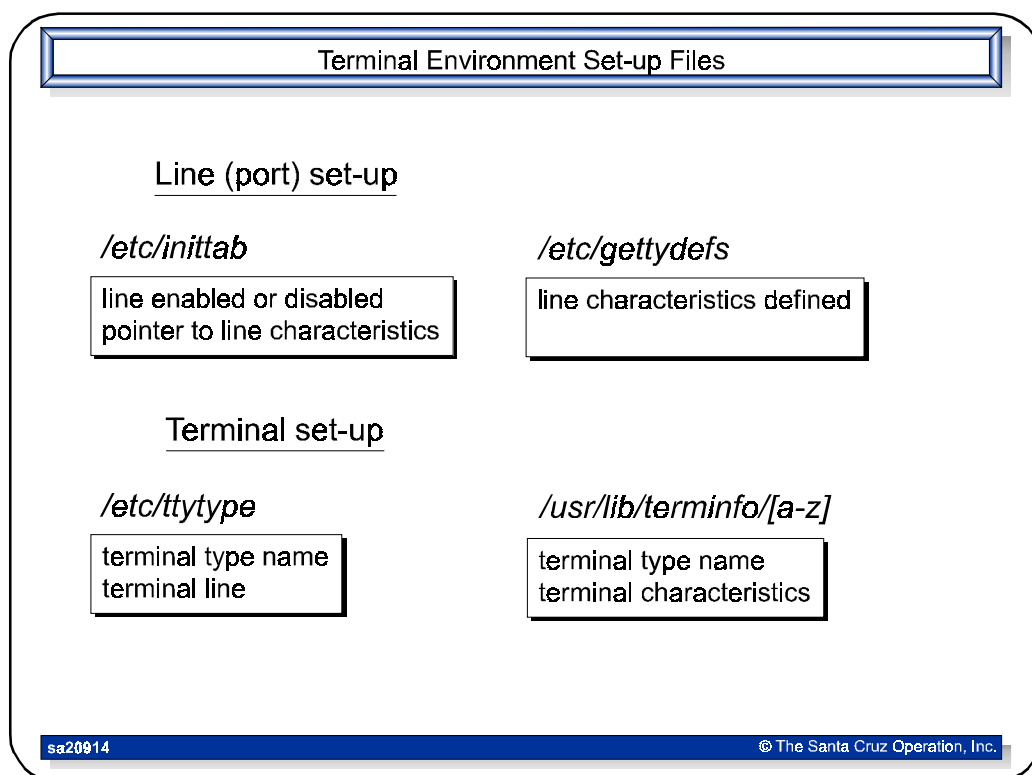
**stty raw -echo**

Type in **date**; echo is turned off so the output does not display.

To restore a normal set of characteristics, enter:

**␣stty sane␣j.**

AV 9-14



## 9.6.4 Setting the Terminal Type

---

UNIX systems require that the terminal type be clearly defined before any work can be performed by the user at the terminal. You can set the terminal type by using the `TERM` variable in the Bourne and Korn shells and the `setenv` parameter in the C shell. You can also use `tset(C)` command to set the terminal type.

### 9.6.4.1 Viewing terminal type source files

---

Numerous terminal definitions are stored in the directories `/usr/lib/terminfo/[a-z]`. These directories contain the terminal names and I/O characteristics so that a terminal will properly display. When setting a terminal type, use a terminal label from the applicable directory.

### 9.6.4.2 Associating serial ports with terminal types

---

To associate terminal types with serial ports, enter the applicable terminal labels from the `/usr/lib/terminfo/[a-z]` files into the `/etc/ttytype` file.

### 9.6.4.3 Setting the terminal type upon login

---

Terminal types are set once the shell executes and reads the user startup files. To assign a terminal type, you can change the `TERM` variable as follows:

- In the user's `.profile` for the Bourne and Korn shells, enter:

```
TERM=termtype;export TERM
```

- In the user's `.login` for the C Shell, enter:

```
setenv TERM termtype
```

Setting the terminal type with the `TERM` variable assumes that the user is always going to log in at the same type of terminal.

The **tset**(C) command can use the terminal type entered into the file */etc/ttytype*. 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* 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 terminal type, the **wy60** label is set.

## 9.7 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.

### 9.7.1 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 all settings match those listed in */etc/gettydefs*.

Step 4. Verify the hardware connection by redirecting output from another terminal:

**date > /dev/tty??**

If the port is enabled, **disable(C)** and **enable(C)** the port.

Step 5. Run the **ps(C)** command to verify that a **getty(C)** process is running on the port.

If this fails, check the documentation that came with your terminal for other troubleshooting suggestions.





AV 9-15

Hung or Scrambled Terminal

| Problem: hung terminal                       | Problem: scrambled terminal                  |
|----------------------------------------------|----------------------------------------------|
| 1. Check physical connections.               | 1. Log out and log back in.                  |
| 2. Type <b>CTRL</b> q                        | 2. Check terminal type.                      |
| 3. Press <b>DEL</b> or <b>CTRL</b> \         | 3. Check terminal setup (baud rate).         |
| 4. Kill process.                             | 4. Type <b>CTRL</b> jstty sane <b>CTRL</b> j |
| 5. Turn terminal off and on.                 |                                              |
| 6. Check terminal configuration.             |                                              |
| 7. Type <b>CTRL</b> jstty sane <b>CTRL</b> j |                                              |

sa20015© The Santa Cruz Operation, Inc.

## 9.7.2 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**(C) 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 `/etc/gettydefs` file
- Check the user's terminal type using the **env**(C) command. If the `TERM` variable is incorrect, either manually set `TERM` or source the user's `.profile` file. Once `TERM` is set correctly, run **tset**(C) with no arguments to reinitialize the terminal.
- Regain a reasonable set of serial-line characteristics with **stty sane**:

```
CTRL j stty sane CTRL j
```

## 9.8 Lesson Summary

---

Serial ports allow other users to access your system. By default, SCO OpenServer provides support for the console and several standard serial ports.

The UNIX login process uses several files, such as */etc/gettydefs* and */etc/inittab*, and invokes several commands, including **init(M)**, **login(M)**, and **getty(M)**.

Hardware devices use two levels of communication when they connect to UNIX servers—line characteristics and escape codes.

If you want to install additional ports, you must install serial adapter cards and configure the software for those cards. Also, after you install the adapter cards and install the terminal, you must perform some additional setup functions to make your terminal known to the operating system.

To install a serial card, use the SCOadmin **Hardware/Kernel Manager** or the command line **mkdev serial**.

If you want to make changes to serial port configurations, you do not edit */etc/inittab* directly. You must recreate */etc/inittab* using several other files.

You can change your terminal environment by editing files and running commands associated with terminal setup. For example, you can edit *gettydefs* to add new terminal settings or change existing ones, and you can use the **tset(C)** command to set your terminal type.

## Review questions

---

1. What is the purpose of two device files being associated with each standard serial port?
2. What happens when you enable a terminal?
3. What are the two levels of hardware communication? What differentiates them from each other?

4. List the files that are accessed during the login sequence.
  
5. After you install a serial adapter card, what do you need to do before UNIX can recognize this card?
  
6. What setup values must be established when you install a serial terminal?
  
7. How do you make permanent changes to the */etc/inittab* file?
  
8. List three common serial terminal problems and their resolutions.

## **9.9 Lab: Rebuilding */etc/inittab***

---

---

In this lab, you will rebuild the */etc/inittab* file.

### **9.9.1 Overview**

---

You have added a terminal that now supports a baud rate of 19200. The */etc/inittab* file must be recreated so that */dev/tty2a* reflects the new baud rate.

## 9.9.2 Procedure

---

- Step 1.        Disable */dev/tty2a*.
- Step 2.        Edit the file */etc/conf/init.d/sio* file and change the baud rate pointer so that */etc/getty* will set a baud rate of 19200 for */dev/tty2a*.
- Step 3.        Run the **idmkernel**(ADM) command to create a new */etc/inittab* file.
- Step 4.        Copy the */etc/conf/cf.d/inittab* file to */etc/inittab*.
- Step 5.        Change the baud rate on the terminal to 19200.
- Step 6.        Enable */dev/tty2a*.
- Step 7.        Log in on */dev/tty2a* and run the **stty**(C) command to verify the new baud rate.

## 9.10 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                         |                                                                                                                                                               |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>custom</b> (ADM)     | Installs, removes, or lists software products and components                                                                                                  |
| <b>disable</b> (C)      | Turns off terminals and printers                                                                                                                              |
| <b>enable</b> (C)       | Configures a login prompt for a serial port                                                                                                                   |
| <b>getty</b> (M)        | Sets terminal type, modes, speed, and line discipline                                                                                                         |
| <b>infocmp</b> (ADM)    | Prints out terminfo descriptions                                                                                                                              |
| <b>init</b> (HW)        | Initializes process control                                                                                                                                   |
| <b>login</b> (M)        | Gives access to the system                                                                                                                                    |
| <b>idmkinit</b> (ADM)   | Reads files containing specifications of <i>/etc/inittab</i> entrees from <i>/etc/conf/init.d</i> and constructs a new inittab file in <i>/etc/conf/cf.d</i>  |
| <b>install</b> (ADM)    | Installs a file in a specific place within a filesystem                                                                                                       |
| <b>installpkg</b> (ADM) | Installs an AT&T <sup>®</sup> UNIX system software package                                                                                                    |
| <b>mkdev</b> (ADM)      | Calls scripts to add peripheral devices; when used with the <b>serial</b> argument, creates a serial port                                                     |
| <b>scoadmin</b> (ADM)   | Invokes SCOadmin applications (in this lesson, the <b>Hardware/Kernel Manager</b> and the <b>Software Manager</b> )                                           |
| <b>stty</b> (C)         | Sets the options for a serial port                                                                                                                            |
| <b>tset</b> (C)         | Allows you to set a terminal's ERASE and KILL characters and define the terminal's type and capabilities by creating values for the TERM environment variable |
| <b>xinstall</b> (ADM)   | Installs XENIX distribution (or application program) floppies                                                                                                 |



## Relevant files

---

|                              |                                                                                                       |
|------------------------------|-------------------------------------------------------------------------------------------------------|
| <i>/dev</i>                  | The special device file directory                                                                     |
| <i>/dev/tty1a</i>            | A sample terminal control special file                                                                |
| <i>/dev/tty1A</i>            | A sample modem control special file                                                                   |
| <i>/dev/lp0</i>              | Serial/parallel adapter #1                                                                            |
| <i>/dev/lp2</i>              | Serial/parallel adapter #2                                                                            |
| <i>/dev/lp1</i>              | Parallel monochrome adapter                                                                           |
| <i>/etc/auth/system/ttys</i> | Terminal Control Database                                                                             |
| <i>/etc/conf/init.d/sio</i>  | A serial device control file used to rebuild the <i>/etc/inittab</i> file when the kernel is relinked |
| <i>/etc/gettydefs</i>        | Database to which <b>getty</b> (M) refers                                                             |
| <i>/etc/inittab</i>          | A serial device control file used by <b>init</b> (M)                                                  |
| <i>/etc/termcap</i>          | Backwards compatible terminal information database                                                    |
| <i>/usr/lib/terminfo/?/*</i> | Terminal information database                                                                         |
| <i>/etc/ttytype</i>          | A serial device control file used by the shell                                                        |

## 9.11 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Listing serial ports

---

Uppercase indicates modem control. If you have a multiport card installed that does not use this convention, point out the difference. COM1 and COM2 are distinguished by an uppercase letter, that is, tty1A and tty2A.

### Exercise: Identify the `getty(M)` process

---

To identify the `getty(M)` process on `tty05`, enter:

```
ps -t tty05
```

### Exercise: Determining baud rates

---

To determine the label field in the `/etc/gettydefs` file that represents a baud rate of 38400, enter:

```
grep '38400' /etc/gettydefs
```

The label that represents 38400 is `O`.

### Exercise: Determining when the `login(M)` program start

---

As soon as there is any type of keyboard input, `getty(C)` execs the `login(M)` program.

### Review questions

---

This section provides suggested answers to the review questions.

1. Two device files are associated with each standard port to provide support for both modem and non-modem control serial devices.
2. When you enable a terminal, the `/etc/inittab` file is updated, and login prompts are sent to the port and the terminal device to which it is attached.

3. The two levels of hardware communication are escape codes and line characteristics. Escape codes relate to the way characters display on the terminal screen and are set by **tty** and **tset**. Line characteristics have to do with the basic terminal communication with the server, and are set up by **getty** and the terminfo database.
4. The following files are accessed during the login sequence: **init** accesses */etc/inittab*; **getty** accesses */etc/gettydefs*, */etc/default/issue*, and */etc/issue*; **login** accesses */etc/utmp*, */etc/wtmp*, and */etc/passwd*; **ksh** accesses */etc/profile*, *\$HOME/.profile*, and *\$HOME/.kshrc*; **csh** accesses */etc/cshrc*, *\$HOME/.login*, and *\$HOME/.cshrc*; **sh** accesses: */etc/profile*, *\$HOME/.profile*.

Additionally, **tset** runs from a shell startup file and associates a terminal type name with a terminal serial port (or line). The **tset** command examines the */usr/lib/terminfo/?/\** database files.

Some UNIX applications access the */etc/termcap* file to obtain terminal characteristics information.

5. Before the UNIX kernel can recognize your serial adapter card, you must create a special device file to represent the board and its switch settings. You can create these new device files for the additional ports by running **mkdev serial** for dumb cards, and by running the **Software Manager** or the manufacturer's suggested program for smart cards.
6. The values you must set when installing a serial terminal vary from terminal to terminal. However, typical values you establish include baud rate, the number of data bits, the number of stop bits, parity, and full or half duplex.
7. To make permanent changes to the */etc/inittab* file, edit the applicable file in the */etc/conf/cf.d* or */etc/conf/init.d* directories, run **/etc/conf/bin/idmkin** to create a new *inittab* file in the */etc/conf/cf.d* directory, then copy */etc/conf/cf.d/inittab* to */etc/inittab*.
8. There are many possible answers to this question. Common serial terminal problems and resolutions include: terminal is hung—check physical connections, kill processes, and so forth; scrambled terminal—log out and back on, check TERM value, and so forth; the terminal has no power—check brightness adjustment, check power cord connection at both ends, check outlet, and so forth.

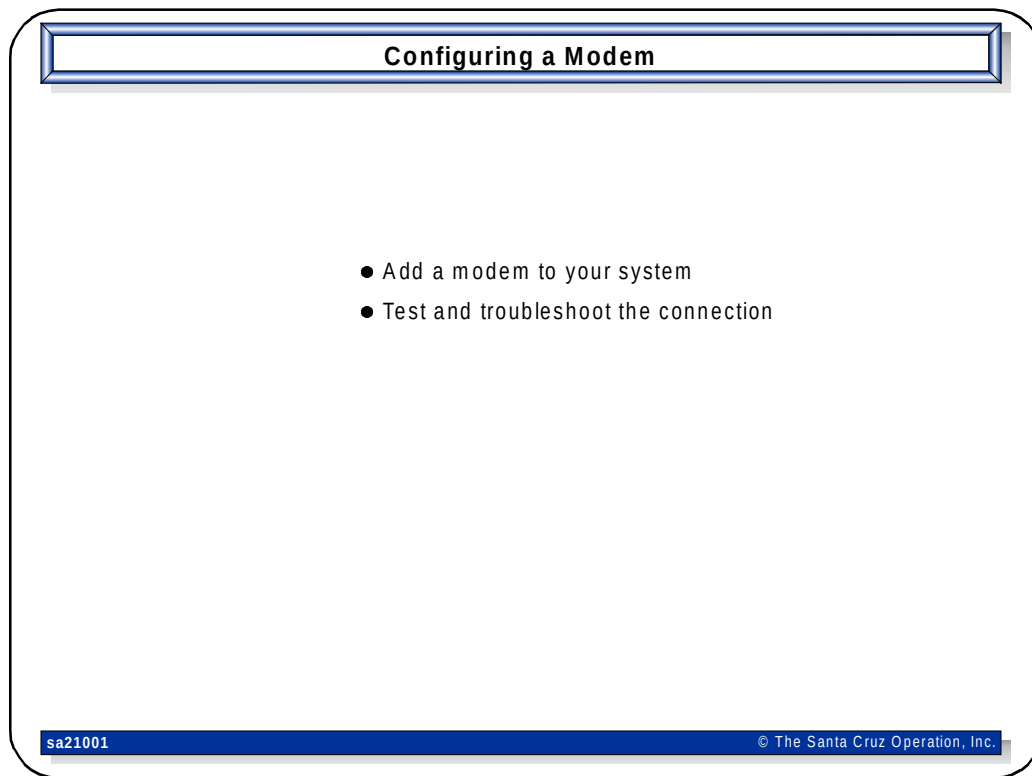


---

# ***10***

## ***Configuring a Modem***

AV 10-1



**Lesson 10**

## **Configuring a Modem**

### **Introduction**

---

As terminals let local users communicate with your system, modems allow you to communicate over phone lines and ISDN channels. Because many modem types are supported by SCO OpenServer Release 5, the **Modem Manager** lets you easily configure your modem.

### **Objectives**

---

This lesson describes modem configuration. By the end of this lesson, you will be able to:

- Add a modem to your system
- Test and troubleshoot the connection

## **10.1 Adding a Modem**

---

Modems allow you to communicate over phone lines with remote systems. To set up a modem on your system, you need to do the following:

- Choose a serial port for connecting the modem
- Configure the modem
- Install the modem



### 10.1.1 Choosing a serial port

---

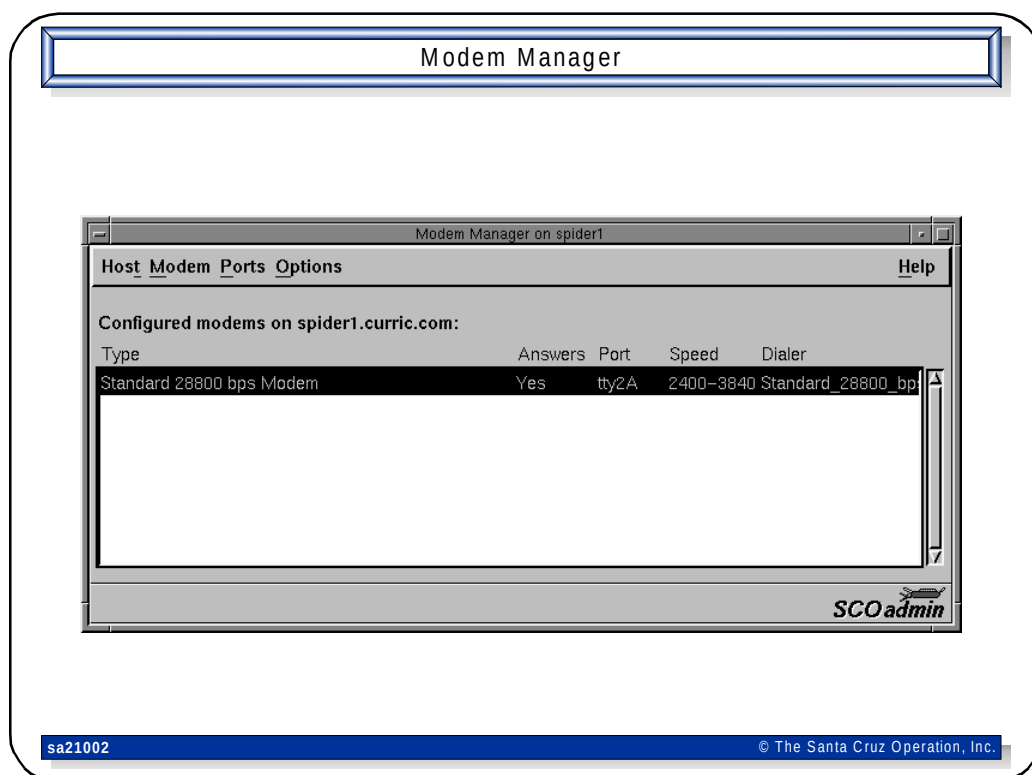
The physical connections between a device and the system depend on the hardware configuration. For specific information about connecting your serial device, refer to the hardware manuals provided with the device and with your computer. Before adding a modem, make sure you have a port available, either directly on COM1 or COM2, or from a multiport card.

Use the **hwconfig(C)** command to verify that the serial port is properly installed.

For systems with non-intelligent (standard or dumb) multiport serial cards, the non-modem control devices are `/dev/tty[1,2][a-m]`, and the modem control devices are `/dev/tty[1,2][A-M]`.

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.

AV 10-2



## 10.1.2 Configuring a modem

---

To configure a modem:

Step 1. Start the **Hardware/Kernel Manager** as **root**.

Step 2. Select the **Modem Manager**.

Step 3. Select **Automatic Detection** or **Manual Configuration**.

With automatic detection, the system queries the ports to determine the make and model of your modems. Manual configuration requires you to specify the vendor, model, and port of your modem.

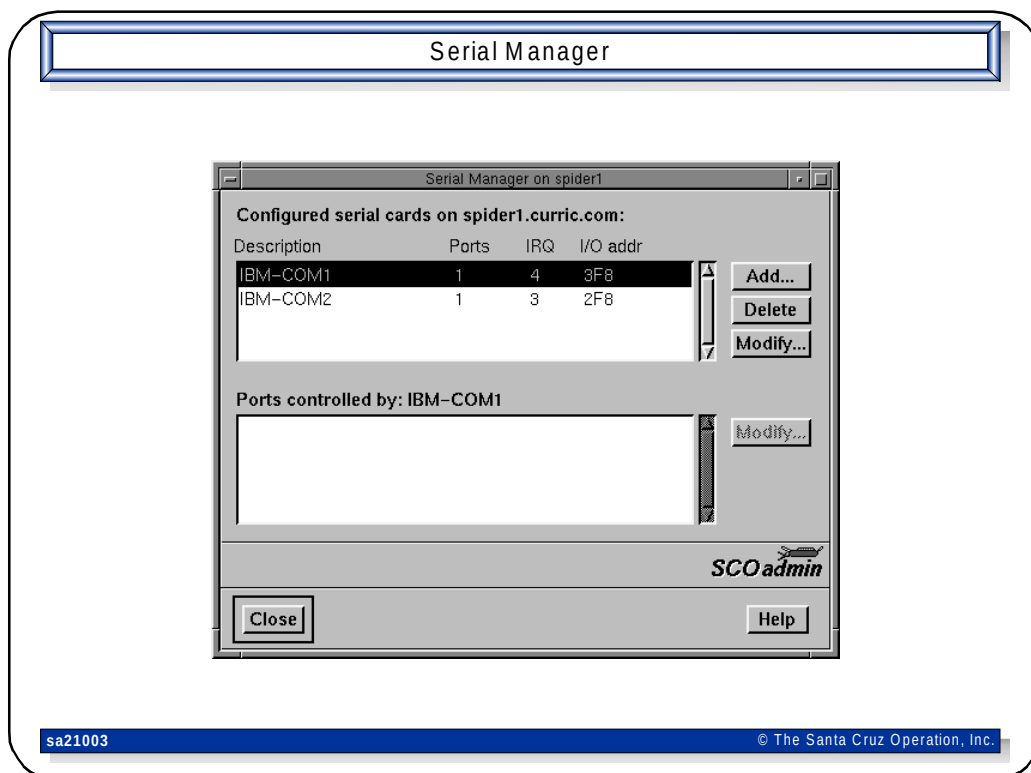
Step 4. Choose **Automatic Detection** and select which port to scan. Click on **Detect**. If automatic detection fails, proceed to Step 5.

Step 5. Configure the modem manually. Select **Modem** ⇒ **Add** ⇒ **Manual configuration** and enter:

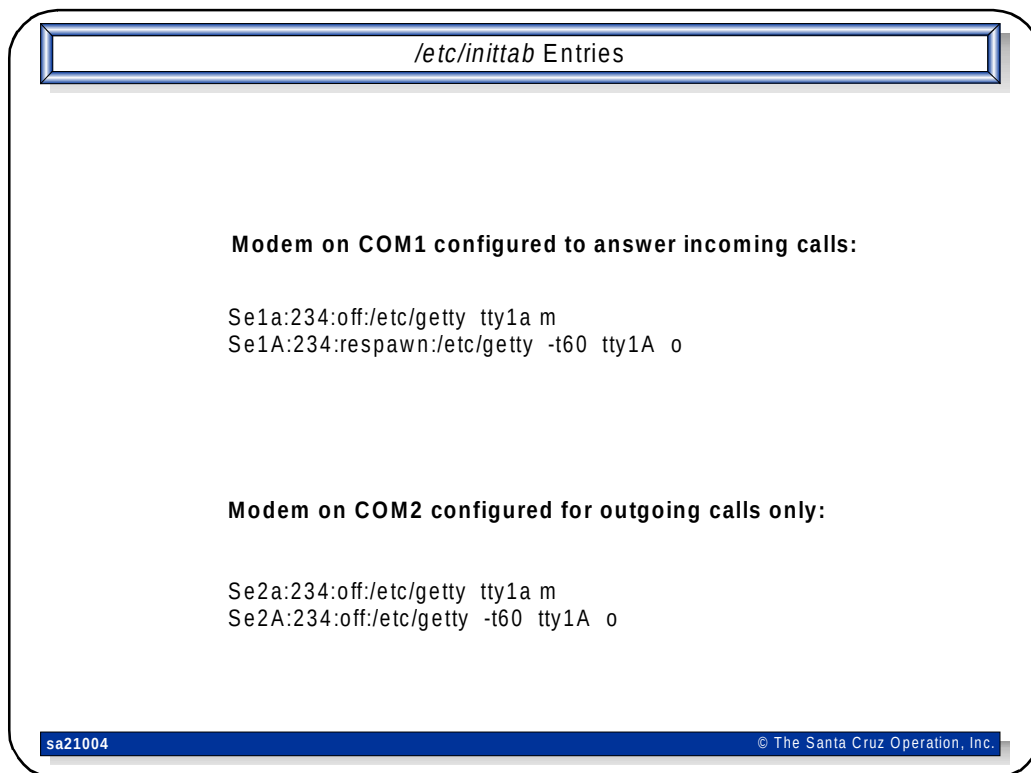
|              |                                                                                                                                                                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modem vendor | If listed, select the manufacturer. Otherwise, select “Standard Modem Types” which works for most Hayes-compatible modes.                                                                                                                                                                       |
| Modem model  | Enter the modem model or select the “Standard Modem” for the speed of your modem.                                                                                                                                                                                                               |
| Line speed   | Select “Any” for the most flexibility. This allows incoming connections from a range of remote modem speeds. For outgoing connections, the speed specified for the serial port is used. Check the serial port speed—set it as high as possible for the type of hardware you have. (See Step 8). |
| Modem port   | Select the modem control serial device (such as tty2A) that corresponds to the port to which the modem is connected.                                                                                                                                                                            |
| Answer mode  | Select “on” if configuring incoming connections. Otherwise select “Off” to prevent your system from accepting calls.                                                                                                                                                                            |

Step 6. Click on **OK** to accept the configuration.

## AV 10-3



## AV 10-4



Step 7. Modify the serial port. Click on **Configure port**. Selecting this option starts the **Serial Manager**, shown in AV 10-3.

Step 8. Select the serial port used for the modem, and click on **Modify**.

When configuring dial out connections, modifications might not be necessary. For dial in connections, use these guidelines:

|            |                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Speed(bps) | Try the highest speed port speed supported by your hardware. Match this speed setting with the modem configuration (Step5). For example, if you set the modem speed to 38400, use that setting here. The setting depends on your hardware: |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|           |                                |
|-----------|--------------------------------|
| 8250 UART | Select no higher than 9600 bps |
|-----------|--------------------------------|

|            |                                 |
|------------|---------------------------------|
| 16450 UART | Select no higher than 19200 bps |
|------------|---------------------------------|

|            |                      |
|------------|----------------------|
| 16550 UART | Set up to 115200 bps |
|------------|----------------------|

|                 |                                                                    |
|-----------------|--------------------------------------------------------------------|
| Line-mode label | Leave as is unless debugging problems with an incoming connection. |
|-----------------|--------------------------------------------------------------------|

|              |                                                        |
|--------------|--------------------------------------------------------|
| Login/answer | Select “On” for dial in access. This enables the port. |
|--------------|--------------------------------------------------------|

|                |                                                                                       |
|----------------|---------------------------------------------------------------------------------------|
| Receive buffer | Defer modifying this control until you need to debug problems or improve performance. |
|----------------|---------------------------------------------------------------------------------------|

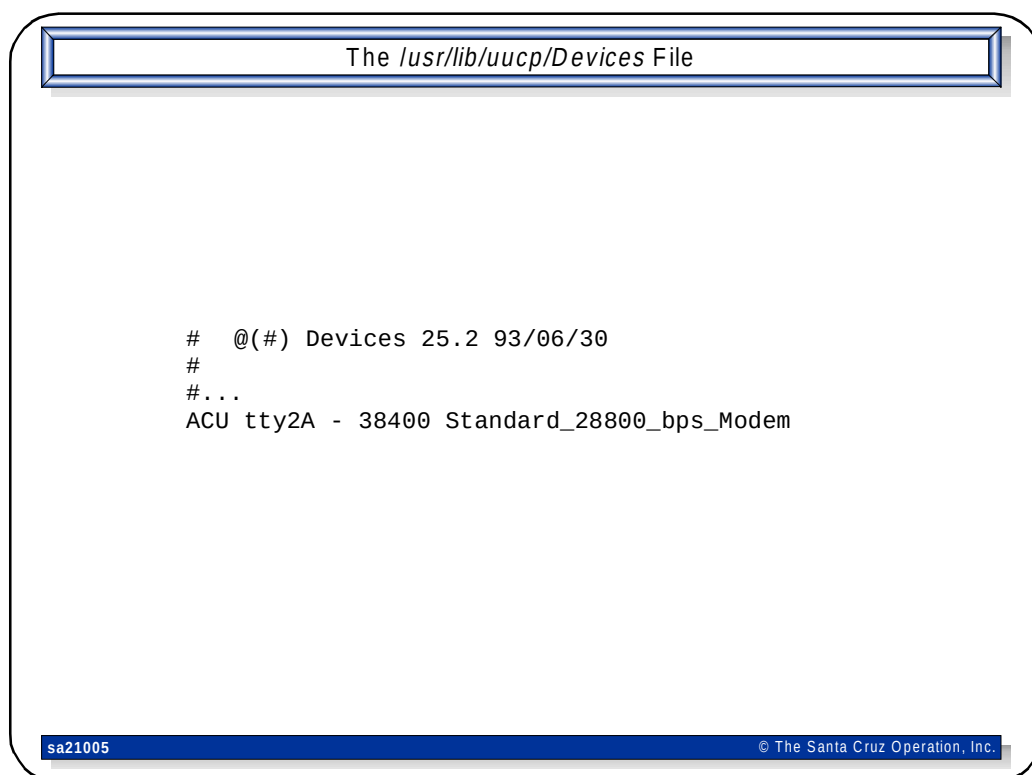
Step 9. Click on **OK** and then **Close** to exit the **Serial Manager**.

The kernel environment is rebuilt and these files are updated:

|                             |                              |
|-----------------------------|------------------------------|
| <i>/etc/conf/init.d/sio</i> | See the examples in AV 10-4. |
|-----------------------------|------------------------------|

|                     |                                          |
|---------------------|------------------------------------------|
| <i>/etc/inittab</i> | Updated from <i>/etc/conf/init.d/sio</i> |
|---------------------|------------------------------------------|

AV 10-5



Step 10. Select **Host** ⇒ **Exit** to exit the **Modem Manager**.

When complete, the **Modem Manager** creates entries in `/usr/lib/uucp/Devices` similar to those shown in AV 10-5. The format of the *Devices(F)* file is:

|                 |                                                                                                                                                                                                                                                     |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection type | Specifies one of two keywords: <ul style="list-style-type: none"><li>● Direct—connects to the modem; useful for testing, debugging, and setting modem configuration</li><li>● ACU—specifies that this is a calling unit</li></ul>                   |
| Line            | Device name of the modem control port                                                                                                                                                                                                               |
| Dialer line     | Used only with 801 dialers; enter a hyphen as a placeholder                                                                                                                                                                                         |
| Speed           | The speed or speed range for the device—matched against the speed field of the entries in <code>/usr/lib/uucp/Systems</code> with ACU as the device field; using a range gives you flexibility to accept calls from modems set at different speeds. |
| Dialer          | Specifies the name of the dialer program (for ACU entries) or direct. Dialer programs must be present in the <code>/usr/lib/uucp</code> directory.                                                                                                  |

### **Exercise: Add a modem**

---

---

Use the **Modem Manager** to configure the modem for dialout connections.

Display `/usr/lib/uucp/Devices` and `/etc/inittab` and view the entries made by the **Modem Manager**.

### 10.1.3 Testing the modem

---

To use **cu**(C) and test a dial out connection, the **uucp** user must own the serial port to which the modem is attached. For example, if the port is attached to */dev/tty1A*, enter:

```
chown uucp tty1A
```

```
chgrp uucp tty1A
```

To test **cu** with a phone number, enter:

```
cu 5551212
```

If you have problems, enable debugging with the **-x9** flag, enter:

```
cu -x9 5551212
```

To drop a connection after it has been established, enter:

```
~.
```

To test a dial-in configuration, use **cu** from a remote UNIX system, or use a Windows terminal emulation program, such as **HyperTerminal** to call the modem.

If **cu** fails, refer **SCOhelp** ⇒ **SCO OpenServer Handbook** ⇒ **Troubleshooting modems**.

#### **Exercise: Test the modem**

---

Test the modem with **cu** using the phone number provided by your instructor:

```
cu -x9 ###-####
```



## 10.2 Lesson Summary

---

---

Use the **Modem Manager** located in the **Hardware/Kernel Manager** to configure a modem for dialin or dialout or both. For testing and debugging your modem configuration, use the **cu(C)** command.

### Review questions

---

1. What command can you run to verify that your modem port is recognized at boot time?
2. When accessing the modem with the **cu(C)** command, which system user must own the serial port to which the modem is attached?

## 10.3 Relevant Commands and Files

---

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                       |                                                                                       |
|-----------------------|---------------------------------------------------------------------------------------|
| <b>chown</b> (C)      | Changes the owner ID of a file                                                        |
| <b>chgrp</b> (C)      | Changes the group ID of a file                                                        |
| <b>cu</b> (C)         | Calls up another UNIX system                                                          |
| <b>custom</b> (ADM)   | Installs, removes, or lists software products and components.                         |
| <b>hwconfig</b> (C)   | Displays hardware information                                                         |
| <b>scoadmin</b> (ADM) | Invokes SCOadmin applications (in this lesson, the <b>Serial and Modem Managers</b> ) |
| <b>uucp</b> (C)       | Copies a file or directory from one UNIX system to another UNIX system                |

### Relevant files

---

|                              |                                                                                                   |
|------------------------------|---------------------------------------------------------------------------------------------------|
| <i>/etc/conf/init.d/sio</i>  | A serial device control file used to rebuild <i>/etc/inittab</i> when the kernel is relinked      |
| <i>/etc/inittab</i>          | A serial device control file used by <b>init</b> (M)                                              |
| <i>/usr/lib/uucp/Devices</i> | Contains information for all the device that can be used to establish a link to a remote computer |

## **10.4 Answers to Review Questions**

---

---

This section provides suggested answers to the review questions.

1. You can run the **hwconfig**(ADM) command to verify that your modem port is recognized at boot time.
2. When accessing the modem with the **cu**(C) command, the **uucp** system user must own the serial port to which the modem is attached.



---

# ***11***

## ***Installing and Managing Printers***

AV 11-1

Installing and Managing Printers

- Install a parallel adapter card
- Install parallel and serial printers
- Add local and remote printers to your system
- Troubleshoot common printer installation problems
- Describe the UNIX print service
- Manage the print request directories
- Set print request priorities
- Manage printer interface scripts
- Troubleshoot the UNIX print service

sa21101© The Santa Cruz Operation, Inc.

**Lesson 11**

## **Installing and Managing Printers**

### **Introduction**

---

After terminals are installed and configured on your server, users can log in and access its data. Users will also need to print their data. You can 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. In addition to printing user and application data files, a printer can also be used to print important system files.

Managing the printers on your SCO OpenServer 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 adapter card
- Install parallel and serial printers
- Add local and remote printers to your system
- Troubleshoot common printer installation problems
- Describe the UNIX print service
- Manage the print request directories
- Set print request priorities
- Manage printer interface scripts
- Troubleshoot the UNIX print service

## 11.1 Installing a Parallel Adapter Card

---

Most servers come with one parallel port installed, but SCO OpenServer supports up to two. The supported ports include:

- Serial/parallel adapter #1:
  - Device special filename: `/dev/lp0`
  - 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 monochrome adapter:
  - Device special file name: `/dev/lp1`
  - Address: 0x3bc-3be
  - IRQ 7

### 11.1.1 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 any other installed card. When possible, select the default device interrupts and settings.

Refer to the **SCO OpenServer Handbook**, “Adding serial and parallel ports,” for a complete discussion of hardware installation.



## 11.1.2 Configuring the software

---

These steps describe how to configure a parallel port using the **Hardware/Kernel Manager** (note that you can also use **mkdev parallel** from the command line):

- Step 1. Start the **Hardware/Kernel Manager**.
- Step 2. Select **Parallel Port**.
- Step 3. Enter **1** to add a parallel port. Press **ENTER**.
- Step 4. Enter the number that represents the address for the parallel port you are installing. Press **ENTER**.
- Step 5. Confirm or change the default interrupt vector (unless you are installing on an MCA server). Press **ENTER**.

The installation program displays the newly installed device file, for example:

The device node is `/dev/lp0`

Each port you select is then added to the kernel using the default settings.

- Step 6. Relink the kernel, rebuild the environment, and reboot your system.

### **Exercise: Configuring a parallel port**

---

Follow the steps outlined in “Configuring the software” to configure the server parallel port.

Be sure to mark the configuration specifics in your system administration log book.

### **11.1.3 Troubleshooting parallel port installation**

---

The port should be recognized during system boot. A message should appear similar to this one:

```
%parallel    0x0378-0x037A  7  -  unit=0
```

If the parallel port is not recognized during boot, recheck the card options using the card documentation.

If the hardware is recognized, attach a parallel printer and redirect the **cal(C)** command to the printer:

```
cal > /dev/lp0
```

If the printer prints the calendar or indicates that data is in the printer buffer, the port is active.

## 11.2 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 standard parallel printer, refer to the printer documentation.

If you are installing a serial printer, note that serial printers must be capable of supporting XON/XOFF or hardware flow control. They must also support and be configured to use the correct handshaking protocol, either RTS/CTS or DTR protocols.

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 standard parallel cable into the printer port and the server 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 server to the serial port on your printer.
- Step 3.        Turn the printer on.
- Step 4.        Power up the system.

Refer to the **SCO OpenServer Handbook**, “Installing printers,” for more information concerning printer hardware installation.

### **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.

### **11.2.1 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/ttynn
```

where *nn* is the number of the serial port you are using, for example, */dev/tty2a*.

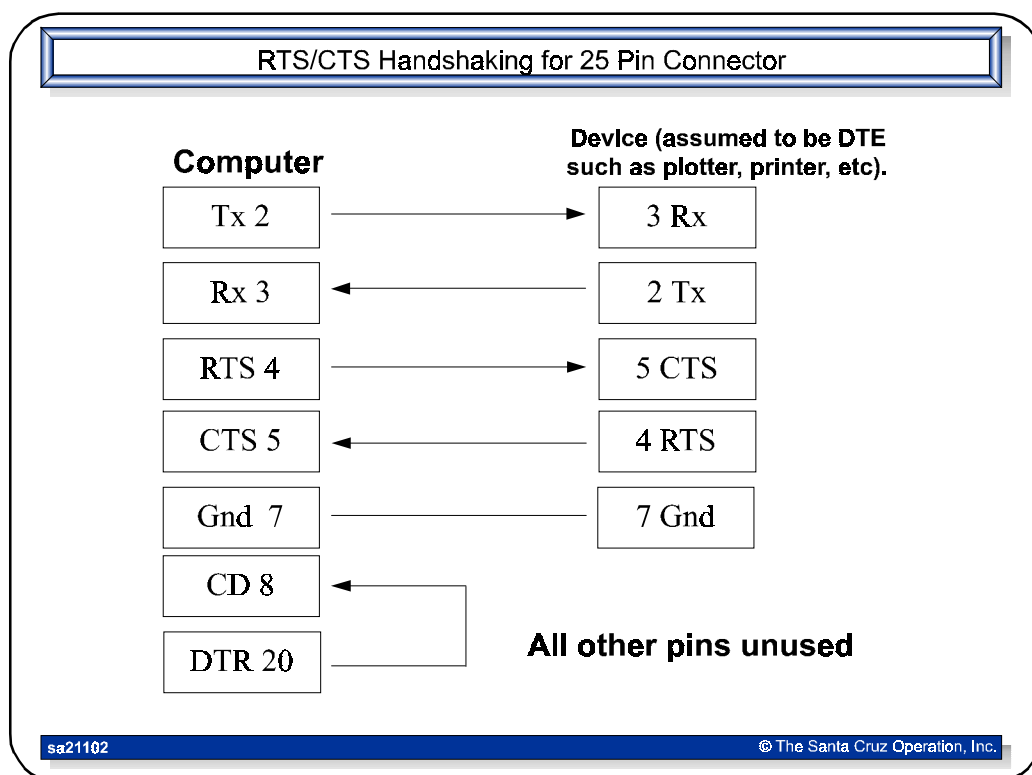
If the calendar prints on the printer, the physical connection is good.

**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 11-2



## AV 11-3

**RTS/CTS and DTR Printer stty Setting Changes**

| RTS/CTS handshaking printers | DTR handshaking printers |
|------------------------------|--------------------------|
| <b>-ixon</b>                 | <b>-ixon</b>             |
| <b>-ixoff</b>                | <b>-ixoff</b>            |
| <b>-clocal</b>               | <b>-clocal</b>           |
| <b>ortsfl</b>                | <b>-ortsfl</b>           |
| <b>rtsflow</b>               | <b>-rtsflow</b>          |
| <b>ctsflow</b>               | <b>ctsflow</b>           |

sa21103 © The Santa Cruz Operation, Inc.

## 11.2.2 Setting up RTS/CTS protocol serial printers

---

If you are setting up an RTS/CTS protocol serial printer, use the modem-control port (such as `/dev/tty1C`) instead of the standard serial device (`/dev/tty1c`).

Devices that use RTS/CTS signals for handshaking require different cables than devices that use the DTR signal for handshaking. Refer to the manufacturer's instructions for specifics.

To enable unidirectional hardware flow control using RTS/CTS handshaking or DTR handshaking, do the following:

- Change the **stty**(C) settings on the port as shown in the slide on the opposite page. A dash indicates that the protocol is turned off.
- Check the printer interface script to verify that there are no settings that conflict with those you just changed; interface scripts are discussed in detail later in this lesson in "Managing Printer Interface Scripts."

To enable the **stty** settings when the system goes to multiuser mode, do the following:

- Place the appropriate command in the `/etc/rc2` or `/etc/rc2.d/P88USRDEFINE` file:

```
stty stty_settings < /dev/ttynn
```

- Remove the `stty` line from the interface script

Refer to the *SCO OpenServer Handbook*, "Installing printers," for more information about setting up RTS/CTS and DTR.

### 11.2.2.1 Verifying the hardware installation of an RTS/CTS protocol serial printer

---

To verify the hardware installation of an RTS/CTS protocol serial printer, enter:

```
cal > /dev/ttynn
```

where `nn` is the number of the serial port you are using, for example, `/dev/tty1c`.

If the calendar prints on the printer, the physical connection is good.

You can specify the printer **stty(C)** settings (speed, parity, data length, and so on) if these differ from the serial line default settings. For example, to test a 4800-baud printer on `/dev/tty1c` that is set to 9600 baud, enter:

```
(stty 4800;date > /dev/tty1c) < /dev/tty1c
```

Refer to the **stty(C)** manual page for more information.

### **Exercise: Verifying the serial printer hardware connection**

---

If you installed an RTS/CTS protocol serial printer, follow the steps outlined in “Verifying the hardware installation of an RTS/CTS protocol serial printer” and verify that the printer hardware connection and cable are correct.

## **11.2.3 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 at bootup. Check the `/usr/adm/messages` file:
  - For parallel cards, you will see a message similar to this:

```
parallel 0x378-0x37A      07      -      unit=0
```

where “unit=0” refers to `/dev/lp0`, and “unit=1” refers to `/dev/lp1`, and so on



- For serial cards, you will see a message similar to this:

```
%serial 0x03F8-0x03FF    04    -    unit=0 type=Standard
nports=1
```

where “unit=0” indicates that the serial port is considered to be COM1, and the “nports=1” indicates that the driver recognized one port on the board.

Smart serial cards using third-party drivers can display different boot messages specific to their drivers. The message displayed at boot when the smart card is recognized may resemble:

```
%onboard 0x0230x0x023F    34    0    unit=0 mem=0x000D0000
nport=16
```

- Confirm that your card is recognized using **hwconfig(HW)**; enter:

**hwconfig -h**

If the card is recognized, an entry is printed to the screen that has similar information to the message on the previous page. 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 server.

- Check and verify that you specified the non-modem control device, such as */dev/tty1c*, for serial printers that use XON/XOFF flow control

## 11.3 Adding Printers

---

---

Use the **Printer Manager** to add a local or remote printer to your system's print services.

### 11.3.1 Adding a local printer

---

To add a local printer (parallel or serial), do the following:

- Step 1. Start the **Printer Manager** as **root**.
- Step 2. Select **Printers** ⇒ **Printer Manager** ⇒ **Printers** ⇒ **Add Local**.
- Step 3. Enter the printer name, model, and device name.
- Step 4. Click on **OK**.

The printer appears in the **Printer Configuration** list. In addition, the printer is automatically enabled. If you want to enable remote print services, you must also do the following:

Select **System** ⇒ **Print Services** ⇒ **Remote UNIX Print Service Enabled**.

When you enable Remote UNIX Print Service for a printer, it is actually enabled for *all* configured local printers, and is automatically enabled whenever you add another local printer. Likewise, if you turn off Remote UNIX Print Service for a printer, it is automatically disabled for that printer and any other configured local printers, and is disabled for any local printers that you add.

In addition, to configure the printer to accept remote jobs, do the following:

Select **Settings** ⇒ **Control** ⇒ **Accept New Remote Jobs**.

- Step 5. Select **Host** ⇒ **Exit** to exit the **Printer Manager**.

### 11.3.2 Adding a remote printer

---

To add a remote printer (parallel or serial), do the following:

- Step 1. Start the **Printer Manager** as **root**.
- Step 2. Select **Printers** ⇒ **Printer Manager** ⇒ **Printer** ⇒ **Add Remote** ⇒ **UNIX**.
- Step 3. Enter the name of the host system, or click on **Select** and choose from a list of available hosts.
- Step 4. Enter the name of the remote printer (or printers), or click on **Select** to choose from a list of printers.
- Step 5. Click on **OK**.  
  
The new remote printer appears in the **Printer Configuration** list on the local host.
- Step 6. Select **Host** ⇒ **Exit** to exit the **Printer 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.

## **11.4 Describing the UNIX Print Service**

---

The background daemon responsible for the print service is **lp sched**(ADM), also known as the print scheduler. The print scheduler starts each time the system goes into multiuser mode. There is an **lp sched** process for each print request currently printing.

### **11.4.1 Who can administer the print service**

---

The following users can administer the print service through **scoadmin**(ADM) or from the command line:

- The **root** account
- The **lp** administrator (a user with **lp** authorization)

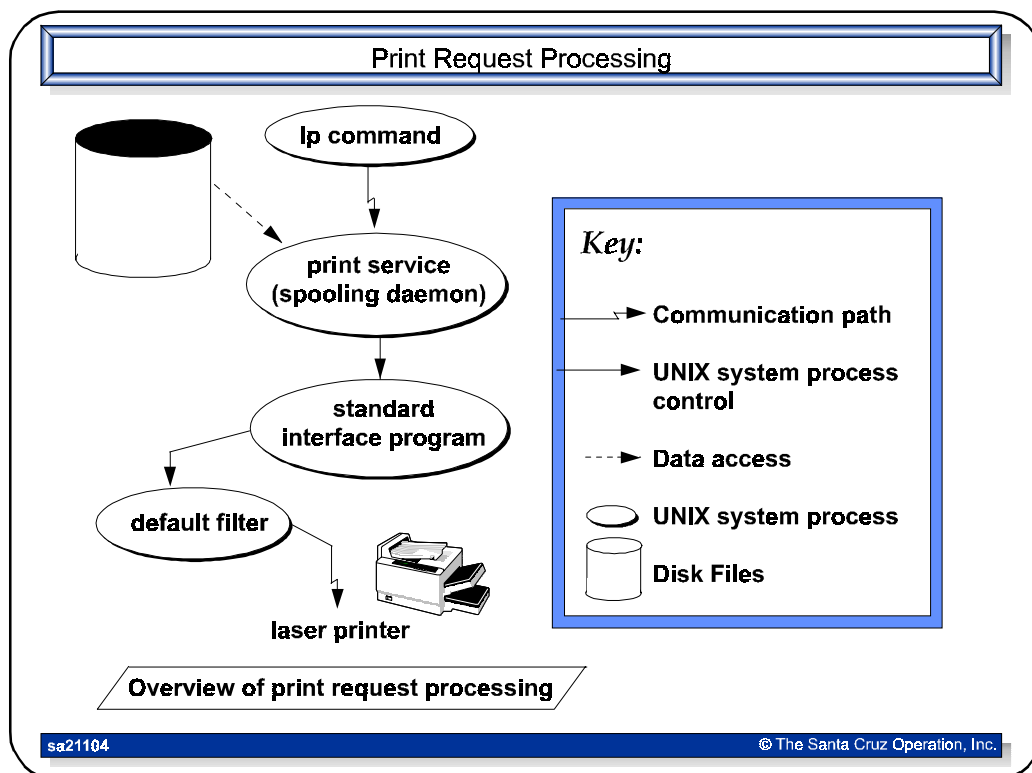
## **11.4.2 Print scheduler functions**

---

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 11-4



### 11.4.3 Print request process

---

The following describes the flow of the 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 11-5

Viewing the Status of the Print Scheduler with lpstat -t

```
scheduler is running
system default destination: laser
device for laser: /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
```

sa21105

© The Santa Cruz Operation, Inc.



### 11.4.4 Viewing the print scheduler

---

Use the **lpstat(C)** command with the **-t** option to view the status of the print scheduler. The following is a description of the information displayed:

|                               |                                                                            |
|-------------------------------|----------------------------------------------------------------------------|
| scheduler is running          | The print service is operational.                                          |
| system default destination    | The name of the 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?

## 11.5 Managing the Print Request Directories

---

The `/usr/spool/lp/temp` and `/usr/spool/lp/requests` directories describe each request that has been submitted to the print service. You can display the information that is stored in the `/usr/spool/lp/requests` directory with the **lpstat(C)** command.

For example, a print request assigned the job number 17 would have the following entries:

|                                          |                                                                                                                                                                                                                         |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/usr/spool/lp/temp/17-0</code>     | Contains the absolute pathname of the file to be printed and scheduling information                                                                                                                                     |
| <code>/usr/spool/lp/temp/17</code>       | Contains error messages encountered while trying to print                                                                                                                                                               |
| <code>/usr/spool/lp/requests/17-0</code> | Contains scheduling information                                                                                                                                                                                         |
| <code>/usr/spool/lp/requests/17-1</code> | When submitting a print request using the <b>-c</b> option to the <b>lp(C)</b> command, or when printing a file using a pipe command, a copy of the file is stored in <code>/usr/spool/lp/temp</code>                   |
| <code>/usr/spool/lp/logs/requests</code> | Once the print job has completed printing, files are removed from <code>/usr/spool/lp/temp</code> and <code>/usr/spool/lp/requests</code> , and the information is appended to <code>/usr/spool/lp/logs/requests</code> |

AV 11-6

**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. <ul style="list-style-type: none"><li>• <b>M</b> by an electronic mail message</li><li>• <b>W</b> by a message written to the user's terminal</li></ul>                                                                                                                                                             |
| O      | Any <b>-o</b> options given to <b>lp(C)</b>                                                                                                                                                                                                                                                                                                                                          |
| 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: <ul style="list-style-type: none"><li>0x0004 Slow filtering finished successfully.</li><li>0x0010 Printing finished successfully.</li><li>0x0040 Request was cancelled.</li><li>0x0100 Request failed filtering or printing.</li></ul> |
| U      | Name of the user who submitted the print request.                                                                                                                                                                                                                                                                                                                                    |

sa21106

© The Santa Cruz Operation, Inc.

### 11.5.1 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**(C) 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 cancelled
- 0x0100 - The request failed printing

Refer to the **System Administrators Guide**, “Managing Printers and Print Jobs,” “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 will remain in the queue because the printer is disabled.
3. Change directory to `/usr/spool/lp/temp`.
4. Examine the contents of your print request ID file.
5. Change the directory to `/usr/spool/lp/requests`.
6. Examine the contents of your print request ID file.
7. Cancel the print request.
8. Examine the contents of `/usr/spool/lp/logs/requests`.

## **11.5.2 Clearing the print request directories**

---

You should examine, archive (if applicable), and remove files in the `/usr/spool/lp/logs` directory. In addition to the `requests` file, you should also manage the `/usr/spool/lp/lpsched` log file, which contains the dates and times the print service was started or stopped.

### **Exercise: Clearing logfiles**

---

Change to the directory `/usr/spool/lp/logs`. Clear the `requests` and `lpsched` files.

**> requests ; > lpsched**

Which program could you use to automate this task?

## 11.6 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.

### 11.6.1 Default priority level and limits

---

You can use the **lpusers**(ADM) command or the **Printer Manager** to set print priorities and limits.

To determine how the current priorities and limits are set, enter:

```
/usr/lib/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.

### 11.6.2 Changing the default priority limit for all users

---

You can assign a default priority limit for the balance of users not assigned a personal limit as follow:

```
/usr/lib/lpusers -d new_priority
```

For example:

```
/usr/lib/lpusers -d 30
```

Users not assigned a personal priority limit can submit print jobs requesting priorities of between 30-39.

### **Exercise: Changing the default priority limit**

---

Change the default priority limit so that all users can request a print priority between 25 and 39 when submitting print requests.

### **11.6.3 Setting a personal priority limit for a user**

---

You can set personal priority limits for a user or group of users as follows:

```
/usr/lib/lpusers -q priority_limit -u username
```

For example:

```
/usr/lib/lpusers -q 10 -u kellyk
```

The user **kellyk** can submit a print request with a priority between 10-39.

### **11.6.4 Restoring the default priority limit**

---

You can restore the default priority limit for a user as follows:

```
/usr/lib/lpusers -u username
```

For example:

```
/usr/lib/lpusers -u kellyk
```

### **11.6.5 Submitting a print job with a different priority**

---

You can submit a print request using a different priority than the default as follows:

```
lp -q new_priority filename
```

For example:

```
lp -q 10 /etc/default/tar
```



### 11.6.6 Changing the priority of print job residing in the queue

---

You can reassign a new priority for print requests that are still waiting to print as follows:

```
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.

### 11.6.7 Bypassing all other awaiting print requests

---

The **root** user and the **lp** administrator are the only users who have the authority to bypass all awaiting print requests.

To bypass all awaiting print requests residing in the queue, enter:

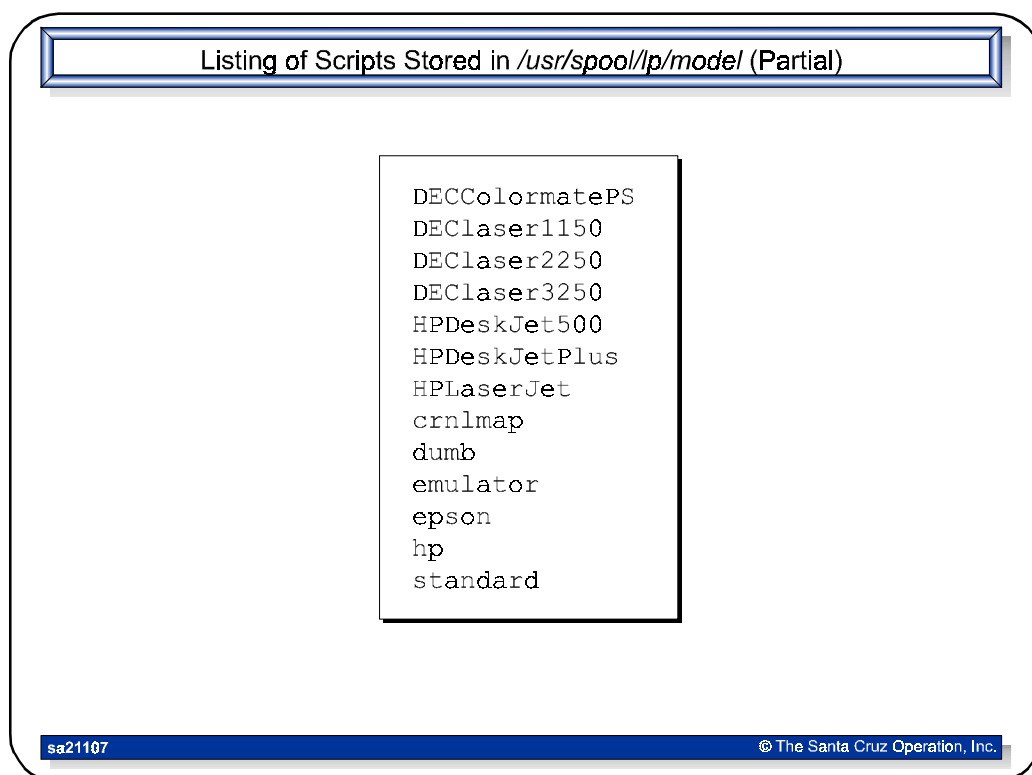
```
lp -H immediate filename
```

For example:

```
lp -H immediate /etc/default/tar
```

If the printer is currently printing, the request is the next job printed.

AV 11-7



## 11.7 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(C)** command
- Reports any errors encountered by the print service

The system contains many 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.

### 11.7.1 Assigning an interface script

---

Model interface scripts are stored in */usr/spool/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:

*/usr/spool/lp/admins/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.

## **11.7.2 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 `/usr/spool/lp/admins/lp/interfaces`. Do not edit the scripts in `/usr/spool/lp/model`.

### **11.7.2.1 Restricting banner pages**

---

You can restrict banner pages from printing as follows:

- At the shell prompt, edit the model interface script *standard*. Change the line that reads:  
`nobanner="no"`  
to read:  
`nobanner="yes"`
- From the **Printer Manager**, select **Advanced** ⇒ **Settings** ⇒ **Banners**. Change the **Number of banner pages** to **0**.

### 11.7.2.2 Skipping page breaks

---

When printing multiple data files, to skip page breaks between files, if you have used the model interface script *standard*, change the line that reads:

```
nofilebreak="no"
```

to read:

```
nofilebreak="yes"
```

### 11.7.2.3 Changing a baud rate

---

It may be necessary to change the 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. Start the **Printer Manager** as **root** or the as the **lp** administrator.
- Step 2. Select the printer for which you want to change the baud rate.
- Step 3. Select **Settings** ⇒ **Serial Comm.**  
  
A window displaying the current serial communications settings for that printer appear.
- Step 4. Click on the small rectangular box within the Baud Rate box. A list of possible values appears.
- Step 5. Select the desired 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" to change the baud rate for your printer.

## 11.8 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.

### 11.8.1 Printer does not print

---

If your printer is not producing printouts, do the following:

- Step 1. Verify that the printer has power, is online, and has no loose cabling.
- Step 2. Run **lpstat -t** to determine if the scheduler is running, and the printer is enabled and is accepting requests.
- Step 3. Run **hwconfig -h** to determine if the hardware is recognized by the operating system.
- Step 4. Redirect the output from the **cal(C)** command to the appropriate **tty** or **lp** port:

```
cal > /dev/tty $nn$ 
```

or

```
cal > /dev/lp $n$ 
```

- 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 or the printer configuration file.

### 11.8.2 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
```

### 11.8.3 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") > lpn
```

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.

### 11.8.4 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 chapter1.ps > /dev/lpn
```

If the printer does not print, check the cable connections both at the printer and the server by removing the cable end and reconnecting it.

### 11.8.5 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, determine if the `/usr/lib/lpsched` daemon is running.

Use the `lpstat(C)` command to determine if the scheduler is running:

```
lpstat -r
```

If it is not running, the system displays: `scheduler is not running`.

To 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`**

### 11.8.6 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 that the device file exists in the `/dev` directory.

**`ls /dev/lp0`**

`crw----- 2 bin bin 6,0 June 13 1993 /dev/lp0`

If the device file exists and has incorrect settings, remove it:

**`rm /dev/lp0`**

Step 2. Recreate the `/dev/lp0` file.

Step 3. Change the file permissions, link count, ownership, and group to that of the original device file:

**`mknod /dev/lp0 c 6 0`**

**`chown bin /dev/lp0`**

**`chgrp bin /dev/lp0`**

**`chmod 600 /dev/lp0`**

**`ln /dev/lp0 /dev/lp`**

Step 4. Run **`hwconfig -h`** to verify that the port is recognized.



Step 5. Redirect the **date**(C) command to the printer:

```
date> /dev/lp0
```

### **11.8.7 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

## 11.9 Lesson Summary

---

To install a parallel card, use the **Hardware/Kernel Manager** or `mkdev parallel`.

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. Additionally, to install, you must know exactly what kind of handshaking protocol it is configured to support (RTS/CTS or DTR).

In every case, the device to which you attach a printer must be recognized at system boot time. If it is not recognized, you will have to install the device using the appropriate **scoadmin**(ADM) utilities.

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 Manager**. 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.

### 11.9.1 Review questions

---

1. What is the procedure for installing and configuring a parallel adapter card?
2. Outline the steps for installing a parallel printer.
3. How would you enable remote print services on a local printer using the **Printer Manager**?

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. What two users have the authority to bypass all waiting print requests?
8. For which tasks are a printer interface script responsible?
9. What command could you use to determine if the print scheduler is running and enabled to accept requests?

## **11.10 Lab: Printer Troubleshooting**

---

---

Your instructor will create a failure in your classroom server and printer setup.

### **11.10.1 Overview**

---

The instructor has created a common failure in your server 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 SCO OpenServer 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 server and show what you did.

If you find and resolve a problem quite quickly, ask your instructor to create another failure for you to isolate and repair.

## 11.11 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                      |                                                                                                                                                                                 |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>disable(C)</b>    | Stops <b>init(M)</b> from allowing logins to the specified terminal(s); activates the specified printer(s) and allows them to accept <b>lp(C)</b> print requests                |
| <b>enable(C)</b>     | Signals <b>init(M)</b> to allows logins to the specified terminal(s); activates the specified printer(s) and allows them to accept <b>lp(C)</b> print requests                  |
| <b>hwconfig(HW)</b>  | Displays the installed drivers                                                                                                                                                  |
| <b>lp(C)</b>         | Sends print requests to a printer                                                                                                                                               |
| <b>lpsched(ADM)</b>  | Starts the UNIX print service                                                                                                                                                   |
| <b>lpshut(ADM)</b>   | Stops the UNIX print service                                                                                                                                                    |
| <b>lpstat(C)</b>     | Prints <b>lp</b> print service status information                                                                                                                               |
| <b>lpusers(ADM)</b>  | Sets printing queue priorities                                                                                                                                                  |
| <b>mkdev(ADM)</b>    | Calls scripts to add peripheral devices; when used with the <b>parallel</b> argument, creates a parallel port; when used with the <b>serial</b> argument, creates a serial port |
| <b>stty(C)</b>       | Sets the options for a serial terminal or printer                                                                                                                               |
| <b>scoadmin(ADM)</b> | Invokes SCOadmin applications (in this lesson, the <b>Printer Manager</b> and the <b>Hardware/Kernel Manager</b> )                                                              |

### Relevant files

---

|                   |                                                                    |
|-------------------|--------------------------------------------------------------------|
| <b>/dev</b>       | The device special file directory                                  |
| <b>/dev/tty1a</b> | Sample device special file for the first serial port               |
| <b>/dev/tty1A</b> | Sample device special file for the first modem control serial port |

---

|                                                 |                                                                           |
|-------------------------------------------------|---------------------------------------------------------------------------|
| <code>/dev/tty2a</code>                         | Sample device special file for the second serial port                     |
| <code>/dev/tty2A</code>                         | Sample device special file for the second modem control serial port       |
| <code>/dev/lp0</code>                           | Device special file for the first parallel port on a server               |
| <code>/dev/lp2</code>                           | Serial/parallel adapter #2                                                |
| <code>/dev/lp1</code>                           | Parallel monochrome adapter                                               |
| <code>/etc/inittab</code>                       | A serial device control file used by <b>init(M)</b>                       |
| <code>/etc/ttytype</code>                       | A serial device control file used by the shell                            |
| <code>/etc/conf/init.d/sio</code>               | A serial device control file used when the kernel is relinked             |
| <code>/etc/auth/system/ttys</code>              | Terminal Control Database                                                 |
| <code>/usr/spool/lp/admins/lp/interfaces</code> | Contains interface scripts for existing printers                          |
| <code>/usr/spool/lp/logs</code>                 | Completed print request log files                                         |
| <code>/usr/spool/lp/model</code>                | Contains model printer interface scripts                                  |
| <code>/usr/spool/lp/requests</code>             | Contains print log information while printing                             |
| <code>/usr/spool/lp/temp</code>                 | Contains print log information and copies of printed files while printing |

## 11.12 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Model interface scripts

---

To list the model interface scripts for your classroom server, enter:

```
cd /usr/spool/lp/model
```

```
l
```

### Exercise: Installed interface scripts

---

To list the installed printer interface scripts for your classroom server, enter:

```
cd /usr/spool/lp/admins/lp/interfaces
```

```
l
```

### Exercise: Clearing logfiles

---

You could use **cron**(C) to automate this procedure.

### Exercise: Changing the default priority limit

---

To change the default priority limit so that all users can request a print priority between 25 and 39 when submitting print requests, enter:

```
/usr/lib/lpusers -d 25
```

## Review questions

---

This section provides suggested answers to the review questions.

1. To install and configure a parallel adapter card, install the hardware using default settings if possible, enter the **Hardware/Kernel Manager** or **mkdev parallel**, select the type of card based on its address and relink the kernel. Check to see if the new port is recognized at boot time.
2. To install a parallel printer, connect the hardware (use the correct parallel cable), configure the parallel port, test the connection, and run the **Printer Manager** to add a new printer.
3. To enable remote print services on a local printer using the **Printer Manager**, you would first highlight the printer, then select **System⇒Print Services⇒Remote UNIX Print Service Enabled**.
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 `/usr/spool/lp/temp` and `/usr/spool/lp/requests` directories contain print request information.
7. The **root** user and the **lp** administrator are the only users who have 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(C)** 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.

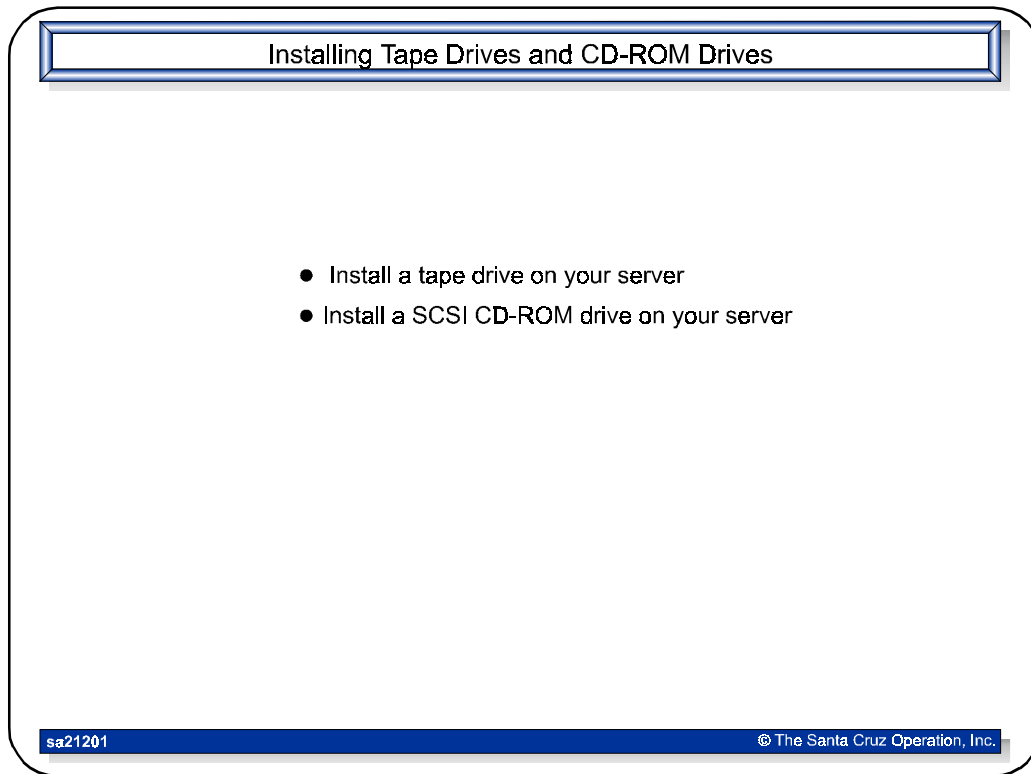


---

# ***12***

## ***Installing Tape Drives and CD-ROM Drives***

AV 12-1



**Lesson 12**

## **Installing Tape Drives and CD-ROM Drives**

### **Introduction**

---

Although your CD-ROM is automatically detected or installed during the operating system installation, you might need to add a second, or change an existing CD-ROM on your server. If you are backing up your server to tape, the device must be installed before doing so.

### **Objectives**

---

The purpose of this lesson is to provide practice installing common server media devices. By the end of this lesson, you will be able to:

- Install a tape drive
- Install a CD-ROM drive

## 12.1 Installing a Tape Drive

---

SCO OpenServer Release 5 supports the installation of tape devices that are attached to an IDE or SCSI controller.

Support for tape drive specifications include:

- Cartridge
- DAT compressing and non-compressing
- Exabyte 8mm (8200 or 8500)
- Generic SCSI 1 and SCSI 2 (including nine track)

Recording capacities vary, but are typically in the upper range (GB and higher). The manual page for SCSI tape drives for **scsitape(HW)**.

### 12.1.1 Hardware settings

---

Hardware settings for non-SCSI tapes can include:

- DMA
- IRQ
- Base I/O address

In addition, if the tape is attached to an IDE controller, you also need to know:

- To which controller it is attached (primary or secondary)
- How the device is configured (primary or secondary)
- Hardware settings for SCSI tapes include: Host adapter type
- Host adapter number
- Target ID
- Bus number
- Logical unit number (LUN)

- Vendor identification
- SCSI version
- Response Data Format

Refer to the *SCO OpenServer Handbook*, “Adding SCSI host adapters and peripherals,” for more information.

## 12.1.2 Configuring the software

---

You can use the **Hardware/Kernel Manager** or **mkdev tape** to add, change, or remove tape drives. The following sections describe the software configuration for IDE and SCSI tape drives.

### 12.1.2.1 Configuring an Enhanced IDE tape drive

---

These steps describe how to configure an IDE tape drive using the **Hardware/Kernel Manager** (note that you can also use **mkdev tape** from the command line):

Step 1. Log in as **root** and verify that the system is in single-user mode.

```
who -r  
. run-level S June2 14:31 S 1 1
```

If you are not in single-user mode, enter **init 1**, then enter the **root** password when prompted to enter System Maintenance Mode.

Step 2. Start the **Hardware/Kernel Manager**.

Step 3. Select **Tape Drive**.

Step 4. Select **1**, Configure a SCSI or Enhanced IDE tape drive.

Step 5. Select **2**, Install an Enhanced IDE Tape Drive.

Step 6. Enter the IDE controller to which the device is attached:

- If the device is attached to the Primary controller, enter **0**.
- If the device is attached to the Secondary controller, enter **1**.

Press .

- Step 7. Enter how the IDE device is configured:
- If the device is configured as a Master, enter **0**.
  - If the device is configured as a Slave, enter **1**.
- Press  .
- Step 8. Enter **y** to update the IDE configuration.
- Step 9. Enter **y** to update the IDE configuration.
- Step 10. Relink the kernel, rebuild the environment, and reboot to use the tape drive.

### Exercise: Installing an Enhanced IDE tape drive

---

If your classroom system has an Enhanced IDE tape drive to install, follow the steps outlined in “Configuring an Enhanced IDE tape drive” and install it.

Be sure to update your system administration log book when you have finished.

## 12.1.2.2 Configuring a SCSI tape drive

---

These steps describe how to configure a SCSI tape drive using the **Hardware/Kernel Manager** (note that you can also use **mkdev tape** from the command line):

- Step 1. Log in as **root** and verify that the system is in single-user mode.
- ```
who -r
. run-level S June2 14:31 S 1 1
```
- If you are not in single-user mode, enter **init 1**, then enter the **root** password when prompted to enter System Maintenance Mode.
- Step 2. Start the **Hardware/Kernel Manager**.
- Step 3. Select **Tape Drive**.
- Step 4. Select **1**, Configure a SCSI or Enhanced IDE tape drive.

- Step 5.        Select **1**, Install a SCSI tape drive.
- Step 6.        Enter the prefix of the SCSI host adapter. Select **h** for a list of supported adapters.
- Step 7.        Enter the host adapter name.
- Step 8.        Specify to which SCSI bus the device is attached.
- Step 9.        Enter the Target ID.
- Step 10.       Enter the LUN of the device.
- Step 11.       Update the SCSI configuration when prompted.
- Step 12.       Enter the vendor identification string or use the default.
- Step 13.       Enter the SCSI version, or use the default.,
- Step 14.       Enter the Response Data Format (or use the default).
- Step 15.       Select the tape drive type.
- Step 16.       Enter the new boot string, or enter **q** for the default.
- Step 17.       Enter **q** twice to exit the menus.
- Step 18.       View the configuration to see if it is correct. Relink the kernel and reboot to use the tape drive. You will be reminded to create a new emergency boot floppy set.

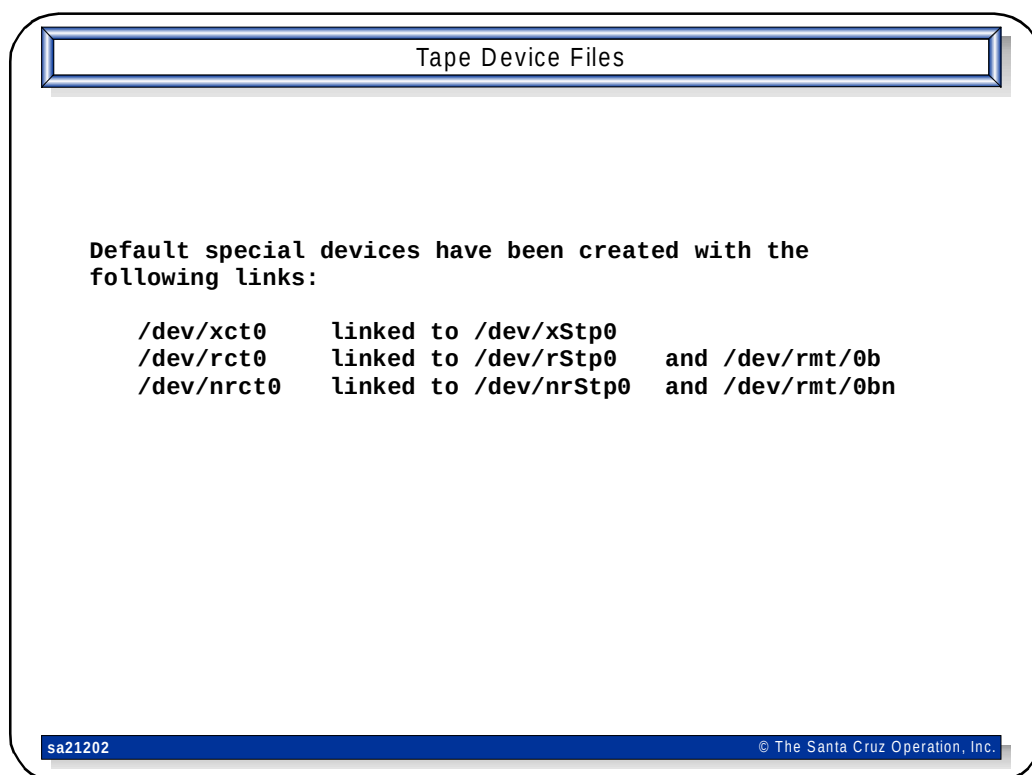
### **Exercise: Installing a SCSI or DAT tape drive**

---

If you have a SCSI or DAT tape drive, follow the steps outlined in “Configuring a SCSI tape drive” and install it now.

Be sure to update your system administration log book when you have finished.

AV 12-2





### 12.1.3 Tape device files

---

Several device files are created during the tape drive installation. Each device has a different purpose. For example:

- The `/dev/xct0` device, linked to `/dev/xStp0` is used with the **tape(C)** command. The **tape(C)** command can be used to rewind and retention your tape device.
- The `/dev/rct0` device, linked to `/dev/rStp0` is used to archive files to the beginning of the tape (rewind).
- The `/dev/nrct0` device, linked to `/dev/nrStp0` is used to archive files where the last archive left off (no rewind).

Note that the `/dev/rmt/0b` and `/dev/rmt/0bn` device files are also created and linked to `/dev/rct0` and `/dev/nrct0` respectively. These file names are sometimes used for application compatibility.

### 12.1.4 Troubleshooting

---

The topics of troubleshooting problems with tape drives in the *SCO OpenServer Handbook* include:

- General problems with tape drives
- Problems with SCSI tape drives

Refer to the *SCO OpenServer Handbook*, “Adding tape drives,” for details on troubleshooting your specific tape installation.

#### **Exercise: Confirming tape drive operation**

---

You can verify the tape drive operation in one of two ways:

- Use the **tape(C)** command to retention a tape, for example:  
  
**tape status /dev/rStp0**
- Use the **tar(C)** command to copy the `/etc/motd` file to the tape.

## 12.2 Installing a CD-ROM Drive

---

When installing a CD-ROM, use an SCO-supported host adapter or controller. Also check with your hardware supplier to verify that the drive works with the host adapter or controller. The device files used for access to the CD-ROM drive are documented on the **cdrom(HW)** manual page.

You can mount High Sierra/ISO 9660 or Rock Ridge format CD-ROMs as read-only filesystems.

CD-ROM filesystems containing extended attribute records are supported. You can access record format information and the file access permission in an extended attribute record using the **-r** option to the **mount(ADM)** command. Refer to the **mount(ADM)** manual page for more information.

### 12.2.1 Installing the hardware

---

AT bus tape drive controllers use jumpers or switches to set the interrupt vector, DMA channel, and so on. Configure EISA and MCA bus settings using the vendor-specific configuration utility provided with the server.

Enhanced IDE (EIDE) devices require that you specify:

- To which controller the CD-ROM is attached (Primary or Secondary)
- How the CD-ROM is configured (Master or Slave)

SCSI CD-ROM drives require that you specify:

- SCSI host adapter type
- Host adapter number
- Target ID
- Logical unit number (LUN)

If the SCSI adapter is not yet installed, you must install it before continuing with the SCSI CD-ROM installation.

Refer to the **SCO OpenServer Handbook**, “Adding SCSI host adapters and peripherals,” for more information.

---

## 12.2.2 Configuring an Enhanced IDE CD-ROM

---

These steps describe how to configure an Enhanced IDE CD-ROM drive using the **Hardware/Kernel Manager** (note that you can also use **mkdev cdrom**) :

Step 1. Log in as **root** and verify that the system is in single-user mode.

```
who -r
. run-level S June2 14:31 S 1 1
```

If you are not in single-user mode, enter **init 1**, then enter the **root** password when prompted to enter System Maintenance Mode.

Step 2. Start the **Hardware/Kernel Manager**.

Step 3. Select **CD-ROM and WORM**.

Step 4. Select **1**, Install an EIDE CD-ROM.

Step 5. Enter the IDE controller to which the device is attached:

- If the device is attached to the Primary controller, enter **0**.
- If the device is attached to the Secondary controller, enter **1**.

Press .

Step 6. Enter how the IDE device is configured:

- If the device is configured as a Master, enter **0**.
- If the device is configured as a Slave, enter **1**.

Press .

Step 7. Enter **y** to update the IDE configuration.

Step 8. Rebuild and relink the kernel.

### 12.2.3 Configuring a SCSI CD-ROM

---

Follow these steps to configure a SCSI CD-ROM using the **Hardware/Kernel Manager** (note that you can also use **mkdev cdrom**):

Step 1. Log in as **root** and verify that the system is in single-user mode.

```
who -r  
. run-level S June2 14:31 S 1 1
```

If you are not in single-user mode, enter **init 1**, then enter the **root** password when prompted to enter System Maintenance Mode.

Step 2. Start the **Hardware/Kernel Manager**.

Step 3. Select **CD-ROM and WORM**.

Step 4. Enter the type and number of the host adapter.

If the host adapter is not already configured into the kernel, enter the requested hardware configuration information. Confirm the values when prompted.

Step 5. Enter the SCSI bus number to which this device is attached.

Step 6. Enter the ID and LUN number of the CD-ROM.

Step 7. Enter **y** to update the SCSI configuration.

Step 8. Relink the kernel and reboot.

#### **Exercise: Installing a SCSI CD-ROM drive**

---

If you have a SCSI CD-ROM drive to install, follow the preceding steps and install it.

Be sure to update your system administration log book when you have finished.

## 12.2.4 Using the CD-ROM

---

Mount the CD-ROM before using it. Enter this command line to mount the first CD-ROM:

```
mount -r /dev/cd0 /mnt
```

If you attempt to mount the drive without inserting a CD-ROM, you see this message:

```
cannot open
```

## 12.2.5 Troubleshooting

---

Refer to the *SCO OpenServer Handbook*, “Adding SCSI host adapters and peripherals,” for details on troubleshooting SCSI CD-ROM drive installations.

## **12.3 Lesson Summary**

---

When installing hardware, such as tape drives and CD-ROM drives, on your server, you must:

- Verify that the controller card for the drive does not conflict with other installed cards
- Configure the software to run with the new drive
- Relink the kernel to access the new drive device drivers
- Troubleshoot the installation, if necessary

## **Review questions**

---

1. List the steps for installing an IDE tape drive on your server.
2. List the required steps for installing a SCSI CD-ROM on your server.

## 12.4 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### 12.4.1 Relevant commands

---

|                       |                                                                                                                                                                                                                                                     |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>mkdev</b> (ADM)    | Calls scripts to add peripheral devices; when used with the <b>cdrom</b> argument, configures a device for installing SCSI CD-ROM drives; when used with the <b>tape</b> argument, configure a device for installing SCSI and non-SCSI tape drives. |
| <b>scoadmin</b> (ADM) | Invokes SCOadmin applications (in this lesson, the <b>Hardware/Kernel Manager</b> )                                                                                                                                                                 |
| <b>tar</b> (C)        | Creates tape archives                                                                                                                                                                                                                               |

### 12.4.2 Relevant files

---

|                   |                                        |
|-------------------|----------------------------------------|
| <i>/dev/rStp0</i> | A SCSI tape drive special file         |
| <i>/dev/rct0</i>  | A 1/4-inch cartridge tape special file |

## 12.5 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Examining the manual page for cartridge(HW)

---

The capacity of a QIC-1350 tape is 1.35GB.

### Exercise: Examining the manual page for scsi(HW)

---

The adapter prefix for the Adaptec 2742A SCSI host adapter is *arad*.

### Review questions

---

This section provides suggested answers to the review questions.

1. To install an IDE tape drive, configure the hardware options and install the hardware, then configure the software by entering the **Hardware/Kernel Manager** or **mkdev tape**, select “Configure a SCSI or Enhanced IDE tape drive,” “Install an Enhanced IDE Tape Drive,” Enter the controller number to which the tape is attached, and if it is attached as the primary or secondary device. Quit the utility. Relink the kernel and reboot.
2. To install a SCSI CD-ROM, configure the hardware options and install the hardware, then configure the software by entering the **Hardware/Kernel Manager** or **mkdev cdrom**, select “Install SCSI CD-ROM Drive.” Enter the SCSI host adapter type, the adapter and BUS numbers, and the ID and LUN of the CD-ROM drive. Confirm the information and quit the utility. Relink the kernel and reboot. You must mount the CD-ROM filesystem before you can access it.



---

# ***13***

## ***Controlling User and System Processes***

AV 13-1

Controlling User and System Processes

- Describe how processes are created
- Distinguish between user- and system-generated processes
- Use system tools to gather information about processes
- Determine the priority of a process
- Use process accounting to monitor processes
- Use **crontab**(C) to automate system processes

sa21301© The Santa Cruz Operation, Inc.

**Lesson 13**

## **Controlling User and System Processes**

### **Introduction**

---

The SCO OpenServer 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.

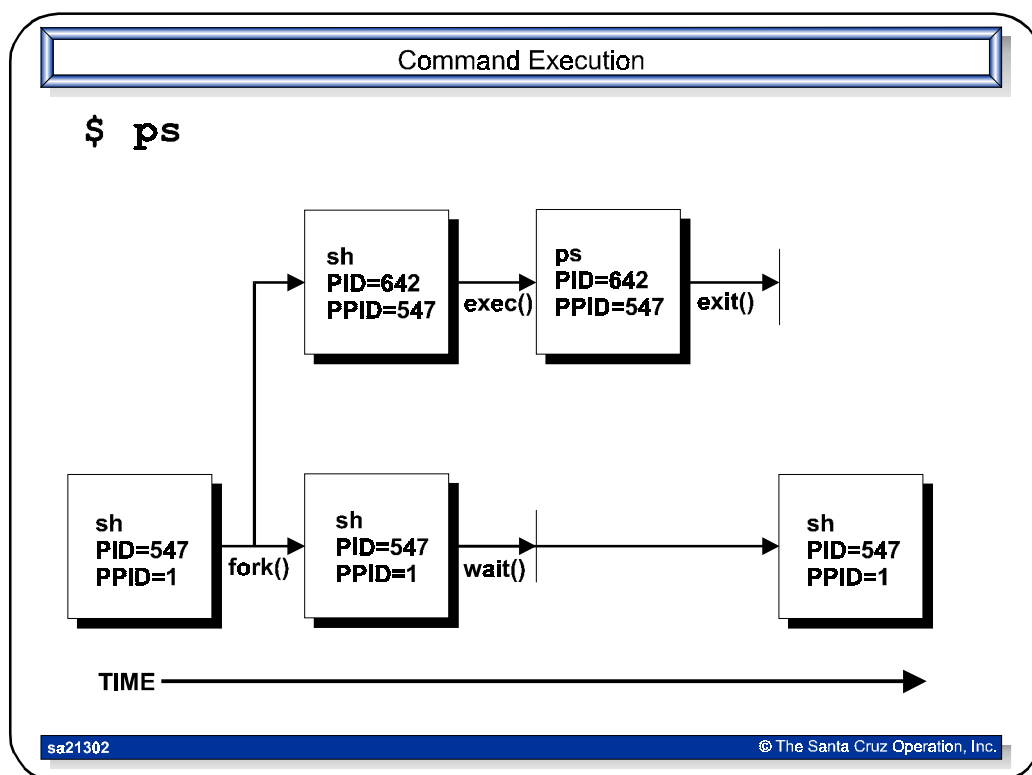
### **Objectives**

---

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
- Use system tools to gather information about processes
- Determine the priority of a process
- Use process accounting to monitor processes
- Use **crontab**(C) to automate system processes

AV 13-2



## 13.1 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(S)** 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(S)** 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(S)** system call that causes the parent to wait until the child process has finished and terminated.

## AV 13-3

| System Processes |     |      |   |        |     |          |                          |
|------------------|-----|------|---|--------|-----|----------|--------------------------|
| UID              | PID | PPID | C | STIME  | TTY | TIME     | CMD                      |
| root             | 0   | 0    | 0 | Mar-24 | ?   | 00:00:00 | sched                    |
| root             | 1   | 0    | 0 | Mar-24 | ?   | 00:00:28 | /etc/init -a             |
| root             | 2   | 0    | 0 | Mar-24 | ?   | 00:00:31 | vhand                    |
| root             | 3   | 0    | 0 | Mar-24 | ?   | 00:01:42 | bdflush                  |
| root             | 4   | 0    | 0 | Mar-24 | ?   | 00:00:00 | kmdaemon                 |
| root             | 5   | 1    | 0 | Mar-24 | ?   | 00:10:29 | htepi_daemon             |
| root             | 6   | 0    | 0 | Mar-24 | ?   | 00:00:00 | vdddaemon                |
| root             | 7   | 0    | 0 | Mar-24 | ?   | 00:00:00 | strd                     |
| root             | 35  | 1    | 0 | Mar-24 | ?   | 00:00:54 | /etc/ifor_pmd            |
| root             | 36  | 35   | 0 | Mar-24 | ?   | 00:00:02 | /etc/sco_cpd             |
| root             | 31  | 1    | 0 | Mar-24 | ?   | 00:00:31 | /etc/syslogd             |
| root             | 28  | 1    | 0 | Mar-24 | ?   | 00:00:02 | htepi_daemon /stand      |
| root             | 446 | 1    | 0 | Mar-24 | ?   | 00:01:25 | /usr/lib/scosh/caldaemon |
| root             | 278 | 1    | 0 | Mar-24 | ?   | 00:01:01 | /usr/lib/lpsched         |
| root             | 267 | 1    | 0 | Mar-24 | ?   | 00:00:14 | /etc/cron                |
| root             | 947 | 364  | 0 | Mar-24 | ?   | 00:00:05 | /usr/bin/X11/scologin    |

sa21303

© The Santa Cruz Operation, Inc.

## 13.2 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.

### 13.2.1 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(C)** command output.

#### **Exercise: Identifying daemon processes**

---

Use **ps(C)** to list all current daemon processes on your classroom system. You might also try using **grep(C)** with it to reduce the amount of extraneous output.

## AV 13-4

| System Processes |     |      |   |        |     |          |                          |
|------------------|-----|------|---|--------|-----|----------|--------------------------|
| UID              | PID | PPID | C | STIME  | TTY | TIME     | CMD                      |
| root             | 0   | 0    | 0 | Mar-24 | ?   | 00:00:00 | sched                    |
| root             | 1   | 0    | 0 | Mar-24 | ?   | 00:00:28 | /etc/init -a             |
| root             | 2   | 0    | 0 | Mar-24 | ?   | 00:00:31 | vhand                    |
| root             | 3   | 0    | 0 | Mar-24 | ?   | 00:01:42 | bdflush                  |
| root             | 4   | 0    | 0 | Mar-24 | ?   | 00:00:00 | kmdaemon                 |
| root             | 5   | 1    | 0 | Mar-24 | ?   | 00:10:29 | htepi_daemon             |
| root             | 6   | 0    | 0 | Mar-24 | ?   | 00:00:00 | vdddaemon                |
| root             | 7   | 0    | 0 | Mar-24 | ?   | 00:00:00 | strd                     |
| root             | 35  | 1    | 0 | Mar-24 | ?   | 00:00:54 | /etc/ifor_pmd            |
| root             | 36  | 35   | 0 | Mar-24 | ?   | 00:00:02 | /etc/sco_cpd             |
| root             | 31  | 1    | 0 | Mar-24 | ?   | 00:00:31 | /etc/syslogd             |
| root             | 28  | 1    | 0 | Mar-24 | ?   | 00:00:02 | htepi_daemon /stand      |
| root             | 446 | 1    | 0 | Mar-24 | ?   | 00:01:25 | /usr/lib/scosh/caldaemon |
| root             | 278 | 1    | 0 | Mar-24 | ?   | 00:01:01 | /usr/lib/lpsched         |
| root             | 267 | 1    | 0 | Mar-24 | ?   | 00:00:14 | /etc/cron                |
| root             | 947 | 364  | 0 | Mar-24 | ?   | 00:00:05 | /usr/bin/X11/scologin    |

sa21304

© The Santa Cruz Operation, Inc.



The following is a partial listing of system daemon processes:

|                            |                                                                                                                                                                          |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>sched</b> —PID 0        | Is the first process created by the UNIX kernel<br><br>Creates other daemon processes at system startup<br><br>Swaps processes out of memory into the swap area and back |
| <b>init</b> —PID 1         | Initializes the boot and login processes<br><br>Is the ancestor of all user processes<br><br>Is the ancestor of non- <b>sched</b> daemon processes                       |
| <b>vhand</b> —PID 2        | Moves infrequently used pages out of memory into the swap area                                                                                                           |
| <b>bdflush</b> —PID 3      | Flushes the buffer cache to disk at specified intervals                                                                                                                  |
| <b>kmdaemon</b> —PID 4     | Maintains a pool of memory that is reserved for immediate availability to interrupt service routines                                                                     |
| <b>htepi_daemon</b> —PID 5 | Handles asynchronous log file updates for the <b>ht</b> filesystem driver                                                                                                |
| <b>strd</b> —PID 6         | Is the streams daemon which allocates and de-allocates streams memory                                                                                                    |

### **13.2.2 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.

## 13.3 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.

### 13.3.1 The `ps(C)` 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.

```
ps -elf unix.old
```

### **Exercise: Viewing all active processes**

---

Use **ps(C)** 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 13-5

| Status of Processes |   |      |     |      |   |     |    |     |          |                   |
|---------------------|---|------|-----|------|---|-----|----|-----|----------|-------------------|
| F                   | S | UID  | PID | PPID | C | PRI | NI | TTY | TIME     | CMD               |
| 71                  | S | root | 0   | 0    | 0 | 95  | 20 | ?   | 00:00:00 | sched             |
| 20                  | S | root | 1   | 0    | 0 | 66  | 20 | ?   | 00:00:05 | /etc/init -a      |
| 71                  | S | root | 2   | 0    | 0 | 95  | 20 | ?   | 00:00:00 | vhand             |
| 71                  | S | root | 3   | 0    | 0 | 95  | 20 | ?   | 00:00:01 | bdf flush         |
| 61                  | O | root | 4   | 0    | 0 | 36  | 39 | ?   | 00:00:00 | CPU1 idle process |
| 61                  | R | root | 5   | 0    | 0 | 36  | 39 | ?   | 00:00:00 | CPU2 idle process |
| 61                  | O | root | 6   | 0    | 0 | 36  | 39 | ?   | 00:00:00 | CPU3 idle process |
| 61                  | R | root | 7   | 0    | 0 | 36  | 39 | ?   | 00:00:00 | CPU4 idle process |

sa21305

© The Santa Cruz Operation, Inc.

Understanding the output of the **ps(C)** 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 octal flags. For example, the <b>sched</b> process has a setting of 71, which is the sum of the flags 1, 10, 20, and 40. This means that the <b>sched</b> process is part of the kernel (1), sleeping at a priority of 77 or more (10), is loaded (20), and locked(40) in primary memory. |
| S (State of the process)       | A process can be in one of four states: S) the process is sleeping; R) the process is ready to run; O) the process is currently running on the processor; Z) the process is a zombie (defunct)                                                                                                                                                             |
| 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.                                                                                                                                                                          |
| C (CPU usage)                  | The recent CPU usage by the process. Used in conjunction with the nice value to calculate the priority of a process. The higher the value, the lower the priority. This value decreases over time when the process is waiting to get the CPU again.                                                                                                        |
| 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.                                                                                                                                                             |

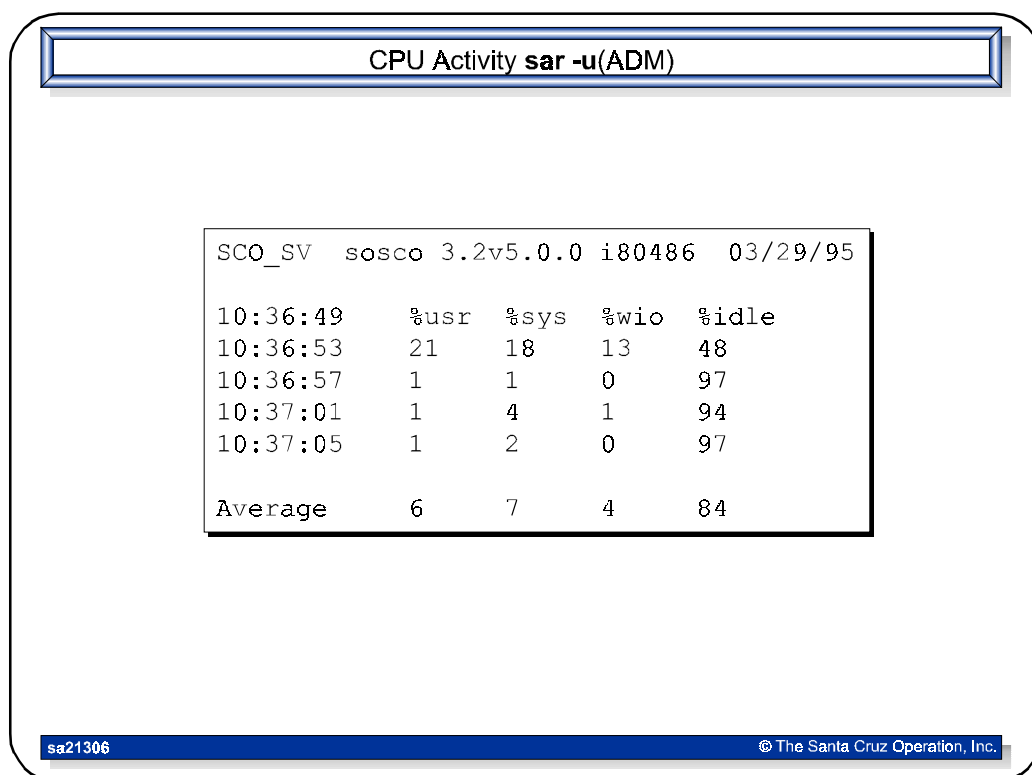
---

**Exercise: Determining the CPU time of a process**

---

Use **ps(C)** to determine which processes are receiving the most CPU time.

AV 13-6



### 13.3.2 The **sar**(ADM) command

---

The **sar**(ADM) command reports system activity on a single processor system. The **cpusar**(ADM) and **mpsar**(ADM) commands report activity on multiprocessor systems. The **cpusar** command reports activity per processor. The **mpsar** command reports activity for all processors.

Although the **sar**(ADM) program reports on many different kinds of system activity, the options discussed in this lesson are related to monitoring process activity.

To determine if a runaway process is “hogging the CPU,” use the **-u** option with **sar** and **mpsar**:

|              |                                                                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>%usr</b>  | Indicates the percentage of time the system is executing user processes                                                                  |
| <b>%sys</b>  | Indicates the percentage of time the system is executing system processes                                                                |
| <b>%wio</b>  | 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 |
| <b>%idle</b> | Indicates the percentage of time the operating system is idle with no runnable processes                                                 |

If **%idle** falls consistently below 10%, the critical resource is processor power. Run the **ps**(C) command to check if the excessive CPU usage is due to a runaway process.

AV 13-7

Examining the Run Queue `sar -q(ADM)`

```
SCO_SV  sosco 3.2v5.0.0 i80486    03/28/95

06:56:55  runq-sz  %runocc  swpq-sz  %swpocc
06:57:00   1.7    98      1.5      36
06:57:05   1.0    63      1.0      31
06:57:10   1.0    58      1.0      49
Average   1.3     74      1.2      39
```

sa21307

© 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**(ADM) or the **mpsar**(ADM) command:

**runq -sz**      Indicates the number of runnable processes in memory

**%runocc**      Indicates the percentage of the run queue in memory

**swpq -sz**      Indicates the number of runnable processes on swap

**%swpocc**      Indicates the percentage of the run queue in swap

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 multiprocessing system might improve response time.

### **13.3.2.1 Running the sar(ADM) 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
```

### **13.3.2.2 Running the sa1(ADM) and sa2(ADM) commands**

---

Two programs have been set up for you to collect and report system activity reporting. These programs are run as a **crontab** and are not automatically started on your system.

The information is stored on a monthly basis in the */usr/adm/sa* directory. For example, the files *sa10* and *sa11* represent the 10th and 11th day of the current month.

The **sa1**(ADM) command collects and stores data in the binary file */usr/adm/sa/sadd*, where *dd* is the current day of the month. It can be started from the sys **crontab** in */usr/spool/cron/crontabs/sys*.

The **sa2**(ADM) command writes a readable daily report to the file */usr/adm/sa/sardd*, where *dd* is the current day of the month. It can be started from the root **crontab** in */usr/spool/cron/crontabs/root*.

To change the running time of these programs, edit the appropriate **crontab** file.

To enable the automatic collecting of system data and print reports:

Step 1. Enter:

```
/usr/lib/sa/sar_enable -y
```

Step 2. Display CPU usages from the file */usr/adm/sa/sa10*:

```
sar -q -f /usr/adm/sa/sa10
```

### **Exercise: Enabling the automatic collection of system data**

Use the steps outlined in “Running the **sa1**(ADM) and **sa2**(ADM) commands” to enable the automatic collection of system reporting activity.

Once enabled, display a report of CPU usage for the current date.

### 13.3.3 The **timex(ADM)** command

---

When you run **timex(ADM)** 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(ADM)** manual page for additional information.

#### **Exercise: Recording the output of the **timex(ADM)** command**

---

How would you record the **timex(ADM)** output of the following command in a data file?

AV 13-8

| Priority Values |                                          |
|-----------------|------------------------------------------|
| Priority        | Meaning                                  |
| 95              | swapping/paging                          |
| 88              | waiting for an inode                     |
| 81              | waiting for i/o                          |
| 80              | waiting for buffer                       |
| 76              | waiting for pipe                         |
| 75              | waiting for tty input                    |
| 74              | waiting for tty output                   |
| 73              | waiting for exit                         |
| 66              | sleeping --- lowest system mode priority |
| 65              | highest user mode priority               |
| 51              | default user mode priority               |
| 0               | lowest user mode priority                |

sa21308

© The Santa Cruz Operation, Inc.

## 13.4 Prioritizing Processes

---

A process's priority can range between 0 (lowest priority) and 127 (highest priority). User processes run at lower priorities than system processes.

Some of the system-mode priorities indicate what a process is waiting for. For example, a priority of 81 indicates that a process is waiting for I/O to complete; a value of 75 means that it is waiting for keyboard input.

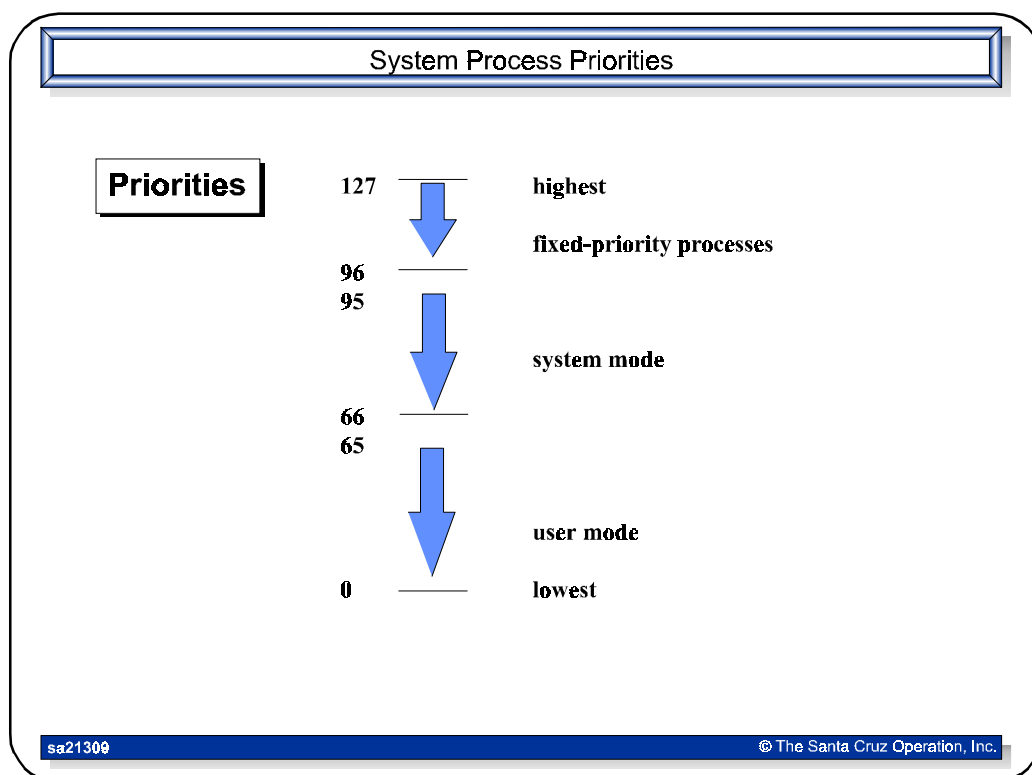
### **Exercise: Determining what a process is waiting for**

---

Use **ps(C)** to determine the priority of a **getty(C)** process.

From the graphic on the adjoining page, determine what the **getty** process is waiting for.

AV 13-9



### 13.4.1 Priority ranges

---

The following list describes the priority ranges for processes:

- User processes run at a lower value than system processes. A user process can have a priority of 0 to 65.
- System processes range from 66 to 95.
- Fixed priority processes range from 96 to 127 and control their own scheduling. An application must be coded to run as a fixed priority process. They are not otherwise created by the operating system.

### 13.4.2 Calculating process priority

---

The operating system varies the priorities of executing processes according to a simple algorithm. This formula ensures that each process on the system gets fair access to the CPU.

Every process receives a base level priority of 51 when it is created. This priority is quickly adjusted. The priorities of kernel daemons such as **sched**, **vhand** and **bdfush**, are not adjusted.

The kernel recalculates the priority of a running process every clock tick. The new priority is based on how much CPU time the process has used and the process's nice value.

When taken off the CPU, the process's lowered priority pushes it down the run queue to decrease the probability of being chosen in the near future.

A process that manages to run for an entire time slice will have its priority reduced by the maximum amount.

### 13.4.3 Controlling the nice value

---

The **nice(C)** 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 a new process. Only **root** can set a lower nice value which results in a higher priority.

The **renice(C)** 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(C)** and **renice(C)** manual pages for information on increasing or decreasing the nice value of a process.



## 13.5 Managing Processes with the Process Manager

---

---

You can also use the **SCOadmin Process Manager** to perform some of the tasks associated with the management of processes on a local or remote system. Use the **Process Manager** to:

- View processes by user, terminal, or PID
- Examine a process
- Change the priority of an existing process
- Delete (kill) processes

### **Exercise: Using the Process Manager**

---

Start the **Process Manager** and view the processes associated with `/dev/tty02`.

## 13.6 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 amount of resources in terms of CPU usage and disk space.

### 13.6.1 Enabling process accounting

---

Process accounting can be started from the command line or automatically at boot time. Process accounting records are stored in the */usr/adm/pacct* file.

To enable process accounting from the command line, enter:

**`/usr/lib/acct/startup`**

To enable process accounting during multiuser initialization, add the following line to the */etc/rc2.d/P88USRDEFINE* file:

**`/usr/lib/acct/startup &`**

To run process accounting on a periodic basis, edit the */usr/spool/cron/crontabs/adm* **crontab** file and add the instructions to start and stop the process accounting.

Refer to **acct\_enable(ADM)** manual page for more information on the *adm crontab* file.

#### **Exercise: Enabling process accounting**

---

Enable process accounting from the command line.

### 13.6.2 Disabling process accounting

---

To disable process accounting, enter:

**/usr/lib/acct/shutacct**

### 13.6.3 Reading process accounting files

---

Accounting takes place at the termination of each process. One record is written to the accounting file for each command executed.

To read the process accounting file and display the contents to your terminal, enter:

**acctcom /usr/adm/pacct**

Process accounting files contain the following seven fields:

- Command name
- User login name
- Name of the tty device
- Starting time of the process
- Ending time of the process
- CPU time in seconds
- Meansize

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?

### 13.6.4 Generating process accounting reports

---

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 /usr/adm/pacct
```

- By group:

```
acctcom -g group_name /usr/adm/pacct
```

- By terminal:

```
acctcom -l /dev/ttyname /usr/adm/pacct
```

- By the most recent process:

```
acctcom -b /usr/adm/pacct
```

#### **Exercise: Generating process accounting reports**

---

Beginning with the most recent process, display all records for the **root** user.  
When completed, turn off process accounting.

### 13.6.5 Clearing process accounting files

---

Process accounting files should be cleared on a regular basis. Before clearing, you may want to print or archive your file.

To manually clear */usr/adm/pacct*, enter:

```
> /usr/adm/pacct
```

To automatically clear */usr/adm/pacct*, see the *adm crontab* in */usr/spool/cron/crontabs*.

## 13.7 Automating Processes Using cron(C)

---

One of the keys to efficient administration is the automation of system tasks. These tasks can include system backups, process accounting, process monitoring using the **sar(C)** command, and clearing log files.

SCO OpenServer systems provide you with five system **crontab** files. These files can be modified to fit your systems needs.

AV 13-10

**System Crontabs**

**System Crontabs**  
**/usr/spool/cron/crontabs**

|            |   |      |      |      |     |    |       |      |
|------------|---|------|------|------|-----|----|-------|------|
| -rw-rw---- | 1 | adm  | cron | 2491 | Jan | 31 | 22:54 | adm  |
| -rw-rw---- | 1 | root | cron | 816  | Mar | 10 | 12:30 | root |
| -rw-rw---- | 1 | sys  | cron | 750  | Mar | 10 | 12:30 | sys  |
| -rw-rw---- | 1 | bin  | cron | 811  | Jan | 31 | 22:55 | uucp |
| -rw-rw---- | 1 | root | cron | 815  | May | 17 | 11:52 | mmdf |

sa21310© The Santa Cruz Operation, Inc.

### 13.7.1 System **crontabs**

---

The system **crontab** files are stored in the */usr/spool/cron/crontabs* directory. They include:

|             |                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>adm</i>  | Manages process accounting and collects log files                                                                                                                                                                                        |
| <i>mmdf</i> | Sends warnings about mail that is undeliverable and removes extraneous file                                                                                                                                                              |
| <i>root</i> | Clears log files and removes core files, checks the size of <i>/usr/lib/cron</i> and copies it to <i>/usr/lib/cron/olog</i> , removes the files in the <i>/tmp</i> directory, sets the system clock, and creates system activity reports |
| <i>sys</i>  | Collects performance information using the <b>sar</b> (ADM) command                                                                                                                                                                      |
| <i>uucp</i> | Automates and performs maintenance tasks for UUCP                                                                                                                                                                                        |

The **cron**(C) command does not execute any **crontab** entry preceded by a pound (#) sign.

Refer to **acct\_enable**(ADM) for information on administering the *adm* **crontab** file.

AV 13-11

**crontab(C) Parameters**

| Field         | Allowable Values                            |
|---------------|---------------------------------------------|
| Minute        | 0-59                                        |
| Hour          | 0-23                                        |
| Day of Month  | 1-31                                        |
| Month of Year | 1-12                                        |
| Day of Week   | 0-6 (0=Sunday)                              |
| Command Name  | Any non-interactive command or shell script |

- A field value can be a single value, range of values, or list of values separated by commas
- An asterisk selects all allowable values

sa21311

© The Santa Cruz Operation, Inc.



## 13.7.2 Editing system crontabs

---

Do not edit the files in `/usr/spool/cron/crontabs`. If you need to add entries, or edit existing entries in a **crontab** file, redirect the contents to a new file, make your changes, and submit the **crontab** file with the **crontab(C)** command.

The **cron(C)** command reads a **crontab** file when the system enters multiuser mode or when a **crontab** is submitted with the **crontab(C)** command.

Before editing any of the system **crontabs**, make sure that you are logged in as that user or use the **su(C)** command.

To edit the root **crontab**, follow these steps:

- Step 1.        Log in as **root**.
- Step 2.        Use **-l** option with the **crontab(C)** command to view the root **crontab**. Save the output in a file, enter:

```
crontab -l > crontab.eg
```

- Step 3.        Edit the file to add or modify entries.

- Step 4.        Use **crontab(C)** to submit your modified entrie, enter:

```
crontab crontab.eg
```

You can also use the **-e** option with the **crontab(C)** command to edit a **crontab** file.

## 13.8 Lesson Summary

---

---

Use the **ps**(C) and **sar**(C) commands to monitor process activity and to distinguish between user- and system-generated processes. You can change process priorities using the **nice**(C) and **renice**(C) commands. Although process accounting is usually started at boot time, it can be stopped and started as necessary. Automate system processes using the **crontab**(C) 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. What command turns on process accounting?
6. Name the five system **crontab** files found in */usr/spool/cron/crontabs*.

## 13.9 Lab: Changing the root crontab

---

At peak periods during the day, the system is experiencing a slow down in operation. A memory shortage is suspected. For one week, you need to monitor memory usage during these peak periods.

### 13.9.1 Overview

---

Edit the root **crontab** and add an entry using the **sar**(ADM) command that will monitor memory usage at five minute intervals for five times. The **sar** command will run at 9:00 a.m., 11:00 a.m., 2:00 p.m, and 3:00 p.m., Monday through Friday. The output from the **sar** command is recorded (appended) into the */usr/sys/memuse* file.

### 13.9.2 Procedure

---

- Step 1. Log in as **root**.
- Step 2. Use **-l** option with the **crontab(C)** command and save the output to a file in */usr/sys/memcron*.
- Step 3. Change to the directory */usr/sys*.
- Step 4. Edit the *memcron* file and on a blank line, add the following entry:  
  
**0 9,11,14,15 \* \* 1-5 /usr/bin/sar -r 300 5 >> /usr/sys/memuse**
- Step 5. Save your changes and quit.
- Step 6. Using the **crontab(C)** command, submit the *memcron* file.

## 13.10 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### 13.10.1 Relevant commands

---

|                       |                                                        |
|-----------------------|--------------------------------------------------------|
| <b>acctcom</b> (ADM)  | Searches for and prints process accounting files       |
| <b>cron</b> (ADM)     | Executes commands submitted by <b>crontab</b> (C)      |
| <b>crontab</b> (C)    | Schedules commands to be executed at certain intervals |
| <b>ps</b> (C)         | Reports process status                                 |
| <b>sar</b> (ADM)      | Reports system activity                                |
| <b>startup</b> (ADM)  | Turns on process accounting                            |
| <b>shutacct</b> (ADM) | Turns off process accounting                           |

### 13.10.2 Relevant files

---

|                                 |                                                    |
|---------------------------------|----------------------------------------------------|
| <i>/usr/adm/pacct</i>           | Default process accounting file                    |
| <i>/usr/adm/sa</i>              | Directory that contains system activity reports    |
| <i>/usr/spool/cron/crontabs</i> | Directory for user and system <b>crontab</b> files |

## 13.11 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Understanding system calls

---

The **fork(S)** system call is not issued.

### Exercise: Generating process accounting reports

---

To generate process accounting reports, enter:

```
acctcom -b /usr/adm/pacct  
/usr/lib/acct/shutacct
```

### Exercise: Reading the process accounting file

---

Commands executed with super user privileges are prefixed with a # sign.

### Exercise: Enabling process accounting

---

To enable process accounting from the command line, enter:

```
/usr/lib/acct/startup
```

### Exercise: Determining what a process is waiting for

---

To determine what a process is waiting for, enter:

```
ps -ef | grep getty
```

### Exercise: Recording the output of the **timex(ADM)** command

---

To record the output of the **timex(ADM)** command, enter:

```
timex grep 'ansi' /etc/termcap 2> filename
```

## Exercise: Enabling the automatic collection of system data

---

To enable the automatic collection of system data, enter:

```
/usr/lib/sa/sar_enable -y
```

```
sar -q -f /usr/adm/sa/sa < current date_of_month
```

## Exercise: Determining the CPU time of a process

---

Compare the **TIME** fields to determine the process that is using the most CPU time.

## Exercise: Viewing all active processes

---

To view all active processes, enter:

```
ps -ef | wc -l
```

## Exercise: Identifying daemon processes

---

To identify daemon processes, enter:

```
ps -e | grep '?' | more
```

## Review questions

---

This section provides suggested answers to the 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 **sched** process swaps data to the hard disk if memory is not available.
3. You can use the **ps(C)** and **sar(ADM)** commands to identify how much CPU time programs are taking.
4. The default nice value for a user process is 20.
5. The `/usr/lib/acct/startup` command turns on process accounting.
6. The five system crontab files found in `/usr/spool/cron/crontabs` are: *root*, *sys*, *adm*, *mmdf*, and *uucp*.

---

# ***14***

## ***Managing System Security***

AV 14-1

Managing System Security

- Describe trusted system concepts
- Describe and select a security profile
- Change your system's security profile
- Review the account database files
- Determine appropriate subsystem and user authorizations
- Configure a user account to execute root commands
- Use advanced protection bits
- Manage, maintain the integrity of, and repair the Trusted Computing Base (TCB)
- Configure additional security measures

sa21401© The Santa Cruz Operation, Inc.



**Lesson 14**

## **Managing System Security**

### **Introduction**

---

Every computer system needs protection from people accessing the computer disks and system files without permission. SCO OpenServer 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 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
- Determine appropriate subsystem and user authorizations
- Configure a user account to execute **root** commands
- Use advanced protection bits
- Manage the Trusted Computing Base (TCB)
- Configure additional security measures



## **14.1 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, SCO has incorporated the C2 level of trust, known as the Trusted Computing Base (TCB). The TCB 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 to what 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.
- Authorizationsspecify user attributes required to perform a certain action.
- Protected Subsystems offer related collections of files, devices, and commands that serve particular functions. C2 provides control of the users who maintain these functions.
- The TCB 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 14-2

| System Security Profiles - Logins                   |          |             |          |      |
|-----------------------------------------------------|----------|-------------|----------|------|
| Security Parameters                                 | Low      | Traditional | Improved | High |
| <b>Logins</b>                                       |          |             |          |      |
| Maximum unsuccessful attempts (account/terminal)    | Infinite | 99          | 5/9      | 3/5  |
| Delay between login attempts (secs) - terminal only | 0        | 1           | 2        | 2    |
| Time to complete login (terminal only)              | 60       | 60          | 60       | 60   |

sa21402 © The Santa Cruz Operation, Inc.

## AV 14-3

| System Security Profiles - Passwords |          |             |             |               |
|--------------------------------------|----------|-------------|-------------|---------------|
| Security Parameters                  | Low      | Traditional | Improved    | High          |
| <b>Passwords</b>                     |          |             |             |               |
| Minimum days between changes         | 0        | 0           | 0           | 14            |
| Expiration time (days)               | Infinite | Infinite    | 42          | 42            |
| Lifetime (days)                      | Infinite | Infinite    | 365         | 90            |
| User can choose own                  | Yes      | Yes         | Yes         | No            |
| User can run generator               | Yes      | Yes         | Yes         | Yes           |
| Maximum generated length             | 8        | 8           | 10          | 10            |
| Minimum Length                       | 1        | 3           | 5           | 8             |
| Password triviality checks           | None     | System V    | Goodpw weak | Goodpw strong |
| Checked for obviousness              | -        | No          | No          | Yes           |
| Password required to login           | No       | No          | Yes         | Yes           |
| Single User Password required        | Yes      | Yes         | Yes         | Yes           |

sa21403 © The Santa Cruz Operation, Inc.

## **14.2 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. After you select a profile, you can switch to another profile, or change any one of the parameters on an individual basis.

### **14.2.1 The high security profile**

---

The high security profile provides the “C2” level of trust. The C2 class is a level of trust defined by the Department of Defense Trusted Computer System Evaluation Criteria.

The High security profile is recommended for systems containing confidential information accessed by many users. With a high security profile:

- Passwords are strictly controlled and generated by the system; users cannot select their own passwords.
- User accounts cannot be unretired or removed.
- All features of C2 are active.

### **14.2.2 The improved security profile**

---

The improved security profile meets the minimum requirements of the C2 level of trust. It is recommended for systems accessed by groups of users who can share information. With an improved security profile:

- Password expiration is more lenient and users can choose their own passwords.
- User accounts cannot be unretired or removed.
- Features of C2 are active, using parameters that are less restrictive than the High security profile.

## AV 14-4

| System Security Profiles - Logins                   |          |             |          |      |
|-----------------------------------------------------|----------|-------------|----------|------|
| Security Parameters                                 | Low      | Traditional | Improved | High |
| <b>Logins</b>                                       |          |             |          |      |
| Maximum unsuccessful attempts (account/terminal)    | Infinite | 99          | 5/9      | 3/5  |
| Delay between login attempts (secs) - terminal only | 0        | 1           | 2        | 2    |
| Time to complete login (terminal only)              | 60       | 60          | 60       | 60   |

sa21404 © The Santa Cruz Operation, Inc.

## AV 14-5

| System Security Profiles - Passwords |          |             |             |               |
|--------------------------------------|----------|-------------|-------------|---------------|
| Security Parameters                  | Low      | Traditional | Improved    | High          |
| <b>Passwords</b>                     |          |             |             |               |
| Minimum days between changes         | 0        | 0           | 0           | 14            |
| Expiration time (days)               | Infinite | Infinite    | 42          | 42            |
| Lifetime (days)                      | Infinite | Infinite    | 365         | 90            |
| User can choose own                  | Yes      | Yes         | Yes         | No            |
| User can run generator               | Yes      | Yes         | Yes         | Yes           |
| Maximum generated length             | 8        | 8           | 10          | 10            |
| Minimum Length                       | 1        | 3           | 5           | 8             |
| Password triviality checks           | None     | System V    | Goodpw weak | Goodpw strong |
| Checked for obviousness              | -        | No          | No          | Yes           |
| Password required to login           | No       | No          | Yes         | Yes           |
| Single User Password required        | Yes      | Yes         | Yes         | Yes           |

sa21405 © The Santa Cruz Operation, Inc.

### **14.2.3 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 C2 security features.

With a traditional security profile:

- Passwords are not required, but are recommended.
- User accounts can be retired, unretired, and removed.
- Features of C2 can be engaged.

### **14.2.4 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 removed.
- Features of C2 are not active.

## 14.3 Viewing and Changing your System Security Profile

---

You are asked to select a security profile at installation time. Later, however, it is possible to select a different profile by using the **Security Profile Manager** or running the **relax**(ADM) command.

After using lower security profiles, it is possible to select the *Improved* or *High* defaults, but this does not mean your system conforms to the requirements of a C2 system. By definition, a C2 system must adhere to the requirements from initial installation. This is because modifications made to the system while at the lower level violate those associated with the higher level.



### 14.3.1 Determining your security profile

---

You can determine your current security profile using **scoadmin**(ADM).

#### **Exercise: Determining your security profile**

---

---

To determine your security profile:

Run **scoadmin**(ADM).

Select **System** ⇒ **Security** ⇒ **Security Profile Manager**.

To also determine your security profile, enter the **secdefs**(ADM) command:

**secdefs**

Other options to the **secdefs** command are discussed later in this lesson.

## AV 14-6

Default Security Profile

```
default:\n  :d_name=default:\n  :u_pwd=*\n  :u_priority#0:u_cmdpriv=audittrail,su,queryspace, printqueue, mem, terminal:\n  :u_syspriv=execsuid,nopromain,chmodsugid,chown:\n  :u_minchg#0:u_maxlen#8:u_exp#0:u_life#0:\n  :u_pickpw:u_genpwd:u_restrict@:u_nullpw:\n  :u_suclog#0:u_unsuclog#0:u_maxtries#99:u_lock:\n  :u_singleuserpswd:U_secclass=c2:u_integrity@:u_tcbpw@:\n  :u_pwseg#1:\n  :t_logdelay#1:t_maxtries#99:t_login_timeout#60:\n  :chkent:
```

/etc/auth/system/default

sa21406© The Santa Cruz Operation, Inc.

### 14.3.2 Selecting a security profile

---

If a different security profile is selected after installation, when you enter the **Security Profile Manager**, the name of the current security profile displays along with the message **The security subsystem has been modified**. This message also appears if a parameter within the profile has been modified.

Files that define each security profile are stored in the */tcb/lib/relax* directory. For example, the file that defines Improved security is */tcb/lib/relax/improved/default*.

The following steps describe how to select *Improved* security:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(ADM)**.
- Step 3. Select **System** ⇒ **Security** ⇒ **Security Profile Manager**.
- Step 4. Select **Current security profile**.
- Step 5. Select **Improved** ⇒ **Security** ⇒ **Save**.

The */tcb/lib/relax/improved/default* file is copied to */etc/auth/system/default*.

- Step 6. Select **Security** ⇒ **Exit** to exit the **Security Profile Manager**.
- Step 7. Reboot your system to invoke the new parameters:

**/etc/shutdown -i6 -g0 -y**

You can also use the **relax(ADM)** 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:

**/etc/shutdown -i6 -g0 -y**

### 14.3.3 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.

You can use the **usermod**(ADM) command to:

- Change a parameter in your security profile that affects all users
- Quickly change parameters on an individual user basis

For example, to change the maximum number of login attempts to 10 for all users, enter:

```
usermod -D -x "{maxLoginAttempts 10}"
```

The **-D** option operates on system defaults instead of an individual account. If any users are currently logged in, changes do not go into effect until the next login.

Always change your security profile using **scoadmin**(ADM) or **usermod**. SCO recommends that you do *not* edit these files from the command line.

If changes have been made to your current security profile, use the **secdefs**(ADM) command to compare the current settings of the system-wide security defaults with the security profiles defined in */tcb/lib/relax* (High, Improved, Traditional, and Low)

### **Exercise: Changing security parameters**

---

View the `/etc/auth/system/default` file. Write down the value of the parameter `u_maxtries#`.

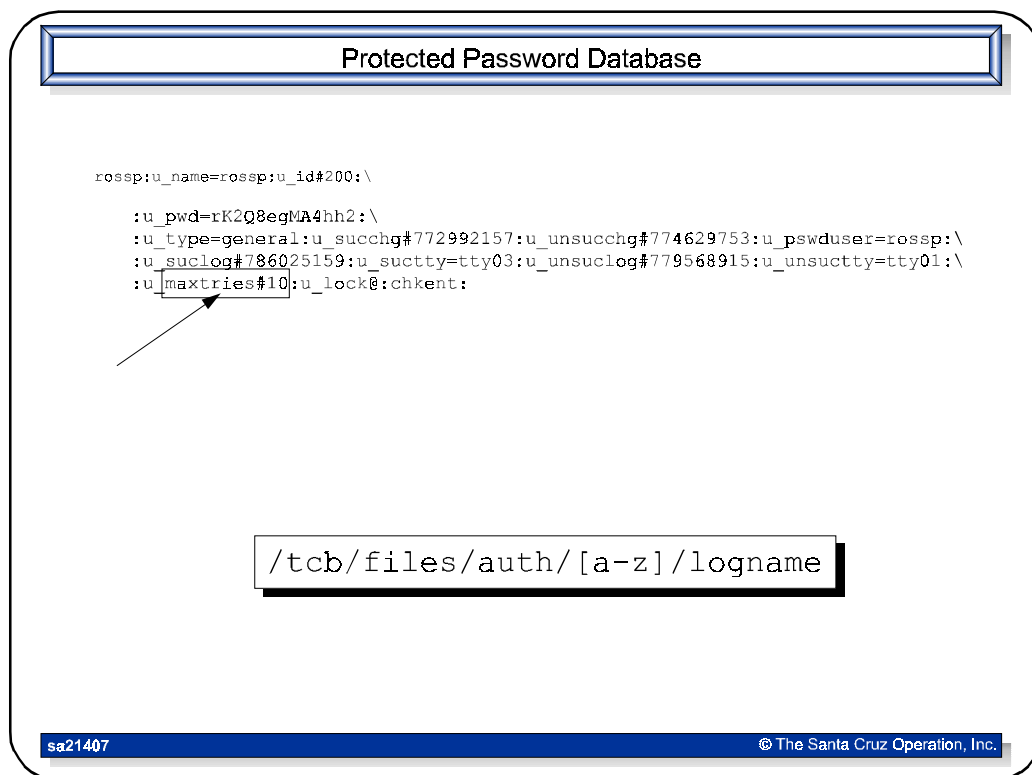
Use **usermod**(ADM) to change the number of login attempts for all users to 20.

To verify the change, enter:

**secdefs -v**

Use the online manual pages to determine the output.

## AV 14-7



### 14.3.4 Changing a parameter for an individual user

---

The maximum number of login attempts for a user is 99 when using the Traditional profile. For security reasons, you may want to decrease the number of login attempts for one or more users.

The following steps describe how to change the number of unsuccessful logins for an individual user:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin**(ADM).
- Step 3. Select **Account Manager**.
- Step 4. Select the user name.
- Step 5. Select **Users** ⇒ **Login Controls**.
- Step 6. Deselect the default value for “Failed login attempts allowed before account is locked.”
- Step 7. Select the box next to “Failed login attempts allowed before account is locked” and enter the desired number.
- Step 8. Click on **OK**.  
  
The message **Changing user login controls** appears.
- Step 9. Select **Host** ⇒ **Exit** to exit the **Account Manager**.

**Exercise: Changing individual parameters**

---

Using the steps outlined in “Changing a parameter for an individual user,” change the number of unsuccessful login attempts to 10 for your student account.



After you complete the exercise:

- The parameter `u_maxtries#` in `/tcb/files/auth/[a-z]/logname` is modified to read:  
`u_maxtries#10`
- The change takes effect on the user's next login

You can also use the **usermod**(ADM) command to change individual security parameters. For example, to change the maximum login attempts to 20 for the user **cindy**:

```
usermod -x "{maxLoginAttempts 20}" cindy
```

Refer to the **usermod**(ADM) manual page for information on changing additional parameters.

## AV 14-8

***/etc/passwd and /etc/group (Partial Contents)******/etc/passwd***

```
root:x:0:3:Superuser:/:  
aaronm:x:200:100:Training Coordinator:/u/aaronm:/bin/sh  
chuckc:x:201:102:Graphic Artist:/u/chuckc:/bin/sh  
sheilage:x:202:101:Sales Department:/u/sheilage:/bin/sh  
barbme:x:203:101:Sales Department:/u/barbme:/bin/sh
```

***/etc/group***

```
root::0:root  
group::50:  
training::100:aaronm  
sales::101:sheilage,barbme  
graphics::102:chuckc
```

sa21408

© The Santa Cruz Operation, Inc.

## AV 14-9

***/etc/default/login and /etc/default/passwd******/etc/default/login***

```
ALTSHELL=YES  
ULIMIT=2097151  
PATH=/bin:/usr/bin  
OVERRIDE=tty01  
PASSREQ=NO  
IDLEWEEKS=0  
ALLOWHUSH=YES  
REUSEUID=YES
```

***/etc/default/passwd***

```
PASSLENGTH=3  
MINWEEKS=0  
MAXWEEKS=0  
RETRIES=3  
ONETRY=YES  
DESCRIBE=/dev/null  
SUMMARY=/dev/null  
CHECKDIR=/usr/lib/goodpw/checks  
GOODPW=YES
```

sa21409

© The Santa Cruz Operation, Inc.

## 14.4 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 Computing Base (TCB) that extends System V Security.

### 14.4.1 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(C)** and **login(C)** manual pages for more information on password restrictions and login controls.

## 14.4.2 Trusted Computing Base Files

---

Trusted Computing Base files are updated automatically when using **scoadmin**(ADM) and the **usermod**(ADM) command to make changes. SCO recommends that you do *not* edit these files from the command line.

Trusted Computing Base files include:

*/tcb/files/auth/[a-z]/username*

The Protected Password database. This file implements the requirements for the C2 level of trust. For the user, it contains the encrypted password, user specific system privileges, and parameters that override the system default database.

*/etc/auth/system/default*

The System Defaults database. This file contains your system security profile.

## 14.5 Delegating System Administration Responsibility

---

Your role as a system administrator 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.

You delegate this responsibility by assigning *authorizations* to users. An authorization grants the user the power to execute the commands and access the files related to one of the following types of *subsystems*:

|                     |                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------|
| Primary subsystem   | A group of related files and commands needed to accomplish all facets of a system administration task |
| Secondary subsystem | A subset of a primary subsystem that grants a user limited capabilities                               |

If you want a user to be responsible for the UNIX print service but do not want that user to perform any other administration duties, you would assign the **lp** primary subsystem authorization. However, if you wanted the user only to enable and disable printers, you would assign the user the **printerstat** secondary subsystem authorization.

When assigning a primary subsystem authorization, secondary subsystem authorizations are automatically assigned.

Primary and secondary authorizations are dependent on user system privileges. For example, when assigning the **lp** subsystem, the user must also be assigned the **chown** user system privilege.

The installed security profile determines which authorizations and privileges are assigned by default.

## AV 14-10

| Subsystem Authorizations |                      |                                                                             |
|--------------------------|----------------------|-----------------------------------------------------------------------------|
| Authorization            | SCO admin<br>Manager | Powers                                                                      |
| <b>mem</b>               | -                    | Access to system data tables, listing all processes on the system           |
| <b>terminal</b>          | -                    | Unrestricted use of the <b>write(C)</b> command                             |
| <b>lp</b>                | Printer Manager      | Administer printers                                                         |
| <b>backup</b>            | Backup Manager       | Perform backups                                                             |
| <b>auth</b>              | Account Manager      | Administer accounts: adding users, changing passwords                       |
| <b>audit</b>             | Audit Manager        | Audit administrator: run system audits and generate reports                 |
| <b>cron</b>              | Job Manager          | Control use of <b>cron(C)</b> , <b>at(C)</b> , and <b>batch(C)</b> commands |
| <b>root</b>              | -                    | Allows use of any command found in <i>/tcb/files/rootcmds</i>               |
| <b>sysadmin</b>          | -                    | Not implemented                                                             |

sa21410

© The Santa Cruz Operation, Inc.

## 14.5.1 Primary subsystem authorizations

---

Default primary authorizations are assigned when the security profile is chosen. The subsystems and the tasks associated with each are as follows:

|                 |                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>auth</b>     | Perform user administration using the <b>Account Manager</b>                                                            |
| <b>backup</b>   | Perform system backups using the <b>Backup Manager</b>                                                                  |
| <b>lp</b>       | Administer the UNIX print system using the <b>Printer Manager</b>                                                       |
| <b>audit</b>    | Determine who and what is audited, generate reports, and maintain audit collection files using the <b>Audit Manager</b> |
| <b>cron</b>     | Control use of the <b>cron(C)</b> , <b>at(C)</b> , and <b>batch(C)</b> commands using the <b>Cron Manager</b>           |
| <b>mem</b>      | View all processes running on the system                                                                                |
| <b>terminal</b> | Send and receive messages using the <b>write(C)</b> and <b>hello(C)</b> commands                                        |
| <b>sysadmin</b> | Alter mount commands using the <b>Filesystem Manager</b>                                                                |
| <b>root</b>     | Execute a root command found in <code>/tcb/files/rootcmds</code>                                                        |

Refer to the **subsystem(M)** manual page for a detailed description of each subsystem. Refer to “Using the Audit Manager” in the *SCO OpenServer System Administration Guide* for information on managing the Audit Subsystem.

### **Exercise: Viewing primary authorizations**

---

View `/etc/auth/system/default` and determine your system's default primary authorizations.



### 14.5.1.1 Assigning primary authorizations

---

The following steps describe how to assign primary authorizations to a user:

- Step 1.        Log in as **root**.
- Step 2.        Run **scoadmin**(ADM).
- Step 3.        Select **Account Manager**.
- Step 4.        Select the user name.
- Step 5.        Select **Users** ⇨ **Authorizations**.
- Step 6.        Deselect the default value for “Use system default authorizations for this user account.”
- Step 7.        Select the primary authorization in the “Not Authorized” list that you want to assign to the user.
- Step 8.        Select **Add** to add that authorization to the “Authorized” list.
- Step 9.        Repeat Steps 7 and 8 for any other primary authorizations that you want to assign to the user. When complete, go to the next step.
- Step 10.       Click on **OK**.  
  
                  The message `Changing user authorizations` appears.
- Step 11.       Select **Host** ⇨ **Exit** to exit the **Account Manager**.

When you assign a primary authorization to a user, the user’s login name with the associated subsystem is written to `/etc/auth/subsystems/subsystem_name`.

This file and `/tcb/files/auth/[a-z]/logname` are checked each time for authorization when a user executes a command associated with a subsystem.

## AV 14-11

| Secondary Authorizations |                      |                                                                                                     |
|--------------------------|----------------------|-----------------------------------------------------------------------------------------------------|
| Secondary Authorization  | Parent Authorization | Description                                                                                         |
| <b>audittrail</b>        | audit                | Ability to generate personal audit reports on one's own activity.                                   |
| <b>backup_create</b>     | backup               | The ability to create (but not restore) backups.                                                    |
| <b>restore</b>           | backup               | The ability to restore (but not create) backups.                                                    |
| <b>queryspace</b>        | backup               | Use of <b>df(C)</b> command to query disk space.                                                    |
| <b>printqueue</b>        | lp                   | View all jobs in queue using <b>lpstat(C)</b> .                                                     |
| <b>printerstat</b>       | lp                   | Use of printer <b>enable/disable</b> commands.                                                      |
| <b>su</b>                | auth                 | Access to the <b>root</b> (superuser) account and other accounts. Access still requires a password. |
| <b>shutdown</b>          | root                 | Ability to use <b>shutdown(ADM)</b>                                                                 |
| <b>passwd</b>            | auth                 | Ability to change user passwords.                                                                   |

sa21411 © The Santa Cruz Operation, Inc.

## AV 14-12

| Installed Security ProfileDefault (Secondary Authorizations) |                                                                              |                                                                    |                                             |            |
|--------------------------------------------------------------|------------------------------------------------------------------------------|--------------------------------------------------------------------|---------------------------------------------|------------|
| Authorizations                                               | Low                                                                          | Traditional                                                        | Improved                                    | High       |
| <i>Subsystem</i>                                             | audittrail,<br>backup,<br>lp, mem,<br>queryspace,<br>shutdown,su<br>terminal | audittrail,<br>mem,<br>printqueue,<br>queryspace,<br>terminal, su, | audittrail,<br>queryspace,<br>printqueue,su | queryspace |

sa21412 © The Santa Cruz Operation, Inc.

**Exercise: Assigning subsystems**

---

Using the steps outlined in “Assigning primary authorizations,” assign your student account the *lp* subsystem. When finished, verify that your name has been added to the */etc/auth/subsystems/lp* file.

---

**14.5.2 Secondary subsystem authorizations**

---

Default secondary authorizations are assigned when the security profile is chosen. Listed are the secondary authorizations and the tasks associated with each:

|                      |                                                            |
|----------------------|------------------------------------------------------------|
| <b>audittrail</b>    | Generate personal audit reports on a user’s own activities |
| <b>backup_create</b> | Create, but not restore backups                            |
| <b>restore</b>       | Restore a backup                                           |
| <b>querspace</b>     | Execute the <b>df(C)</b> (disk space free) command         |
| <b>printqueue</b>    | View all jobs in the printqueue using <b>lpstat(ADM)</b>   |
| <b>printerstat</b>   | Enable and disable printers                                |
| <b>su</b>            | Switch user to <b>root</b> or other user accounts          |
| <b>shutdown</b>      | Use the <b>shutdown(ADM)</b> command                       |
| <b>passwd</b>        | Change user passwords using the <b>Account Manager</b>     |

## AV 14-13

| User System Privileges |                                                     |
|------------------------|-----------------------------------------------------|
| Authorizations         | Action                                              |
| configaudit            | configure audit subsystem parameters                |
| writeaudit             | write audit records to the audit trail              |
| execsuid               | ability to run set-UID programs                     |
| chmodsugid             | ability to set the set-UID and set-GID bit on files |
| chown                  | ability to change the owner of an object            |
| suspendaudit           | suspend operating system auditing of the process    |

sa21413 © The Santa Cruz Operation, Inc.

## AV 14-14

| User System Privileges Assigned Based on Installed Security Profile |                                                    |                                  |                                   |                    |
|---------------------------------------------------------------------|----------------------------------------------------|----------------------------------|-----------------------------------|--------------------|
| Kernel                                                              | Low                                                | Traditional                      | Improved                          | High               |
|                                                                     | chmodsugid,<br>chown,<br>execsuid,<br>suspendaudit | chmodsugid,<br>chown<br>execsuid | chmodsugid,<br>chown,<br>execsuid | chown,<br>execsuid |

sa21414 © The Santa Cruz Operation, Inc.

### 14.5.2.1 Assigning secondary authorizations

---

The steps for assigning secondary authorizations are the same as in assigning primary authorizations.

### 14.5.3 Using authorizations

---

System administration tasks are accomplished using the **scoadmin**(ADM) managers or the command-line interface.

### 14.5.4 User system privileges

---

You must assign system privileges to users before they can run commands associated with primary and secondary authorizations. For example, if you assign the **auth** subsystem authorization to a user, you must also give them the **execsuid** user system privilege before they can execute the **passwd**(C) command.

Default user system privileges are determined by the selected security profile. As with authorizations, privileges can also be added or removed.

User system privileges include:

|                     |                                              |
|---------------------|----------------------------------------------|
| <b>configaudit</b>  | Configure audit subsystem parameters         |
| <b>writeaudit</b>   | Write audit records to the audit trail       |
| <b>execsuid</b>     | Ability to run set-UID programs              |
| <b>chmodsugid</b>   | Ability to set UID and set GID bits on files |
| <b>chown</b>        | Ability to change the owner of a file        |
| <b>suspendaudit</b> | Suspend operating system auditing            |

The user system privileges *configaudit*, *suspendaudit*, and *writeaudit* should only be given when the *audit* subsystem has been assigned to a user.

AV 14-15

| Subsystems Privilege Requirements |                                   |
|-----------------------------------|-----------------------------------|
| Subsystem Authorization           | Privilege Required                |
| audit                             | configaudit, execsuid, writeaudit |
| auth                              | execsuid, chown                   |
| backup                            | execsuid                          |
| lp                                | chown                             |
| cron                              | chmodsugid, chown, execsuid       |
| sysadmin                          | chmodsugid, chown, execsuid       |

sa21415

© The Santa Cruz Operation, Inc.

## 14.5.5 Subsystem privilege requirements

---

When assigning a subsystem, certain user system privileges are required. Most subsystem privileges are assigned by default under the Low and Traditional security profile. The following privileges are required when assigning a subsystem to a user:

|                 |                                       |
|-----------------|---------------------------------------|
| <b>audit</b>    | configaudit, execsuid, and writeaudit |
| <b>auth</b>     | execsuid and chown                    |
| <b>backup</b>   | execsuid                              |
| <b>lp</b>       | chown                                 |
| <b>cron</b>     | chmodsugid, chown, and execsuid       |
| <b>sysadmin</b> | chmodsugid, chown, and execsuid       |

### 14.5.5.1 Changing user system privileges

---

The following steps describe how to add or remove a system privilege for a user:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(ADM)**.
- Step 3. Select **Account Manager**.
- Step 4. Select a user name.
- Step 5. Select **Users** ⇨ **Privileges**.
- Step 6. Deselect the default value for “Use system default privileges for this user account.”
- Step 7. Select the privilege:
  - If you are adding a privilege, select the privilege in the “Not Allowed” list that you want to assign to the user.
  - If you are removing a privilege, select the privilege in the “Allowed” list that you want to take away from the user.
- Step 8. Select **Add** or **Remove**.

- Step 9. Repeat Steps 7 and 8 for any other privileges that you want to add or remove. When complete, go to the next step.
- Step 10. Click on **OK**.
- The message `Changing user privileges` appears.
- Step 11. Select **Host** ⇒ **Exit** to exit the **Account Manager**.



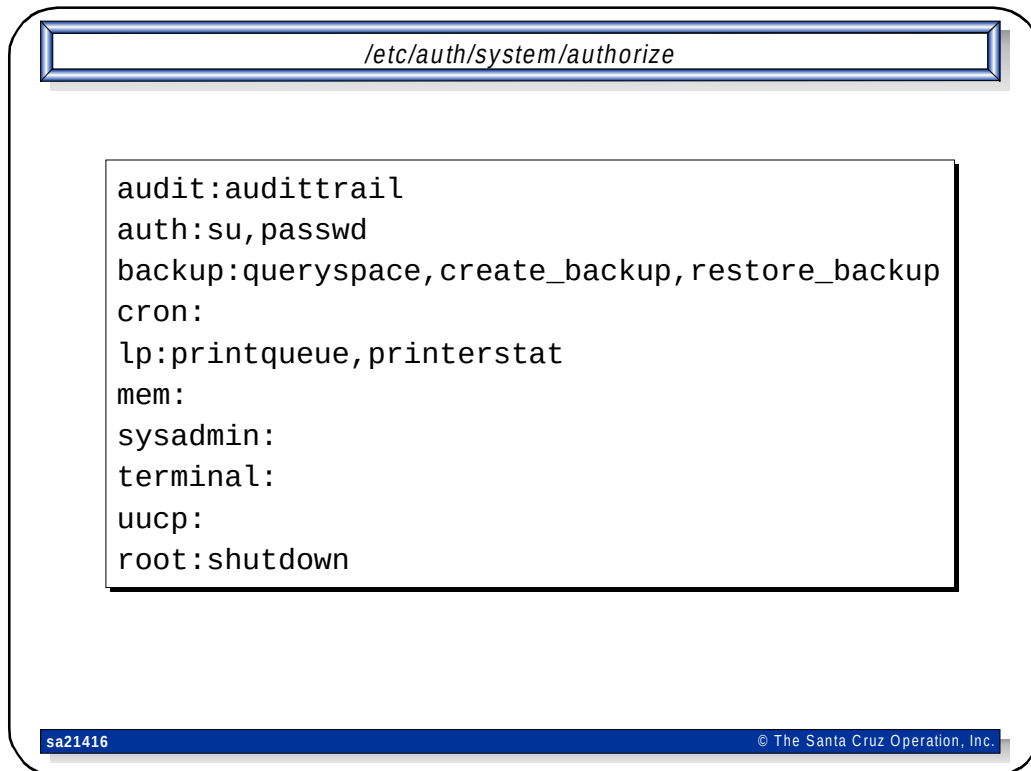
## 14.6 Allowing Users to Execute Root Commands

---

You can assign users the ability to execute a command that is only executable by the **root** user. The user does not have to log in or **su(C)** to the **root** user, ensuring a more secure system.

The user must have the *execsuid* system privilege to execute the **root** command using the **asroot(ADM)** command.

AV 14-16



### 14.6.1 Configuring asroot(ADM)

---

Before a user can execute commands using the **asroot(ADM)** command, you must first add the root command file to the account database, then assign the root command to the user. The following steps describe how to do this:

- Step 1. Login in as **root**. Copy the root command file (the command to execute) to */tcb/files/rootcmds*.
- Step 2. Run **fixmog -i** to ensure the correct file attributes.
- Step 3. Edit the */etc/auth/system/authorize* file and add the *root* command to the line **root:shutdown**.  
  
Verify that your entry is entered correctly. If your syntax contains errors, this will not work.
- Step 4. Run **scoadmin(ADM)**. Select the **Account Manager**.
- Step 5. Select the user name.
- Step 6. Select **Users** ⇒ **Authorization**. Deselect the default value for “Use system default authorizations for this user account.”
- Step 7. Select the root command entered in Step 2.
- Step 8. Click on **Add**.
- Step 9. Click on **OK**.
- Step 10. The message Changing user authorizations appears.
- Step 11. Select **Host** ⇒ **Exit** to exit the **Account Manager**.

## 14.6.2 Executing the root command

---

To execute the root command, the user types:

**/tcb/bin/asroot root\_command**

If your system has been installed with High security, the user is prompted to enter his or her login password before executing the command.

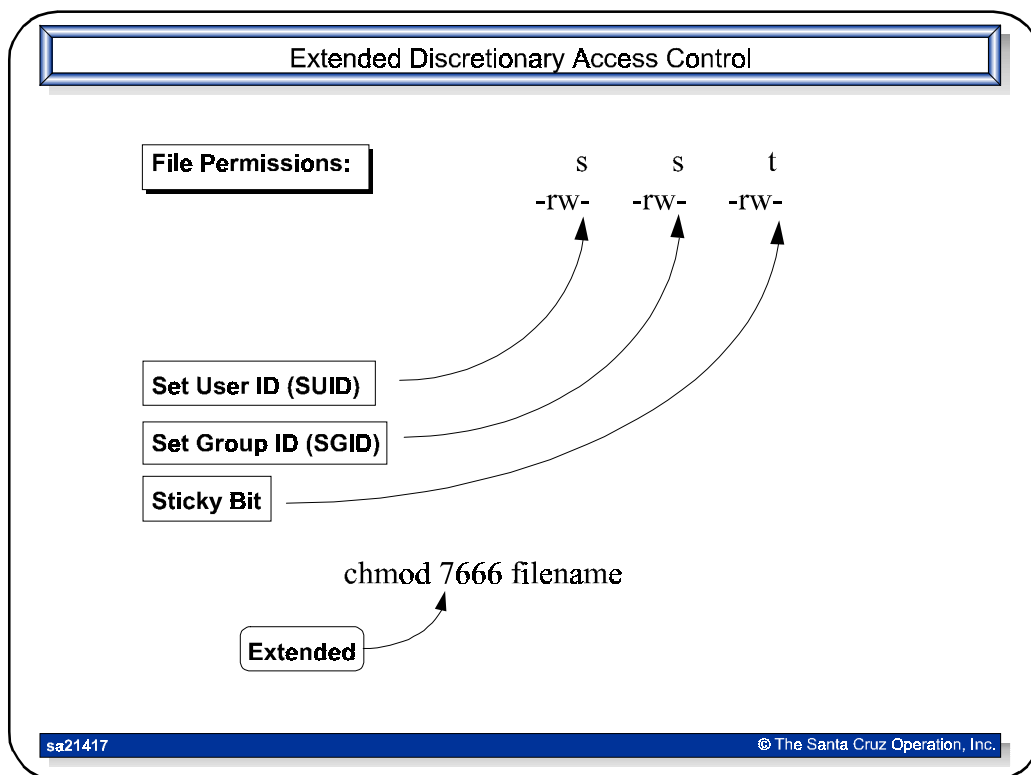
## 14.7 Advanced Protection Bits

---

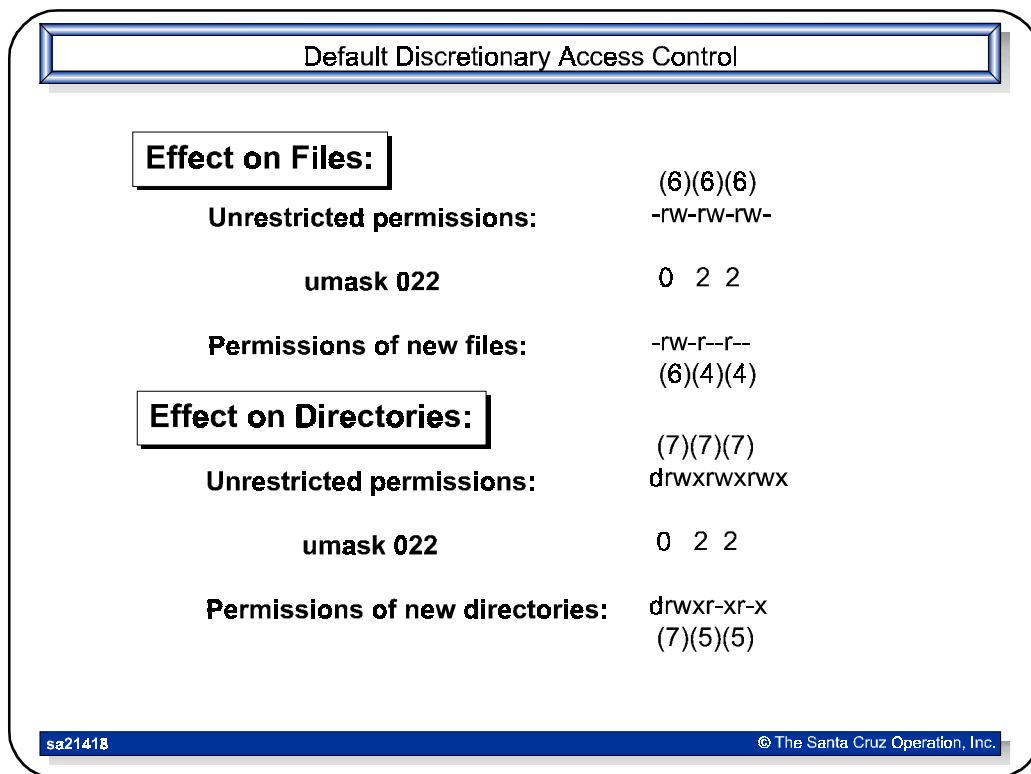
The primary data protection on your system is the use of standard UNIX system permissions on files and directories. Your SCO OpenServer system also includes features that extend this protection. These features greatly enhance the security of the system:

- 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.
- A user must be assigned the **chmodsugid** user system privilege and own the file to set advance protection bits.
- A user must be assigned the **execsuid** user system privilege to execute a command which has the protection bit set.

AV 14-17



AV 14-18



### 14.7.1 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 **auth** can modify the password file. Because the SGID bit is set on the **passwd(C)** command, when the user executes the **passwd** command, the user assumes the group identity **auth**. This enables the user to change his or her password.

```
---x--s--x 1 bin  auth ... /opt/K/SCO/Unix/5.0.0Bh/bin/passwd  
-rw-rw---- 1 root auth ... /tcb/files/auth/d/donnas
```

The SUID bit appears as:

```
---s--x--x 1 root  bin ... /opt/K/SCO/Unix/5.0.0Bh/bin/newgrp
```

Use the **chmod(C)** command to set the SUID bits, enter:

```
chmod 4111 binary_file
```

To set the SGID bit, enter:

```
chmod 2111 binary_file
```

To remove the SUID and SGID bits, enter:

```
chmod 111 binary_file
```

### 14.7.2 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 2 sys sys 1024 Mar 27 10:14 /tmp
```

To add a sticky bit to a directory, enter:

```
chmod 1777 dirname
```

To remove a sticky bit from a directory, enter:

```
chmod 777 dirname
```

Users can only set and remove sticky bits from directories they own. The **root** user can set and remove sticky bits from any directory.

## 14.8 Managing the Trusted Computing Base

---

Several database files store the characteristics of the system itself and its users, administrators, and subsystems so that a site can control its own security parameters. The complete database resides in two hierarchies:

|                                |                                  |
|--------------------------------|----------------------------------|
| <code>/tcb/files/auth/*</code> | Contains user specific files     |
| <code>/etc/auth/*</code>       | Contains system-wide information |

### 14.8.1 Authentication database

---

This database contains authentication and identity information for users, terminals, and Trusted Computing Base files as well as system-wide parameters. It is intended to be used by programs to interrogate users and system values, as well as by authentication programs to update that information. The files are not meant to be edited from the command line.

- Protected Password database—`/tcb/files/auth/[a-z]/username`—user authentication profiles (including encrypted password)
- Terminal Control database—`/etc/auth/system/ttys`—terminal access control (who accessed which terminal)
- Subsystem database—`/etc/auth/subsystems`—authorizations (one per subsystem)
- Device Assignment database—`/etc/auth/system/devassign`—stores device names relating to the same physical device

### 14.8.2 Other database files

---

The File Control Database—`/etc/auth/system/files`—helps maintain the integrity of the Trusted Computing Base. It is used by the TCB database administration tools **integrity**(ADM) and **fixmog**(ADM).

### 14.8.3 Maintaining the integrity of the Trusted Computing Base

---

Several programs maintain the integrity of the Trusted Computing Base. The `/etc/auth/system/files` file maintains a record of ownership, group ownership, permissions, and other important attributes of files that are important to the TCB's operation.



**Exercise: Viewing the File Control database**

---

View `/etc/auth/system/files`.

If you suspect that actual file attributes have been changed or files are missing, run the **integrity** command to compare the actual files to the File Control database, `/etc/auth/system/files`.

The **integrity**(ADM) command reports files that are missing. Most often this is not an error. The file has not been created. For example, if the **integrity** command reports:

```
/usr/lib/cron/cron.allow (entry 261) is missing
```

then the file that allows users to submit **crontabs** to **cron**(C) has not been created.

Run the **fixmog**(ADM) command to make system-specific files consistent with the File Control database or manually make the repair.

When running **fixmog**, use the **-i** option to perform an interactive repair.

**fixmog -i**

**Exercise: Repairing inconsistencies in the File Control database**

---

Run the command `/tcb/bin/integrity` to report any inconsistencies in the File Control database. Redirect the output of the command to the `/tmp/int.report` file.

Review the `/tmp/int.report` file and interactively repair the inconsistencies using **fixmog -i**.

## 14.8.4 Repairing the Trusted Computing Base database

---

When adding users, assigning subsystems, and executing root commands, several files are updated. Should you experience a system problem during one of these tasks, all files might not be updated, which can result in the user not being able to log in.

The **authck**(ADM) command checks the database for any inconsistencies and the **tcbck**(ADM) command checks and prompts for the restoration of any missing files.

### 14.8.4.1 Using */tcb/bin/authck*

---

The **authck**(ADM) command checks for and can interactively repair inconsistencies found in these files:

- The Protected Password database, */tcb/files/auth/[a-z]/\**, and */etc/passwd* for completeness so that neither contains entries not found in the other
- The Terminal Control database, */etc/auth/system/ttys*, for valid terminal entries
- The Protected Subsystem databases, */etc/auth/subsystems/\** and */etc/auth/system/default*, to ensure they correctly reflect the subsystem authorization entries found in the Protected Password database

When running the **authck**(ADM) command:

- Use the **-a** option to check all Password, Terminal Control, and Protected Subsystem databases
- Use the **-v** to display running diagnostics and warnings

***/tcb/bin/authck -av***

### 14.8.4.2 Using tcbck(ADM)

---

The **tcbck**(ADM) command checks the files in the Trusted Computing Base for files that were caught in the process of being updated when the system went down and for files that have been removed. If files are missing, **tcbck** prompts for the restoration of those files.

The **tcbck** command is invoked:

- By */etc/smmck* during system maintenance mode
- By */etc/authckrc* when the system enters multiuser mode
- Using **scoadmin**(ADM)

The following directories and files are checked:

- */etc/auth/system*
- */etc/auth/subsystems*
- */tcb/files/auth/\**
- */etc/auth/system/default*
- */etc/auth/system/files*
- */etc/auth/system/devassign*
- */etc/auth/system/authorize*
- */tcb/files/auth/r/root*
- */etc/group*
- */etc/passwd*

Refer to the **SCO OpenServer Handbook**, “Checking the security databases,” and the **tcbck**(ADM) manual page for additional information about running **tcbck**.

These steps describe how to check the security databases:

- Step 1. Log in as **root**.
- Step 2. Run **scoadmin(ADM)**.
- Step 3. Select **System** ⇒ **Security** ⇒ **Reports Manager** ⇒ **Reports**.
- Step 4. Select **Checks and Verifications**.
- Step 5. Select **Check consistency of account databases**.
- Step 6. Click on **OK**. The report data appears in the Reports Manager window.
- Step 7. Select **File** ⇒ **Exit** to exit the **Reports Manager**.

### **Exercise: Check for database inconsistencies**

---

---

Using the steps in the preceding procedure for “Using **tcbk(ADM)**,” check for and repair any security database inconsistencies on your classroom system.

## 14.9 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 SCO OpenServer system also provides additional tools to monitor system security.

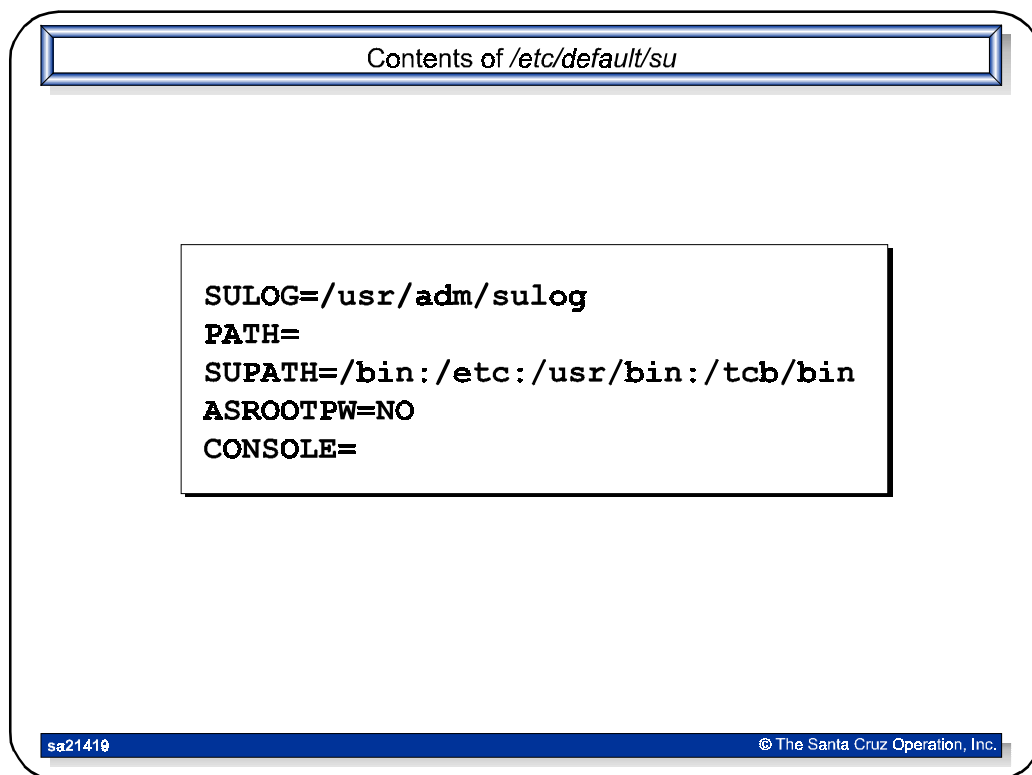
### 14.9.1 Restricting root logins

---

You can restrict the **root** login to the console, eliminating entry from any other port. These steps describe how to do this:

- Step 1.        Log in as **root**.
- Step 2.        Edit the `/etc/default/login` file.
- Step 3.        Add the entry:
- `CONSOLE=/dev/tty01`

AV 14-19



## 14.9.2 Monitoring the use of the su(C) command

---

If you give a user **su** secondary authorization, that user is 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(C)** attempts to the `/usr/adm/sulog` file:

- Step 1.        Log in as **root**.
- Step 2.        Edit the file `/etc/default/su`.
- Step 3.        Verify that the entry `SULOG=/usr/adm/sulog` exists. If not, add the entry.
- Step 4.        Uncomment the entry `CONSOLE=` and edit so that it reads:
- `CONSOLE=/dev/tty01`

### **Exercise: Reporting su attempts to the console**

---

Edit the `/etc/default/su` file and add an entry to report **su(C)** attempts to the console.

Log into your student account and **su** to the **root** account.

View `/usr/adm/sulog`.

### 14.9.3 Additional dial-in security

---

If a modem is connected to your system, you can set up an additional password. Users are prompted for a password when the connection is made and again for their user password when logging in.

The name of the serial port to which the modem is attached and the login shell for the users are required. If users are logging into different shells, an entry is required for each type of shell.

- Step 1.        Log in as **root**.
- Step 2.        Edit the `/etc/dialups` file and enter the absolute pathname to the serial device. For example, if a modem is connected to COM1, enter `/dev/tty1A`.
- Step 3.        Edit the `/etc/d_passwd` file. For the Korn shell, add the entry :  
  
                `/bin/ksh::`
- Step 4.        Assign a password, enter:  
  
                **`passwd -m ksh`**



## 14.10 Lesson Summary

---

The SCO OpenServer 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 primary and secondary authorizations. Additional security can be provided by restricting **root** logins and adding dialup passwords.

## Review Questions

---

1. What security features are provided by *Identification and Authentication*?
2. Which security profile(s) allows the removal of user accounts?
3. Using the command-line interface, how would you change your security profile from Improved to Traditional?
4. In which two files is the encrypted password stored when using traditional security on an SCO OpenServer System?
5. Which secondary authorization allows a user to back up, but not restore, a filesystem?
6. Which command executes a root command for an ordinary user?
7. Which user system privilege is required to set a UID on a binary file?

8. To which database file does the **integrity**(ADM) command compare actual files?
  
9. In which file are both successful and unsuccessful **su**(C) attempts to the **root** account recorded?

## 14.11 Lab: Configuring and Using asroot(ADM)

---

You will authorize your student account to change the system time.

### 14.11.1 Overview

---

Authorize your student account to change the system time using the **asktime**(ADM) command. Login as your student account and execute the **asktime** command.

### 14.11.2 Procedure

---

The following steps describe how to authorize your student account to change the system time:

- Step 1. Log in as **root**.
- Step 2. Copy the file `/etc/asktime` to `/tcb/files/rootcmds`.
- Step 3. Run **fixmog -i** to compare the file attributes to the File Control Database.
- Step 4. Edit the file `/etc/auth/system/authorize` and add **asktime** to the root entry.
- Step 5. Use **scoadmin**(ADM) and assign the **asktime** authorization to your student account.
- Step 6. Log in to your student account.
- Step 7. Enter:  
  
**/tcb/bin/asroot asktime**
- Step 8. Change the system time.

## 14.12 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                        |                                                                                                                                                     |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>asroot</b> (ADM)    | Runs a command as <b>root</b>                                                                                                                       |
| <b>authck</b> (ADM)    | Checks internal consistency of the authentication database                                                                                          |
| <b>fixmog</b> (ADM)    | Makes all or specific system files consistent with the File Control Database                                                                        |
| <b>integrity</b> (ADM) | Examines system files against the authentication database                                                                                           |
| <b>scoadmin</b> (ADM)  | Invokes SCOadmin applications (in this lesson, the <b>Accounts Manager</b> , the <b>Reports Manager</b> , and the <b>Security Profile Manager</b> ) |
| <b>tcbck</b> (ADM)     | Checks the Trusted Computing Base for missing files                                                                                                 |
| <b>relax</b> (ADM)     | Changes the system default security profile                                                                                                         |

### Relevant files

---

|                                   |                              |
|-----------------------------------|------------------------------|
| <i>/etc/auth/system/authorize</i> | Subsystem authorization file |
| <i>/etc/auth/system/default</i>   | System default database file |
| <i>/etc/auth/system/files</i>     | File Control Database        |
| <i>/tcb/files/auth/[a-z]/*</i>    | Protected Password Database  |

## 14.13 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Viewing primary authorizations

---

To view the primary authorizations assigned by default, enter:

```
more /etc/auth/system/default
```

The `u_cmdpriv` field indicates primary authorizations.

### Exercise: Changing security parameters

---

To change security parameters, enter:

```
usermod -D -x "[maxLoginAttempts 20]"
```

### Exercise: Repairing inconsistencies in the File Control Database

---

To repair inconsistencies in the File Control database, enter:

```
/tcb/bin/integrity -e -m > /tmp/int.report
```

```
fixmog -i
```

### Review questions

---

This section provides suggested answers to the review questions.

1. Identification and Authentication provides greater control over login attempts and the login process.
2. The Traditional and Low security profiles allow the removal of user accounts.
3. To change your security profile from Improved to Traditional from the command line, enter: **relax traditional**.
4. The `/tcb/files/auth/[a-z]/*` and `/etc/shadow` files store encrypted passwords.

5. The **backup\_create** authorization allows a user to back up, but not restore, a filesystem.
6. The **asroot**(ADM) command executes a root command for an ordinary user.
7. The **chmodsuid** user system privilege is required to set a UID on a binary file.
8. The **integrity**(ADM) command compares actual files to the */etc/auth/system/files* (the File Control database).
9. The */etc/default/su* file contains both successful and unsuccessful **su**(C) attempts.

---

# ***15***

## ***Performance Tuning***

AV 15-1

Performance Tuning

- Describe factors that influence performance
- Define performance goals
- Collect performance goals
- Identify dynamically configured kernel tables
- Interpret kernel error messages
- Configure kernel parameters

sa21501 © The Santa Cruz Operation, Inc.



**Lesson 15**

## **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 the users, providing optimal performance. System hardware, a network environment, and interactive and graphical programs can all have an impact on kernel parameters. To provide optimal performance, you need to know what the kernel parameters are and how they can affect system performance. This understanding enables you to diagnose, troubleshoot, and make adjustments effectively.

### **Objectives**

---

This lesson covers the hardware and software factors that influence system performance. By the end of this lesson, you will be able to:

- Describe factors that influence performance
- Define performance goals
- Collect performance data
- Identify dynamically configured kernel tables
- Interpret kernel error messages
- Configure kernel parameters

## 15.1 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 programs are accessed also affect system performance.

### 15.1.1 Hardware factors

---

Your system's hardware has the greatest influence on its performance. It is the ultimate limiting factor on how fast a process can 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

On multiprocessor systems, these 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.

### **15.1.2 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 can 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?
- Are 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.

### **15.1.3 System organization**

---

Proper organization of your hard disk can improve its performance. Use the following guidelines for organizing your hard disk:

- Organize filesystems according to use (for example, by department, frequency or data use, and so forth)
- Create small filesystems when possible (small filesystems tend to be more efficient than large filesystems)
- Keep the root filesystem as small as possible (leaving room for expansion in the future)
- 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

## 15.2 Defining 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 set the goals needed to improve performance. To accomplish this:

- Define your goals
- List your goals
- Understand your goals

### 15.2.1 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, which can improve overall performance, to tuning the system to run a single application.

### 15.2.2 Listing your goals

---

The performance goals are listed in order of priority. Often goals can conflict. For example, a system can be running a database that uses a large cache and also requires a large portion of memory to compile programs during software development.

Assigning a priority to your goals can help you decide if database performance is more important than the speed of compilations.

### 15.2.3 Understanding your goals

---

Attempt to understand all goals. If possible, note what resources are affected by each goal. Do you have any conflicts?

## 15.3 Collecting Performance Data

---

After identifying the 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

You should collect data over a duration that is long enough for you to establish normal patterns of usage.

Analysis should focus on three potential performance problems:

- CPU saturation
- I/O system saturation
- Memory saturation

Poor system performance can result from any combination of these three problems.

Refer to the ***SCO OpenServer Performance Guide***, “Appendix A,” “Tools reference,” to establish your system's current performance.



## 15.3.1 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:

- Turn on process accounting to track CPU usage over a period of time
- Use the **ps(C)** and **sar(ADM)** commands to help identify CPU saturation

### 15.3.1.1 Using the **ps(C)** command

---

You can run the **ps(C)** command at regular intervals and check the **TIME** field. Processes showing an increased CPU time can be an indication of CPU saturation.

## AV 15-2

**CPU Activity `sar -u`(ADM)**

```
SCO_SV  sosco 3.2v5.0.0 i80486  03/29/95

10:36:49      %usr  %sys  %wio  %idle
10:36:53      21    18    13    48
10:36:57       1     1     0    97
10:37:01       1     4     1    94
10:37:05       1     2     0    97

Average       6     7     4    84
```

sa21502© The Santa Cruz Operation, Inc.

## AV 15-3

**Examining the Run Queue `sar -q`(ADM)**

```
SCO_SV  sosco 3.2v5.0.0 i80486    03/28/95

06:56:55  runq-sz  %runocc  swpq-sz  %swpocc
06:57:00   1.7    98      1.5     36
06:57:05   1.0    63      1.0     31
06:57:10   1.0    58      1.0     49

Average   1.3     74      1.2     39
```

sa21503© The Santa Cruz Operation, Inc.



### 15.3.1.2 Using the sar(ADM) command

---

The **sar**(ADM) command reports activity on single processor systems. To check for CPU saturation with **sar**, use the **-u** and **-q** options:

```
sar -u 60 10
```

If the **%idle** field is less than or equal to 10%, it indicates 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.

## 15.3.2 Tuning CPU resources

---

### 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**(ADM) 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  to terminate **sartest -cpu**.

If your system is CPU-bound, you can do several things to improve CPU resources. It usually involves hardware; however, you can also tune a limited amount of kernel parameters.

### 15.3.2.1 Hardware

---

Changing your hardware to improve CPU resources includes:

- 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

### 15.3.2.2 Kernel parameters

---

To improve CPU resources, you can change the following kernel parameters:

|          |                                                                                                                                                                                                                                  |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAXSLICE | Sets the maximum time slice for a process. The default is 100 clock ticks per second. Values range between 25 and 100. The effect of reducing MAXSLICE is to allow each process to run more often but for a shorter period time. |
| MAXUP    | Controls the number of processes that a user can run at any given time. The default is 100. Reduce this parameter to limit the number of processes a user can run.                                                               |

### 15.3.3 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.

### 15.3.4 The buffer cache

---

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.

The buffer cache is created in an area of kernel memory and is never swapped out. Although it can be regarded as a memory resource, it is primarily an I/O resource due to its use in mediating data transfer.

The buffer cache is automatically configured at system startup based on the amount of memory available. This parameter is tunable should you need to increase or decrease the size of the buffer cache.

### **Exercise: Determining the size of the buffer cache**

---

Use **grep(C)** to search for “bufs” in the `/usr/adm/messages` file to determine the size of your system’s buffer cache.

#### **15.3.4.1 Viewing buffer cache activity**

---

To determine the activity in the buffer cache, use the **-b** option with the **sar(ADM)** 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.

Increasing the size of the buffer cache may help alleviate a disk I/O bottleneck.

#### **15.3.4.2 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**(ADM) command:

```
sar -d 60 10
```

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.

Watch for unbalanced loads if you have multiple disks. Individual disks can be overloaded before the entire I/O system approaches capacity. A **%busy** average over 50% can indicate I/O saturation.

High values of the ratio of **avserv** also suggest that the disk is saturated with requests.

### **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 multiscreeen.

On a second multiscreeen, run the **sar**(ADM) 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  to terminate the **sartest -io** shell script.



## 15.3.5 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.

### 15.3.5.1 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

Refer to the ***SCO OpenServer Performance Guide***, “Tuning I/O resources” for more information on tuning disk and other I/O activity.

AV 15-4

| Buffer Managment - Tunable Parameters |                                                                |
|---------------------------------------|----------------------------------------------------------------|
| <b>NBUF</b>                           | <b>Size of the buffer cache</b>                                |
| <b>NHBUF</b>                          | <b>Number of hash queues</b>                                   |
| <b>NMPBUF</b>                         | <b>Number of 4KB pages of memory</b>                           |
| <b>PLOWBUFS</b>                       | <b>Buffer cache located in first 16MB of memory</b>            |
| <b>PUTBUFSZ</b>                       | <b>Size of the circular buffer</b>                             |
| <b>NHINODE</b>                        | <b>Size of the inode hash table</b>                            |
| <b>BDFLUSHR</b>                       | <b>Rate in which the buffer cache is written to disk</b>       |
| <b>NAUTOUP</b>                        | <b>Age of data in the buffer cache</b>                         |
| <b>BFREEMIN</b>                       | <b>The number of buffers that must remain in the free list</b> |

sa21504

© The Santa Cruz Operation, Inc.



### 15.3.5.2 Kernel Parameters

---

To tune I/O performance, you can change these kernel parameters:

**NBUF** Controls the buffer cache. Increasing the size of **NBUF** reduces the writes to the disk. The value of **NBUF** is 0, which means the system automatically configures this value at boot time based on the amount of available memory.

When changing the size of **NBUF**, for optimal performance, you must also change the **NHBUF** parameter. **NHBUF** is used to search for a buffer in the hash queue as opposed to searching the entire list of buffers. For single processor systems, **NHBUF** is set to the power of 2 that is less than or equal to half the value of **NBUF**.

Refer to the *SCO OpenServer Performance Guide*, “Configuring kernel parameters,” “Configuration tools,” to set these parameters for a multiprocessor system.

**BDFLUSHR** Controls how often data in the buffer cache is written to disk. The default is every 30 seconds. Increasing this parameter can reduce disk activity, while increasing the chance of data loss.

**NAUTOUP** Specifies the buffer age in seconds before the system daemon, **bdflusher**, flushes the buffer cache to disk. The value of this parameter must be chosen in conjunction with **BDFLUSHR**. Setting a high value for **BDFLUSHER** and a low value for **NAUTOUP** could result in data loss during a system crash.

Refer to the *SCO OpenServer Performance Guide*, “Configuring kernel parameters,” “Configuration tools,” for more information on how to effectively configure these parameters.

### 15.3.6 Identifying memory saturation

---

SCO OpenServer systems uses 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 UNIX system.

UNIX is designed to provide programs with as much memory as they require, even if the requirements exceed physical memory. When physical memory is exhausted, UNIX memory management begins to use swap areas on the disk for swapping and paging.

*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 uses the swapper daemon, **schd**, to free assigned memory pages by copying modified process pages to the swap area.

### **Exercise: Identifying the amount of physical memory**

---

View the `/usr/adm/messages` file and determine how much memory is required by kernel and user processes. Combine these to identify the total amount of memory installed on your classroom system.

#### **15.3.6.1 The swap(ADM) command**

---

To quickly determine if memory is being exhausted, run the command:

**swap -l**

Blocks are displayed in 512-bytes. If the swap area is being used, look further to see if memory saturation has occurred.

#### **15.3.6.2 The sar(ADM) command**

---

Use the **r** option with the **sar** (ADM) command to display the amount of free memory and free swap space on your system.

**sar -r 60 10**

The **freemem** field is a count of 4KB pages. The **freeswp** field is displayed in 512-byte blocks.

**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**(ADM) command to determine the amount of free memory on your classroom system.

On a multiscreeen, run the **sartest -swap** script.

On a second multiscreeen, run the **sar** command and view the memory usage and swapping activity.

When finished, use the  key to terminate the **sartest -swap** script.

AV 15-5

| Paging - Tunable Parameters |                                                                                              |
|-----------------------------|----------------------------------------------------------------------------------------------|
| <b>GPGSLO</b>               | <b>Low value of free memory before vhand runs</b>                                            |
| <b>GPGSHI</b>               | <b>High value of free memory before vhand stops</b>                                          |
| <b>MAXSLICE</b>             | <b>Maximum time slice for user processes</b>                                                 |
| <b>MINARMEM</b>             | <b>Minimum amount of physical memory needed for text and data segments of user processes</b> |
| <b>MINASMEM</b>             | <b>Minimum size that virtual memory is allowed to reach</b>                                  |
| <b>SPTMAP</b>               | <b>Size of the map entry array</b>                                                           |

sa21505

© The Santa Cruz Operation, Inc.

### 15.3.7 Tuning memory saturation

---

If you have determined that your system's memory is saturated, you can increase the amount of physical memory or change these kernel parameters:

|      |                                                                                                                       |
|------|-----------------------------------------------------------------------------------------------------------------------|
| NBUF | Controls the buffer cache. Decrease the size of NBUF. Remember to also set the size of NHBUF to an appropriate value. |
|------|-----------------------------------------------------------------------------------------------------------------------|

|                   |                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| GPGSLO and GPGSHI | Specify the low value and high value, respectively, of free memory pages at which <b>vhand</b> will stop stealing pages from processes. |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------|

If the system appears to be constantly paging, this may be the result of GPGSLO and GPGSHI being too high. This causes **vhand** to swap out pages while a large number of free pages still exist. Lowering these values could delay the onset of paging.

## 15.4 Identifying Kernel Tables

---

---

The kernel sets up a number of tables in memory. For example, these tables accommodate the number of open processes, files, and inodes.

System tables are allocated in memory dynamically by the kernel. This allows them to grow to accommodate maximum demand. You can specify a maximum size to which a table can grow; however, this can hinder your system if you select too low a value.

Before determining the number of entries you need for certain kernel tables, you need to look at the parameters that control how much memory can be allocated for use with dynamic kernel tables.

### 15.4.1 Setting table limits

---

The following list describes the parameters that control table limits:

|             |                                                                                                                                                                          |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TBLPAGES    | The maximum number of pages for dynamic tables. The default is 0, which means the kernel configures the value based on the amount of memory available at system startup. |
| TBLDMAPAGES | The maximum number of pages that can be used for DMA access. This is configured in the same manner as TBLPAGES.                                                          |
| TBLLIMIT    | The percentage of TBLPAGES and TBLDMAPAGES that a single table can grow. The default is 70% of the allocated pages.                                                      |
| TBLSYSLIMIT | The percentage of memory allowed for dynamic tables if TBLPAGES is set to 0. The default is 25% of memory.                                                               |

AV 15-6

| Number of Bytes per Unit Parameter (Partial Listing) |                                                                               |
|------------------------------------------------------|-------------------------------------------------------------------------------|
| NBUF                                                 | 1096 per buffer on most filesystems<br>4168 per buffer on the DTFS filesystem |
| MAX_INODE                                            | 76 per entry                                                                  |
| MAX_PROC                                             | 344 per entry                                                                 |
| MAX_FILE                                             | 12 per entry                                                                  |
| MAX_REGION                                           | 76 per entry                                                                  |

sa21506 © The Santa Cruz Operation, Inc.



## 15.4.2 Determining table memory size

---

Each table entry, active or empty, occupies space in memory. For example, if you determined that your system has been averaging 300 active processes, each process will occupy 344 bytes. The space occupied in memory would be:

300 processes X 344 bytes = requires approximately 10KB of memory

Refer to the *SCO OpenServer Performance Guide*, “Configuring kernel parameters,” “When to change system parameters,” for a complete listing of sizes in bytes.

## 15.4.3 Dynamic kernel tables

---

These parameters control the maximum grown size of dynamic kernel tables. These tables are initially set to 0 allowing them to dynamically grow provided that sufficient TBLPAGES of memory are allocated.

|            |                                                                                                                                                                             |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAX_DISK   | The number of disk drives that can be attached to the system                                                                                                                |
| MAX_INODE  | The number of allocated inode table entries for all filesystems. This entry depends on the number of open files.                                                            |
| MAX_PROC   | The number of allocated process table entries                                                                                                                               |
| MAX_FILE   | The number of allocated open files for all filesystems                                                                                                                      |
| MAX_REGION | The number of allocated region table entries. Should you configure this parameter, as a general rule, set its value to slightly more than three times the size of MAX_PROC. |
| MAX_MOUNT  | The number of allocated mount table entries. Each entry represents a mounted filesystem.                                                                                    |
| MAX_FLCREC | The number of allocated lock table entries. This determines the number of file regions that can be locked by the system.                                                    |

### 15.4.3.1 Displaying table values

---

You can use the **getconf(C)** command to display the actual configured value and the maximum value to which a parameter can dynamically grow.

To display the configured value of the process table, MAX\_PROC, enter:

### **getconf KERNEL\_PROC**

To view the maximum configurable value, enter:

### **getconf KERNEL\_PROC\_MAX**

Use the **sar**(ADM) command with the **-v** option to display actual entries being used by the process, file, inode, and lock tables. For example, enter:

```
sar -v 60 10
```

When running the **sar -v** command, the first number in the “proc-sz” field indicates the actual number of active processes.

#### **Exercise: Displaying kernel parameter values**

---

Use **getconf**(C) to display the maximum configured value that the process table can reach.

Use **sar**(C) to display the value currently used by the process table.

### **15.4.3.2 Displaying kernel parameters**

---

You can use several commands to display kernel parameters. If a parameter is auto-configured, a value of 0 displays.

To display the name, description, and current value of each parameter, enter:

```
/etc/conf/cf.d/configure -x | pg
```

To display the name and current value of certain parameters, enter:

```
sysdef | pg
```

## 15.5 Interpreting Kernel Error Messages

---

Kernel error messages display on the console and append the `/usr/adm/messages` file.

Each error message reports:

- What happened
- What parameter is affected
- How much the parameter is exceeded by

For example, if the number of processes is exceeded on your system or the system is out of swap space, these error messages display:

`CONFIG: newproc-process table overflow (MAX_PROC = number exceeded)`

`CONFIG: swapdel-total swap area is too small (MINAMEM = number exceeded)`

Refer to *SCO OpenServer Performance Guide*, “Configuring kernel parameters,” “Configuration tools,” for a complete description of error messages along with the appropriate kernel parameter.



## 15.6 Configuring Kernel Parameters

---

---

You might need to configure 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

AV 15-7

**Commands to Configure Kernel Parameters**

**configure (ADM)**  
**Hardware/Kernel Manager (scoadmin (ADM))**  
**idtune (ADM)**  
**iddeftune (ADM)**  
**setconf(ADM)**

sa21507 © The Santa Cruz Operation, Inc.

## 15.6.1 Configuration tools

---

You can use these configuration tools for examining or changing tunable parameters:

|                         |                                                                                                                                                                                          |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>configure</b> (ADM)  | A menu-driven program that allows you to examine and modify the value of tunable kernel parameters. You can also access this program using the SCOadmin <b>Hardware/Kernel Manager</b> . |
| <b>id tune</b> (ADM)    | Changes parameters that cannot be changed using the <b>configure</b> (ADM) command                                                                                                       |
| <b>iddef tune</b> (ADM) | Increases the value of certain parameters when you increase the amount of physical memory beyond 32MB                                                                                    |
| <b>setconf</b> (ADM)    | Increases dynamic kernel table sizes. The new value only remains in force until the system is rebooted. Use <b>configure</b> (ADM) to make the change permanent.                         |

### **Exercise: Changing the value of a kernel table**

---

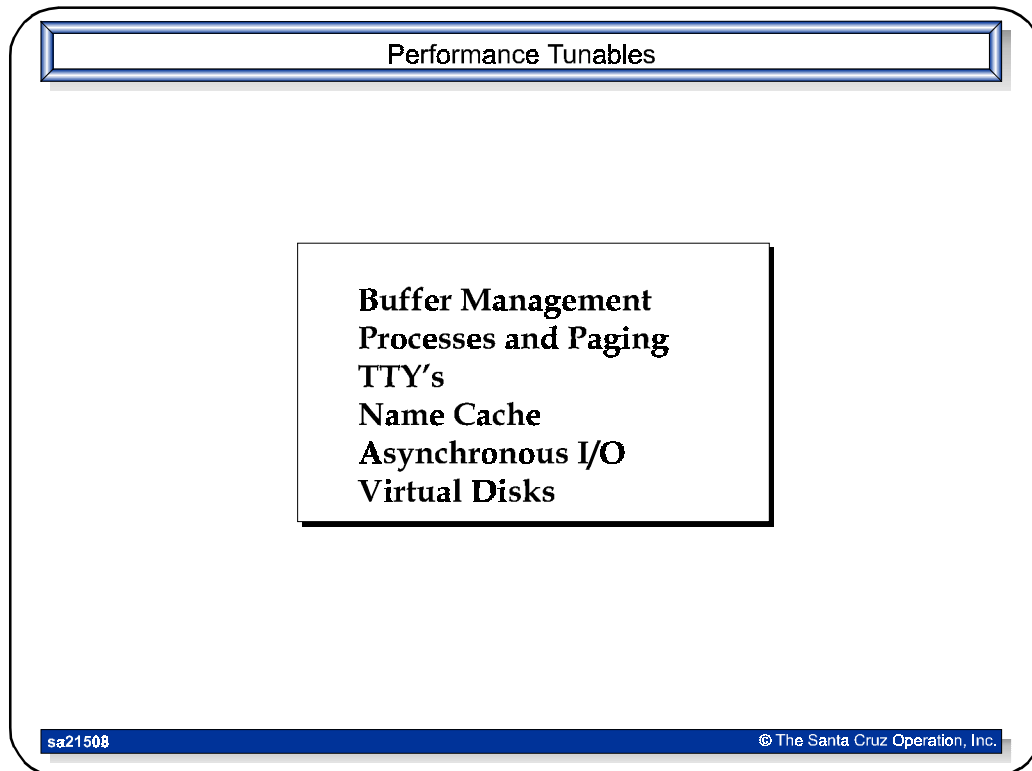
---

Use **getconf**(ADM) to view the size of the process table.

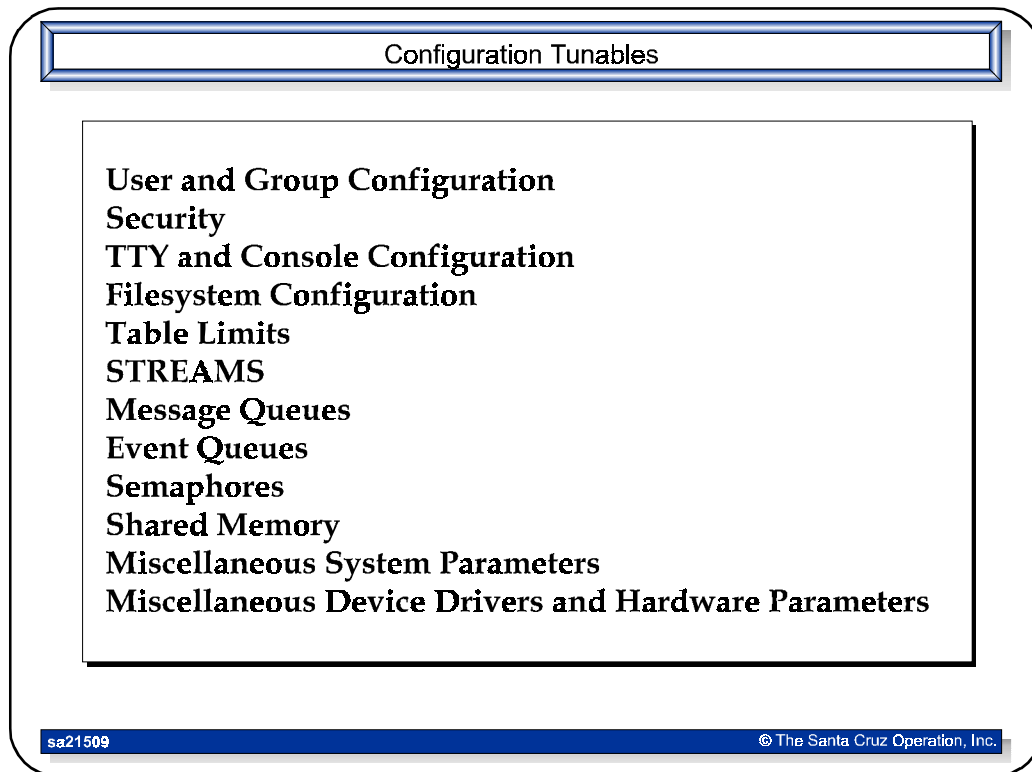
Use **setconf**(ADM) to increase the size of the process table by 50 entries.

Use **getconf**(ADM) to verify your changes.

AV 15-8



AV 15-9





## 15.6.2 Using **configure**(ADM)

---

The tunable parameters that you can change using **configure**(ADM) are grouped into two sets of categories. The first category deals with performance tunables such as buffer management, processes and paging. The second relates to configuration tunables such as table limits and filesystems.

Before using the **configure** command, 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 reconfigure.

These steps describe how to use **configure**(ADM):

- Step 1. Shut down your system to single-user mode.
- Step 2. Make sure your boot filesystem is mounted as a read/write filesystem. If it is mounted as read-only, unmount it:

```
umount /dev/boot
```

Then, mount it as a read/write filesystem:

```
mount /dev/boot /stand
```

- Step 3. Copy */stand/unix* to */stand/unix.before*.
- Step 4. Change to the */etc/conf/cf.d* directory.
- Step 5. Enter **./configure** to display the kernel configuration menu.
- Step 6. Select a category.
- Step 7. Press  until the desired parameter displays.
- Step 8. Enter the new value.
- Step 9. Press  until the main menu displays.
- Step 10. Enter **q** to quit. The following message appears:

```
Would you like to update the system configuration
files with your changes? (y/n)
```

- Step 11. Enter **y**. When complete, the following message appears:
- `Configuration files updated.` You are also given instructions for relinking the kernel.
- Step 12. Relink the kernel, enter:
- `./link_unix`
- After the kernel is rebuilt, the following messages appear:
- `The UNIX kernel has now been rebuilt.`
- `Do you want this kernel to boot by default?`
- Step 13. Enter **y**.
- The system backs up the old kernel by moving the current `/stand/unix` to `/stand/unix.old`. When complete, the following message appears:
- `Do you want the kernel environment rebuilt?`
- Step 14. Enter **y**. The system rebuilds `/etc/inittab`.
- Step 15. Reboot the system to load the new kernel.

## 15.7 Lesson Summary

---

---

You now have the tools to monitor your systems for CPU, I/O, and memory saturation. If a performance issue warrants, use the **configure**(ADM) command or the **Hardware/Kernel Manager** to reconfigure the UNIX kernel.

### Review questions

---

1. Describe three hardware factors that influence system performance.
2. What are the three steps when setting performance goals?
3. To which file is the size of the buffer cache written on system startup?
4. What command would you run to display the current sizes of your process, file, inode, and lock tables?
5. Name three components that comprise a kernel error message.
6. In which directory are programs and files relating to kernel configuration stored?

## 15.8 Lab: Reconfiguring the kernel

---

---

You have determined that the number of processes a user can execute simultaneously on your system should be limited to 25. The default value is 100.

### 15.8.1 Overview

---

Make a copy of the existing kernel in the */stand* directory before changing a kernel parameter. Run the **configure**(ADM) command and decrease the **MAXUP** parameter by 25. Rebuild and reboot the new kernel.

## 15.8.2 Procedure

---

- Step 1. Shut down your system to single-user mode.
- Step 2. Unmount the boot filesystem, enter:
- ```
umount /dev/boot
```
- Step 3. Mount the boot filesystem as */stand*, enter:
- ```
mount /dev/boot /stand
```
- Step 4. Copy */stand/unix* to */stand/unix.before*, enter:
- ```
cp /stand/unix /stand/unix.before
```
- Step 5. Change to the */etc/conf/cf.d* directory, enter:
- ```
cd /etc/conf/cf.d
```
- Step 6. Display the kernel configuration menu, enter:
- ```
./configure
```
- Step 7. Select “User and group configuration.”
- Step 8. Press  until the MAXUP parameter displays.
- Step 9. Enter **25**.
- Step 10. Press  until the main menu displays.
- Step 11. Enter **q** to quit.
- Step 12. Enter **y** when prompted to update the system configuration files.
- Step 13. Rebuild a new kernel, enter:
- ```
./link_unix
```
- Step 14. Enter **y** when prompted with the following message:
- ```
Do you want this kernel to boot by default?
```

- Step 15. Enter **y** when prompted with the following message:
- Do you want the kernel environment rebuilt?
- Step 16. Shut down your system, enter:
- /etc/shutdown -i6 -g0 -y**
- Step 17. Boot the new kernel.

## 15.9 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                        |                                                                                                  |
|------------------------|--------------------------------------------------------------------------------------------------|
| <b>configure</b> (ADM) | Changes kernel configuration parameters                                                          |
| <b>getconf</b> (C)     | Displays actual configured value and the maximum value to which a parameter can dynamically grow |
| <b>idtune</b> (ADM)    | Changes parameters that cannot be changed using the <b>configure</b> (ADM) command               |
| <b>ps</b> (C)          | Reports process status                                                                           |
| <b>sar</b> (ADM)       | Reports system activity                                                                          |
| <b>setconf</b> (ADM)   | Increases or decreases (temporarily) the size of dynamic kernel tables                           |
| <b>scoadmin</b> (ADM)  | Invokes SCOadmin applications (in this lesson, the <b>Hardware/Kernel Manager</b> )              |
| <b>swap</b> (ADM)      | Adds, deletes, and monitors the system swap areas                                                |
| <b>sysdef</b> (ADM)    | Displays the name and current value of certain parameters                                        |

### Relevant files

---

|                       |                                                                                     |
|-----------------------|-------------------------------------------------------------------------------------|
| <i>/unix</i>          | The UNIX kernel                                                                     |
| <i>/unix.old</i>      | The UNIX kernel created before a new kernel is rebuilt                              |
| <i>/etc/conf/cf.d</i> | The directory which contains the program and the files to rebuild a new UNIX kernel |

## 15.10 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Determining the size of the buffer cache

---

To determine the size of the buffer cache, enter:

```
grep 'bufs' /usr/adm/messages
```

### Exercise: Identifying the amount of physical memory

---

Use the **grep(C)** command to search for “mem”—the `mem: total` field indicates the total amount of physical memory installed on your system.

### Exercise: Determining free memory

---

The `freemem` field of **sar -r** output indicates the amount of available physical memory on your system.

### Exercise: Displaying kernel parameters

---

To display kernel parameters, enter:

```
getconf KERNEL_PROC_MAX
```

```
sar -v
```

### Review questions

---

This section provides suggested answers to the 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 buffer cache writes to the */usr/adm/messages* file during startup.
4. You would run **sar -v** to display the current sizes of process, file, inode, and lock tables.
5. The three components that comprise a kernel error message are what happened, what parameter was exceeded, and how much the parameter was exceed by.
6. The */etc/conf/cf.d* directory contains programs and files related to kernel configuration.



---

# ***16***

## ***Disaster Recovery***

AV 16-1

Disaster Recovery

- Prepare for recovery
- Recover system files
- Restore a non-root filesystem
- Restore the root and boot filesystems
- Replace the root hard disk

sa21601 © The Santa Cruz Operation, Inc.

**Lesson 16**

## **Disaster Recovery**

### **Introduction**

---

Now that you are familiar with the files and directories that make up your SCO OpenServer 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 root disk. 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
- Recover system files
- Restore a non-root filesystem
- Restore the root and boot filesystems
- Replace the root hard disk

## 16.1 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 complete system recovery.

The following components should be available:

- Original media

On rare occasions if you cannot recover a system file from your backup media, your only alternative is to recover it from your original distribution media. Original media is also required if you are replacing a primary hard disk or if an emergency boot floppy set is not available.

- Registration information

If you need to call SCO for technical support, have your serial numbers ready. SCO needs this information before they can assist you.

- Emergency boot floppy set

Required when one or several boot files are missing from your hard disk. They are also needed to restore a boot and root filesystem.

- 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 full system recovery.

- Backups

Your most recent system and data backups.

## 16.2 Recovering a System File

---

The recovery of a system file from a backup is very similar to recovering any other type of file. Recovering from your distribution media involves knowing where the file is stored and what data transfer method is involved in extracting the file.

### 16.2.1 Product packaging

---

Your distribution media is comprised of a Boot Disk and CD-ROM. Each SCO product is arranged in the following units:

|           |                                                                                           |
|-----------|-------------------------------------------------------------------------------------------|
| Product   | SCO OpenServer system, Enterprise system, and so forth                                    |
| Component | System Administration, Extended Utilities, Mail Utilities, Backup Utilities, and so forth |
| Package   | The actual files that make up a Component                                                 |

To list the contents of your distribution media, use **scoadmin(ADM)** to access the **Software Manager**.

#### Exercise: Using the Software Manager

---

List the files which comprise the **Backup Subsystem**:

Run **scoadmin(ADM)**.

Select **Software Manager** ⇒ **SCO OpenServer Enterprise System** ⇒ **SCO OpenServer Enterprise System UNIX** ⇒ **System Administration** ⇒ **Backup Subsystem**.

Select **System backup and recovery tools**.

Select **Software** ⇒ **Examine** ⇒ **Files**.

### 16.2.2 Extracting a system file

---

Use the **customextract(ADM)** command to extract system files from your distribution media. The files are compressed on your distribution media. Other data transfer programs cannot be used to extract the data.

Files are extracted relative to the current working directory; when extracting a file, use the absolute pathname.

If you want to install a file to its original location, first change directory to the root (/) directory.

If you want to install a file to a location different than its original location, first change directory to that location, then enter the **customextract**(ADM) command line, specifying the absolute pathname.

For example, to install the */opt/K/SCO/Unix/5.0.5.Eb/etc/mount* file to your */tmp* directory, enter:

```
cd /tmp
```

```
customextract -m /dev/cd0 /opt/K/SCO/Unix/5.0.5Eb/etc/mount
```

The **-m** option specifies the media device.



## 16.3 Restoring a Non-root Filesystem

---

If you are receiving errors regarding a non-root filesystem, first run the **fsck**(ADM) command to correct any inconsistencies. If running **fsck** does not resolve the problem, you can restore the filesystem.

### 16.3.1 Running fsck(ADM)

---

Remember when running **fsck**, the filesystem must be unmounted. To run **fsck** from the command line:

**fsck /dev/name\_of\_filesystem**

Or, use the **scoadmin**(ADM) **Filesystem Manager**.

#### **Exercise: Running fsck(ADM)**

---

Run **fsck**(ADM) on your **/u** filesystem.

### 16.3.2 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. Run the **divvy**(ADM) command.
- Step 4. Select **c[reate]**.
- Step 5. Enter the division # of the non-root filesystem.
- Step 6. Enter **q** to quit.
- Step 7. Select **i** to install.

### 16.3.3 Procedure for restoring a non-root filesystem

---

After the filesystem has been recreated with the **divvy**(ADM) program, to restore the filesystem (assuming you use the **Backup Manager** to back up your system), do the following:

- Step 1. Log in as **root**.
- Step 2. Mount the filesystem.
- Step 3. Run **scoadmin**(ADM).
- Step 4. Select **Backup Manager**.
- Step 5. Select the filesystem.
- Step 6. Select **Restore** ⇨ **From Backup History**.
- Step 7. Select **Complete Filesystem**.
- Step 8. Click on **Start**.
- Step 9. Insert the media.
- Step 10. Click on **OK**.
- Step 11. Select **Host** ⇨ **Exit** to exit the **Backup Manager**.

#### **Exercise: Recreating and restoring a non-root filesystem**

---

Recreate your **/u** filesystem using the steps outlined in “Recreating a non-root filesystem” and “Restoring a non-root filesystem.”

After you recreate your **/u** filesystem, use the steps outlined in “Restoring a non-root filesystem” to restore your **/u** filesystem.

## 16.4 Restoring the root and boot filesystems

---

If your root filesystem is so corrupt that, when you boot the system, **fsck**(ADM) cannot run, or the system continues to display error messages, you must restore the root filesystem from backups. Before restoring your root filesystem, many of your recovery components must be available. If not, a partial installation of the operating system may be necessary.

### 16.4.1 Recovery components

---

To ensure a complete recovery of your root and boot filesystems, the following components are required:

- An emergency boot floppy set

This set must have a kernel with the correct tape drivers installed. If you have changed the type of tape drive since creating your emergency boot floppy set, you will need to configure the tape at the **BOOT :** prompt.

Refer to the *SCO OpenServer Handbook*, “Basic Hardware Configuration,” “Using bootstrings,” to configure your tape at boot.

- Your system log

This should contain the parameters for your **divvy**(ADM) and **fdisk**(ADM) tables. This is necessary if the hard disk or filesystem information is invalid.

- Your filesystem(s) backups

Additional filesystems are restored once you can boot from your hard disk. You must restore your root and boot filesystems from your emergency boot floppy set.

### 16.4.2 Reconfiguring the root hard disk

---

Use **mkdev hd** to reconfigure your root hard disk with your emergency boot floppy set. This program runs all the programs necessary to initialize your disk as described in Lesson 8, “Filesystem Configuration and Disk Management,” including:

- Setting disk parameters with **dkinit**(ADM), if necessary
- Partitioning the disk with **fdisk**(ADM). If you create a new partition during the re-creation of your root filesystem, you will be prompted to re-register your SCO OpenServer software on your next reboot of the system.

- Mapping bad blocks or sectors with **badtrk**(ADM)
- Recreating the filesystems and swap space with **divvy**(ADM)

If your disk parameters and partition information are valid, quit from **dkinit** and **fdisk**. Unless your system is displaying bad track or bad block errors, quit from **badtrk**.

During installation, you can accept the defaults as to the size of your swap space and additional filesystems. These can be changed and additional filesystems can be created by answering yes when **divvy** displays:

Do you want to make any manual adjustments?

Enter the size of your filesystems and swap space from your system log book.

### 16.4.3 Procedure for restoring the root and boot filesystems

---

The following steps describe how to restore your boot and root filesystems:

- Step 1. Shut down your system.
- Step 2. Insert your boot floppy disk.
- Step 3. At the **BOOT :** prompt, press .
- Step 4. When prompted, insert the root floppy **disk**, and press , and wait for the **root** prompt(**#**) to appear.
- Step 5. Enter the following to reconfigure your *root* hard disk:  
  
**mkdev hd**
- Step 6. Enter **y** when prompted to continue.
- Step 7. Select **q** to quit if **dkinit**(ADM) is run on your system.
- Step 8. Select **q** to quit from **fdisk**(ADM).
- Step 9. Select **q** to quit from **badtrk**(ADM).
- Step 10. Press  to use the existing value of the bad track table.
- Step 11. Enter **n** when prompted to preserve your additional filesystems.
- Step 12. Press  to accept the default swap-space allocation.

- Step 13. Press **ENTER** to accept the default boot-space allocation.
- Step 14. Enter **y** when prompted for a separate `/u` filesystem.
- Step 15. Enter the size of your `/u` filesystem in 1KB blocks.
- Step 16. Enter **y** when prompted with:  
  
do you want to make any manual adjustments?  
  
The **divvy**(ADM) program is invoked.
- Step 17. Enter or change the information required for each filesystem and swap space.
- Step 18. Select **q** to quit.
- Step 19. Select **i** to install the division setup shown.
- Step 20. Mount the root filesystem after the filesystems are created:  
  
**mount /dev/hd0root /mnt**
- Step 21. Change to the directory `/mnt`:  
  
**cd /mnt**
- Step 22. Restore the root filesystem.  
  
**cpio -ivmkBud -I/dev/rct0**
- Step 23. Wait for the restore to complete, then list the contents of the `/mnt` directory and make sure there is a `/stand` directory. If not, create it in the `/mnt` directory.
- Step 24. Mount the *boot* filesystem to `/mnt/stand`:  
  
**mount /dev/boot /mnt/stand**
- Step 25. Change to the directory `/mnt/stand`:  
  
**cd /mnt/stand**
- Step 26. Insert your boot filesystem tape and restore the boot filesystem:  
  
**cpio -ivmkBud -I/dev/rct0**
- Step 27. Wait for the restore to complete, then unmount your boot filesystem:

```
cd /  
umount /dev/boot
```

Step 28. Unmount your *root* filesystem:

```
umount /dev/hd0root
```

Step 29. Remove your floppy and tape and power down your system and boot normally.

Step 30. Restore any of your secondary filesystems with the **scoadmin**(ADM) **Backup Manager**.

## 16.5 Replacing the Primary Root Hard Disk

---

You may need to replace the root hard disk if an existing disk is no longer large enough for your needs or it suffers from some type of irreparable failure.

### 16.5.1 Recovery components

---

Replacing a root disk requires a partial installation. In addition to the components used when restoring a root filesystem, you will need the following:

- Original distribution media
- A backup of root (/) and /stand
- License number, activation key, and license data. You will need to re-register your SCO OpenServer software.

### 16.5.2 Partial installation

---

After you have installed your new disk, you will need to partially reinstall your operating system so that you can create disk partitions and filesystem divisions. The following steps describe how to partially install your operating system:

- Step 1. Start a fresh installation. Wait for the “Preparing your disk and choosing software” screen to appear.
- Step 2. Select the “Hard Disk Setup” field by pressing . Wait for the “Disk 0 Partitions and filesystems” screen to appear.
- Step 3. Press  and select **Customize**.
- Step 4. Recreate each of your old filesystems. Make the sizes of your filesystems at least as large as the originals, or your backups will not fit when you restore them.
- Step 5. Do the following when you are returned to the “Preparing your disk and choosing software” screen:
  - Press  at the “Optional Software” field
  - Omit the installation of all optional software by pressing **n** for “Operating system services,” “Graphical environment”, and so on

- Step 6.        Boot with your emergency boot floppy set when prompted to reboot.
- Step 7.        Insert your root diskette when prompted.
- Step 8.        Perform Steps 21 through 30 of the lesson, “Restoring your root and boot filesystems.”



## 16.6 Lesson Summary

---

---

Restoring filesystems is much easier when all recovery components are available. The key components are:

- Emergency boot floppy set, which includes the current tape driver
- Recent data and system backups

## Review questions

---

1. List the recovery components.
2. How can you list the contents of your distribution media?
3. List the steps to recreate a non-root filesystem.
4. List the components that you should have available when restoring your root filesystem?
5. Why is it important to have a system log available when replacing your root hard disk?

## 16.7 Lab: Recovering your root and boot filesystems

---

You will recover your root and boot filesystems in this lab.

### 16.7.1 Overview

---

After trying to repair the root filesystem with **fsck**(ADM) you have decided to recreate and restore both the root and boot filesystems. In doing so, your user filesystems should be recreated as well. You have current backups of your root, boot, and */u* filesystems.

Your instructor will provide you with the following:

- The **fdisk**(ADM) and **divvy**(ADM) parameters
- A complete backup of your root, boot and */u* filesystems

### 16.7.2 Procedure

---

The following steps describe how to restore your boot and root filesystems:

- Step 1. Shut down your system.
- Step 2. Insert your boot floppy.
- Step 3. At the **BooT :** prompt, press .
- Step 4. Insert the *root* floppy and press  and wait for the root prompt (**#**) to appear.
- Step 5. Enter the following to reconfigure your root hard disk:
- mkdev hd**
- Step 6. Enter **y** when prompted to continue.
- Step 7. Select **q** to quit if **dkinit**(ADM) is run on your system.
- Step 8. Select **q** to quit from **fdisk**(ADM).
- Step 9. Select **q** to quit from **badtrk**(ADM)
- Step 10. Press  to use the existing value of the bad track table.

- Step 11. Enter **n** when prompted to preserve your additional filesystems.
- Step 12. Press  to accept the default swap-space allocation.
- Step 13. Press  to accept the default boot-space allocation.
- Step 14. Enter **y** when prompted for a separate `/u` filesystem.
- Step 15. Enter the size of your `/u` filesystem in 1KB blocks.
- Step 16. Enter **y** when prompted with:
- Do you want to make any manual adjustments?
- The **divvy**(ADM) program is invoked.
- Step 17. Enter or change the information required for each filesystem and swap space.
- Step 18. Select **q** to quit.
- Step 19. Select **i** to install the division set-up shown.
- Step 20. Mount the root filesystem after the filesystems are created.
- mount /dev/hd0root /mnt**
- Step 21. Change to the `/mnt` directory:
- cd /mnt**
- Step 22. Restore the root filesystem:
- cpio -ivmkBud -I/dev/rct0**
- Step 23. Wait for the restore to complete, then list the contents of the `/mnt` directory and make sure there is a `stand` directory. If not, create it in the `/mnt` directory.
- Step 24. Mount the boot filesystem to `/mnt/stand`:
- mount /dev/boot /mnt/stand**
- Step 25. Change to the `/mnt/stand` directory:
- cd /mnt/stand**

- Step 26. Insert your *boot* filesystem tape and restore the boot filesystem:
- ```
cpio -ivmkBud -I/dev/rct0
```
- Step 27. Wait for the restore to complete, then unmount your boot filesystem:
- ```
cd /  
umount /dev/boot
```
- Step 28. Unmount your root filesystem:
- ```
umount /dev/hd0root
```
- Step 29. Remove your floppy and tape.
- Step 30. Enter **haltsys** and reboot your system.
- Step 31. Restore any of your secondary filesystems using the **Backup Manager** in **scoadmin**(ADM).

## 16.8 Relevant Commands and Files

---

The following commands and files were presented in this lesson.

### Relevant commands

---

|                       |                                                                                                                                              |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>divvy</b> (ADM)    | Invokes the utility for dividing a disk                                                                                                      |
| <b>fdisk</b> (ADM)    | Creates UNIX partitions                                                                                                                      |
| <b>fsck</b> (ADM)     | Checks and repairs filesystems                                                                                                               |
| <b>mkdev hd</b> (ADM) | Adds a hard disk to the system                                                                                                               |
| <b>scoadmin</b> (ADM) | Invokes SCOadmin applications (in this lesson, the <b>Software Manager</b> , the <b>Filesystem Manager</b> , and the <b>Backup Manager</b> ) |

### Relevant files

---

|                  |                                       |
|------------------|---------------------------------------|
| <i>/dev/boot</i> | The boot filesystem block device file |
| <i>/dev/root</i> | The root filesystem block device file |

## 16.9 Answers to Exercises and Review Questions

---

This section provides suggested answers to the exercises and review questions.

### Exercise: Running fsck(ADM)

---

To run **fsck**(ADM) on the `/u` filesystem, enter:

```
umount /dev/u
```

```
fsck /dev/u
```

### Review questions

---

This section provides suggested answers to the review questions.

1. Recovery components include the original media, registration information, emergency boot floppy set, system and backup logs, and system and data backups.
2. You can use the **Software Manager** to list the contents of your distribution media.
3. To recreate a non-root filesystem, shut down your system to single-user mode, unmount the filesystem, run the **divvy**(ADM) command, select **c** to create, enter the division number, enter **q** to quit, and select **i** to install.
4. When you are restoring the root filesystem, you should have the following components available: an emergency boot floppy set, a current system log, and filesystem backups.
5. It is important to have a system log available when replacing the root hard disk so that you have your original hard disk parameters and division sizes.