

Intermediate System
Administration for the Solaris™ 10
Operating System
SA-200-S10

Student Guide



Sun Microsystems, Inc.
UBRM05-104
500 Eldorado Blvd.
Broomfield, CO 80021
U.S.A.

Revision A.1

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Ultra, SunOS, Sun StorEdge, ToolTalk, SunSolve, SunService, Sun Blade, Sun Enterprise, OpenBoot, Sun Fire, and JumpStart are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

PostScript is a trademark or registered trademark of Adobe Systems, Incorporated, which may be registered in certain jurisdictions.

Federal Acquisitions: Commercial Software – Government Users Subject to Standard License Terms and Conditions

Export Laws. Products, Services, and technical data delivered by Sun may be subject to U.S. export controls or the trade laws of other countries. You will comply with all such laws and obtain all licenses to export, re-export, or import as may be required after delivery to You. You will not export or re-export to entities on the most current U.S. export exclusions lists or to any country subject to U.S. embargo or terrorist controls as specified in the U.S. export laws. You will not use or provide Products, Services, or technical data for nuclear, missile, or chemical biological weaponry end uses.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

THIS MANUAL IS DESIGNED TO SUPPORT AN INSTRUCTOR-LED TRAINING (ILT) COURSE AND IS INTENDED TO BE USED FOR REFERENCE PURPOSES IN CONJUNCTION WITH THE ILT COURSE. THE MANUAL IS NOT A STANDALONE TRAINING TOOL. USE OF THE MANUAL FOR SELF-STUDY WITHOUT CLASS ATTENDANCE IS NOT RECOMMENDED.

Export Commodity Classification Number (ECCN) assigned: 12 December 2001



Please
Recycle



Adobe PostScript™

Copyright 2005 Sun Microsystems Inc. 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Ultra, SunOS, Sun StorEdge, ToolTalk, SunSolve, SunService, Sun Blade, Sun Enterprise, OpenBoot, Sun Fire, et JumpStart sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

UNIX est une marques déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

PostScript est une marque fabrique d'Adobe Systems, Incorporated, laquelle pourrait être déposée dans certaines juridictions.

Législation en matière d'exportations. Les Produits, Services et données techniques livrés par Sun peuvent être soumis aux contrôles américains sur les exportations, ou à la législation commerciale d'autres pays. Nous nous conformerons à l'ensemble de ces textes et nous obtiendrons toutes licences d'exportation, de ré-exportation ou d'importation susceptibles d'être requises après livraison à Vous. Vous n'exporterez, ni ne ré-exporterez en aucun cas à des entités figurant sur les listes américaines d'interdiction d'exportation les plus courantes, ni vers un quelconque pays soumis à embargo par les Etats-Unis, ou à des contrôles anti-terroristes, comme prévu par la législation américaine en matière d'exportations. Vous n'utiliserez, ni ne fournirez les Produits, Services ou données techniques pour aucune utilisation finale liée aux armes nucléaires, chimiques ou biologiques ou aux missiles.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

CE MANUEL DE RÉFÉRENCE DOIT ÊTRE UTILISÉ DANS LE CADRE D'UN COURS DE FORMATION DIRIGÉ PAR UN INSTRUCTEUR (ILT). IL NE S'AGIT PAS D'UN OUTIL DE FORMATION INDÉPENDANT. NOUS VOUS DÉCONSEILLONS DE L'UTILISER DANS LE CADRE D'UNE AUTO-FORMATION.



Please
Recycle



Adobe PostScript

Table of Contents

About This Course	Preface-xix
Course Goals	Preface-xix
Course Map	Preface-xx
Topics Not Covered	Preface-xxi
How Prepared Are You?	Preface-xxii
Introductions	Preface-xxiii
How to Use Course Materials	Preface-xxiv
Conventions	Preface-xxv
Icons	Preface-xxv
Typographical Conventions	Preface-xxvi
Notes to the Instructor.....	Preface-xxvii
Installing the Solaris™ 10 Operating System	1-1
Objectives	1-1
Identifying the Fundamentals of the CD-ROM Installation	1-2
Solaris 10 OS Installation and Upgrade Options.....	1-2
Solaris Installation Command Line Interpreter (CLI)	1-2
Custom JumpStart™ Installation.....	1-3
Solaris Flash Archive Installation	1-3
Solaris WAN Boot	1-4
Standard Upgrade to the Solaris OS	1-4
Solaris Live Upgrade Software	1-5
Hardware Requirements for Installation of the Solaris 10 OS	1-5
Software Components of the Solaris OS.....	1-6
Solaris OS Software Groups	1-7
Installing the Solaris 10 OS From a CD-ROM or DVD	1-11
Pre-Installation Information	1-11
Demonstration: Performing an Interactive Installation	1-13
Introducing the Solaris™ 10 OS Directory Hierarchy	2-1
Objectives	2-1
Introducing / (root) Subdirectories.....	2-2
Introducing Important System Directories	2-3
Introducing Important In-Memory System Directories	2-4

Introducing File Components	2-9
File Names.....	2-9
Inodes.....	2-9
Data Blocks.....	2-10
Identifying File Types.....	2-11
Regular Files	2-12
Directories	2-13
Symbolic Links	2-14
Device Files	2-16
Using Hard Links.....	2-20
Introducing Hard Links	2-20
Creating New Hard Links	2-21
Removing Hard Links	2-22
Performing the Exercises	2-23
Exercise: Identifying File Types (Level 1).....	2-24
Preparation.....	2-24
Tasks	2-24
Exercise: Identifying File Types (Level 2).....	2-26
Preparation.....	2-26
Task Summary.....	2-26
Tasks	2-28
Exercise: Identifying File Types (Level 3).....	2-29
Preparation.....	2-29
Task Summary.....	2-29
Tasks and Solutions	2-31
Exercise Summary.....	2-34
Managing Local Disk Devices.....	3-1
Objectives	3-1
Introducing the Basic Architecture of a Disk	3-2
Physical Disk Structure	3-2
Data Organization on Disk Platters.....	3-3
Disk Slices.....	3-4
Introducing Solaris OS Device Naming Conventions	3-10
Logical Device Names	3-10
Physical Device Names	3-11
Instance Names	3-12
Listing a System's Devices	3-13
The /etc/path_to_inst File	3-13
The prtconf Command.....	3-15
The format Command.....	3-16
Reconfiguring Devices	3-17
Performing a Reconfiguration Boot.....	3-17
Using the devfsadm Command	3-18
Performing the Exercises	3-20

Exercise: Configuring and Naming Devices (Level 1).....	3-21
Preparation.....	3-21
Tasks	3-22
Exercise: Configuring and Naming Devices (Level 2).....	3-23
Preparation.....	3-23
Task Summary	3-24
Tasks	3-24
Exercise: Configuring and Naming Devices (Level 3).....	3-26
Preparation.....	3-26
Task Summary	3-27
Tasks and Solutions	3-27
Exercise Summary.....	3-30
Partitioning the Hard Disk	3-31
Introducing the Fundamentals of Disk Partitioning.....	3-31
Recognizing Disk Space and Undesirable Conditions	3-32
Recognizing Wasted Disk Space.....	3-33
Recognizing Overlapping Disk Slices	3-34
Introducing Disk Partition Tables	3-35
Using the <code>format</code> Command	3-36
Partitioning a Disk	3-38
Managing Disk Labels.....	3-45
Viewing the Disk VTOC	3-45
Relabeling a Disk.....	3-47
Performing the Exercises	3-48
Exercise: Working With Disks and Partitions (Level 1)	3-49
Preparation.....	3-49
Tasks	3-49
Exercise: Working With Disks and Partitions (Level 2)	3-51
Preparation.....	3-51
Task Summary	3-51
Tasks	3-52
Exercise: Working With Disks and Partitions (Level 3)	3-56
Preparation.....	3-56
Task Summary	3-56
Tasks	3-57
Introducing the Solaris TM Management Console	3-64
Starting the Solaris Management Console	3-64
Using the Solaris Management Console Tools	3-65
Restarting the Solaris Management Console	3-66
Identifying the Functional Areas of the Solaris Management Console	3-67
Partitioning a Disk by Using the Solaris Management Console Disks Manager Tool.....	3-71
Partitioning the Disk Using the Disks Tool.....	3-71
Performing the Exercises	3-80

Exercise: Working With the Solaris Management Console (Level 1).....	3-81
Preparation.....	3-81
Tasks	3-81
Exercise: Working With the Solaris Management Console (Level 2).....	3-82
Preparation.....	3-82
Task Summary.....	3-82
Tasks	3-83
Exercise Summary.....	3-84
Managing Solaris OS File Systems	4-1
Objectives	4-1
Introducing Solaris OS File Systems	4-2
Disk-based File Systems.....	4-2
Distributed File Systems	4-2
Pseudo File Systems	4-3
Creating a New ufs File System.....	4-4
Viewing the Solaris OS ufs File System	4-4
Using the newfs Command.....	4-14
Checking the File System by Using the fsck Command.....	4-17
Data Inconsistencies Checked by the fsck Command	4-17
Superblock Consistency	4-17
Cylinder Group Block Consistency	4-17
Inode Consistency	4-18
Data Block Consistency	4-18
The lost+found Directory	4-18
Noninteractive Mode.....	4-18
Interactive Mode	4-19
Resolving File System Inconsistencies	4-20
Reconnecting an Allocated Unreferenced File.....	4-20
Adjusting a Link Counter	4-21
Salvaging the Free List	4-21
Using Backup Superblocks	4-22
Monitoring File System Use	4-25
Using the df Command	4-25
Using the du Command	4-28
Using the quot Command	4-30
Using the Solaris Management Console Usage Tool.....	4-31
Performing the Exercises	4-33
Exercise: Creating and Maintaining ufs File Systems (Level 1)4-34	
Preparation.....	4-34
Tasks	4-35
Exercise: Creating and Maintaining ufs File Systems (Level 2)4-36	
Preparation.....	4-36
Task Summary	4-37
Tasks	4-38

Exercise: Creating and Maintaining ufs File Systems (Level 3).....	4-40
Preparation.....	4-40
Task Summary	4-41
Tasks and Solutions	4-42
Exercise Summary.....	4-47
Performing Mounts and Unmounts	5-1
Objectives	5-1
Working With Mounting Basics	5-2
Determining Which File Systems Are Currently Mounted	5-4
Mounting a File System Automatically	5-4
Introducing the Virtual File System Table: /etc/vfstab..	5-5
Introducing the /etc/mnttab File	5-8
Performing Mounts.....	5-11
Mounting a Local File System Manually	5-11
Using the mount Command Options	5-12
Mounting All File Systems Manually	5-14
Mounting a New File System	5-15
Mounting Different Types of File Systems.....	5-16
Performing Unmounts	5-18
Unmounting a File System	5-18
Unmounting All File Systems	5-19
Unmounting a Busy File System.....	5-19
Repairing Important Files if Boot Fails	5-21
Accessing Mounted Diskettes, CD-ROMs or DVDs	5-23
Using Volume Management (vold).....	5-24
Restricting Access to Mounted Diskettes, CD-ROMs, or DVDs	5-26
Stopping Volume Management (vold)	5-26
Troubleshooting Volume Management (vold) Problems .	5-26
Accessing a Diskette, CD-ROM, or DVD Without Volume	
Management (vold)	5-27
Using the mount Command.....	5-27
Performing the Exercises	5-28
Exercise: Mounting File Systems (Level 1).....	5-29
Preparation.....	5-29
Tasks	5-29
Exercise: Mounting File Systems (Level 2).....	5-31
Preparation.....	5-31
Task Summary.....	5-31
Tasks	5-32
Exercise: Mounting File Systems (Level 3).....	5-34
Preparation.....	5-34
Task Summary	5-34
Tasks and Solutions	5-35
Exercise Summary.....	5-38

Performing Solaris 10 OS Package Administration.....	6-1
Objectives	6-1
Introducing the Fundamentals of Package Administration	6-2
Software Packages	6-2
The /var/sadm/install/contents File	6-2
Package Formats	6-4
Administering Packages From the Command Line.....	6-6
Translating Package Formats	6-6
Displaying Information About Installed Software Packages	6-7
Adding a Software Package	6-9
Checking a Package Installation	6-12
Removing a Software Package.....	6-14
Adding Packages by Using a Spool Directory.....	6-15
Streaming One or More Packages	6-17
Reviewing Package Administration.....	6-18
Performing the Exercises	6-19
Exercise: Manipulating Software Packages (Level 1)	6-20
Preparation.....	6-20
Tasks	6-20
Exercise: Manipulating Software Packages (Level 2)	6-21
Preparation.....	6-21
Task Summary	6-21
Tasks	6-22
Exercise: Manipulating Software Packages (Level 3)	6-24
Preparation.....	6-24
Task Summary.....	6-24
Tasks and Solutions	6-25
Exercise Summary.....	6-29
Managing Software Patches on the Solaris 10 OS	7-1
Objectives	7-1
Preparing for Patch Administration.....	7-2
Introducing Solaris OS Patches	7-2
Checking Patch Levels	7-4
Obtaining Patches	7-5
Preparing Patches for Installation	7-6
Installing and Removing Patches	7-9
Installing a Patch.....	7-9
Removing a Patch	7-10
Installing Patch Clusters	7-11
The smpatch Utility	7-14
Performing the Exercises	7-15
Exercise: Maintaining Patches (Level 1)	7-16
Preparation.....	7-16
Tasks	7-16

Exercise: Maintaining Patches (Level 2)	7-17
Preparation.....	7-17
Task Summary.....	7-17
Tasks	7-18
Exercise: Maintaining Patches (Level 3)	7-19
Preparation.....	7-19
Task Summary.....	7-19
Tasks and Solutions	7-20
Exercise Summary.....	7-23
Executing Boot PROM Commands.....	8-1
Objectives	8-1
Introducing Boot PROM Fundamentals.....	8-2
Goal of the OpenBoot™ Architecture Standard.....	8-3
Boot PROM	8-3
System Configuration Information	8-5
Disabling the Abort Sequence.....	8-8
Displaying POST to the Serial Port	8-9
Using Basic Boot PROM Commands	8-11
Identifying the System Boot PROM Version.....	8-12
Booting the System	8-12
Accessing More Detailed Information	8-14
Listing NVRAM Parameters	8-15
Changing NVRAM Parameters	8-16
Restoring Default NVRAM Parameters	8-17
Displaying Devices Connected to the Bus.....	8-17
Identifying the System's Boot Device	8-20
The show-devs Command	8-22
The devalias Command	8-23
Creating and Removing Custom Device Aliases	8-24
The nvalias Command.....	8-24
The nvunalias Command	8-25
Viewing and Changing NVRAM Parameters From the OS	8-26
Using the eeprom Command	8-26
Interrupting an Unresponsive System.....	8-27
Aborting an Unresponsive System.....	8-27
Performing the Exercises	8-28
Exercise: Using the OpenBoot PROM Commands (Level 1)	8-29
Preparation.....	8-29
Tasks	8-29
Exercise: Using the OpenBoot PROM Commands (Level 2)	8-31
Preparation.....	8-31
Task Summary	8-31
Tasks	8-33

Exercise: Using the OpenBoot PROM Commands (Level 3)	8-36
Preparation.....	8-36
Task Summary.....	8-36
Tasks and Solutions	8-38
Exercise Summary.....	8-42
Performing Boot and Shutdown Procedures	9-1
Objectives	9-1
The Service Management Facility (SMF)	9-2
SMF Service.....	9-2
Service Identifiers.....	9-3
Service States	9-6
Milestones	9-7
The svc.startd Daemon.....	9-10
The Service Configuration Repository.....	9-10
Identifying Legacy Run Level Fundamentals.....	9-12
Determining a System's Current Run Level	9-13
Changing Run Levels	9-13
Identifying the Phases of the Boot Process.....	9-14
Boot PROM Phase	9-15
Boot Programs Phase.....	9-16
The kernel Initialization Phase	9-16
The /etc/system File and Kernel Configuration.....	9-18
The init Phase.....	9-22
The svc.startd Daemon.....	9-24
Controlling Legacy Boot Processes	9-25
The /sbin Directory	9-25
The /etc/rc#.d Directories	9-27
Start Run Control Scripts	9-28
Stop Run Control Scripts	9-28
The /etc/init.d Directory	9-29
Stopping and Starting Services Using SMF Commands ..	9-30
Using svcs to Determine Why Services are Not Running	9-34
Creating New Service Scripts.....	9-36
Performing System Shutdown Procedures	9-44
The /usr/sbin/init Command	9-45
The /usr/sbin/shutdown Command	9-45
“Ungraceful” Shutdown Commands.....	9-47
The Service Repository Database.....	9-48
Performing the Exercises	9-50
Exercise: Controlling the Boot Process (Level 1)	9-51
Preparation.....	9-51
Tasks	9-52
Exercise: Controlling the Boot Process (Level 2)	9-53
Preparation.....	9-53
Task Summary	9-53
Tasks	9-54

Exercise: Controlling the Boot Process (Level 3)	9-57
Preparation.....	9-57
Task Summary	9-57
Tasks and Solutions	9-58
Exercise Summary.....	9-63
Performing User Administration	10-1
Objectives	10-1
Introducing User Administration.....	10-2
Main Components of a User Account.....	10-2
System Files That Store User Account Information.....	10-3
Managing User Accounts.....	10-14
Introducing Command-Line Tools.....	10-14
Creating a User Account.....	10-15
Modifying a User Account	10-20
Deleting a User Account	10-22
Creating a Group Entry.....	10-23
Modifying a Group Entry	10-24
Deleting a Group Entry	10-26
Using the Solaris Management Console Users Tool.....	10-27
Troubleshooting Login Issues	10-36
Performing the Exercises	10-40
Exercise: Adding User Accounts and Group Entries (Level 1).....	10-41
Preparation.....	10-41
Tasks	10-43
Exercise: Adding User Accounts and Group Entries (Level 2).....	10-45
Preparation.....	10-45
Task Summary	10-45
Tasks	10-46
Exercise: Adding User Accounts and Group Entries (Level 3).....	10-51
Preparation.....	10-51
Task Summary	10-51
Tasks and Solutions	10-52
Exercise Summary.....	10-58
Managing Initialization Files.....	10-59
Introducing System-Wide Initialization Files	10-59
Introducing User Initialization Files	10-60
Customizing the User's Work Environment.....	10-61
Performing the Exercises	10-64
Exercise: Modifying Initialization Files (Level 1).....	10-65
Preparation.....	10-65
Tasks	10-65
Exercise: Modifying Initialization Files (Level 2).....	10-67
Preparation.....	10-67
Task Summary	10-67
Tasks	10-68
Exercise: Modifying Initialization Files (Level 3).....	10-71
Preparation.....	10-71

Task Summary	10-71
Tasks and Solutions	10-72
Exercise Summary.....	10-76
Performing System Security.....	11-1
Objectives	11-1
Monitoring System Access.....	11-2
Displaying Users on the Local System.....	11-2
Displaying Users on Remote Systems.....	11-3
Displaying User Information	11-4
Displaying a Record of Login Activity	11-5
Recording Failed Login Attempts	11-6
Switching Users on a System	11-8
Introducing the su Command.....	11-8
Switching to Another Regular User	11-10
Becoming the root User	11-11
Monitoring su Attempts	11-12
Controlling System Access	11-14
The /etc/default/login File.....	11-14
File Transfer Protocol (FTP) Access.....	11-16
The /etc/hosts.equiv and \$HOME/.rhosts Files	11-17
The /etc/hosts.equiv File Rules	11-19
The \$HOME/.rhosts File Rules.....	11-20
Performing the Exercises	11-21
Exercise: User Access (Level 1)	11-22
Preparation.....	11-22
Tasks	11-23
Exercise: User Access (Level 2)	11-24
Preparation.....	11-24
Task Summary	11-25
Tasks	11-25
Exercise: User Access (Level 3)	11-28
Preparation.....	11-28
Task Summary	11-29
Tasks and Solutions	11-30
Exercise Summary.....	11-35
Restricting Access to Data in Files.....	11-36
Determining a User's Group Membership.....	11-36
Identifying a User Account.....	11-37
Changing File and Directory Ownership	11-37
Changing File and Directory Group Membership	11-40
Using File Permissions	11-41
Performing the Exercises	11-44
Exercise: Restricting Access to Data on Systems (Level 1).....	11-45
Preparation.....	11-45
Tasks	11-45

Exercise: Restricting Access to Data on Systems (Level 2).....	11-47
Preparation.....	11-47
Task Summary	11-47
Tasks	11-48
Exercise: Restricting Access to Data on Systems (Level 3).....	11-51
Preparation.....	11-51
Task Summary	11-51
Tasks and Solutions	11-52
Exercise Summary.....	11-57
Configuring and Using Printer Services	12-1
Objectives	12-1
Introducing Network Printing Fundamentals.....	12-2
Raster Image Processor (RIP)	12-2
PostScript Printer Description (PPD)	12-2
Print Management Tools.....	12-3
Client-Server Model.....	12-3
Types of Printer Configurations	12-3
Basic Functions of the Solaris OS LP Print Service	12-5
LP Print Service Directory Structure	12-6
Print Requests From the Network.....	12-10
Solaris OS Printing Process.....	12-12
Configuring Printer Services	12-19
Using the Solaris OS Print Manager.....	12-19
Configuring a New Network Printer	12-22
Administering Printer Services.....	12-30
Setting the System's Default Printer.....	12-31
Removing a Client's Printer Configuration	12-31
Removing a Server's Printer Configuration.....	12-32
Starting and Stopping the LP Print Service.....	12-33
Starting the LP Print Service.....	12-33
Stopping the LP Print Service	12-33
Specifying a Destination Printer	12-34
Using the lp Command	12-34
Using the lpr Command	12-34
Using the LP Print Service	12-35
Accepting Print Jobs	12-35
Rejecting Print Jobs.....	12-36
Enabling Printers.....	12-36
Disabling Printers	12-37
Moving Print Jobs	12-38
Performing the Exercises	12-40
Exercise: Using the LP Print Service (Level 1)	12-41
Preparation.....	12-41
Tasks	12-41

Exercise: Using the LP Print Service (Level 2)	12-43
Preparation.....	12-43
Task Summary.....	12-43
Tasks	12-44
Exercise: Using the LP Print Service (Level 3)	12-48
Preparation.....	12-48
Task Summary.....	12-48
Tasks and Solutions	12-49
Exercise Summary.....	12-53
Controlling System Processes	13-1
Objectives	13-1
Viewing System Processes.....	13-2
Using the CDE Process Manager	13-2
Using the prstat Command	13-4
Using the Solaris Management Console Process Tool.....	13-7
Killing Frozen Processes	13-9
Using the kill and pkill Commands	13-9
Performing a Remote Login	13-11
Suspending and Terminating Processes with SMC	13-12
Scheduling an Automatic One-Time Execution of a Command.....	13-14
Using the at Command	13-14
Controlling Access to the at Command.....	13-16
Scheduling an Automatic Recurring Execution of a Command.....	13-17
Introducing the crontab File Format	13-17
Using the crontab Command	13-19
Controlling Access to the crontab Command.....	13-21
Using the Solaris™ Management Console Job Scheduler Tool	13-22
Performing the Exercises	13-24
Exercise: Using Process Control (Level 1)	13-25
Preparation.....	13-25
Tasks	13-26
Exercise: Using Process Control (Level 2)	13-27
Preparation.....	13-27
Task Summary.....	13-28
Tasks	13-28
Exercise: Using Process Control (Level 3)	13-30
Preparation.....	13-30
Task Summary.....	13-31
Tasks and Solutions	13-32
Exercise Summary.....	13-35

Performing File System Backups	14-1
Objectives	14-1
Introducing the Fundamentals of Backups	14-2
Importance of Routine File System Backups	14-2
Tape Media Types.....	14-3
Tape Drive Naming	14-4
Tape Drive Control	14-5
Strategies for Scheduled Backups.....	14-6
The /etc/dumpdates File.....	14-9
Backing Up an Unmounted File System.....	14-10
The ufsdump Command.....	14-10
Options for the ufsdump Command	14-11
Tape Back Ups	14-12
Remote Backups to a Tape.....	14-13
Performing the Exercises	14-14
Exercise: Backing Up a File System (Level 1).....	14-15
Preparation.....	14-15
Tasks	14-15
Exercise: Backing Up a File System (Level 2).....	14-17
Preparation.....	14-17
Task Summary	14-17
Tasks	14-18
Exercise: Backing Up a File System (Level 3).....	14-19
Preparation.....	14-19
Task Summary	14-19
Tasks and Solutions	14-20
Exercise Summary.....	14-22
Performing File System Restores	15-1
Objectives	15-1
Restoring a ufs File System.....	15-2
Restoring a Regular File System	15-2
Restoring the /usr File System	15-4
Performing a Special Case Recovery of the / (root)	
File System	15-6
Invoking an Interactive Restore	15-7
Performing an Incremental Restore	15-9
Performing the Exercises	15-14
Exercise: Recovering Backup Files and File Systems (Level 1) 15-15	
Preparation.....	15-15
Tasks	15-15
Exercise: Recovering Backup Files and File Systems (Level 2) 15-17	
Preparation.....	15-17
Task Summary	15-17
Tasks	15-19
Exercise: Recovering Backup Files and File Systems (Level 3) 15-21	
Preparation.....	15-21

Task Summary	15-21
Tasks and Solutions	15-22
Exercise Summary.....	15-27
Backing Up a Mounted File System With a UFS Snapshot..... 16-1	
Objectives	16-1
Creating a UFS Snapshot	16-2
Using the fssnap Command	16-2
Limiting the Size of the Backing-Store File	16-4
Displaying Information for a ufs File System Snapshot .	16-5
Backing Up the UFS Snapshot File	16-6
Performing a Backup of a UFS Snapshot.....	16-6
Performing an Incremental Backup Using a UFS Snapshot.....	16-7
Restoring Data From a UFS Snapshot Backup	16-10
Deleting a UFS Snapshot.....	16-10
Performing the Exercises	16-11
Exercise: Working With UFS Snapshots (Level 1).....	16-12
Tasks	16-12
Exercise: Working With UFS Snapshots (Level 2).....	16-13
Task Summary.....	16-13
Tasks	16-13
Exercise: Working With UFS Snapshots (Level 3).....	16-15
Task Summary.....	16-15
Tasks and Solutions	16-15
Exercise Summary.....	16-17

About This Course

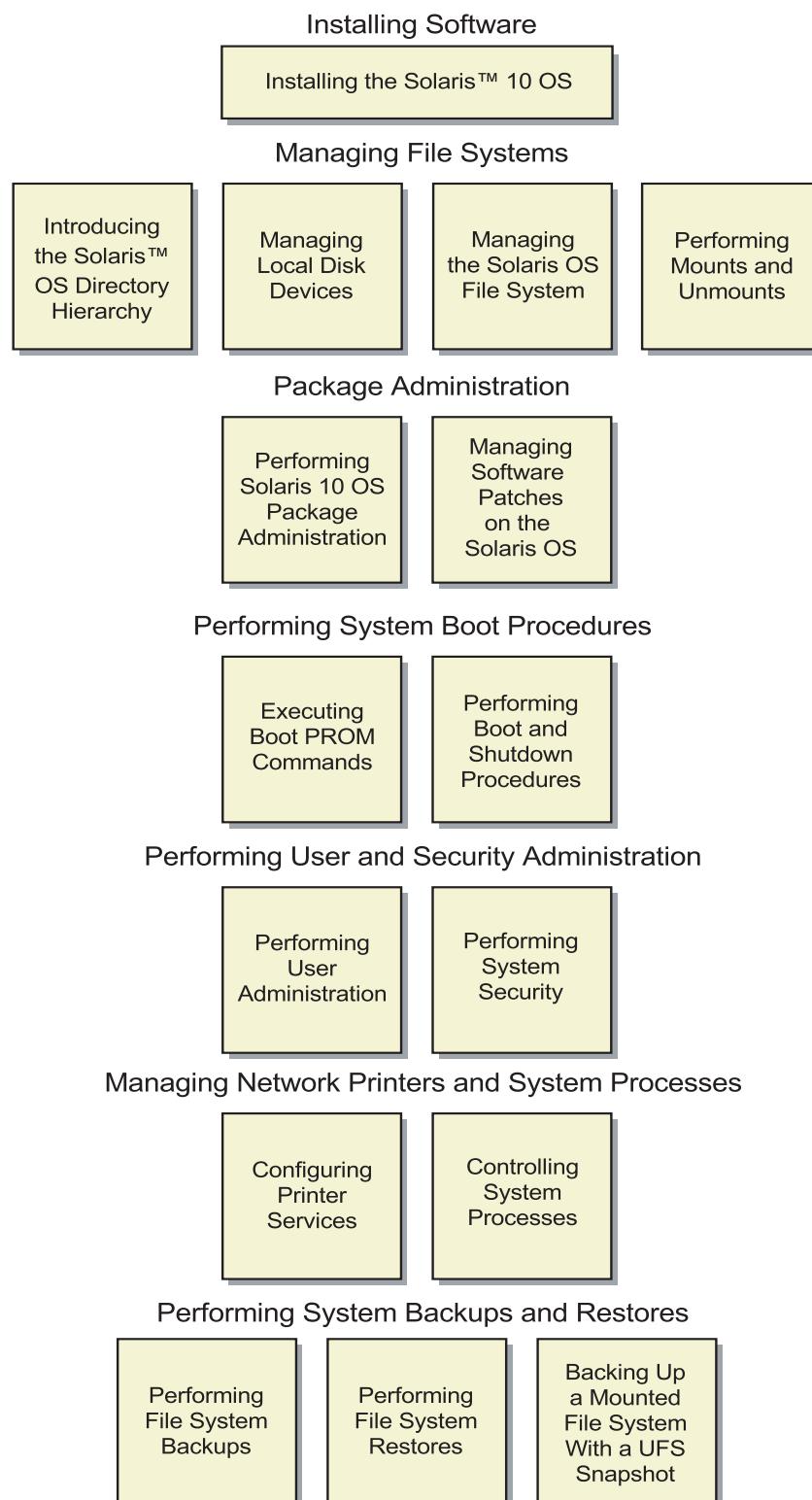
Course Goals

Upon completion of this course, you should be able to:

- Manage file systems
- Install software
- Perform system boot procedures
- Perform user and security administration
- Manage network printers and system processes
- Perform system backups and restores

Course Map

The course map enables you to see what you have accomplished and where you are going in reference to the course goals.



Topics Not Covered

This course does not cover the following topics. Many of these topics are covered in other courses offered by Sun Educational Services:

- Basic UNIX® commands – Covered in SA-100: *UNIX® Essentials Featuring the Solaris™ 10 Operating System*
- The vi editor – Covered in SA-100: *UNIX® Essentials Featuring the Solaris™ 10 Operating System*
- Basic UNIX file security – Covered in SA-100: *UNIX® Essentials Featuring the Solaris™ 10 Operating System*
- JumpStart™ procedure– Covered in SA-202: *Advanced System Administration for the Solaris™ 10 Operating System*
- Network File System (NFS) environment configuration – Covered in SA-202: *Advanced System Administration for the Solaris™ 10 Operating System*
- All the new features in Solaris 10 – Covered in SA-225-S10: *Solaris™ 10 for Experienced System Administrators*
- Naming services – Covered in SA-202: *Advanced System Administration for the Solaris™ 10 Operating System*
- Troubleshooting – Covered in ST-350: *Sun™ Systems Fault Analysis Workshop*
- System tuning – Covered in SA-400: *Solaris™ System Performance Management*

Refer to the Sun Educational Services catalog for specific information and registration.

How Prepared Are You?

To be sure you are prepared to take this course, can you answer yes to the following questions?

- Perform basic UNIX tasks
- Understand basic UNIX commands
- Use the `vi` text editor
- Interact with a windowing system

Introductions

Now that you have been introduced to the course, introduce yourself to the other students and the instructor, addressing the items shown below:

- Name
- Company affiliation
- Title, function, and job responsibility
- Experience related to topics presented in this course
- Reasons for enrolling in this course
- Expectations for this course

How to Use Course Materials

To enable you to succeed in this course, these course materials employ a learning module that is composed of the following components:

- Objectives – You should be able to accomplish the objectives after completing a portion of instructional content. Objectives support goals and can support other higher-level objectives.
- Lecture – The instructor will present information specific to the objective of the module. This information will help you learn the knowledge and skills necessary to succeed with the activities.
- Activities – The activities take on various forms, such as an exercise, self-check, discussion, and demonstration. Activities are used to facilitate mastery of an objective.
- Visual aids – The instructor might use several visual aids to convey a concept, such as a process, in a visual form. Visual aids commonly contain graphics, animation, and video.

Note – Many system administration tasks for the Solaris™ Operating System can be accomplished in more than one way. The methods presented in the courseware reflect recommended practices used by Sun Services.



Conventions

The following conventions are used in this course to represent various training elements and alternative learning resources.

Icons



Discussion – Indicates a small-group or class discussion on the current topic is recommended at this time.



Demonstration – Indicates a demonstration of the current topic is recommended at this time.



Note – Indicates additional information that can help students but is not crucial to their understanding of the concept being described. Students should be able to understand the concept or complete the task without this information. Examples of notational information include keyword shortcuts and minor system adjustments.



Caution – Indicates that there is a risk of personal injury from a nonelectrical hazard, or risk of irreversible damage to data, software, or the operating system. A caution indicates that the possibility of a hazard (as opposed to certainty) might happen, depending on the action of the user.

Typographical Conventions

Courier is used for the names of commands, files, directories, user names, host names, programming code, and on-screen computer output; for example:

Use the `ls -al` command to list all files.

```
host1# cd /home
```

Courier bold is used for characters and numbers that you type; for example:

To list the files in this directory, type the following:

```
# ls
```

Courier italics is used for variables and command-line placeholders that are replaced with a real name or value; for example:

To delete a file, use the `rm filename` command.

Courier italic bold is used to represent variables whose values are to be entered by the student as part of an activity; for example:

Type `chmod a+rwx filename` to grant read, write, and execute rights for *filename*.

Palatino italics is used for book titles, new words or terms, or words that you want to emphasize; for example:

Read Chapter 6 in the *User's Guide*.

These are called *class* options.

Module 1

Installing the Solaris™ 10 Operating System

Objectives

Upon completion of this module, you should be able to:

- Identify the fundamentals of the Solaris™ 10 Operating System (Solaris 10 OS) installation from a CD-ROM or DVD
- Install Solaris 10 OS from a CD-ROM or DVD

The course map in Figure 1-1 shows how this module fits into the current instructional goal.

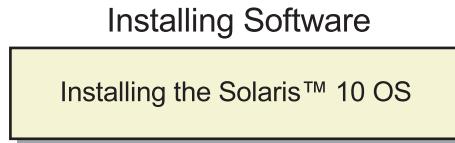


Figure 1-1 Course Map

Identifying the Fundamentals of the CD-ROM Installation

The following section describes the CD-ROM or DVD installation of the Solaris 10 OS.

Solaris 10 OS Installation and Upgrade Options

There are two ways to install the Solaris 10 OS on your system, suninstall and Flash installation. There are a number of different ways the installation can take place:

- Solaris installation Graphical User Interface (GUI)
- Solaris installation Command Line Interpreter (CLI)
- Solaris Custom JumpStart™ software (JumpStart) installation
- Solaris Flash Archives
- Solaris WAN boot installation
- Solaris Upgrade method

This module focuses on the text installation. The default installation method is graphical if the host has sufficient memory, and is using a graphical interface.

Solaris Installation Command Line Interpreter (CLI)

Hosts which do not have a graphical screen cannot run the GUI installation. Starting the installation with the nowin argument allows all the questions and answers to be completed in a text-only environment. Options are provided in menu format with the spacebar being used to select options and F2, (or the equivalent escape key sequence), being used to accept selected options.

- 64-127 Mbytes starts with nowin
- 128-383 Mbytes starts a GUI window with a text-based install running in it
- 384-511 Mbytes starts up the GUI interface
- 512 Mbytes and higher starts the installation kiosk

Custom JumpStart™ Installation

The Solaris JumpStart procedure installs Solaris OS software on a system by referencing a user-defined profile. You can customize profiles for different types of systems.

A JumpStart installation provides a hands-off installation across the network and is based on a central-configured server. The JumpStart procedure is a command-line interface that enables you to incorporate shell scripts. The shell scripts include pre-installation and post-installation tasks. You choose the profile and the scripts to use for installation or upgrade. Then the custom installation method installs or upgrades the system.

Solaris Flash Archive Installation

The Solaris Flash Archive Installation enables you to install many systems based on a configuration that you install on a master system. After you have installed and configured the master system, you create a flash archive from the master system. You create as many flash archives as you need and choose which flash archive to install on each system.

The standard Solaris OS installation methods install each Solaris OS package individually. This method of package-based installation is time consuming because the installation must update the package map for each package. The Solaris Flash archive installs on your system much faster than when you install each of the individual Solaris OS packages, because you are only producing a copy of an already installed system.

Solaris WAN Boot

The WAN boot installation method enables you to boot and install software over a wide area network (WAN) by using HTTP. The WAN boot installation method enables you to transmit an encrypted Solaris Flash archive over a public network to a remote SPARC®-based client. The WAN boot programs then install the client system by performing a custom JumpStart installation.

To protect the integrity of the installation, you can use private keys to authenticate and encrypt data. You can also transmit your installation data and files over a secure HTTP connection by configuring your systems to use digital certificates.

Solaris upgrade options include both the standard upgrade and the live upgrade.

Standard Upgrade to the Solaris OS

A standard upgrade merges the new version of the Solaris OS with the existing files on the system's disk. The methods available for a standard upgrade are Solaris GUI installation, the CLI installation, and the custom JumpStart procedure.

A standard upgrade saves many of the modifications that were made to the OS with the previous version of the Solaris OS. Because the Solaris OS is unavailable to users during the standard upgrade, the standard upgrade results in longer periods of downtime.

Solaris Live Upgrade Software

The Solaris Live Upgrade Software upgrades a duplicate boot environment while the active boot environment is still running. This method eliminates downtime of the production environment. The Solaris Live Upgrade method can be run with either a GUI or a command-line interface. First, create a duplicate boot environment. After that has been created, upgrade or install a Solaris Web Start Flash archive on the inactive boot environment. When you are ready, activate the inactive boot environment. During the next reboot, the inactive boot environment becomes the active boot environment. If there is a failure, you can recover your original boot environment by reactivating it and rebooting the system.

Solaris Live Upgrade Software requires enough available disk space to create a duplicate of your boot environment. To estimate the file system size needed to create a boot environment, start the creation of the new boot environment. The file system size is calculated, and you can then abort the process.

Hardware Requirements for Installation of the Solaris 10 OS

A Solaris 10 OS installation requires the following:

- 256 Mbytes of memory minimum recommended
- At least 5 Gbytes of disk space
- Access to a CD-ROM/DVD drive or an installation server

Software Components of the Solaris OS

The Solaris OS software is organized into three components:

- Software packages
- Software clusters
- Software groups

Figure 1-2 shows the relationship among the Common Desktop Environment (CDE) software components.

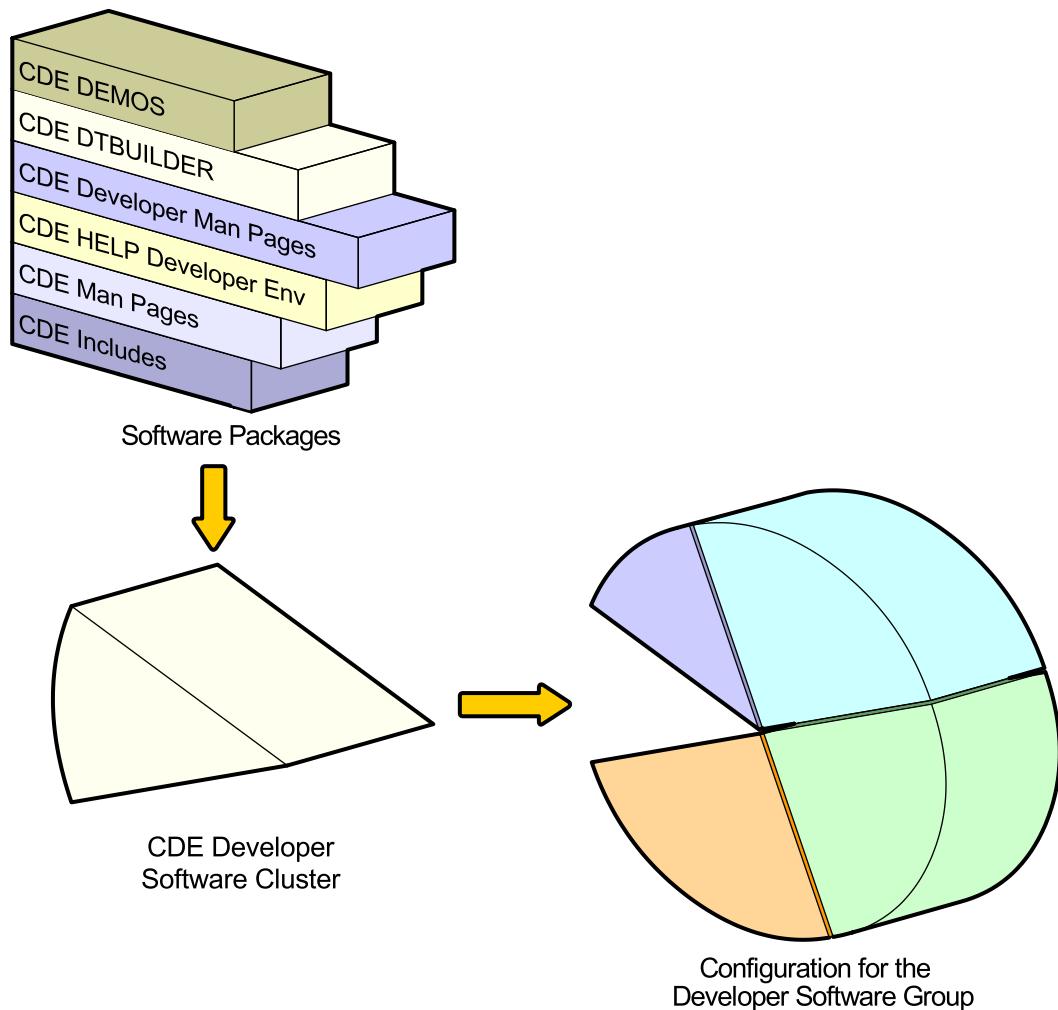


Figure 1-2 Solaris OS Software Components

Software Packages

A software package contains a group of software files and directories. The package also contains the related software installation scripts.

Software Clusters

During the software installation process, software clusters group logical collections of software packages together. Table 1-1 shows the software packages that are grouped into the CDE software cluster.

Table 1-1 Packages Included in the CDE Software Cluster

SUNWdtwm	SUNWdthez	SUNWdtbas	SUNWdtab
SUNWdtdst	SUNWdtjxt	SUNWdtdmr	SUNWdthed
SUNWdtscm	SUNWpdas	SUNWdtdmn	SUNWdtinc
SUNWdthe	SUNWdtim	SUNWdtdte	SUNWdtmad
SUNWdthev	SUNWdtezt	SUNWdtlog	SUNWdtma
SUNWdticn	SUNWscgui	SUNWdtdem	SUNWdtmaz

Some software clusters contain only one software package.

Solaris OS Software Groups

Software groups are collections of Solaris OS software packages. Each software group includes support for different functions and hardware drivers. The Solaris OS is made up of six software groups:

- Reduced Networking Support software group
- Core System Support software group
- End User Solaris software group
- Developer Solaris software group
- Entire Solaris software group
- Entire Solaris software group plus Original Equipment Manufacturers (OEM) support

Figure 1-3 shows the software groups that compose the Solaris OS.

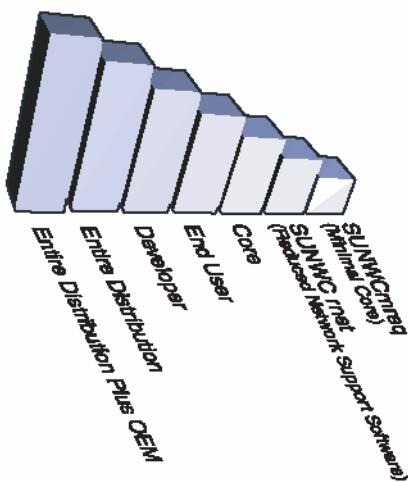


Figure 1-3 Solaris OS Software Groups

Minimal Core Metacluster (SUNWCmreq)

The metacluster SUNWCmreq is a hidden metacluster. It allows you to create a minimal core metacluster by deselecting packages from the core metacluster.

Reduced Network Support Software Group (SUNWCrnet)

This group contains the minimum software that is required to boot and run a Solaris system with limited network service support. The Reduced Networking software group provides a multiuser text-based console and system administration utilities. This software group also enables the system to recognize network interfaces, but does not activate network services.

A system installed with the Reduced Networking software group could, for example, be used as a *thin-client* host in a network.

Core Software Group (SUNWCreq)

The Core software group contains the minimum software required to boot and run the Solaris OS in a minimum configuration, without the support to run many server applications. The Core software group includes a minimum of networking software, including Telnet, File Transfer Protocol (FTP), Network File System (NFS), Network Information Service (NIS) clients, and Domain Name Service (DNS). This software group also includes the drivers required to run the Common Desktop Environment (CDE) but does not include the CDE software. The Core software group also does not include online manual pages.

End User System Support Software Group (SUNWCuser)

The End User System Support software group contains the Core software group and also contains the recommended software for an end user plus the CDE.

Developer System Support Software Group (SUNWCprog)

The Developer System Support software group contains the End User System Support software group. It also contains the libraries, the include files, the online manual pages, and the programming tools for developing software.

Entire Distribution Software Group (SUNWCall)

The Entire Distribution software group contains the Developer System Support software group. It also contains additional software needed for servers. The software that is in the Entire Distribution software group is the entire Solaris OS software release minus OEM support.

Entire Distribution Plus OEM Support Software Group (SUNWCXall)

The Entire Distribution Plus OEM Support software group contains the entire Solaris OS software release. It also contains additional hardware support for OEMs and hardware not on the system at the time of installation. This software group is recommended when you are installing the Solaris OS software on non-Sun servers that use UltraSPARC processors.

To view the names of the cluster configurations, perform the command:

```
# grep METACLUSTER /var/sadm/system/admin/.clustertoc
METACLUSTER=SUNWCXall
METACLUSTER=SUNWCall
METACLUSTER=SUNWCprog
METACLUSTER=SUNWCuser
METACLUSTER=SUNWCreq
METACLUSTER=SUNWCrnet
METACLUSTER=SUNWCmreq
```

To determine which cluster configuration has been installed on the system, perform the command:

```
# cat /var/sadm/system/admin/CLUSTER
CLUSTER=SUNWCXall
```

Installing the Solaris 10 OS From a CD-ROM or DVD

Pre-Installation Information

Consider the following general guidelines while planning an installation:

- Allocate additional disk space for each language that you install.
- Allocate additional space in the /var file system if you plan to have your system support printing or mail.
- Allocate double the amount of physical memory in the /var file system if you plan to use the crash dump feature savecore on your system.
- Allocate additional space in the /export or /export/home file system if you plan to provide a home directory file system for users.
- Allocate space for the Solaris OS software group you want to install.
- Allocate an additional 30 percent more disk space for each file system that you create, and create a minimum number of file systems. This leaves room for upgrades to future software releases.

Note – By default, the Solaris OS installation methods create only the / (root) file system, /export/home, and swap partitions.



- Allocate additional disk space for additional software or third-party software.

Before installing the Solaris OS software on a networked stand-alone system, you must provide the following information:

Host name	Determine a unique, and usually, short name for the networked system. You can use the command <code>uname -n</code> to find the host name on an existing system.
Host Internet Protocol (IP) address	Determine the software address that represents the host address and network address. You can use the <code>ifconfig interface</code> command (for example, <code>ifconfig hme0</code>) to display your current IP address.

Name service type	Determine if the networked system is to be included in one of the following types of name service domains: Lightweight Directory Access Protocol (LDAP), NIS, Network Information Service Plus (NIS+), DNS, or none.
Subnet mask	Determine if the networked system is included in a particular subnet. The subnet mask is stored in the /etc/netmasks file.



Note – A subnet is used to partition network traffic. Segmenting network traffic over many different subnets increases the bandwidth available to each host.

Geographic location and time zone	Determine the specific region where the networked system physically resides.
root password	Determine a password assigned to the root user. Use the root password to gain access to root privileges on the networked system.
Language	Determine the language with which to install the Solaris OS. Use the CD-ROM labeled Solaris 10 Installation SPARC® Platform Edition. The installation software enables the user to choose from a list of languages. Prompts, messages, and other installation information are displayed in the chosen language. The language choices include English, German, Spanish, French, Italian, Japanese, Korean, Swedish, Simplified Chinese, and Traditional Chinese.

As the last step in the pre-installation process, make sure the following Solaris 10 OS CD-ROM set is available:

- Solaris 10 OS Software 1 – This CD is the only bootable CD. From this CD, you can access both the Solaris OS installation graphical user interface (GUI) and the console-based installation.
- Solaris 10 OS Software 2 - This CD contains Solaris OS packages which the software prompts you to install if necessary.
- Solaris 10 OS Software 3 - This CD contains Solaris OS packages which the software prompts you to install if necessary.
- Solaris 10 OS Software 4 - This CD contains Solaris OS packages which the software prompts you to install if necessary and ExtraValue software.
- Solaris 10 OS Languages CD - This CD contains translated message files and other software in languages other than English.

Before performing a software installation, always back up any modifications or data that exist in the previous version of the Solaris OS, and restore them after completing the installation.

Demonstration: Performing an Interactive Installation

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

For this particular installation exercise, install only one of the two systems in your assigned pod. After you finish the installation, you need to retrieve course files that exist on the other system in the pod. Use the `ftp` command from the newly installed system and retrieve the two files from the other system's root directory that have names similar to:

`SA200-S10_A_SunOS_student.tar.gz` and
`vnc_solaris_sun4_en-us_1.2.8r1.3_install.tar.gz`. Position these files on your newly installed system in the root directory and follow this procedure to process each of the files:

1. Uncompress the VNC (Virtual Network Computing) bundle:

```
# gunzip vnc_solaris_sun4_en-us_1.2.8r1.3_install.tar.gz
```

2. Process the VNC bundle:

```
# /tmp/postinstall
```

3. Uncompress the student file bundle:

```
# gunzip SA200-S10_A_SunOS_student.tar.gz
```

4. Process the student file bundle:

```
# /tmp/postinstall
```

 **Note** – It is very important to process one bundle completely before processing the next bundle. It does not matter which one you process first as long as you process one completely (executing the postinstall script), before starting to process the other.

Interactive Installation in the Classroom

In this demonstration, your instructor leads you through an interactive installation of the Solaris 10 OS.

Preparation

The interactive installation demonstration requires a networked, standalone system configured with a 5-Gbyte, or larger, boot disk. Depending on the speed of CD-ROM devices in use, the complete installation process requires approximately two hours.

 **Note** – Examples of the installation sequence presented in this module are from a text-based, DVD installation. Your classroom may have CD-ROM media available for the installation. The installation steps are much the same.

 **Note** – When using the remote web learning lab environment, the installation is performed using the single, Solaris 10 OS DVD.

Boot the system from the Solaris Software 1 CD-ROM or DVD, and install the Solaris 10 OS software. Create a configuration as follows:

- Assign host name, IP address, subnet mask, routing, time zone, and naming service information compatible with the classroom configuration.
- Perform an initial installation, and use the Entire Distribution configuration cluster.

- Select the appropriate boot disk, and manually lay out your file system.
- Create slices for the / (root), swap, /opt, /usr, and /export/home file systems.
- Elect not to install additional software products.
- Set the root password to cangetin.



Demonstration Instructions

The following demonstration uses the DVD to install a local host using the text-based installation. The text you see is exactly the same as the text in the graphical windows when installing with CD-ROMs. Complete the following steps:

1. Insert the Solaris Software 1 CD-ROM or DVD into the CD-ROM drive. If the current version of the Solaris OS is running, log in as root, and bring the system to run level 0.

```
# init 0
```

You can also abort the Solaris OS by pressing the Stop-A key sequence.

2. Boot the system from the CD-ROM or DVD. Ignore error messages, such as cable problem messages, that relate to network interfaces that are not attached.

```
ok boot cdrom -nowin
```

The installation program is loaded into random access memory (RAM), and the installation process begins automatically. However, no changes are made to the disk until you click the Begin Installation button at the end of the installation process.

If a frame buffer is detected and the system has enough RAM, it uses the GUI. If a frame buffer is not detected or there is insufficient RAM, the command-line interface (CLI) is used. The content and sequence of instructions in both are generally the same.

3. When the installation software is finished loading, the system will attempt to determine the locale:

Detecting local, please wait

4. A list of languages appears. Prompts, messages, and other installation information are displayed in the chosen language. English is the default language choice. You can select a different language from the list of available languages.

Select a Language

- 0 English
- 1 French
- 2 German
- 3 Italian
- 4 Japanese
- 5 Korean
- 6 Simplified Chinese
- 7 Spanish
- 8 Swedish
- 9 Traditional Chinese

Please make a choice (0-9), or press h or ? for help:

Respond by making your language selection.

5. The next screen queries your terminal type.

What type of terminal are you using?

- 1) ANSI Standard CRT
- 2) DEC VT52
- 3) DEC VT100
- 4) Heathkit 19
- 5) Lear Siegler ADM31
- 6) PC Console
- 7) Sun Command Tool
- 8) Sun Workstation
- 9) Televideo 910
- 10) Televideo 925
- 11) Wyse Model 50
- 12) X Terminal Emulator (xterms)
- 13) CDE Terminal Emulator (dtterm)
- 14) Other

Type the number of your choice and press Return:

Respond by selecting your terminal type.

The system begins the Solaris Installation Program.

--The Solaris Installation Program

The Solaris installation program is divided into a series of short sections where you'll be prompted to provide information for the installation. At the end of each section, you'll be able to change the selections you've made before continuing.

About navigation...

- The mouse cannot be used
- If your keyboard does not have function keys, or they do not respond, press ESC; the legend at the bottom of the screen will change to show the ESC keys to use for navigation.

F2_Continue F6_Help

Press F2, or the appropriate escape sequence for your terminal type to continue. For the purposes of this installation, the escape sequence for Continue will be Escape and 2 (Esc-2). The Identify This System window appears.



Note – The text that appears in the installation sequence that follows has been edited to fit into this document. For the rest of this installation sequence, the text at the bottom of the window for Esc-2_Continue and Esc-6_Help does not appear.

--Identify This System

On the next screens, you must identify this system as networked or non-networked, and set the default time zone and date/time.

If this system is networked, the software will try to find the information it needs to identify your system; you will be prompted to supply any information it cannot find.

- > To begin identifying this system, press F2.

6. Read the description of the identification process. When you are finished, press Esc-2.
The next window asks you to select Yes or No in response to the question: "Is the system networked?"
7. Select Yes if your system is connected to a network. Use the arrow keys to move between choices, and press Return to select the choice. Use these steps to make selections for the remainder of the demonstration. When you have made your selection, press Esc-2.
If your system has more than one network interface, you are prompted to select each network interface that you want to configure, and select which network interface you want to be your primary interface.
8. Select which network interfaces you want to configure. Press Esc-2 to continue.
9. The Primary Network Interface window appears. You must specify which network adapter you want to be the primary interface, then press Esc-2 to continue.

Note – If your system has multiple network interfaces that you want to configure, you will be prompted to supply information for each interface. The remainder of this installation assumes a single interface.



10. The next screen asks you whether or not to use the Dynamic Host Configuration Protocol (DHCP). Select No to confirm that the system is not using DHCP for network interface information. To continue, press Esc-2.
11. Enter the assigned host name for the system in the Host name field. To continue, press Esc-2.
12. Enter the assigned IP address in the IP address field. To continue, press Esc-2.
13. Select Yes to confirm that the system is part of a subnet. To continue, press Esc-2.
14. For this demonstration, accept the default subnet mask of 255.255.255.0. To continue, press Esc-2.
15. When the next window is displayed, confirm that the system does not use Internet Protocol version 6. To continue, press Esc-2. There will be a short delay.

The Set the Default Route window appears. In this window, you can let the operating system try to find a default route, or you can specify one.

16. Select *Specify one*. To continue, press Esc-2.
17. Enter the default route IP address provided to you by your instructor. To continue, press Esc-2.
18. The *Confirm Information for Your Interface* Window appears.

Confirm Information for eri0

- > Confirm the following information. If it is correct, press F2; to change any information, press F4.

Networked: Yes
Use DHCP: No
Host name: sys70
IP address: 192.168.30.70
System part of a subnet: Yes
Netmask: 255.255.255.0
Enable IPv6: No
Default Route: Specify one
Router IP Address: 192.168.30.30

19. Verify your system configuration. Press Esc-4 to go back and make changes or to correct errors. To continue, press Esc-2.
The next window relates to the use of Kerberos security controls.
20. Select *No* to configure the Solaris 10 OS to use standard UNIX security. To continue, press Esc-2.
A window appears requesting confirmation.
21. Press Esc-2 to confirm the *No* response and to display the next window.
22. Select *None* as your name service. To continue, press Esc-2.
A window appears requesting confirmation.
23. To change the information, press Esc-4. To continue, press Esc-2.
A *TimeZone* window displays.
24. Select the appropriate time zone continent. To continue, press Esc-2.
After the continent has been selected, another window appears in which the specific country can be selected.
25. Select the appropriate time zone country or region. To continue, press Esc-2.
A time zone window appears.
26. Select the appropriate time zone for your area. To continue, press Esc-2.

27. Accept the default date and time, or enter new values. To continue, press Esc-2.
A confirmation window appears.
28. Review the information. To change the information, press Esc-4. To continue, press Esc-2.
29. A window appears in which you must set the password for the root user. Enter your root password in both areas and then press Esc-2 to continue.
A message, System identification is completed, appears and the console window displays additional information while generating a software table of contents, checking the rules.ok file, and executing scripts. The Solaris Interactive Installation window appears.

Solaris Interactive Installation

On the following screens, you can accept the defaults or you can customize how Solaris software will be installed by:

- Selecting the type of Solaris software to install
- Selecting disks to hold software you've selected
- Selecting unbundled products to be installed with Solaris
- Specifying how file systems are laid out on the disks

After completing these tasks, a summary of your selections (called a profile) will be displayed.

There are two ways to install your Solaris software:

- "Standard" installs your system from a standard Solaris Distribution. Selecting "Standard" allows you to choose between initial install and upgrade, if your system is upgradable.
- "Flash" installs your system from one or more Flash Archives.

The Standard method and the Flash method are the two methods available for installing the Solaris 10 OS.

30. Select the Standard method for this demonstration. To continue, press Esc-2.
The Solaris Interactive Installation Loading Install Media window appears briefly to inform you that the suninstall program is loading the software.
31. Choose to have the system eject the CD-ROM/DVD automatically or do it manually. Select to have an automatic ejection of the CD/DVD. To continue, press Esc-2.

32. The Reboot After Installation window enables you to choose between an automatic or a manual reboot. Select to have an automatic reboot. To continue, press Esc-2.

Solaris Interactive Installation

This system is upgradable, so there are two ways to install the Solaris software.

The Upgrade option updates the Solaris software to the new release, saving as many modifications to the previous version of Solaris software as possible. Back up the system before using the Upgrade option.

The Initial option overwrites the system disks with the new version of Solaris software. This option allows you to preserve any existing file systems. Back up any modifications made to the previous version of Solaris software before starting the Initial option.

After you select an option and complete the tasks that follow, a summary of your actions will be displayed.

If you have previously installed a version of the Solaris OS software on the system, the installation program advises you that the system can be upgraded. The upgrade procedure attempts to preserve local modifications to the system whenever possible. An upgrade procedure generally takes two or three times longer than the initial installation procedure because it does file comparisons.

The first thing the upgrade program does is analyze the current Solaris OS files and disk configuration. The upgrade program then calculates the size of replacement packages to determine if the disk partitioning is adequate for the new software. If adequate space is allocated, the program prompts you to customize the software for the upgrade.

The upgrade program attempts to mount all file systems listed in the /etc/vfstab file. If any file system cannot be mounted, the upgrade program reports the failure and then exits.

If there is no need to preserve existing data on the system, press Esc-4 to perform the initial installation. The Initial option destroys the existing file systems as it performs an installation of the Solaris 10 OS.

33. For this demonstration, use the Initial installation method. Press Esc-4 to select the Initial installation method. There is a short wait while the media is being read.

A license window appears.

34. The cursor keys can be used to scroll down the license agreement. To accept the license, press Esc-2.

The Select Geographic Regions window appears. Geographic regions are composed of locales and languages. You can select support for a portion of a region or an entire region. You can also select support for more than one region. An X means support for a region or locale is selected. A slash (/) means the region or locale is partially selected.

35. Make the appropriate selections. To continue, press Esc-2.

The next window allows the selection of the primary system Locale details.

A choice of which locale to use after installation is presented.

36. Select the most appropriate locale.

The next window shows details of extra products that can be installed.

37. There might be extra products on the Solaris Software DVD or the Solaris Software 3 CD-ROM, that can be automatically installed at the end of the Solaris OS installation. Do not select any extra products, and press Esc-2 to continue.

38. Additional products can be installed at the end of the Solaris OS installation from a variety of sources. You can select the software group that most closely fits the specific needs of your system. Notice the recommended or estimated disk file size required to install each of the software groups. These sizes vary based on the system type and kernel architecture.

Select None and press Esc-2 to continue.

39. The Select Software window allows you to choose which software group you want to install. Select Entire Distribution. To continue, press Esc-2.

40. Select the disk or disks on which you are installing the software.

The window displays values that reflect available space on the disk and the suggested minimum space. Recall that the size of the clusters varies and that there are other general considerations for determining disk slices and sizes. If you choose to change your boot drive, the installation program prompts you to verify the change and makes changes to your nonvolatile random access memory (NVRAM) parameters.

41. To continue, press Esc-2.

The initial installation preserves data only on demand.

-
42. To continue, press Esc-2.



Note – If you select F4 to preserve data, the installation program displays a window that enables you to preserve data on a specific partition of the disk. If your system was previously a home directory server, you might want to preserve the /export/home file system.

The installation program can automatically arrange the file system, or you can select disks and slices manually.

43. Press Esc-4 to select Manual Layout.

This window summarizes the current file system and disk layout. The window reflects the overlap partition of the boot drive.

44. Press Esc-4 to select Customize.

The Customize Disk window is a tool you use to reconfigure disk partitions for each disk selected. There are numerous ways to partition slices and to name file systems. Your instructor should inform you of the number of partitions and their sizes for this demonstration.

45. Select the disk slice you want to change. Enter the mount point for the file system that will reside on the slice and the size you want to apply to the slice. Press Return.

The Size (MB) column reflects your changes. The Allocated and Free Space variables change as you configure each slice of the disk. Recommendations and minimum size requirements are displayed at the upper right.

46. When you have finished reconfiguring the disk, press Esc-4.

The Disk Editing Options window enables you to choose how disks are displayed and computed.

47. Make your selections. To continue, press Esc-2.

The Customize Disk Finished window enables you to review and modify your changes.

48. To continue, press Esc-2.

The File System and Disk Layout summary window is your final confirmation of what the disk layout looks like. Check that your disk layout is the same as the instructor's directions.

If you have made any errors, this window is named Warning and details which file systems are incorrect.

49. To continue, press Esc-2.

50. In the Mount Remote File Systems Window, you can Press Esc-4 to open a window that prompts you to enter a server name, an IP address, and a mount point to a location where you have stored data. To continue, press Esc-2.

The Profile window displays the installation choices you made in previous windows.

Profile

The information shown below is your profile for installing Solaris software.

It reflects the choices you've made on previous screens.

=====

Installation Option: Initial

Boot Device: c0t0d0

Client Services: None

Locales: U.S.A. (UTF-8)

U.S.A. (en_US.ISO8859-1)

System Locale: U.S.A. (en_US.ISO8859-1) (en_US.IS

Software: Solaris 10, Entire Distribution plus OEM su

File System and Disk Layout:	/	c0t0d0s0	500 MB
	swap	c0t0d0s1	512 MB
	/var	c0t0d0s3	512 MB
	/opt	c0t0d0s5	500 MB
	/usr	c0t0d0s6	7000 MB

Esc-2_Begin Installation F4_Change F5_Exit F6_Help

This is the last window that enables you to change the options you have selected.

51. After making your selection, press Esc-2 to begin the installation process.

The system begins the installation by writing a Volume Table of Contents (VTOC) on the disk or disks selected and creating file systems.

Preparing system for Solaris install

Configuring disk (c0t0d0)

- Creating Solaris disk label (VTOC)

Creating and checking UFS file systems

- Creating / (c0t0d0s0)

- Creating /var (c0t0d0s3)

- Creating /opt (c0t0d0s5)
- Creating /usr (c0t0d0s6)
- Creating /export/home (c0t0d0s7)

Beginning Solaris Installation

The Solaris Initial Install window displays the software cluster currently being installed. The window indicates how many megabytes of the cluster have been installed and how many megabytes of the cluster remain to be installed.

If you are installing with a DVD, the installation runs through until completion without having to change the DVD.

If you are installing with CD-ROM media, at the end of the installation of the Solaris Software 1 CD-ROM, the system reboots.

During the next phase of the installation, if you are installing with a GUI, the Common Desktop Environment (CDE) starts, and the remainder of the windows are displayed. Otherwise, the installation continues in a text-based mode.

After the system reboots, you are prompted for the next media choice.

52. Select the CD option. Insert the Solaris Software 2 CD-ROM. To continue, click OK or press Enter.

The Launching Installer window displays.

A list of the items to install is displayed. To continue, select Install Now.

While the system installs the packages from the second CD-ROM, the progress bar displays the progress of the installation.

53. If you are working in the GUI, you can click Details to display the log file of the installation. The log file contains information about the packages installed.
54. Click Exit when you are finished reviewing the log file.

A window appears in which you specify the media to use.

55. The installation continues in the same manner with prompts appearing for loading and unloading the remaining CDs. Follow the prompts until the final phase of the installation process.

With the text-based installation, a final message before rebooting appears:

Pausing for 90 seconds at the "Reboot" screen. The wizard will continue to the next step unless you select "Pause". Enter 'p' to pause. Enter 'c' to continue. [c] c

56. Press Enter to continue with the reboot.

If you are running the GUI, a window appears even if automatic reboot has been selected. An automatic reboot occurs unless the Pause button is selected in a dialog box that displays.

Remove the CD-ROM and click Reboot Now to continue.

The host reboots. A message appears during the boot asking whether or not you need to override the NFS domain:

This system is configured with NFS version 4, which uses a domain name that is automatically derived from the system's name services. The derived domain name is sufficient for most configurations. In a few cases, mounts that cross different domains might cause files to be owned by "nobody" due to the lack of a common domain name.

Do you need to override the system's default NFS version 4 domain name (yes/no) ? [no] :

57. Respond with No to continue.

 **Note** – More detail about NFS is discussed in SA-202-S10, *Advanced System Administration for the Solaris 10 OS*.

The installation asks two more questions about configuring the energy saving option:

=====

This system is configured to conserve energy.

=====

After 30 minutes of idle time on this system, your system state will automatically be saved to disk, and the system will power-off.

Later, when you want to use the system again, and you turn the power back on, your system will be restored to its previous state, including all the programs that you were running.

Do you want this automatic power-saving shutdown?
(If this system is used as a server, answer n) [y,n,?] **n**

Autoshutdown has been disabled.

Do you want the system to ask about this again, when you next reboot?
(This gives you the chance to try it before deciding whether
to keep it.) [y,n,?] **n**

The "Power Management" chapter in the "Solaris Common Desktop Environment: User Guide" describes more about how to change and set workstation energy-saving features.

58. After the system completes the reboot process, log in and verify that the system is operational. You can review additional log file information after the system has rebooted by looking at the `/var/sadm/install_data/install_log` file.

You have now completed the installation demonstration.

Module 2

Introducing the Solaris™ 10 OS Directory Hierarchy

Objectives

Upon completing this module, you should be able to:

- Describe / (root) subdirectories
- Describe file components
- Describe file types
- Use hard links

The course map in Figure 2-1 shows how this module fits into the current instructional goal.

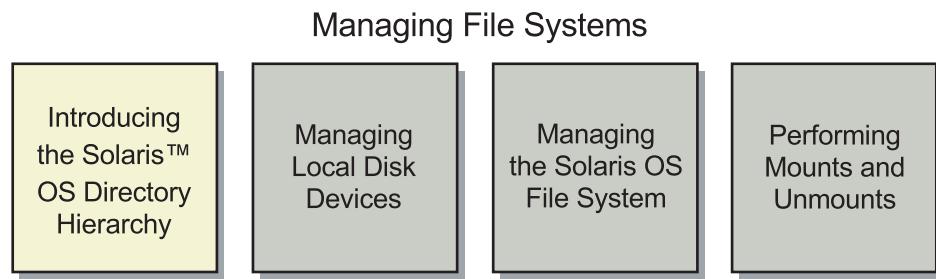


Figure 2-1 Course Map

Introducing / (root) Subdirectories

The directory hierarchy of the Solaris™ Operating System (Solaris OS) is organized for administrative convenience. Branches within this directory tree segregate directories that are used for different purposes. For example, directories exist to hold files that are private to the local system, files to share with other systems, and home directories.

Logically, all directories fall below the / (root) directory. Physically, however, directories can be located on a single file system or divided among multiple file systems. Every Solaris OS must have a root file system but can also have other file systems attached at points within the directory hierarchy. Most file systems are structures created on disk slices that contain or hold files and directories. Some file systems reside in areas of virtual memory and are managed by the Solaris kernel.

Note – Refer to `man -s5 filesystem` for information on file system organization.



Introducing Important System Directories

The Solaris OS consists of a hierarchy of critical system directories and files that are necessary for the operating system to function properly. Table 2-1 lists some of the critical, disk-based, system directories and subdirectories that are found in the Solaris OS.

Table 2-1 Critical Directories

/	The root of the overall file system namespace.
/bin	A symbolic link to the /usr/bin directory. It is the directory location for the binary files of standard system commands.
/dev	The primary directory for logical device names. The contents of this directory are symbolic links that point to device files in the /devices directory.
/etc	The directory that holds host-specific configuration files and databases for system administration.
/export	The default directory for commonly shared file systems, such as users' home directories, application software, or other shared file systems.
/home	The default directory or mount point for a user's home directory.
/kernel	The directory of platform-independent loadable kernel modules that are required as part of the boot process.
/lib	The contents of this directory are shared executable files and Service Management Facility executables.
/mnt	A convenient, temporary mount point for file systems.
/opt	The default directory or mount point for add-on application packages.
/platform	The directory of platform-dependent loadable kernel modules.
/sbin	The single-user bin directory that contains essential executables that are used during the booting process and in manual system-failure recovery.
/usr	The directory that contains programs, scripts, and libraries that are used by all system users.

Table 2-1 Critical Directories (Continued)

/var	<p>The directory for varying files, which usually includes temporary, logging, or status files.</p> <p>Following the introduction of the Service Management Facility and Zones, in the Solaris 10 OS, the /var directory hierarchy is more heavily used than in previous releases.</p> <p>It is important that the /var directory has sufficient disk space available to store software package information, log files, spool files, and so on.</p>
------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Introducing Important In-Memory System Directories

Table 2-2 lists of some of the important in-memory system directories and subdirectories that are found in the Solaris 10 OS.

Table 2-2 In-Memory System Directories

/dev/fd	The directory that contains special files relating to current file-descriptors in use by the system.
/devices	The primary directory for physical device names.
/etc/mnttab	A memory-based file, in its own file system, that contains details of current file system mounts.
/etc/svc/volatile	The directory that contains log files and reference files relating to the current state of system services.
/proc	The directory that stores current process-related information. Every process has its own set of subdirectories below the /proc directory.

Table 2-2 In-Memory System Directories (Continued)

/system/contract	<p>CTFS (the contract file system) is the interface for creating, controlling, and observing contracts. A contract enhances the relationship between a process and the system resources it depends on by providing richer error reporting and (optionally) a means of delaying the removal of a resource.</p> <p>The service management facility (SMF) uses process contracts to track the processes which compose a service, so that a failure in a part of a multi-process service can be identified as a failure of that service.</p> <p>The contract file system supports all the SMF services.</p>
/system/object	The OBJFS (object) file system describes the state of all modules currently loaded by the kernel. This file system is used by debuggers to access information about kernel symbols without having to access the kernel directly. It is used primarily for Dtrace activity.
/tmp	The directory for temporary files. This directory is cleared during the boot sequence.
/var/run	The directory that contains lock files, special files, and reference files for a variety of system processes and services.



Note – These in-memory directories are maintained by the kernel and system services. Users should never attempt to manually create, alter, or remove files from these directories.

The following tables list primary subdirectories under key directories.

Table 2-3 Primary Subdirectories Under the /dev Directory

Subdirectory	Description
/dev/dsk	Block disk devices

Table 2-3 Primary Subdirectories Under the /dev Directory (Continued)

Subdirectory	Description
/dev/fd	File descriptors
/dev/md	Logical volume management metadisk devices
/dev/pts	Pseudo terminal devices
/dev/rdsk	Raw disk devices
/dev/rmt	Raw magnetic tape devices
/dev/term	Serial devices

Table 2-4 Primary Subdirectories Under the /etc Directory

Subdirectory	Description
/etc/acct	Configuration information for the accounting system
/etc/cron.d	Configuration information for the cron utility
/etc/default	Default information for various programs
/etc/inet	Configuration files for network services
/etc/init.d	Scripts for starting and stopping services
/etc/lib	Dynamic linking libraries needed when the /usr file system is not available
/etc/lp	Configuration information for the printer subsystem
/etc/mail	Configuration information for the mail subsystem
/etc/nfs	Configuration file for NFS server logging
/etc/opt	Configuration information for optional packages
/etc/rc#.d	Legacy scripts that are executed when entering or leaving a specific run level
/etc/security	Control files for Role Based Access Control and security privileges
/etc/skel	Default shell initialization files for new user accounts

Table 2-4 Primary Subdirectories Under the /etc Directory (Continued)

Subdirectory	Description
/etc/svc	The Service Management Facility database and log files
/etc/zones	Initialization and reference files for the Solaris 10 OS Zones facility

Table 2-5 Contents of the /usr Directory

Subdirectory	Description
/usr/bin	Standard system commands
/usr/ccs	C-compilation programs and libraries
/usr/demo	Demonstration programs and data
/usr/dt	Directory or mount point for Common Desktop Environment (CDE) software
/usr/include	Header files (for C programs, and so on)
/usr/jdk	Directories that contain Java™ technology programs and libraries
/usr/kernel	Platform-independent loadable kernel modules that are not generally required during the boot process
/usr/lib	Architecture-dependent databases, various program libraries, and binaries that are not invoked directly by the user
/usr/opt	Configuration information for optional packages
/usr/sbin	System administration commands
/usr/spool	Symbolic link to the /var/spool directory

Table 2-6 Primary Subdirectories Under the /var Directory

Subdirectory	Description
/var/adm	Log files (for syslog, system accounting, and so on).

Table 2-6 Primary Subdirectories Under the /var Directory (Continued)

Subdirectory	Description
/var/crash	For storing crash dump files following a catastrophic system failure. Files from this directory can be analyzed by Help Desk staff to determine the cause of the system crash.
/var/spool	Spoooled files (for mail, print services, and so on).
/var/svc	Service Management Facility control files and logs.
/var/tmp	Long-term storage of temporary files across a system reboot, as an alternative to the /tmp directory.

Introducing File Components

All files in the Solaris OS make use of a file name and a record called an inode. Most files also make use of data blocks. In general, a file name is associated with an inode, and an inode provides access to data blocks.

Figure 2-2 shows the relationship between the file components.

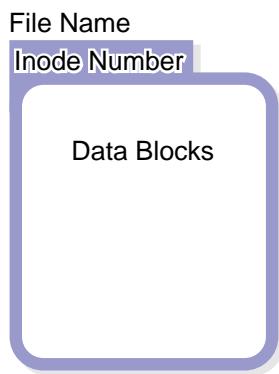


Figure 2-2 File Names, Inodes, and Data Blocks

File Names

File names are the objects most often used to access and manipulate files. A file must have a name that is associated with an inode.

Inodes

Inodes are the objects the Solaris OS uses to record information about a file. In general, inodes contain two parts. First, inodes contain information about the file, including its owner, its permissions, and its size. Second, inodes contain pointers to data blocks associated with the file content.

Inodes are numbered, and each file system contains its own list of inodes. When a new file system is created, a complete list of new inodes is also created in that file system.

Data Blocks

Data blocks are units of disk space that are used to store data. Regular files, directories, and symbolic links make use of data blocks. Device files do not hold data.

Identifying File Types

The Solaris OS supports a standard set of file types that are found in nearly all UNIX-based operating systems. In general, files provide a means of storing data, activating devices, or allowing inter-process communication. Of the different types of files that exist in the Solaris OS, there are four main file types:

- Regular or ordinary files
- Directories
- Symbolic links
- Device files

Regular files, directories, and symbolic links all store one or more types of data. Device files do not store data. Instead, device files provide access to devices.

Use the `ls` command to distinguish different file types from one another. The character in the first column of information that the `ls -l` command displays indicates the file type.

The following examples, taken from a Sun Ultra™ 5 workstation, show partial listings of directories that contain a variety of different file types:

```
# cd /etc
# ls -l
total 573
drwxr-xr-x  2 adm      adm          512 Sep 19 17:21 acct
lrwxrwxrwx  1 root    root          14 Sep 19 16:00 aliases ->
./mail/aliases
drwxr-xr-x  7 root    bin          512 Sep 19 17:55 apache
drwxr-xr-x  2 root    other         512 Sep 19 16:59 apoc
-rw-r--r--  1 root    bin          194 Sep 19 15:55 auto_home
(output truncated)

# cd /devices/pci@1f,0/pci@1,1/ide@3
# ls -l
total 4
drwxr-xr-x  2 root    sys          512 Sep 19 20:13 dad@0,0
brw-r----- 1 root    sys          136,   8 Sep 23 08:35 dad@0,0:a
crw-r----- 1 root    sys          136,   8 Sep 23 12:51 dad@0,0:a,raw
(output truncated)
```

The character in the first column identifies each file type, as follows:

- Regular files
- d Directories
- l Symbolic links
- b Block-special device files
- c Character-special device files

Regular Files

Perhaps the most common file types found in the Solaris OS are regular files, which enable the user to store many different types of data. Regular files can hold American Standard Code for Information Interchange (ASCII) text or binary data, including image data, database data, application-related data, and more.

There are many ways to create regular files. For example, a user could use the vi editor to create an ASCII text file, or a user could use a compiler to create a file that contains binary data. As another example, a user could use the touch command with a non-existent file name to create a new, empty, regular file.

Figure 2-3 shows a regular file called `file1`. As illustrated, the name `file1` is associated with inode number 1282. The data blocks associated with `file1` can hold one of many types of data, and the file could have been created in one of many different ways.

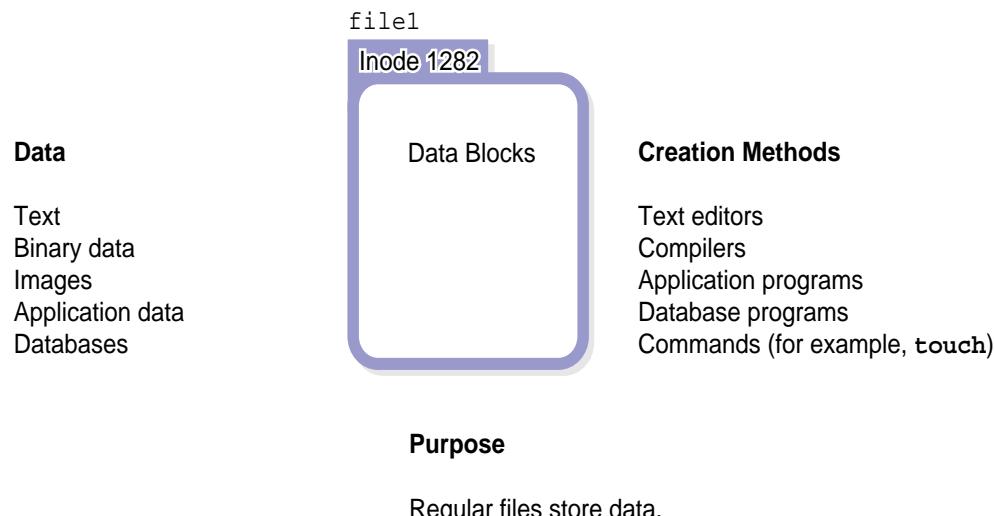


Figure 2-3 Regular Files

Directories

Directories store information that associates file names with inode numbers. Unlike regular files, which can hold many different types of data, directories only hold file name-to-inode associations.

A directory contains entries for files of all types that are logically found within that directory.

Figure 2-4 shows information about a directory called `dir1`. As illustrated in the figure, the name `dir1` is associated with inode number 4221. The data blocks associated with the `dir1` directory hold a list of file names and their associated inode numbers. The `mkdir` command is one way to create new directories.

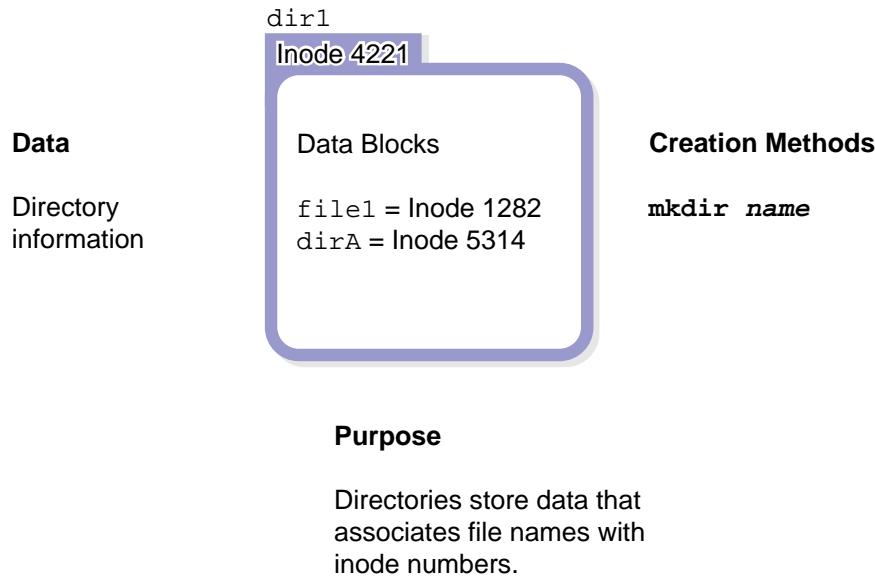


Figure 2-4 Directories

Think of the information that directories hold as a list. Each entry in this list accounts for one file name. If the file called `file1` and the directory called `dirA` were logically located in the directory called `dir1`, then the `dir1` directory would contain an entry that associates the name `file1` with inode number 1282 and an entry that associates the name `dirA` with inode number 5314.

Symbolic Links

A symbolic link is a file that points to another file. Like directories, which contain only directory information, symbolic links contain only one type of data.

A symbolic link contains the path name of the file to which it points. Because symbolic links use path names to point to other files, they can point to files in other file systems.

The size of a symbolic link always matches the number of characters in the path name it contains.

In the following example, the symbolic link called `/bin` points to the directory `./usr/bin`. The size of the symbolic link is 9 bytes because the path name `./usr/bin` contains nine characters.

```
# cd /
# ls -l bin
lrwxrwxrwx 1 root      root          9 Sep 19 15:41 bin -> ./usr/bin
```

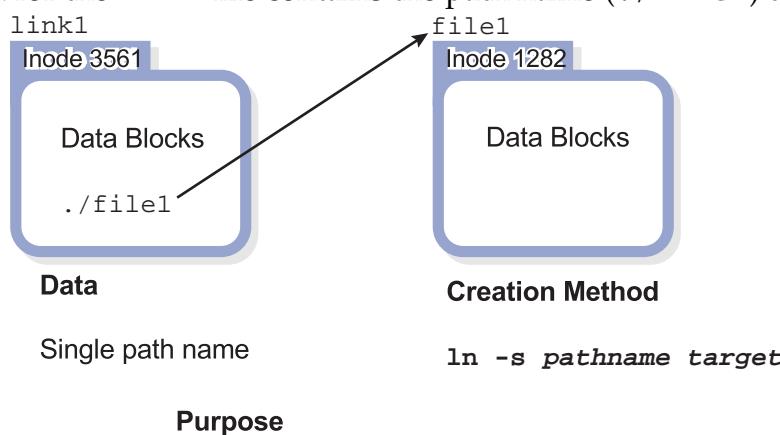
Symbolic links can point to regular files, directories, other symbolic links, and device files. They can use absolute or relative path names.

The `ln` command with the `-s` option creates a symbolic link.

```
# ln -s file1 link1
```

Symbolic links direct read and write operations to the file to which they point. The preceding command shows how using `link1` as a command-line argument causes the `ln` command to refer to the file called `file1`.

Figure 2-5 shows a symbolic link file called `link1`. As shown in the following figure, the `link1` file is associated with inode number 3561. The data block for the `link1` file contains the path name (`./file1`) to `file1`.



Symbolic links refer to other file names.
A symbolic link contains the path name
of the file to which it points.

Figure 2-5 Symbolic Links

Device Files

A device file provides access to a device. Unlike regular files, directories, and symbolic links, device files do not use data blocks. Instead, the inode information of device files holds numbers that refer to devices. Use the `ls -l` command to display these numbers.

For example, a long listing of a regular file shows the file's size in the fifth field of output.

```
# cd /etc
# ls -al |more
total 599
drwxr-xr-x  77 root      sys        4096 Sep 23 08:36 .
drwxr-xr-x  26 root      root       1024 Sep 23 08:40 ..
-rw-r--r--   1 root      root      2236 Sep 23 08:36 .cpr_config
drwxr-xr-x  3 root      bin       512  Sep 19 16:39 .java
-rw-r--r--   1 root      sys       524  Sep 19 15:41 .login
-rw-r--r--   1 root      other     18   Sep 19 16:30 .sysidconfig.apps
-rw-r--r--   1 root      other     284  Sep 19 16:00 .sysIDtool.state
(output truncated)
```

A long listing of a device file shows two numbers, separated by a comma, where the file size details would normally have been displayed. These two numbers are called major and minor device numbers. In the following example, the device file `dad@0,0:a` refers to major device number 136 and minor device number 8.

```
# cd /devices/pci@1f,0/pci@1,1/ide@3
# ls -l dad@0*
total 4
drwxr-xr-x  2 root      sys        512 Sep 19 20:13 dad@0,0
brw-r----  1 root      sys      136,  8 Sep 23 08:35 dad@0,0:a
crw-r----  1 root      sys      136,  8 Sep 23 12:51 dad@0,0:a,raw
brw-r----  1 root      sys      136,  9 Sep 23 08:35 dad@0,0:b
crw-r----  1 root      sys      136,  9 Sep 23 12:51 dad@0,0:b,raw
brw-r----  1 root      sys      136, 10 Sep 23 12:51 dad@0,0:c
crw-r----  1 root      sys      136, 10 Sep 23 12:51 dad@0,0:c,raw
(output truncated)
```

A major device number identifies the specific device driver required to access a device. A minor device number identifies the specific unit of the type that the device driver controls.

The device file `dad@0,0:a`, shown in Figure 2-6, occupies inode number 90681. The inode contains the major and minor device numbers that refer to a specific device, in this case, a slice on a disk.

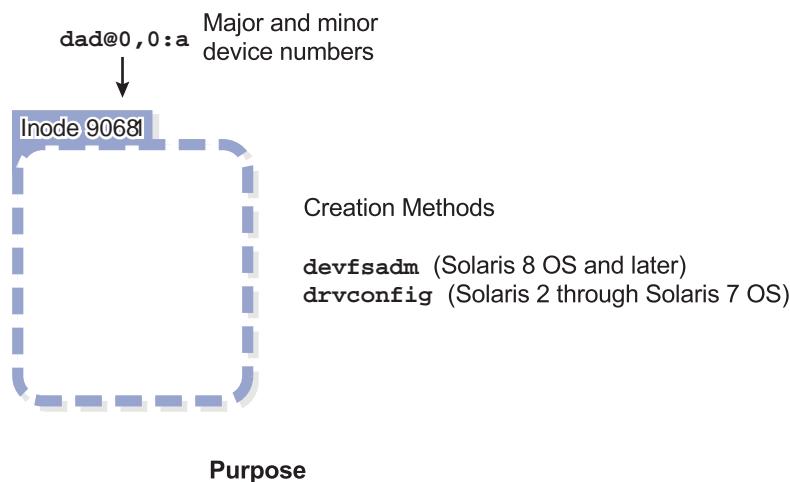


Figure 2-6 Device Files

In general, a reconfiguration boot creates device files and symbolic links to the device files automatically. In the Solaris OS, you can use the `devfsadm` command to create new device files.

A direct relationship exists between the device file and the device it controls. The major and minor device numbers contained in the inode establish this relationship.

Figure 2-7 shows the relationship between the device file `dad@0,0:a` and the disk device it controls. The inode information for `dad@0,0:a` contains major number 136 and minor number 8. Major number 136 identifies the dad device driver. The dad device driver controls integrated device electronics (IDE) disk drives. Minor number 8, in this case, identifies Slice 0.

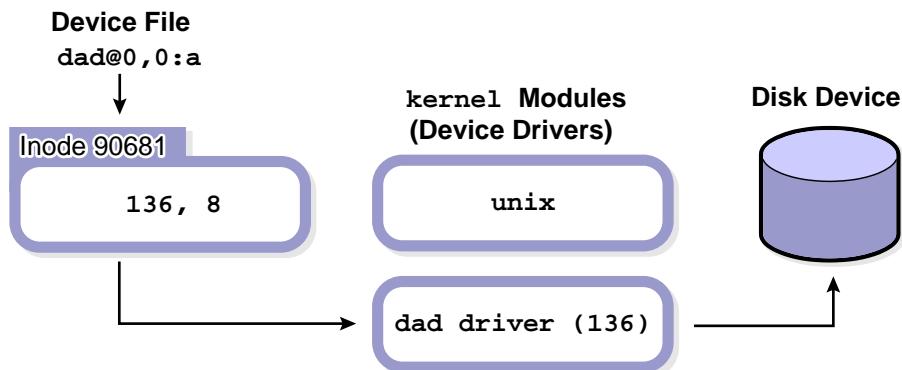


Figure 2-7 Device File Example

It is possible to verify the device driver is available as a kernel module:

```
# modinfo -w | grep -w dad
21 122e118 7b48 136 1 dad (DAD Disk Driver 1.86)
# modinfo -c | grep -w dad
21                               LOADED/INSTALLED
1 dad
```

Device files fall into two categories: character-special devices and block-special devices. Character-special devices are also called character or raw devices. Block-special devices are often called block devices. Device files in these two categories interact with devices differently.

Character-Special Device Files

The file type “c” identifies character-special device files. Data is accessed as a data stream.

The following example shows a character-special device file.

```
crw-r----- 1 root      sys      136,   8 Sep 23 12:51 dad@0,0:a,raw
```

Block-Special Device Files

The file type "b" identifies block-special device files. For disk devices, block-special device files call for I/O operations based on a defined block size. The block size depends on the particular device.

The following example shows a block-special device file.

```
brw-r----- 1 root      sys        136,   8 Sep 23 08:35 dad@0,0:a
```

Data transferred between a process and a block-special device is first stored in a kernel-managed memory-based cache. This provides better performance when data is being accessed from block-special devices in a repetitive manner. Also, block devices allow random seeks to be performed, and character devices do not.

Using Hard Links

This section defines hard links and describes how to use them.

Introducing Hard Links

A hard link is the association between a file name and an inode. A hard link is not a separate type of file. Every type of file uses at least one hard link. Every entry in a directory constitutes a hard link. Think of every file name as a hard link to an inode. When you create a file, using the `touch` command, for example, a new directory entry is created that links the file name you specified with a particular inode. In this way, creating a new file creates a hard link.

In Figure 2-8, the file called `file1` is listed in the directory `dir1`. In `dir1`, the name `file1` is associated with inode number 1282. The hard link is the association between `file1` and inode number 1282.

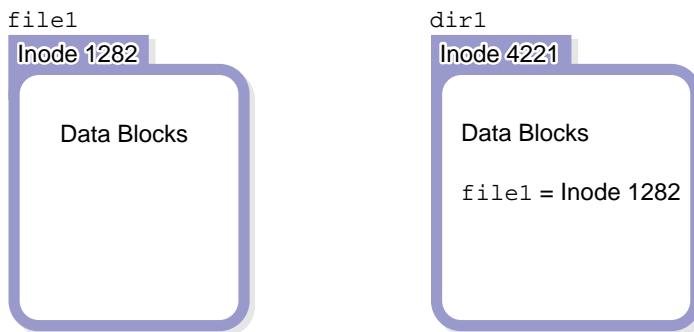


Figure 2-8 Hard Link

Information in each inode keeps count of the number of file names associated with it. This is called a link count. In the output from the `ls -l` command, the link count appears between the column of file permissions and the column identifying the owner. In the following example, the file called `file1` uses one hard link.

```
# cd dir1
# touch file1
# ls -l
total 0
-rw-r--r--    1 root      root          0 Sep 23 13:19 file1
```

Creating New Hard Links

A new hard link for a file name increments the link count in the associated inode.

In the following example, inode 1282 now has two hard links, one for `file1` and the other for `file2`. The `ls -li` command lists the inode number in the left-most column. The `find -inum` command locates files and directories that have the same inode numbers.

```
# ln file1 file2
# ls -l
total 0
-rw-r--r--  2 root      root           0 Sep 23 13:19 file1
-rw-r--r--  2 root      root           0 Sep 23 13:19 file2
# ls -li
total 0
    1282 -rw-r--r--  2 root      root           0 Sep 23 13:19 file1
    1282 -rw-r--r--  2 root      root           0 Sep 23 13:19 file2
# find . -inum 1282
./file1
./file2
```

The `ln` command creates new hard links to regular files.

For example, the `ln file1 file2` command creates a new directory entry called `file2`. The `file2` file is associated with the same inode that is associated with `file1`.

Figure 2-9 shows the result of the `ln` command. Two file names are associated with inode number 1282. Unlike symbolic links, hard links cannot span file systems.

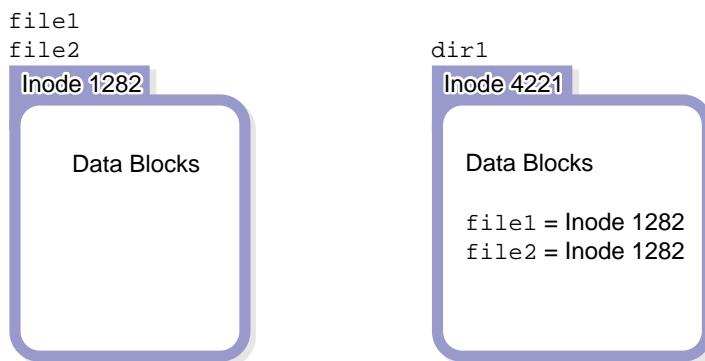


Figure 2-9 File Names Associated With an Inode Number

Removing Hard Links

Deleting one of the files has no effect on the other file. The link count decrements accordingly.

The following example shows how deleting `file1` from the previous example has no effect on `file2`.

```
# rm file1
# ls -li
total 0
1282 -rw-r--r--    1 root      root          0 Sep 23 13:19 file2
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Identifying File Types (Level 1)

In this exercise, you complete the following tasks:

- Navigate within the directory hierarchy
- Identify different types of files

Preparation

Refer to the lecture notes as necessary to perform the following steps and answer the following questions.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Identify the first symbolic link listed in the / (root) directory. Record the symbolic link's size and the name of the file it references. Identify the types of files found in the /dev/dsk directory and the types of files that the symbolic links reference, if any. Identify the types of files found in the /dev/pts directory and the types of files that the symbolic links reference, if any.

(Steps 1–5 in Level 2 lab)

- Identify the types of files found in the /etc/init.d directory. Record the inode number and link count for the volmgt file. Use the find command to locate all other files below the /etc directory that use the same inode as volmgt.
(Steps 6–8 in Level 2 lab)
- Create a directory called /testdir. In this directory, create a file and a symbolic link that points to the file. Determine if the two files use the same or a different inode.
Create a directory called newdir within the /testdir directory. Identify the inode it uses, its link count, and the name of any other file that uses the same inode as the newdir directory.
Create another directory below the newdir directory. Determine how the link count for the newdir directory changes, and find any new file that uses the same inode as the newdir directory.
(Steps 9–14 in Level 2 lab)

Exercise: Identifying File Types (Level 2)

In this exercise, you complete the following tasks:

- Navigate within the directory hierarchy
- Identify different types of files

Preparation

Refer to the lecture notes as necessary to perform the following steps and answer the following questions.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

Complete the following tasks:

- Identify the first symbolic link listed in the / (root) directory. Record the symbolic link's size and the name of the file it references. Identify the types of files found in the /dev/dsk directory and the types of files that the symbolic links reference, if any. Identify the types of files found in the /dev/pts directory and the types of files that the symbolic links reference, if any.
- Identify the types of files found in the /etc/init.d directory. Record the inode number and link count for the volmgt file. Use the find command to locate all other files below the /etc directory that use the same inode as volmgt.
- Create a directory called /testdir. In this directory, create a file and a symbolic link that points to the file. Determine if the two files use the same or a different inode.

Create a directory called newdir within the /testdir directory. Identify the inode it uses, its link count, and the name of any other file that uses the same inode as the newdir directory.

Create another directory below the newdir directory. Determine how the link count for the newdir directory changes, and find any new file that uses the same inode as the newdir directory.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. In the / (root) directory, perform a long listing and record the name of the first symbolic link listed.
2. What is the size in bytes of the link you found in Step 1? How many characters are there in the name of the file to which this link points?
3. Change to the /dev/dsk directory. Record the file types that you find in this directory.
4. Use the appropriate options for the ls command to display information for the files that are referenced by the files in the /dev/dsk directory. Record the file types reported.
5. Change to the /dev/pts directory, and use the same commands you used in Steps 3 and 4 for the /dev/dsk directory. Record the file types you find.
6. Change to the /etc/init.d directory, and identify the type of file in this directory.
7. How many hard links are associated with the /etc/init.d/volmgt file? What is the inode number associated with this file?
8. Find the number of files in the /etc directory, or below, that have the same inode number as that used by the /etc/init.d/volmgt file.
9. Create a new directory called /testdir. Create a file in this directory called file1. Create a symbolic link called link1 that points to file1.
10. List file1 and the link1 symbolic link. Do these files use the same or different inodes?
11. In the /testdir directory, create a new directory called newdir. What is the number of hard links associated with the newdir directory? What is the inode number associated with the newdir directory?
12. List all files, including hidden files, that exist in the newdir directory. Which of these files uses the same inode as the newdir directory?
13. Create a new directory called dir2 below the newdir directory. What happens to the link count for the newdir directory?
14. Use the ls command with appropriate options to find the new file name that uses the same inode as the newdir directory. Record the name of the new file.

Exercise: Identifying File Types (Level 3)

In this exercise, you complete the following tasks:

- Navigate within the directory hierarchy
- Identify different types of files

Preparation

Refer to the lecture notes as necessary to perform the following steps and answer the following questions.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Identify the first symbolic link listed in the / (root) directory. Record the symbolic link's size and the name of the file it references. Identify the types of files found in the /dev/dsk directory and the types of files that the symbolic links reference, if any. Identify the types of files found in the /dev/pts directory and the types of files that the symbolic links reference, if any.
- Identify the types of files found in the /etc/init.d directory. Record the inode number and link count for the volmgt file. Use the find command to locate all other files below the /etc directory that use the same inode as volmgt.
- Create a directory called /testdir. In this directory, create a file and a symbolic link that points to the file. Determine if the two files use the same or a different inode.

Exercise: Identifying File Types (Level 3)

Create a directory called `newdir` within the `/testdir` directory.
Identify the inode it uses, its link count, and the name of any other file that uses the same inode as the `newdir` directory.

Create another directory below the `newdir` directory. Determine how the link count for the `newdir` directory changes, and find any new file that uses the same inode as the `newdir` directory.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. In the / (root) directory, perform a long listing, and record the name of the first symbolic link listed.

```
# cd /
# ls -l
```

The /bin symbolic link should be the first link listed in the / (root) directory.

2. What is the size in bytes of the link you found in Step 1? How many characters are there in the name of the file to which this link points?

The /bin symbolic link contains 9 bytes of data and points to ./usr/bin.

3. Change to the /dev/dsk directory. Record the file types that you find in this directory.

```
# cd /dev/dsk
# ls -l
```

The /dev/dsk directory contains symbolic links.

4. Use the appropriate options of the ls command to display information for the files referenced by the files in the /dev/dsk directory. Record the file types reported.

```
# ls -lL
```

The symbolic links in the /dev/dsk directory point to block-special device files.

5. Change to the /dev/pts directory, and use the same commands you used in Steps 3 and 4 for the /dev/dsk directory. Record the file types you find.

```
# cd /dev/pts
# ls -l
# ls -lL
```

The /dev/pts directory contains symbolic links.

The symbolic links in the /dev/pts directory point to character-special device files.

6. Change to the /etc/init.d directory, and identify the type of file in this directory.

```
# cd /etc/init.d ; ls -l
```

The /etc/init.d directory contains regular files.

Exercise: Identifying File Types (Level 3)

7. How many hard links are associated with the /etc/init.d/volmgt file? What is the inode number associated with this file?

```
# ls -li volmgt
```

The /etc/init.d/volmgt file has six hard links associated with it. The inode number varies among different systems.

8. Find the number of files in the /etc directory or below that have the same inode number as that used by the /etc/init.d/volmgt file. In this example, the inode number is 21449.

```
# ls -i /etc/init.d/volmgt
```

21449	-rwxr--r--	6	root	sys	473	Sep	3	15:37	volmgt
-------	------------	---	------	-----	-----	-----	---	-------	--------

```
# find /etc -inum 21449 -exec ls -i {} \;
```

Six files, including /etc/init.d/volmgt, use the same inode number. They are:

21449	/etc/init.d/volmgt
21449	/etc/rc0.d/K05volmgt
21449	/etc/rc1.d/K05volmgt
21449	/etc/rc2.d/K05volmgt
21449	/etc/rc3.d/S81volmgt
21449	/etc/rcS.d/K05volmgt

9. Create a new directory called /testdir. Create a file in this directory called file1. Create a symbolic link called link1 that points to file1.

```
# mkdir /testdir
# cd /testdir
# touch file1
# ln -s file1 link1
```

10. List file1 and the link1 symbolic link. Do these files use the same or different inodes?

```
# ls -li
```

These two files use two different inodes.

11. In the /testdir directory, create a new directory called newdir. What is the number of hard links associated with the newdir directory? What is the inode number associated with the newdir directory?

```
# mkdir newdir
# ls -ldi newdir
```

The link count for the newdir directory is two. The inode number varies among different systems.

12. List all files, including hidden files, that exist in the newdir directory. Which of these files uses the same inode as the newdir directory?

```
# ls -lia newdir
```

The file called dot (.) uses the same inode as the newdir directory.

13. Create a new directory called dir2 below the newdir directory. What happens to the link count for the newdir directory?

```
# mkdir newdir/dir2
```

```
# ls -ldi newdir
```

The link count increases from two to three.

14. Use the ls command with appropriate options to find the new file name that uses the same inode as the newdir directory. Record the name of the new file.

```
# ls -laRi newdir
```

The newdir/dir2/.. file uses the same inode as the newdir directory.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 3

Managing Local Disk Devices

Objectives

Upon completion of this module, you should be able to:

- Describe the basic architecture of a disk
- Describe the naming conventions for devices
- List devices
- Reconfigure devices
- Perform hard disk partitioning
- Manage disk labels
- Describe the Solaris Management Console
- Partition a disk by using the Solaris Management Console

The course map in Figure 3-1 shows how this module fits into the current instructional goal.

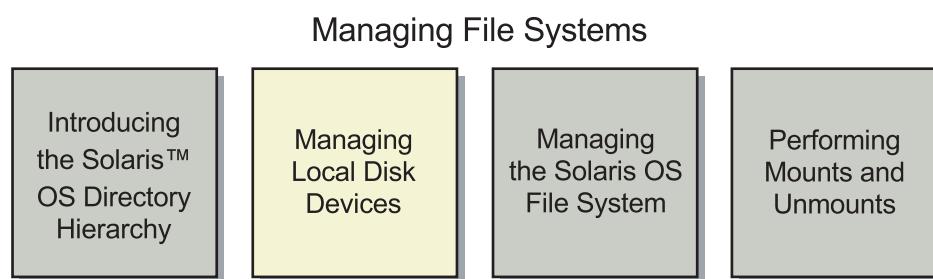


Figure 3-1 Course Map

Introducing the Basic Architecture of a Disk

A disk device has physical components and logical components. The physical components include disk platters and read/write heads. The logical components include disk slices, cylinders, tracks, and sectors.

Physical Disk Structure

A disk is physically composed of a series of flat, magnetically coated platters that are stacked on a spindle. The spindle turns while the read/write heads move as a single unit radially, reading and writing data on the platters.

Figure 3-2 identifies the parts of a disk.

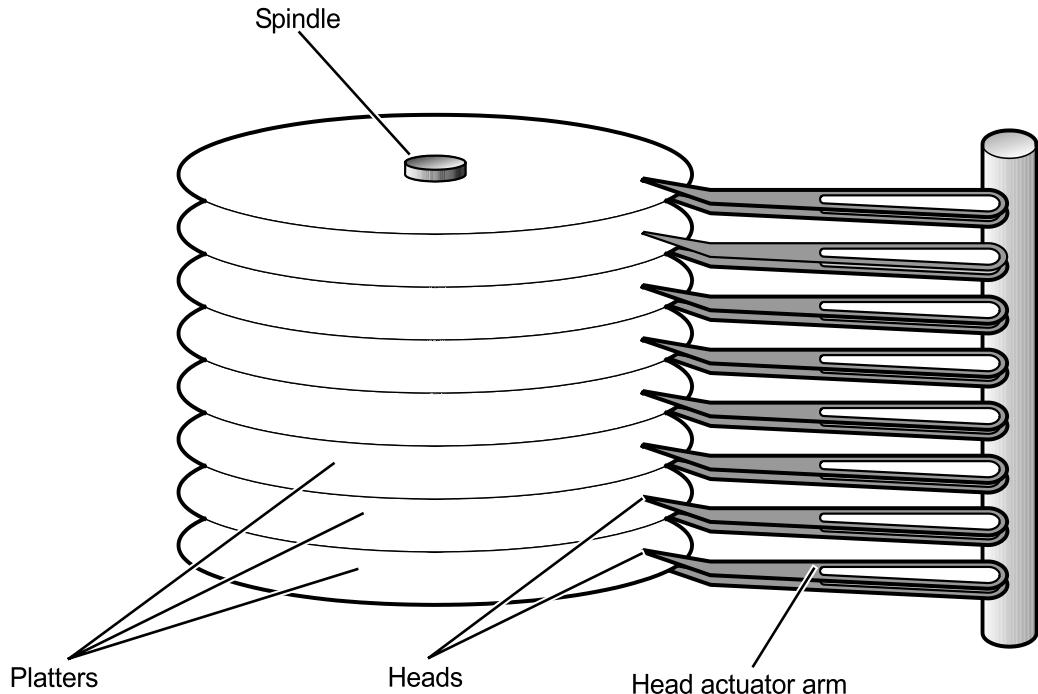


Figure 3-2 Components of a Disk

The following list describes the physical components of a disk:

- The disk storage area is composed of one or more platters.
- The platters rotate.
- The head actuator arm moves the read/write heads as a unit radially.
- The read/write heads read and write data on the magnetic surface on both sides of the platters.

Data Organization on Disk Platters

Figure 3-3 shows the logical components of a disk platter.

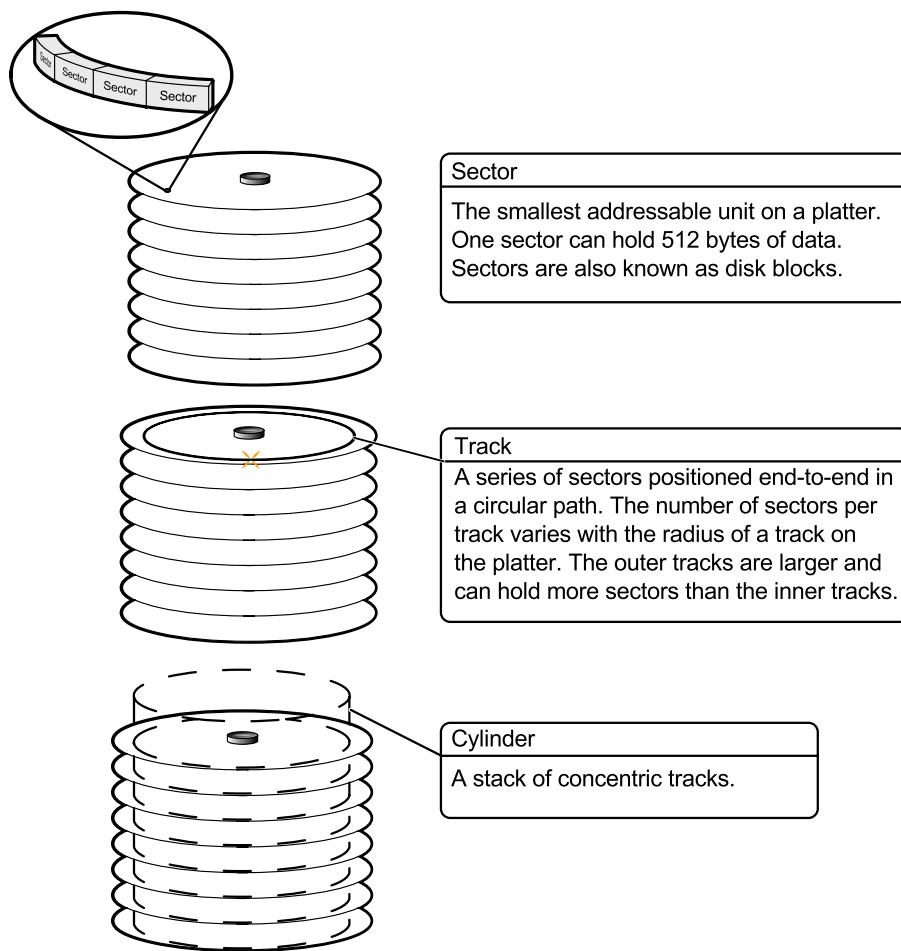


Figure 3-3 Data Organization on Disk Platters

A disk platter is divided into sectors, tracks, and cylinders.

Sector	The smallest addressable unit on a platter. One sector can hold 512 bytes of data. Sectors are also known as disk blocks.
Track	A series of sectors positioned end-to-end in a circular path.
Cylinder	A stack of tracks.

The number of sectors per track varies with the radius of a track on the platter. The outermost tracks are larger and can hold more sectors than the inner tracks.

Because a disk spins continuously and the read/write heads move as a single unit, the most efficient seeking occurs when the sectors to be read from or written to are located in a single cylinder.

Disk Slices

Disks are logically divided into individual partitions known as disk slices. Disk slices are groupings of cylinders that are commonly used to organize data by function.

For example, one slice can store critical system files and programs while another slice on the same disk can store user-created files.

Note – Grouping cylinders into slices is done to organize data, facilitate backups, and provide swap space.



A disk under the Solaris OS can be divided into eight slices that are labeled Slice 0 through Slice 7.

By convention, Slice 2 represents the entire disk. Slice 2 maintains important data about the entire disk, such as the size of the actual disk and the total number of cylinders available for the storage of files and directories.

A starting cylinder and an ending cylinder define each slice. These cylinder boundaries determine the size of a slice.

Figure 3-4 shows how disk slices might reside on a disk.

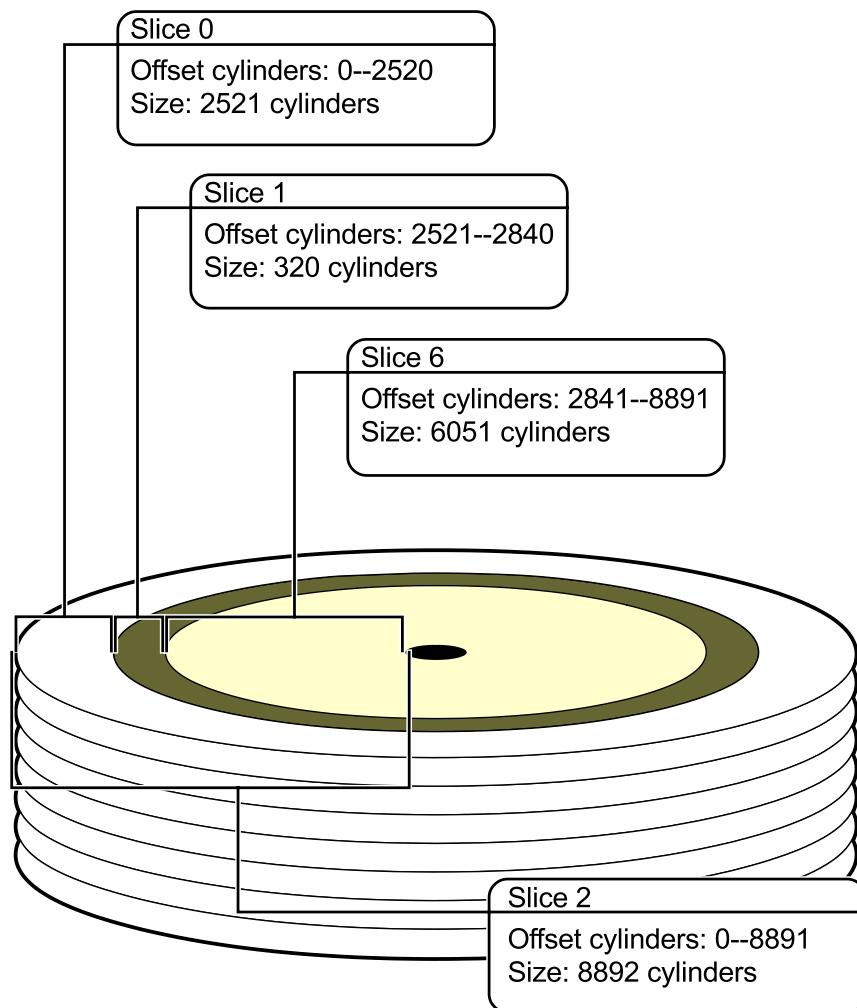


Figure 3-4 Cylinders and Slices

Table 3-1 shows disk slices and the different file systems they could hold.

Table 3-1 Disk Slices

Slice	Name	Function
0	/	The root directory's system files
1	swap	Swap area
2		Entire disk
5	/opt	Optional software
6	/usr	System executables and programs
7	/export/home	User files and directories

Figure 3-5 shows a possible configuration convention for organizing data. The example disk is divided into slices that logically organize the data on the boot disk.

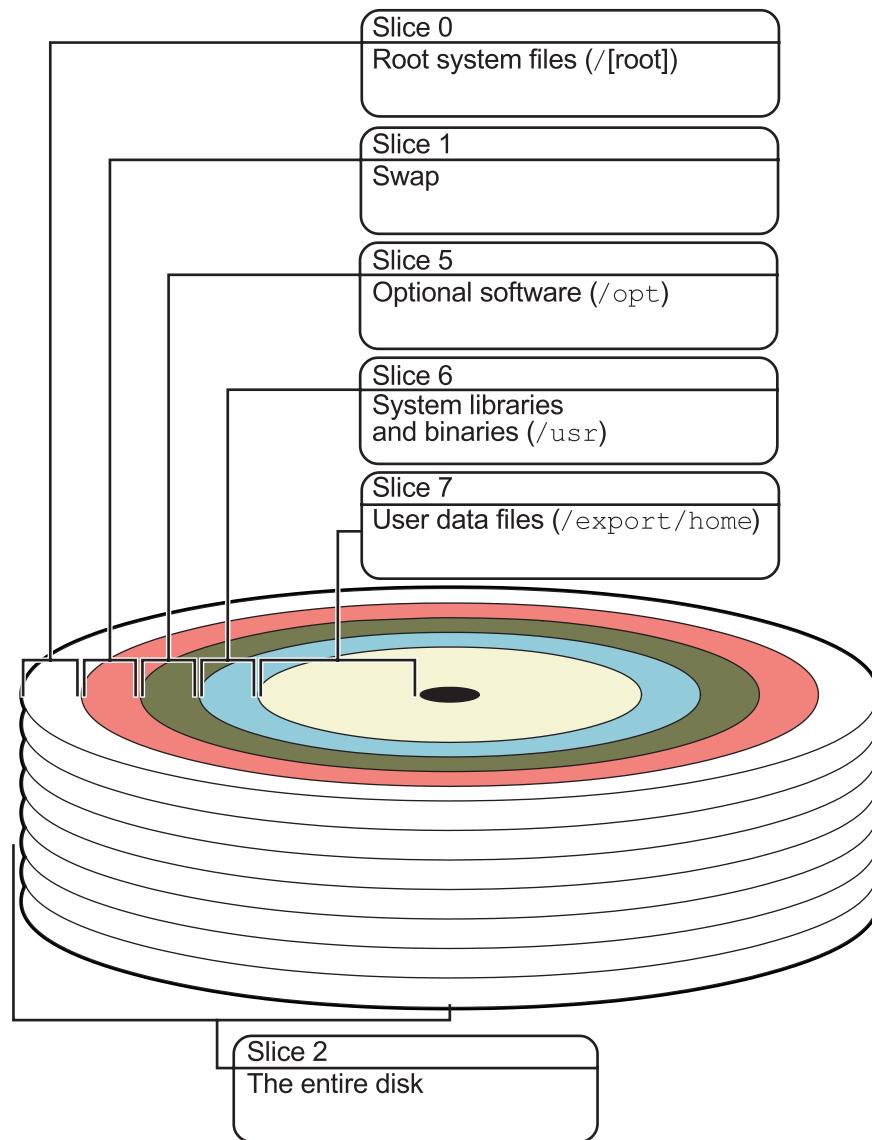


Figure 3-5 Top View of Five Configured Disk Slices

Disk Slice Naming Convention

An eight-character string typically represents the full name of a slice. The string includes the controller number, the target number, the disk number, and the slice number.

Controller number	Identifies the host bus adapter (HBA), which controls communications between the system and disk unit. The HBA takes care of sending and receiving both commands and data to the device. The controller number is assigned in sequential order, such as c0, c1, c2, and so on.
Target number	Target numbers, such as t0, t1, t2, and t3, correspond to a unique hardware address that is assigned to each disk, tape, or CD-ROM. Some external disk drives have an address switch located on the rear panel. Some internal disks have address pins that are jumpered to assign that disk's target number.
Disk number	The disk number is also known as the logical unit number (LUN). This number reflects the number of disks at the target location.
Slice number	A slice number ranging from 0 to 7.

The embedded SCSI configuration and the integrated device electronics (IDE) configuration represent the disk slice naming conventions across two different architectures. The disk number is always set to d0 with embedded SCSI disks.

Figure 3-6 shows the string that represents the full name of a disk slice.

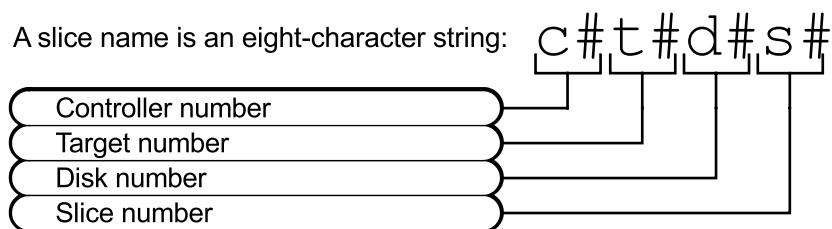


Figure 3-6 Disk Slice Naming Conventions

Figure 3-7 shows the configuration of the SCSI architecture.

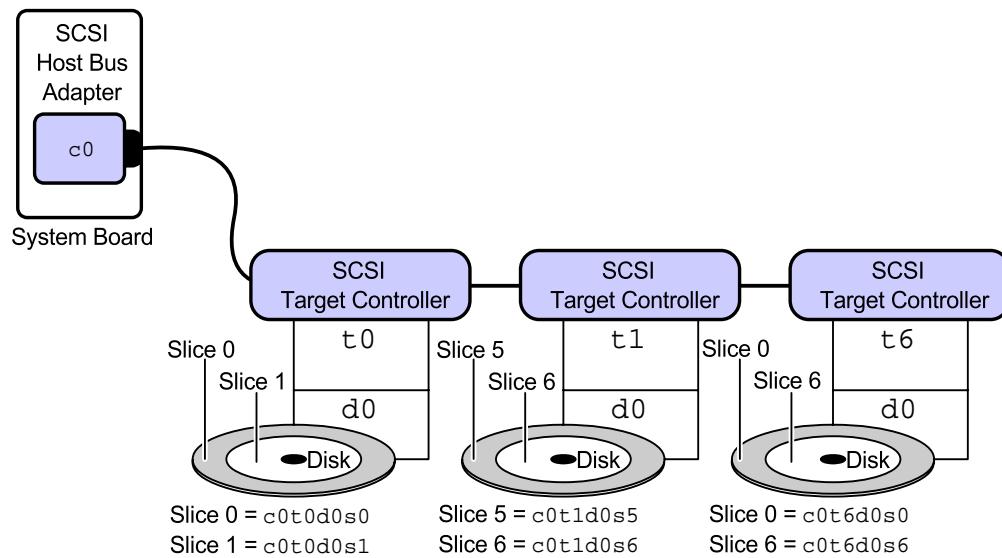


Figure 3-7 Embedded SCSI Configuration

Figure 3-8 shows the configuration of the IDE architecture.

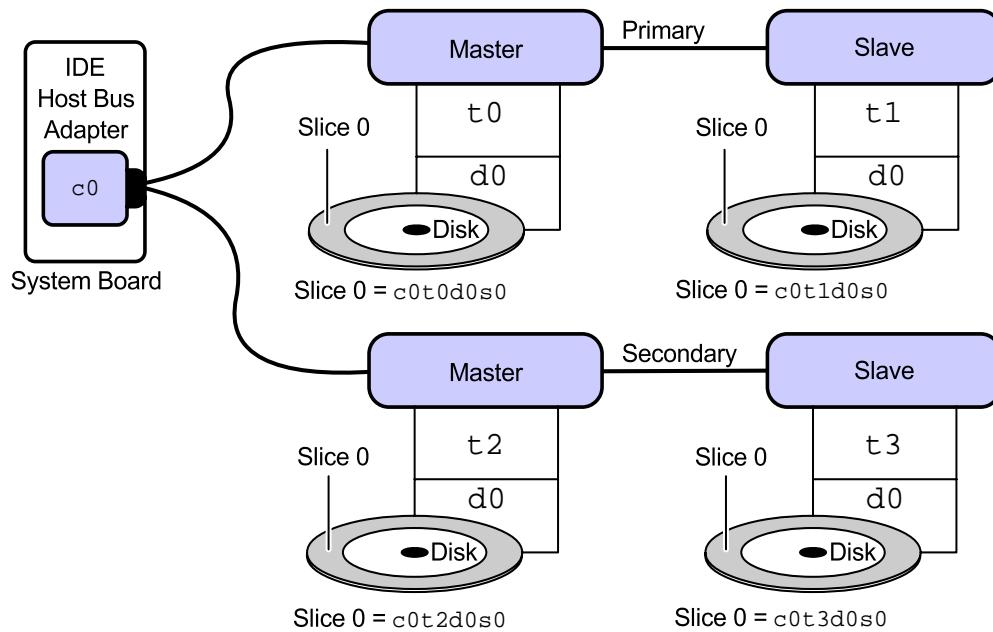


Figure 3-8 IDE Configuration

Introducing Solaris OS Device Naming Conventions

In the Solaris OS, all devices are represented by three different types of names, depending on how the device is being referenced:

- Logical device names
- Physical device names
- Instance names

Logical Device Names

Logical disk device names are symbolic links to the physical device names kept in the /devices directory. Logical device names are used primarily to refer to a device when you are entering commands on the command line. All logical device names are kept in the /dev directory. The logical device names contain the controller number, target number, disk number, and slice number.

Every disk device has an entry in both the /dev/dsk and /dev/rdsk directories for the block and character disk devices, respectively. To display the entries in the /dev/dsk directory, perform the command:

```
# ls /dev/dsk
c0t0d0s0  c0t0d0s4  c0t2d0s0  c0t2d0s4  c1t1d0s0  c1t1d0s4
c0t0d0s1  c0t0d0s5  c0t2d0s1  c0t2d0s5  c1t1d0s1  c1t1d0s5
c0t0d0s2  c0t0d0s6  c0t2d0s2  c0t2d0s6  c1t1d0s2  c1t1d0s6
c0t0d0s3  c0t0d0s7  c0t2d0s3  c0t2d0s7  c1t1d0s3  c1t1d0s7
```

- c0t0d0s0 through c0t0d0s7 – Identifies the device names for disk Slices 0 through 7 for a disk that is attached to Controller 0, at Target 0, on Disk Unit 0.
- c0t2d0s0 through c0t2d0s7 – Identifies the device names for disk Slices 0 through 7 for a disk that is attached to Controller 0, at Target 2, on Disk Unit 0.
- c1t1d0s0 through c1t1d0s7 – Identifies the device names for disk Slices 0 through 7 for a disk that is attached to Controller 1, at Target 1, on Disk Unit 0.

Physical Device Names

Physical device names uniquely identify the physical location of the hardware devices on the system and are maintained in the /devices directory.

A physical device name contains the hardware information, represented as a series of node names, separated by slashes, that indicate the path to the device. To display a physical device name, perform the command:

```
# ls -l /dev/dsk/c0t0d0s0
lrwxrwxrwx 1 root      root      46 Sep 24 10:59 /dev/dsk/c0t0d0s0 ->
.../..../devices/pci@lf,0/pci@1,1/ide@3/dad@0,0:a
```

FC-AL disks will appear slightly different because they also display a World Wide Name (WWN). The following example was taken from a Sun™ Enterprise 3500 server:

```
# ls -l /dev/rdsk/c0t0d0s0
lrwxrwxrwx 1 root      root      78 Jun 16 2000 /dev/rdsk/c0t0d0s0 ->
.../..../devices/sbus@2,0/SUNW,socal@d,10000/sf@0,0/ssd@w21000020375b9ab6,0:a,raw
```

Figure 3-9 shows the device configuration hierarchy of an Ultra 5 workstation. Not all possible devices are included.

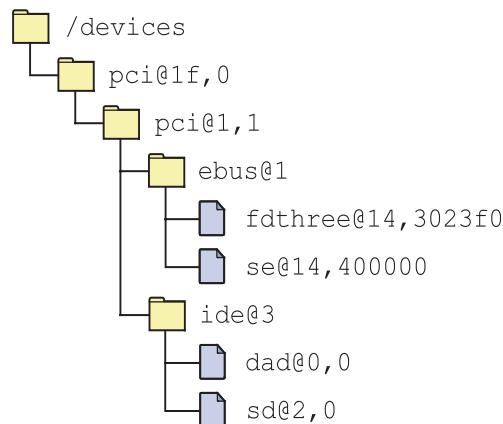


Figure 3-9 The /devices Directory Structure

The device tree can be thought of as existing on two levels. First, there is the device tree as recognized by the hardware at boot time. This device tree consists of all devices that were powered on and accessible to the hardware testing at power-on time.

The second device tree is that known to the Solaris OS kernel. This tree is managed using the various software controls available to the Solaris OS. The devices in this tree must have appropriate device files. If no device file exists for a powered-on device that is physically attached to the system, the kernel does not recognize this device in its device tree until a new device probe is initiated.

 **Note –** Various hardware platforms have different device trees.

The top-most directory in the hierarchy is called the root node of the device tree. The bus nexus nodes and the leaf nodes below the root object have device drivers associated with them.

A device driver is the software that communicates with the device. This software must be available to the kernel so that the system can use the device.

During system initialization, the kernel identifies the physical location of a device. The kernel associates a node with an address, *nodename@address*, which is the physical device name. In Figure 3-9, dad@0 is the direct access disk device at address 0.

Instance Names

Instance names are abbreviated names assigned by the kernel for each device on the system.

An instance name is a shortened name for the physical device name. Two examples are shown:

- `sdn`
where `sd` is the disk name and `n` is the number, such as `sd0` for the first SCSI disk device.
- `dadn`
where `dad` (direct access device) is the disk name and `n` is the number, such as `dad0` for the first IDE disk device.

Listing a System's Devices

In the Solaris OS, there are several ways to list a system's devices, including:

- Using the `/etc/path_to_inst` file
- Using the `prtconf` command
- Using the `format` command

The `/etc/path_to_inst` File

For each device, the system records its physical name and instance name in the `/etc/path_to_inst` file. These names are used by the kernel to identify every possible device. This file is read only at boot time.

The `/etc/path_to_inst` file is maintained by the kernel, and it is generally not necessary, nor is it advisable, for the system administrator to change this file.

The following example shows entries in the `/etc/path_to_inst` file. The text within the parentheses indicates what device is referred to by the entry and does not appear in the actual file.

```
# cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/pci@lf,0" 0 "pcipsy"
"/pci@lf,0/pci@1,1" 0 "simba"
"/pci@lf,0/pci@1,1/ide@3" 0 "uata"
"/pci@lf,0/pci@1,1/ide@3/sd@2,0" 3 "sd"
"/pci@lf,0/pci@1,1/ide@3/dad@0,0" 1 "dad"
"/pci@lf,0/pci@1,1/ebus@1" 0 "ebus"
"/pci@lf,0/pci@1,1/ebus@1/power@14,724000" 0 "power"
...
"/pci@lf,0/pci@1/pci@3" 0 "pci_pci"
"/pci@lf,0/pci@1/pci@3/SUNW,qfe@0,1" 0 "qfe"
"/pci@lf,0/pci@1/pci@3/SUNW,qfe@1,1" 1 "qfe"
"/pci@lf,0/pci@1/pci@3/SUNW,qfe@2,1" 2 "qfe"
"/pci@lf,0/pci@1/pci@3/SUNW,qfe@3,1" 3 "qfe"
```

(Some text was omitted.)

Listing a System's Devices

The device instance number, shown in the preceding example, appears to the left of the device instance name when recorded in this file.

Note – Different systems have different physical device paths. The preceding example shows an on-board peripheral component interconnect (PCI) bus configuration.

The following is a /etc/path_to_inst file from a system that has a different bus architecture. In this case, it is an example of a system that has an on-board Sun System bus (SBus).

```
# cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
"/sbus@1f,0" 0 "sbus"
"/sbus@1f,0/espdma@e,8400000" 0 "dma"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000" 0 "esp"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@3,0" 3 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@2,0" 2 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@1,0" 1 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@0,0" 0 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@6,0" 6 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@5,0" 5 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@4,0" 4 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@3,0" 3 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@2,0" 2 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@1,0" 1 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@0,0" 0 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@6,0" 6 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@5,0" 5 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@4,0" 4 "st"
... < remaining lines removed > ...
```

The following example is an /etc/path_to_inst file with an FC-AL entry:

```
"/sbus@2,0/SUNW,socal@d,10000/sf@0,0/ssd@w2100020375b9ab6,0" 0 "ssd"
```

The prtconf Command

Use the `prtconf` command to display the system's configuration information, including the total amount of memory installed and the configuration of system peripherals, which is formatted as a device tree.

The `prtconf` command lists all possible instances of devices, whether the device is attached or not attached to the system. To view a list of only attached devices on the system, perform the command:

```
# prtconf | grep -v not
System Configuration: Sun Microsystems sun4u
Memory size: 256 Megabytes
System Peripherals (Software Nodes):

SUNW,Ultra-5_10
    scsi_vhci, instance #0
        options, instance #0
        pci, instance #0
            pci, instance #0
                ebus, instance #0
                    power, instance #0
                    su, instance #0
                    su, instance #1
                    fdthree, instance #0
                network, instance #0
                SUNW,m64B, instance #0
                ide, instance #0
                    sd, instance #3
                    dad, instance #1
            pci, instance #1
                scsi, instance #0
        pseudo, instance #0
#
#
```

Note – The `grep -v not` command is used to omit all lines containing the word “not” from the output (such as driver not attached).



The format Command

Use the format command to display both logical and physical device names for all currently available disks. To view the logical and physical devices for currently available disks, perform the command:

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
 0. c0t0d0 <ST38410A cyl 16706 alt 2 hd 16 sec 63>
    /pci@1f,0/pci@1,1/ide@3/dad@0,0
 1. c1t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
    /pci@1f,0/pci@1/scsi@1/sd@3,0
Specify disk (enter its number):^D
#
```

Note – Press Control-D to exit the format command without selecting a disk.



Reconfiguring Devices

Two of the ways the system recognizes a newly added peripheral device is if a reconfiguration boot is invoked or if the `devfsadm` command is run.

Performing a Reconfiguration Boot

For example, you can use a boot process to add a new device to a newly generated `/etc/path_to_inst` file and to the `/dev` and `/devices` directories.

The following steps reconfigure a system to recognize a new disk.

1. Create the `/reconfigure` file. This file causes the system to check for the presence of any newly installed devices the next time it is powered on or booted.

```
# touch /reconfigure
#
```

2. Shut down the system by using the `init 5` command. This command safely powers off the system, allowing for addition or removal of devices. (If the device is already attached to your system, you can shut down to the `ok` prompt with the command `init 0`.)

```
# init 5
```

3. Install the peripheral device. Make sure that the address of the device being added does not conflict with the address of other devices on the system.
4. Turn on the power to all external devices.
5. Verify that the peripheral device has been added by issuing either the `prtconf` command or the `format` command.

After the disk is recognized by the system, begin the process of defining disk slices.



Note – If the `/reconfigure` file was not created before the system was shut down, you can invoke a manual reconfiguration boot with the programmable read-only memory (PROM) level command: `boot -r`.

Using the devfsadm Command

Many systems are running critical customer applications on a 24-hour, 7-day-a-week basis. It might not be possible to perform a reconfiguration boot on these systems. In this situation, you can use the `devfsadm` command.

The `devfsadm` command performs the device reconfiguration process and updates the `/etc/path_to_inst` file and the `/dev` and `/devices` directories during reconfiguration events.

The `devfsadm` command attempts to load every driver in the system and attach all possible device instances. It then creates the device files in the `/devices` directory and the logical links in the `/dev` directory. In addition to managing these directories, the `devfsadm` command also maintains the `/etc/path_to_inst` file.

```
# devfsadm
```

To restrict the operation of the `devfsadm` command to a specific device class, use the `-c` option.

```
devfsadm -c device_class
```

The values for `device_class` include `disk`, `tape`, `port`, `audio`, and `pseudo`. For example, to restrict the `devfsadm` command to the `disk` device class, perform the command:

```
# devfsadm -c disk
```

Use the `-c` option more than once on the command line to specify multiple device classes. For example, to specify the `disk`, `tape`, and `audio` device classes, perform the command:

```
# devfsadm -c disk -c tape -c audio
```

To restrict the use of the `devfsadm` command to configure only devices for a named driver, use the `-i` option.

```
devfsadm -i driver_name
```

The following examples use the **-i** option.

- To configure only those disks supported by the **dad** driver, perform the command:

```
# devfsadm -i dad
```

- To configure only those disks supported by the **sd** driver, perform the command:

```
# devfsadm -i sd
```

- To configure devices supported by the **st** driver, perform the command:

```
# devfsadm -i st
```

For a verbose output of changes to the device tree, perform the command:

```
# devfsadm -v
```

To invoke cleanup routines that remove unreferenced symbolic links for devices, perform the command:

```
# devfsadm -C
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab is more difficult. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Configuring and Naming Devices (Level 1)

In this exercise, you complete the following tasks:

- Identify logical, physical, and instance names for disk devices
- View the /etc/path_to_inst file for information about your boot disk
- Add a new disk or tape drive to a system
- Create new device files for the new disk or tape

Preparation

This exercise requires a system that is configured with an external disk or tape drive. During system boot, this external disk must remain powered off to avoid creating links and device files.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Identify the logical device name of your boot disk. Locate the logical device files in the /dev/dsk and /dev/rdsk directories for Slice 0 on this disk, and record their true file types.
- Locate the physical device names that are associated with both logical device names that you found for your boot disk. Record their true file types. (Steps 1–5 in the Level 2 lab)
- In the /etc/path_to_inst file, identify and record the instance name for your boot disk. (Steps 6–7 in the Level 2 lab)
- Confirm that no links or device files exist for the disk or tape device that you want to connect. Halt the system, and power on the device. Boot the system to its default run state. Run the devfsadm command in verbose mode to create new links and device files, and check the directories in which you created them to confirm that they exist. (Task 1, Steps 1–2, and Task 2, Steps 1–4, in the Level 2 lab)

Exercise: Configuring and Naming Devices (Level 2)

In this exercise, you complete the following tasks:

- Identify logical, physical, and instance names for disk devices
- View the /etc/path_to_inst file for information about your boot disk
- Add a new disk or tape drive to a system
- Create new device files for the new disk or tape

Preparation

This exercise requires a system that is configured with an external disk or tape drive. During system boot, this external disk must remain powered off to avoid creating links and device files.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

Complete the following tasks:

- Identify the logical device name of your boot disk. Locate the logical device files in the /dev/dsk and /dev/rdsk directories for Slice 0 on this disk, and record their true file types.
- Locate the physical device names that are associated with both logical device names that you found for your boot disk. Record their true file types.
- In the /etc/path_to_inst file, identify and record the instance name for your boot disk.
- Confirm that no links or device files exist for the disk or tape device that you want to connect. Halt the system, and power on the device. Boot the system to its default run state. Run the devfsadm command in verbose mode to create new links and device files, and check the directories in which you created them to confirm that they exist.

Tasks

Complete the following tasks.

Task 1 – Identifying Device Files

Complete the following steps:

1. Log in as the root user, and open a terminal window. Expand the window so that it occupies the entire screen area. Change to the /dev/dsk directory.
2. List the files in this directory. Identify the files related to the boot disk of your system. Most systems use c0t0d0. Locate the item related to Slice 0 on this disk, and display a long listing of it.

Which type of file did you just locate? The file type indicator is the first character on the left side of the long listing.

Record the full path name to which this file points.

3. Issue a long listing command of the path name you recorded.

Which type of file is this?

The command `ls -lL c0t0d0s0` displays the same information but shows only the link file name, not the real device file name.

4. Change to the /dev/rdsk directory. Display a long listing of the same file name you selected in Step 2.

Which type of file is this?

Record the full path name to which this file points.

5. Issue a long listing of the path name you recorded in the previous step.

Which type of file is this?

The ls -lL c0t0d0s0 command displays the same information but shows only the link file name, not the real device file name.

6. Change the directory to the /etc directory. Display the contents of the path_to_inst file.

7. Use the information from the previous steps to locate and record the entry for your boot disk. An Ultra 5 workstation, for example, would use c0t0d0 as its boot disk. This relates to the device file called dad@0,0 and is listed in the /etc/path_to_inst directory.

The instance name is composed of the dad or sd tag and the number that precedes it in the /etc/path_to_inst file. What is the instance name for the device listed in this step?

Task 2 – Adding a New Disk or Tape Device

Complete the following steps:

1. In the /dev/dsk and /dev/rmt directories, confirm that no files exist for your external disk or tape device, for example, /dev/dsk/c1t0d0s0 or /dev/rmt/0. If files for the external device do exist, ask your instructor to provide directions to remove them.
2. Shut down your system to run state 0.
3. Power on the external disk or tape drive attached to your system.
4. Boot the system to its default run state.
5. Log in as the root user, and open a terminal window. Run the devfsadm command with the -v option to create new links and device files for the new disk or tape drive. Observe the messages that the devfsadm command displays.
6. Confirm that new links and device files exist in the /dev/dsk and /dev/rdsk directories for disks or /dev/rmt for tape drives.

Exercise: Configuring and Naming Devices (Level 3)

In this exercise, you complete the following tasks:

- Identify logical, physical, and instance names for disk devices
- View the /etc/path_to_inst file for information about your boot disk
- Add a new disk or tape drive to a system
- Create new device files for the new disk or tape

Preparation

This exercise requires a system that is configured with an external disk or tape drive. During system boot, this external disk must remain powered off to avoid creating links and device files.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

Complete the following tasks:

- Identify the logical device name of your boot disk. Locate the logical device files in the /dev/dsk and /dev/rdsk directories for Slice 0 on this disk, and record their true file types.
- Locate the physical device names that are associated with both logical device names that you found for your boot disk. Record their true file types.
- In the /etc/path_to_inst file, identify and record the instance name for your boot disk.
- Confirm that no links or device files exist for the disk or tape device that you want to connect. Halt the system, and power on the device. Boot the system to its default run state. Run the devfsadm command in verbose mode to create new links and device files, and check the directories in which you created them to confirm that they exist.

Tasks and Solutions

Complete the following tasks.

Task 1 – Identifying Device Files

Complete the following steps:

1. Log in as the root user, and open a terminal window. Expand the window so that it occupies the entire screen area. Change to the /dev/dsk directory.

```
# cd /dev/dsk
```

2. List the files in this directory. Identify the files related to the boot disk of your system. Most systems use c0t0d0. Locate the item related to Slice 0 on this disk, and display a long listing of it.

```
# ls  
# ls -l c0t0d0s0
```

Which type of file did you just locate? The file type indicator is the first character on the left side of the long listing.

Files in this directory are symbolic links. The letter l in the left-most column identifies a symbolic link.

Record the full path name to which this file points.

Exercise: Configuring and Naming Devices (Level 3)

Systems that use PCI bus architectures list path names similar to the following:

```
.../.../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a
```

Systems that use SBus architectures list path names similar to the following:

```
.../.../devices/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,80000000/sd@3,0:a
```

3. Issue a long listing command of the path name you recorded.

```
# ls -l pathname
```

Which type of file is this?

Files in this directory are device files. The b character in the left-most column identifies a block-special device file.

The command `ls -lL c0t0d0s0` displays the same information but shows only the link file name, not the real device file name.

4. Change to the `/dev/rdsk` directory. Display a long listing of the same file name you selected in Step 2.

```
# cd /dev/rdsk  
# ls -l c0t0d0s0
```

Which type of file is this?

Files in this directory are symbolic links. The letter l in the left-most column identifies a symbolic link.

Record the full path name to which this file points.

Systems that use PCI bus architectures list path names similar to the following:

```
.../.../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a,raw
```

Systems that use SBus architectures list path names similar to the following:

```
.../.../devices/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,80000000/sd@3,0:a,raw
```

5. Issue a long listing command of the path name you recorded.

```
# ls -l pathname
```

Which type of file is this?

Files in this directory are device files. The c character in the left-most column identifies a character-special device file.

The `ls -lL c0t0d0s0` command displays the same information but shows only the link file name, not the real device file name.

6. Change to the /etc directory. Display the contents of the path_to_inst file.

```
# cd /etc
# more path_to_inst
```

7. Use the information from the previous steps to locate and record the entry for your boot disk. An Ultra 5 workstation, for example, would use c0t0d0 as its boot disk. This relates to the device file called dad@0,0 and is listed in the /etc/path_to_inst directory.

Systems that use PCI bus architectures list path names similar to the following:

/pci@1f,0/pci@1,1/ide@3/dad@0,0

Systems that use SBus architectures list path names similar to the following:

/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0

The instance name is composed of the dad or sd tag and the number that precedes it in the /etc/path_to_inst file. What is the instance name for the device listed in this step?

dad0, sd3, or sd0, depending on the system architecture.

Task 2 – Adding a New Disk or Tape Device

Complete the following steps:

1. In the /dev/dsk and /dev/rmt directories, confirm that no files exist for your external disk or device, for example, /dev/dsk/c1t0d0s0 or /dev/rmt/0. If files for the external device do exist, ask your instructor for guidance.
2. Shut down your system to run state 0.

```
# init 0
```

3. Power on the external disk or tape drive attached to your system.
4. Boot the system to its default run state.

```
ok boot
```

5. Log in as the root user, and open a terminal window. Run the devfsadm command with the -v option to create new links and device files for the new disk or tape drive. Observe the messages that the devfsadm command displays.

```
# devfsadm -v
```

6. Confirm that new links and device files exist in the /dev/dsk and /dev/rdsk directories for disks or /dev/rmt for tape drives.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Partitioning the Hard Disk

The `format` utility is a system administration tool used primarily to prepare hard disk drives for use in the Solaris OS.

Although the `format` utility also performs a variety of disk management activities, the main function of the `format` utility is to divide a disk into disk slices.

Note – You do not need to partition the disk before you install the Solaris OS.



Introducing the Fundamentals of Disk Partitioning

To divide a disk into slices:

1. Identify the correct disk.
2. Plan the layout of the disk.
3. Use the `format` utility to divide the disk into slices.
4. Label the disk with new slice information.

Only a user with privileges can use the `format` utility. If a regular user runs the `format` utility, the following error message appears:

```
$ /usr/sbin/format
Searching for disk...done
No permission (or no disk found)!
```

Recognizing Disk Space and Undesirable Conditions

Disk slices are defined by an offset and a size in cylinders. The offset is the distance from Cylinder 0. Figure 3-10 shows an example of disk slice sizes and offsets.

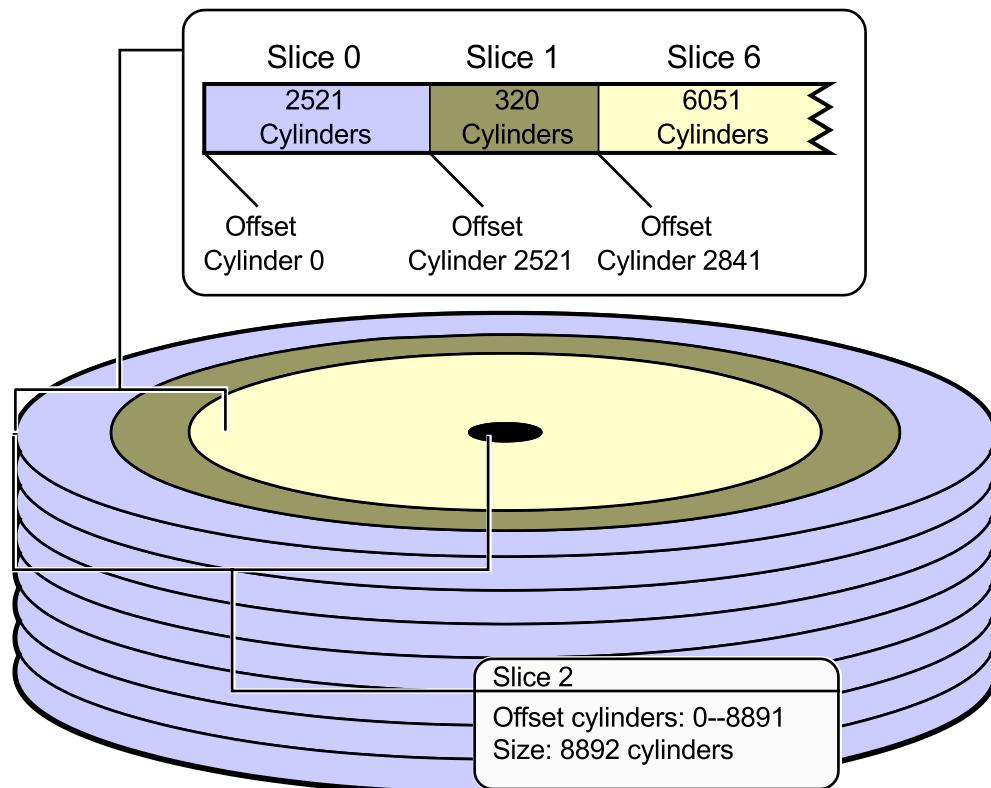


Figure 3-10 Offsets and Sizes for Disk Partitions

The offset for Slice 0 is 0 cylinders, and its size is 2521 cylinders. Slice 0 begins on Cylinder 0 and ends on Cylinder 2520.

The offset for Slice 1 is 2521 cylinders, and its size is 320 cylinders. Slice 1 begins on Cylinder 2521 and ends on Cylinder 2840.

The offset for Slice 6 is 2841 cylinders, and its size is 6051 cylinders. Slice 6 begins on Cylinder 2841 and ends on the last available cylinder, which is Cylinder 8891.

Recognizing Wasted Disk Space

Wasted disk space occurs when one or more cylinders are not allocated to a disk slice. Figure 3-11 shows a disk with cylinders that are not allocated.

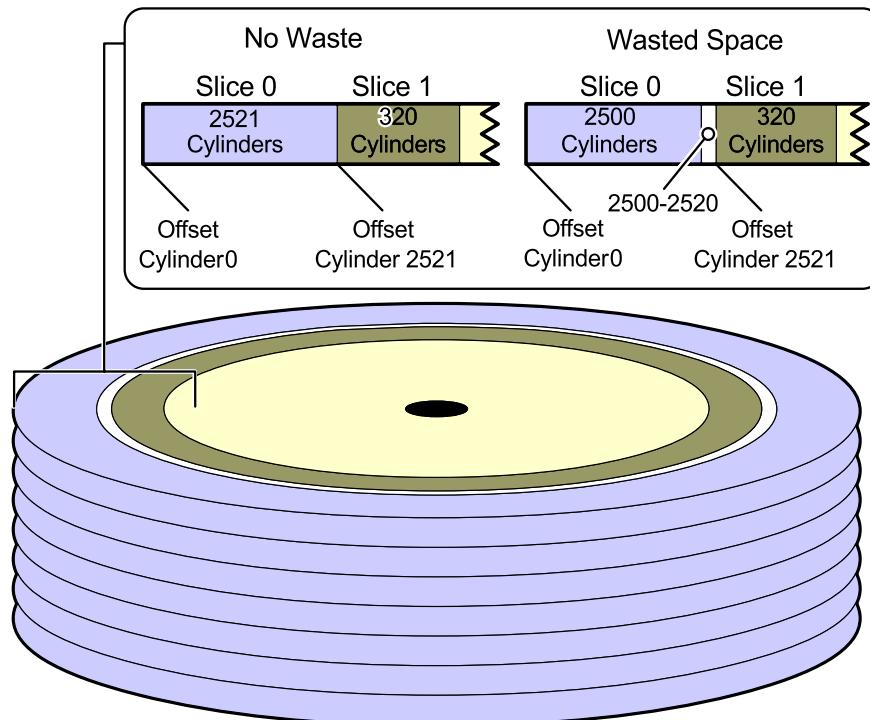


Figure 3-11 A Disk Slice With Wasted Space

Because the cylinders are not allocated to the disk slice, Cylinders 2500 through 2520 are unusable.

Wasted disk space occurs during partitioning when one or more cylinders have not been allocated to a disk slice. This might happen intentionally or accidentally. If there are unallocated slices available, then wasted space can possibly be assigned to a slice later on.

Recognizing Overlapping Disk Slices

Overlapping disk slices occur when one or more cylinders are allocated to more than one disk slice. Figure 3-12 shows a disk with cylinders allocated to more than one disk slice.

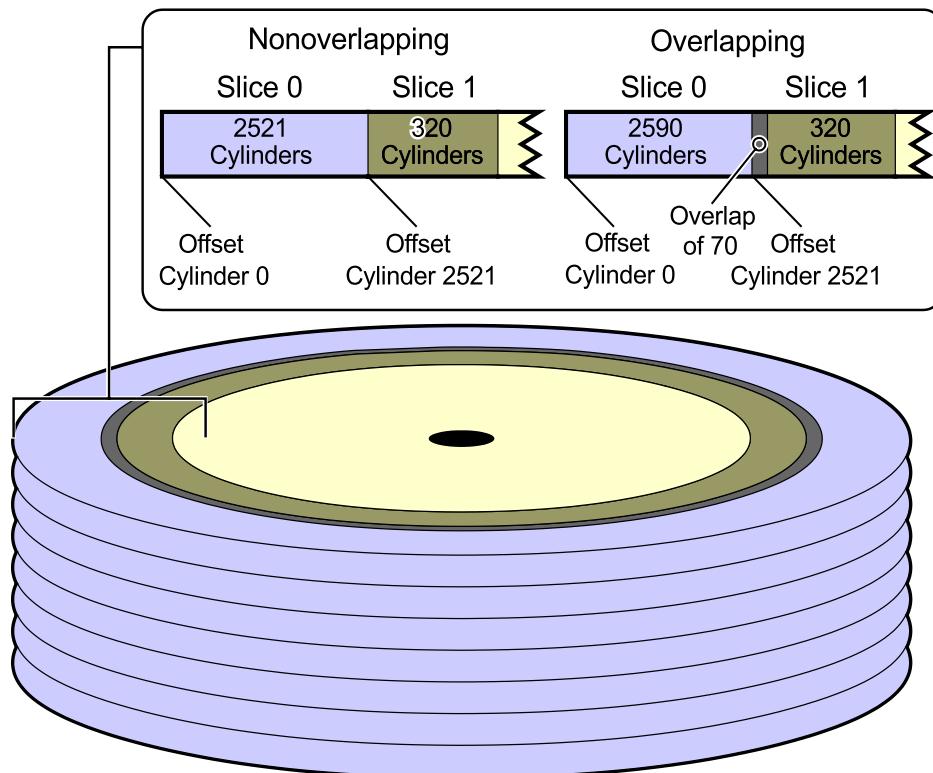


Figure 3-12 Disk Slices With Overlapping Cylinders

In Figure 3-12, Cylinders 2521 through 2590 are overlapping two disk slices.

This type of condition occurs when the size of one disk slice is increased and the starting cylinder number of the next disk slice is not adjusted. Only the *format* utility's *modify* command warns you of overlapping disk slices.

```
partition> modify
Select partitioning base:
  0. Current partition table (unnamed)
  1. All Free Hog
Choose base (enter number) [0]? 0
```

Warning: Overlapping partition (1) in table.

Warning: Fix, or select a different partition table.



Caution – Do not change the size of disk slices that are currently in use. When a disk with existing slices is repartitioned and relabeled, any existing data can become inaccessible. Copy existing data to backup media before the disk is repartitioned, and restore the data to the disk after the disk is relabeled and contains a new file system.



Note – If two partitions create the overlap, when data is saved into one of the partitions, data could be overwritten in the other partition located on the tracks in the same disk cylinder.

Introducing Disk Partition Tables

As the root user, when you use the `format` utility and select a disk to partition, a copy of the disk's partition table is read from the label on the disk into memory and is displayed as the current disk partition table.

The `format` utility also works with a file called `/etc/format.dat`, which is read when you invoke the `format` utility.

The `/etc/format.dat` file is a table of available disk types and a set of predefined partition tables that you can use to partition a disk quickly.

Introducing Disk Labels

The disk's label is the area set aside for storing information about the disk's controller, geometry, and slices. Another term used to describe a disk label is the volume table of contents (VTOC). The disk's label or VTOC is stored on the first sector of the disk.

To label a disk means to write slice information onto the disk. If you fail to label a disk after defining slices, the slice information is lost.

An important part of the disk label is the partition table, which identifies a disk's slices, the slice boundaries in cylinders, and the total size of the slices.



Note – The terms disk slice and disk partition are interchangeable.

Figure 3-13 shows the relationship among the label on the disk, the current label in memory, and the predefined label in the /etc/format.dat file.

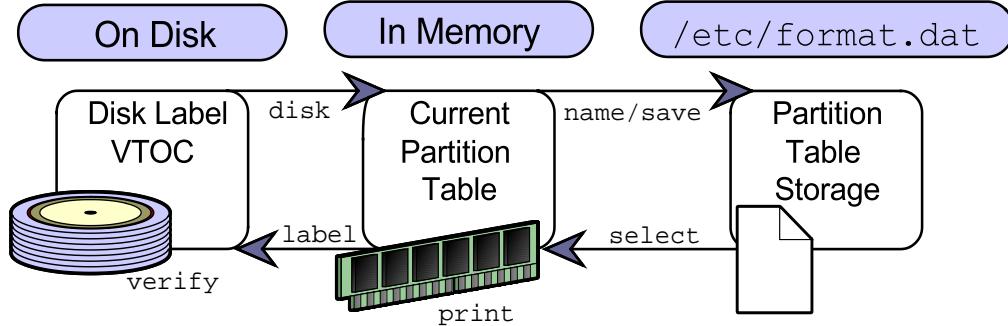


Figure 3-13 Partition Table Locations

Using the format Command

The format utility is organized into two tiers of commands.

When you type format on the command line, the first tier of commands appears. This set of commands allow you to, among other functions, select a disk, select a partition, save new disk and partition definitions, and write the label to the disk. The top tier of commands is denoted by the `format>` prompt.

A second tier of commands appears when you type other functions, for example, `partition`, from the `format>` prompt. This set of commands allows you to define the characteristics of the individual slices, print the existing partition table, and write the partition map and label to the disk.

Table 3-2 describes the terminology for disk partitioning.

Table 3-2 Partition Table Terms and Usage

Term	Description
Part	The slice number. Valid slice numbers are 0 through 7.
Tag The Tag field is historical in meaning and not used anymore.	A value that indicates how the slice is being used. 0 = unassigned 1 = boot 2 = root 3 = swap 4 = usr 5 = backup 6 = stand 8 = home Veritas Volume Manager array tags: 14 = public (region) 15 = private (region)
Flag The Flag field is historical in meaning and not used anymore.	00 wm = The disk slice is writable and mountable. 01 wu = The disk slice is writable and unmountable. <i>This is the default state of slices dedicated for swap areas.</i> 10 rm = The disk slice is read-only and mountable. 11 ru = The disk slice is read-only and unmountable.
Cylinders	The starting and ending cylinder number for the disk slice.
Size	The slice size: Mbytes (MB), Gbytes (GB), blocks (b), or cylinders (c).
Blocks	The total number of cylinders and the total number of sectors per slice.

Partitioning a Disk

Caution – Do not change the size of disk slices that are currently in use.



The following steps demonstrate how to divide a disk into slices:

1. As the root user, type format at the prompt, and press Return.

```
# format
```

```
Searching for disks...done
```

AVAILABLE DISK SELECTIONS:

0. c0t0d0 <ST38410A cyl 16706 alt 2 hd 16 sec 63>
/pci@1f,0/pci@1,1/ide@3/dad@0,0
1. c1t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
/pci@1f,0/pci@1/scsi@1/sd@3,0

Specify disk (enter its number):

The format utility searches for all attached disks that are powered on. For each disk it finds, the format utility displays the logical device name, Sun marketing name, physical parameters, and physical device name.

2. Choose the second disk by selecting the number located to the left of that disk's logical device name. From the preceding display, the number chosen is 1. The format utility's main menu appears.

Specify disk (enter its number): 1

```
selecting c1t3d0
```

```
[disk formatted]
```

FORMAT MENU:

disk	- select a disk
type	- select (define) a disk type
partition	- select (define) a partition table
current	- describe the current disk
format	- format and analyze the disk
repair	- repair a defective sector
label	- write label to the disk
analyze	- surface analysis
defect	- defect list management
backup	- search for backup labels
verify	- read and display labels
save	- save new disk/partition definitions
inquiry	- show vendor, product and revision
scsi	- independent SCSI mode selects
cache	- enable, disable or query SCSI disk cache

```

volname      - set 8-character volume name
!<cmd>       - execute <cmd>, then return
quit
format>

```

The specific menu selections that you can use to view, change, or commit disk slices include the following:

partition	Displays the Partition menu
label	Writes the current partition definition to the disk label
verify	Reads and displays the disk label
quit	Exits the format utility

3. Type **partition** at the **format** prompt. The Partition menu appears.

```
format> partition
```

PARTITION MENU:

0	- change '0' partition
1	- change '1' partition
2	- change '2' partition
3	- change '3' partition
4	- change '4' partition
5	- change '5' partition
6	- change '6' partition
7	- change '7' partition
select	- select a predefined table
modify	- modify a predefined partition table
name	- name the current table
print	- display the current table
label	- write partition map and label to the disk
!<cmd>	- execute <cmd>, then return
quit	

The Partition menu enables you to perform the following functions:

0–7	Specify the offset and size of up to eight slices
select	Choose a predefined partition table from the /etc/format.dat file
modify	Change the current partition table in memory
name	Provide a means to identify the partition table in the /etc/format.dat file
print	Display the current partition table in memory
label	Write the current partition table to the disk label
!<cmd>	Escape from the utility and execute a command from the shell

4. Type print at the partition prompt to display the disk label that was copied to random access memory (RAM) when the format utility was invoked.

partition> **print**

Current partition table (original):

Total disk cylinders available: 4924 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks
0	unassigned	wm	0	0	(0/0/0) 0
1	unassigned	wm	0	0	(0/0/0) 0
2	backup	ru	0 - 4923	8.43GB	(4924/0/0) 17682084
3	unassigned	wu	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	unassigned	wm	0	0	(0/0/0) 0

The name of the partition table appears in parentheses in the first line of the table.

The columns of the table have the following meanings:

Part	The disk slice number
Tag	The predefined, optional tag
Flag	The predefined, optional flag
Cylinders	The starting and ending cylinder number for the slice
Size	The slice size in blocks (b), cylinders (c), Mbytes (MB), or Gbytes (GB)
Blocks	The total number of cylinders and the total number of sectors per slice

5. Select Slice 0 (zero) by entering 0.

```
partition> 0
```

Part	Tag	Flag	Cylinders	Size	Blocks
0 unassigned		wm	0	0	(0/0/0) 0

6. When prompted for the ID tag, type a question mark (?), and press Return to list the available choices. You can change a tag by entering a new tag name.

Enter partition id tag[unassigned]: ?

Expecting one of the following: (abbreviations ok):

unassigned	boot	root	swap
usr	backup	stand	var
home	alternates	reserved	

Enter partition id tag[unassigned]:

7. Type the tag alternates, and press Return.

Enter partition id tag[unassigned]: **alternates**

8. When prompted for the permission flags, type a question mark (?), and press Return to list the available choices. You can change a flag by entering the new flag name.

Enter partition permission flags[wm]: ?

Expecting one of the following: (abbreviations ok):

wm	- read-write, mountable
wu	- read-write, unmountable
rm	- read-only, mountable
ru	- read-only, unmountable

Enter partition permission flags[wm]:

Partitioning the Hard Disk

9. Press Return to accept the default flag.

Enter partition permission flags[wm]: <return>

10. Press Return to accept the starting cylinder of 0 (zero).

Enter new starting cyl[0]: <return>

11. Enter 400mb for the new partition size for Slice 0.

Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: **980mb**

12. Type print, and press Return. The Partition table appears.

partition> **print**

Current partition table (unnamed):

Total disk cylinders available: 1965 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks
0	alternates	wm	0 - 558	980.16MB	(559/0/0) 200736
1	unassigned	wm	0	0	(0/0/0) 0
2	backup	ru	0 - 4923	8.43GB	(4924/0/0) 17682084
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	unassigned	wm	0	0	(0/0/0) 0

The current partition table shows the change to Slice 0.

Now adjust the starting cylinder for Slice 1.

13. Select slice number 1 by typing **1**.

partition> **1**

Part	Tag	Flag	Cylinders	Size	Blocks
1	unassigned	wm	0	0	(0/0/0) 0

14. Type the tag swap, and press Return.

Enter partition id tag[unassigned]: **swap**

15. Type wu at the permission flags selection, and press Return.

Enter partition permission flags[wu]: **wu**

16. Enter the new starting cylinder for Slice 1.

Enter new starting cyl[0]: **559**

17. Enter the new partition size for Slice 1.

Enter partition size[0b, 0c, 603e, 0.00mb, 0.00gb]: **512mb**

18. Type print, and press Return.

```
partition> print
```

Current partition table (unnamed):

Total disk cylinders available: 1965 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks
0	alternates	wm	0 - 558	980.16MB	(559/0/0) 2007369
1	swap	wu	559 - 851	513.75MB	(293/0/0) 1052163
2	backup	ru	0 - 4923	8.43GB	(4924/0/0) 17682084
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	unassigned	wm	0	0	(0/0/0) 0

The current partition table shows the change to Slice 1.

The new starting cylinder for Slice 1 is one greater than the ending cylinder for Slice 0.

Now adjust the starting cylinder for Slice 7.

19. Type 7 to select Slice 7.

```
partition> 7
```

Part	Tag	Flag	Cylinders	Size	Blocks
7	unassigned	wm	0	0	(0/0/0) 0

20. Type the tag **home**, and press Return.

Enter partition id tag[unassigned]: **home**

21. Press Return to select the default flag.

Enter partition permission flags[wm]: <return>

22. Type the new starting cylinder for Slice 7.

Enter new starting cyl[0]: **852**

23. Type the new partition size for Slice 7 by typing a dollar (\$) sign.

Enter partition size[0b, 0c, 694e, 0.00mb, 0.00gb]: **\$**

partition>

Note – Enter a dollar (\$) sign as a value for the last partition size to automatically assign the remaining space on the disk to this slice.



Partitioning the Hard Disk

24. Type **print** to display the partition table.

```
partition> print
```

```
Current partition table (unnamed):
```

```
Total disk cylinders available: 1965 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	alternates	wm	0 - 558	980.16MB	(559/0/0) 2007369
1	swap	wu	559 - 851	513.75MB	(293/0/0) 1052163
2	backup	ru	0 - 4923	8.43GB	(4924/0/0) 17682084
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	home	wm	852 - 4923	6.97GB	(4072/0/0) 14622552

Add up the cylinders in the Blocks column for Slice 0, Slice 1, and Slice 7. The number should equal the total number of cylinders contained in Slice 2.

25. After checking the partition table to ensure that there are no errors, label the disk by typing **label**.

```
partition> label
```

```
Ready to label disk, continue? y
```

```
partition>
```

Managing Disk Labels

Every disk in the Solaris OS has a label set aside for storing information about the disk's controller, geometry, and slices.

Viewing the Disk VTOC

You can use two methods for locating and viewing a disk's label or VTOC:

- Use the `verify` command from the `format` utility
- Invoke the `prtvtoc` command from the command line

Reading a Disk's VTOC Using the `verify` Command

The `verify` command enables you to view a disk's VTOC from within the `format` utility. To read a disk's VTOC, perform the following steps:

1. At the `format` prompt, enter the `verify` command, and press Return.

```
format> verify
```

Primary label contents:

```
Volume name = <           >
ascii name  = <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
pcyl       = 4926
ncyl       = 4924
acyl       =    2
nhead      =    27
nsect      =   133
Part       Tag     Flag      Cylinders          Size            Blocks
  0 alternates   wm      0 -  558    980.16MB    (559/0/0)  2007369
  1      swap     wu      559 -  851    513.75MB    (293/0/0)  1052163
  2    backup     ru      0 - 4923     8.43GB     (4924/0/0) 17682084
  3 unassigned   wu      0           0      (0/0/0)        0
  4 unassigned   wm      0           0      (0/0/0)        0
  5 unassigned   wm      0           0      (0/0/0)        0
  6 unassigned   wu      0           0      (0/0/0)        0
  7      home     wm     852 - 4923    6.97GB     (4072/0/0) 14622552
```

2. Type `quit` or `q`, and press Return to exit the `format` menu.

Reading a Disk's VTOC Using the `prtvtoc` Command

The `prtvtoc` command enables you to view a disk's VTOC from the command line. To view a disk's VTOC from the command line, type the following:

```
# prtvtoc /dev/dsk/c1t3d0s0
* /dev/dsk/c1t3d0s0 partition map
*
* Dimensions:
*      512 bytes/sector
*      133 sectors/track
*      27 tracks/cylinder
*    3591 sectors/cylinder
*    4926 cylinders
*  4924 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*          First        Sector       Last
* Partition Tag  Flags    Sector     Count    Sector Mount Directory
* 0         9   00          0  2007369  2007368
* 1         3   01  2007369  1052163  3059531
* 2         5   11          0  17682084 17682083
* 7         8   00  3059532 14622552 17682083
```

The disk label information includes the following fields:

Dimensions	Describes the logical dimensions of the disk.
Flags	Describes the flags that are listed in the partition table.
Partition	A slice number. It is described further in Table 3-2 on page 3-37.
Tag	A value used to indicate how the slice is being used. It is described further in Table 3-2 on page 3-37.
Flags	The 00 flag is read/write, mountable; 01 is read/write, unmountable; and 10 is read only. These are described further in Table 3-2 on page 3-37.
First Sector	Defines the first sector of the slice.

Sector Count	Defines the total number of sectors in the slice.
Last Sector	Defines the last sector number in the slice.
Mount Directory	If the field is empty, the slice is currently not mounted and no entry exists in the /etc/vfstab file.

Relabeling a Disk

Save a disk's VTOC to a file by using the `prtvtoc` command. This allows you to relabel the disk by using the `fmthard` command if one of the following situations occurs:

- The VTOC on the disk has been destroyed.
- You accidentally changed the partition information on the disk and did not save a backup label in the `/etc/format.dat` file.

To save a disk's VTOC to a file, perform the command:

```
# prtvtoc /dev/dsk/c1t3d0s0 > /var/tmp/c1t3d0.vtoc
```

The `fmthard` Command

To relabel a disk, you can save the output of the `prtvtoc` command into a file on another disk and use it as the `datafile` argument to the `fmthard` command.

```
fmthard -s datafile /dev/rdsck/c#t#d#s2
```



Caution – The `fmthard` command cannot write a disk label on an unlabeled disk. Use the `format` utility for this purpose.

If the need to relabel a disk arises and the VTOC was previously saved to a file, the following options are available:

- Run `format`, select the disk, and label it with the default partition table.
- Use the `fmthard` command to write the desired label information, previously saved to a `datafile` back to the disk.

```
# fmthard -s /var/tmp/c1t3d0.vtoc /dev/rdsck/c1t3d0s2
      ● Use the fmthard command to initialize the VTOC of a disk.
# fmthard -s /dev/null /dev/rdsck/c1t3d0s2
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab is more difficult. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Working With Disks and Partitions (Level 1)

In this exercise, you complete the following tasks:

- Use the `format` utility to partition a disk
- Use the `prtvtoc` and `fmthard` commands to repair a corrupted disk label

Preparation

This exercise requires a system configured with an external disk.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Use the `format` command to list the disks currently attached to your system. Use the `prtvtoc` command to identify a disk that does not currently hold any mounted file systems. Examine the information that the `prtvtoc` command displays. Record the name of a disk that has no mount directory listed.
(Steps 1–4 in the Level 2 lab)
- Use the `format` command to divide the unused disk into four slices of equal size. Use Slices 0, 1, 3, and 4. Set all other slices to size 0. Manually change the size of Slice 0 so that it ends 25 Mbytes into the space assigned to Slice 1.
(Steps 4–11 in the Level 2 lab)

Exercise: Working With Disks and Partitions (Level 1)

- Attempt to correct the overlap by using the Modify menu. Record the message that appears. Then correct the overlap by using the `all free hog` option. Verify your disk label with the `prtvtoc` command.
(Steps 12–18 in the Level 2 lab)
- Create a directory called `/vtoc`. Run the `prtvtoc` command to read the label of the disk you modified, and save its output in a file in the `/vtoc` directory. Use the `dd` command from Step 21 of the Level 2 lab to destroy the label on the same disk. Attempt to read the disk label by using the `prtvtoc` command, and record the result. If required, use the `format` command to write a default label to the disk. Use the `fmthard` command to restore the label by using the output from the `prtvtoc` command that you saved earlier. Verify that the new label exists.
(Steps 19–25 in the Level 2 lab)

Exercise: Working With Disks and Partitions (Level 2)

In this exercise, you complete the following tasks:

- Use the `format` utility to partition a disk
- Use the `prtvtoc` and `fmthard` commands to repair a corrupted disk label

Preparation

This exercise requires a system configured with an external disk.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Use the `format` command to list the disks currently attached to your system. Use the `prtvtoc` command to identify a disk that does not currently hold any mounted file systems. Examine the Mount Directory field in the information that the `prtvtoc` command displays. Record the name of a disk that has no mount directory listed.
- Use the `format` command to divide the unused disk into four slices of equal size. Use Slices 0, 1, 3, and 4. Set all other slices to size 0. Manually change the size of Slice 0 so that it ends 25 Mbytes into the space assigned to Slice 1.

- Attempt to correct the overlap by using Option 0 from the Modify menu. Record the message that appears. Then correct the overlap by using the all free hog option. Verify your disk label with the prtvtoc command.
- Create a directory called /vtoc. Run the prtvtoc command to read the label of the disk you modified, and save its output in a file in the /vtoc directory. Use the dd command to destroy the label on the same disk. Attempt to read the disk label by using the prtvtoc command, and record the result. If required, use the format command to write a default label to the disk. Use the fmthard command to restore the label by using the output from the prtvtoc command that you saved earlier. Verify that the new label exists.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. Run the format command.
2. Record the list of disks presented by the format command, for example, c0t0d0 and c1t3d0.

Press the Control-D keys to exit the format utility.

3. Use the prtvtoc command to list the VTOC for each of the disks that you found in the previous step. Examine the Mount Directory field in the information that the prtvtoc command displays. Record the name of a disk that has no mount directory listed. For your environment, this is an unused disk.
4. Run the format command again. Select the unused disk from the list of disks presented.
5. Display the Partition menu. Print the current partition table, and record the number of megabytes assigned to Slice 2. For example, if the disk reports 8.4 Gbytes, record 8600 Mbytes. (8.4 x 1024, rounded to the nearest 10 Mbytes).

Mbytes:

6. Divide the number of megabytes by 4. Use the result as the number of megabytes to assign as disk space to four slices. Round down to the next whole megabyte if the result includes a fraction.

Mbytes/4:

7. Display the Partition menu again. Select Slice 0. Accept the defaults for tags and flags. Start this first slice on Cylinder 0. Enter the resulting number of megabytes from the previous step for the slice size. Print the partition table again to verify the change.
8. Set the sizes of Slices 1, 3, and 4 so that they are the same as Slice 0. Begin each successive slice on the cylinder that follows the ending cylinder of the previous slice.
9. Set Slices 5, 6, and 7 to start at Cylinder 0, and assign them 0 Mbytes.
10. Print the partition table. Is there any overlap of ending and beginning cylinders for any of the slices listed? Proceed to the following steps to introduce this problem.
11. Add 25 to the number Mbytes/4 value listed in Step 6.
 $(\text{Mbytes}/4) + 25:$
Change Slice 0 so that it uses the new size listed above.
The partition table should now show that Slice 0 ends after the starting cylinder for Slice 1.
12. Use the modify command from the Partition menu to attempt to fix this problem. Select Item 0 to modify the current partition table.
Which warnings appear?
13. Modify the partition table. Select Item 1 to use the All Free Hog method.
14. The partition table appears. Observe the Cylinders and Size columns, and notice that they are all zero.
15. Respond to the prompts to continue the process. Select Slice 4 as the All Free Hog slice. Use the size listed in Step 6 for Slices 0, 1, and 3. Set the other slices to Size 0.
At the end of this process, you should have three slices of equal size, where Slice 4 takes up any extra room if it exists.
16. Name the partition table "MYDISKpartition", then label the disk.
17. Quit the partition menu, and save your new partition table to the /etc/format.dat file. Carefully read the message that is displayed by the format utility, and enter the correct file name. Quit the format utility when you have finished. Use the cat command to view the contents of the /etc/format.dat file. Note that your information is appended to the file.
18. Verify your new partition table with the prtvtoc command.
19. Create a directory called /vtoc.

Exercise: Working With Disks and Partitions (Level 2)

20. Use the `prtvtoc` command to print the partition table that you just created, and save its output to a file in the `/vtoc` directory. Name the file so that it corresponds with the disk you are examining. Use the `cat` command to verify that valid information exists in the file that you create.
21. Use the following `dd` command to destroy the disk label. Be certain to specify the correct disk device name for the `of=` argument. Enter all other arguments exactly as listed.

```
# dd if=/dev/zero of=/dev/rdsck/c1t0d0s2 bs=512 count=1
1+0 records in
1+0 records out
#
```

22. Attempt to read the label from the same disk by using the `prtvtoc` command.

What happens?

23. If the `prtvtoc` command reported an "Unable to read Disk geometry" message, use the `format` command to place a default label on the disk for which you destroyed the label earlier.

If the `prtvtoc` command reports that only Slice 2 exists on the disk, skip to the next step. Otherwise, perform the commands:

```
# format
Searching for disks...done

c1t3d0: configured with capacity of 8.43GB
```

AVAILABLE DISK SELECTIONS:

0. c0t0d0 <ST38410A cyl 16706 alt 2 hd 16 sec 63>
/pci@1f,0/pci@1,1/ide@3/dad@0,0
1. c1t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
/pci@1f,0/pci@1/scsi@1/sd@3,0

Specify disk (enter its number): **1**

selecting c1t3d0

[disk formatted]

Disk not labeled. Label it now? **y**

(format menu)

```
format> q
#
# prtvtoc /dev/rdsck/c1t3d0s2
```

24. Use the `fmthard` command to write to the disk the label information you saved earlier.
25. Attempt to read the label from the same disk.
Was this successful?

Exercise: Working With Disks and Partitions (Level 3)

In this exercise, you complete the following tasks:

- Use the format utility to partition a disk
- Use the prtvtoc and fmthard commands to repair a corrupted disk label

Preparation

This exercise requires a system configured with an external disk.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Use the format command to list the disks currently attached to your system. Use the prtvtoc command to identify a disk that does not currently hold any mounted file systems. Examine the Mount Directory field in the information that the prtvtoc command displays. Record the name of a disk that has no mount directory listed.
- Use the format command to divide the unused disk into four slices of equal size. Use Slices 0, 1, 3, and 4. Set all other slices to size 0. Manually change the size of Slice 0 so that it ends 25 Mbytes into the space assigned to Slice 1.

- Attempt to correct the overlap using Option 0 from the Modify menu. Record the message that appears. Then correct the overlap by using the `all free hog` partition. Verify your disk label with the `prtvtoc` command.
- Create a directory called `/vtoc`. Run the `prtvtoc` command to read the label of the disk you modified, and save its output in a file in the `/vtoc` directory. Use the `dd` command to destroy the label on the same disk. Attempt to read the disk label by using the `prtvtoc` command, and record the result. If required, use the `format` command to write a default label to the disk. Use the `fmthard` command to restore the label by using the output from the `prtvtoc` command that you saved earlier. Verify that the new label exists.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Run the `format` command.

```
# format
```

2. Record the list of disks presented by the `format` command, for example, `c0t0d0` and `c1t3d0`.

Press Control-D to exit the `format` utility.

```
format> Control-D
```

```
#
```

3. Use the `prtvtoc` command to list the VTOC for each of the disks you found in the previous step. Examine the `Mount Directory` field in the information that the `prtvtoc` command displays. Record the name of a disk that has no mount directory listed. For your environment, this is an unused disk.

```
# prtvtoc /dev/rdsck/c1t3d0s2
```

Unused disk: *Your entry will depend on your system.*

4. Run the `format` command again. Select the unused disk from the list of disks presented.

```
# format
```

```
(list of disks)
```

```
Specify disk (enter its number): x
```

Exercise: Working With Disks and Partitions (Level 3)

5. Display the Partition menu. Print the current partition table, and record the number of megabytes assigned to Slice 2. For example, if the disk reports 8.4 Gbytes, record 8600 Mbytes. (8.4 x 1024, rounded to the nearest 10 Mbytes).

```
format> part  
partition> print
```

Mbytes: *Your entry will depend on your system.*

6. Divide the number of megabytes by 4. Use the result as the number of megabytes to assign as disk space to four slices. Round down to the next whole megabyte if the result includes a fraction.

Mbytes/4: *Your entry will depend on your system.*

7. Display the Partition menu again. Select Slice 0. Accept the defaults for tags and flags. Start this first slice on Cylinder 0. Enter the resulting number of megabytes from the previous step for the slice size. Print the partition table again to verify the change.

```
partition> 0  
Part      Tag     Flag      Cylinders      Size          Blocks  
 0 unassigned    wm        0            0      (0/0/0)          0
```

Enter partition id tag[unassigned]: <Return>

Enter partition permission flags[wm]: <Return>

Enter new starting cyl[0]: 0

Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: 2150mb

```
partition> print  
(partition table)
```

8. Set the sizes of Slices 1, 3, and 4 so that they are the same as Slice 0. Begin each successive slice on the cylinder that follows the ending cylinder of the previous slice.

```
partition> ?  
(Partition menu)  
partition> 1  
Part      Tag     Flag      Cylinders      Size          Blocks  
 1 unassigned    wm        0            0      (0/0/0)          0  
Enter partition id tag[unassigned]: <Return>  
Enter partition permission flags[wm]: <Return>  
Enter new starting cyl[0]: 1227  
Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: 2150mb  
partition> print  
(partition table)
```

9. Set Slices 5, 6, and 7 to start at Cylinder 0, and assign them 0 Mbytes.

```
partition> ?
(Partition menu)
partition> 5
Part      Tag      Flag      Cylinders      Size          Blocks
 5 unassigned    wm          0           0   (0/0/0)          0

Enter partition id tag[unassigned]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 0
Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: 0m
partition>
```

10. Print the partition table. Is there any overlap of ending and beginning cylinders for any of the slices listed? Proceed to the following steps to introduce this problem.

```
partition> print
11. Add 25 to the number Mbytes/4 value listed in Step 6.
(Mbytes/4) + 25: Your entry will depend on your system.
Change Slice 0 so that it uses the new size listed previously.
```

```
partition> ?
(Partition menu)
partition> 0
Part      Tag      Flag      Cylinders      Size          Blocks
 0 unassigned    wm      0 - 1226      2.10GB      (1227/0/0)  4406157

Enter partition id tag[unassigned]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 0
Enter partition size[4406157b, 1227c, 1226e, 2151.44mb, 2.10gb]: 2175mb
partition> print
(partition table)
```

The partition table should now indicate that Slice 0 ends after Slice 1 begins.

Exercise: Working With Disks and Partitions (Level 3)

12. Use the modify command from the Partition menu to attempt to fix this problem. Select Item 0 to modify the current partition table.

```
partition> ?
(Partition menu)
partition> modify
Select partitioning base:
 0. Current partition table (unnamed)
 1. All Free Hog
Choose base (enter number) [0]? 0
```

Which warnings display?

Warning: Overlapping partition (1) in table.

Warning: Fix, or select a different partition table.

13. Modify the partition table. Select Item 1 to use the All Free Hog option.

```
partition> modify
Select partitioning base:
 0. Current partition table (original)
 1. All Free Hog
Choose base (enter number) [0]? 1
```

14. The partition table appears. Observe the Cylinders and Size columns, and notice that they are all zero; for example:

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0	0	(0/0/0) 0
1	swap	wu	0	0	(0/0/0) 0
2	backup	wu	0 - 4923	8.43GB	(4924/0/0) 17682084
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	usr	wm	0	0	(0/0/0) 0
7	unassigned	wm	0	0	(0/0/0) 0

15. Respond to the prompts to continue the process. Select Slice 4 as the All Free Hog partition. Use the size listed in Step 6 for Slices 0, 1, and 3. Set the other slices to Size 0.

Do you wish to continue creating a new partition
table based on above table[yes]? **yes**

```
Free Hog partition[6]? 4
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 2150m
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 2150m
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]: 2150m
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]: 0
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]: 0
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]: 0
```

At the end of this process, you should have three slices of equal size, where Slice 4 takes up any extra room if it exists.

16. Name the partition table "MYDISKpartition", then label the disk.

```
Okay to make this the current partition table[yes]? y
Enter table name (remember quotes): "MYDISKpartition"
```

```
Ready to label disk, continue? y
```

```
partition>
partition> quit
(format menu)
format>
```

17. Save your new partition table to the /etc/format.dat file. Carefully read the message that is displayed by the format utility, and enter the correct file name. Quit the format utility when you have finished. Use the cat command to view the contents of the /etc/format.dat file. Note that your information is appended to the file.

```
format> save
Saving new disk and partition definitions
Enter file name["./format.dat"]: /etc/format.dat
format> quit
#
# cat /etc/format.dat
```

18. Verify your new partition table with the prtvtoc command.

```
# prtvtoc /dev/rdsck/c1t3d0s2
```

19. Create a directory called /vtoc.

```
# mkdir /vtoc
```

20. Use the prtvtoc command to print the partition table that you just created, and save its output to a file in the /vtoc directory. Name the file so that it corresponds with the disk you are examining. Use the cat command to verify that valid information exists in the file that you create.

```
# prtvtoc /dev/rdsck/c1t3d0s2 > /vtoc/c1t3d0
# cat /vtoc/c1t3d0
```

Exercise: Working With Disks and Partitions (Level 3)

21. Use the following dd command to destroy the disk label. Be certain to specify the correct disk device name for the of= argument. Enter all other arguments exactly as listed.

```
# dd if=/dev/zero of=/dev/rdsck/c1t3d0s2 bs=512 count=1
1+0 records in
1+0 records out
#
```

22. Attempt to read the label from the same disk by using the prtvtoc command.

```
# prtvtoc /dev/rdsck/c1t3d0s2
```

What happens?

Different disk types present different results. SCSI disks might report messages that indicate that the disk label is unreadable, for example:

prtvtoc: /dev/rdsck/c1t3d0s2: Unable to read Disk geometry errno = 0x16

IDE disks might report a partition table where only Slice 2 remains defined, for example:

Partition	Tag	Flags	Sector	Count	Sector
Mount	Directory				
	2		5	01	0
17801280		17801279			

23. If the `prtvtoc` command reported an "Unable to read Disk geometry" message, use the `format` command to place a default label on the disk for which you destroyed the label earlier.

If the `prtvtoc` command reports that only Slice 2 exists on the disk, skip to the next step. Otherwise, perform the commands:

```
# format
```

```
Searching for disks...done
```

```
c1t3d0: configured with capacity of 8.43GB
```

```
AVAILABLE DISK SELECTIONS:
```

- 0. c0t0d0 <ST38410A cyl 16706 alt 2 hd 16 sec 63>
 /pci@1f,0/pci@1,1/ide@3/dad@0,0
- 1. c1t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
 /pci@1f,0/pci@1/scsi@1/sd@3,0

```
Specify disk (enter its number): 1
```

```
selecting c1t3d0
```

```
[disk formatted]
```

```
Disk not labeled. Label it now? y
```

```
(format menu)
```

```
format> q
```

```
#
```

```
# prtvtoc /dev/rdsck/c1t3d0s2
```

24. Use the `fmthard` command to write to the disk the label information you saved earlier.

```
# fmthard -s /vtoc/c1t3d0 /dev/rdsck/c1t3d0s2
```

```
fmthard: New volume table of contents now in place.
```

```
#
```

25. Attempt to read the label from the same disk.

```
# prtvtoc /dev/rdsck/c1t3d0s2
```

Was this successful?

This command should successfully read the disk label.

Introducing the Solaris™ Management Console

The Solaris Management Console is a Java technology-based tool for the administration of systems. It provides a central integration point for the configuration and administration of important applications and services.

The Solaris Management Console software simplifies the job of configuring and administering servers. With point-and-click graphical user interface (GUI) tools, the Solaris Management Console makes the Solaris OS easy to administer, especially for administrators who are not familiar with the UNIX environment.

Starting the Solaris Management Console

The Solaris Management Console can be started from the command line or from within the Application Manager by clicking the Solaris Management Console icon.

Log in to your system as root, and type `smc&` in a terminal window. You can start the Solaris Management Console as a normal user, but some tools and applications are not available to you. When you initiate the Solaris Management Console for the first time, it can take a few minutes to launch.

Note – The information provided in this course is only a small subset of the overall capabilities of the Solaris Management Console.



Using the Solaris Management Console Tools

The default toolbox for a Solaris Management Console server includes the following folders and tools:

System Status	This category includes System Information, Log Viewer, Processes, and Performance.
System Configuration	This category includes Users, Projects, Computers and Networks, and Patches.
Services	This category includes Scheduled Jobs.
Storage	This category includes Mounts and Shares, Disks, and Enhanced Storage.
Devices and Hardware	This category includes Serial Ports.

The Solaris Management Console enables local users and administrators to register remote Solaris Management Console servers and applications on the network they want to administer. When you access the Solaris Management Console, it dynamically configures tree views of those registered hosts and services. Point and click with the mouse to invoke an application remotely on a selected Solaris Management Console server and view the application's GUI on the local display.

Introducing the Help Screen

The online help for the Solaris Management Console provides an alternative to standard documentation. The information panes that appear in both the Solaris Management Console and the Solaris Management Console Toolbox Editor provide the steps necessary to perform the tasks executed within these windows. In addition, the Help menu item Contents displays a window that further describes the features and functions of the window components.

Figure 3-14 shows the help functionality of the Solaris Management Console.

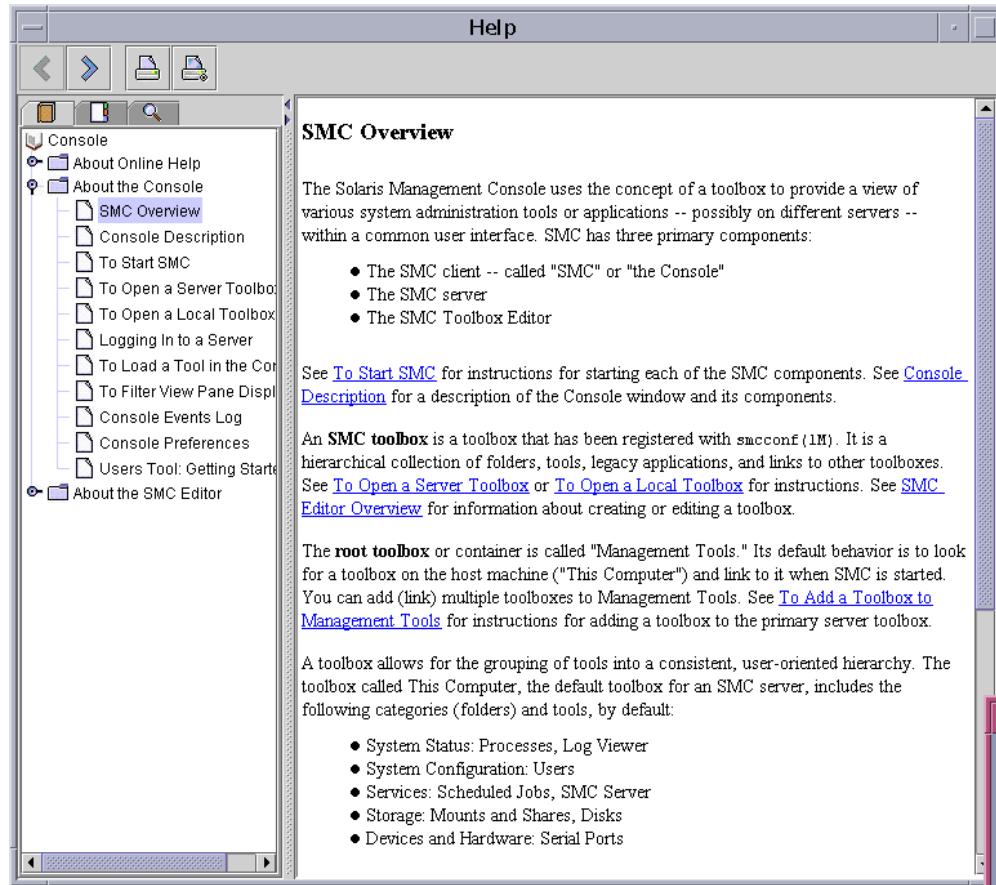


Figure 3-14 Solaris Management Console Help View

Restarting the Solaris Management Console

If you have trouble accessing Solaris Management Console, the reason might be that the Solaris Management Console server is not running or is in a problem state.

To determine if the Solaris Management Console server is running, perform the command:

```
# /etc/init.d/init.wbem status
```

If the Solaris Management Console server is running, a response similar to the following returns: "Solaris Management Console server version 2.1.0 running on port 898."



Note – If this is the first time SMC has been run after a reboot, this command may show an error.

To stop the Solaris Management Console server, as the root user, perform the command:

```
# /etc/init.d/init.wbem stop
```

The following response returns: "SMC stopped."

To start the Solaris Management Console server, as the root user, perform the command:

```
# /etc/init.d/init.wbem start
```

After a short time, the following response returns: "SMC server started."

Identifying the Functional Areas of the Solaris Management Console

The Solaris Management Console and the Solaris Management Console Toolbox Editor windows are divided into functional areas as follows:

- Navigation pane
- View pane
- Information pane
- Location bar
- Status bar

Figure 3-15 shows these divisions.

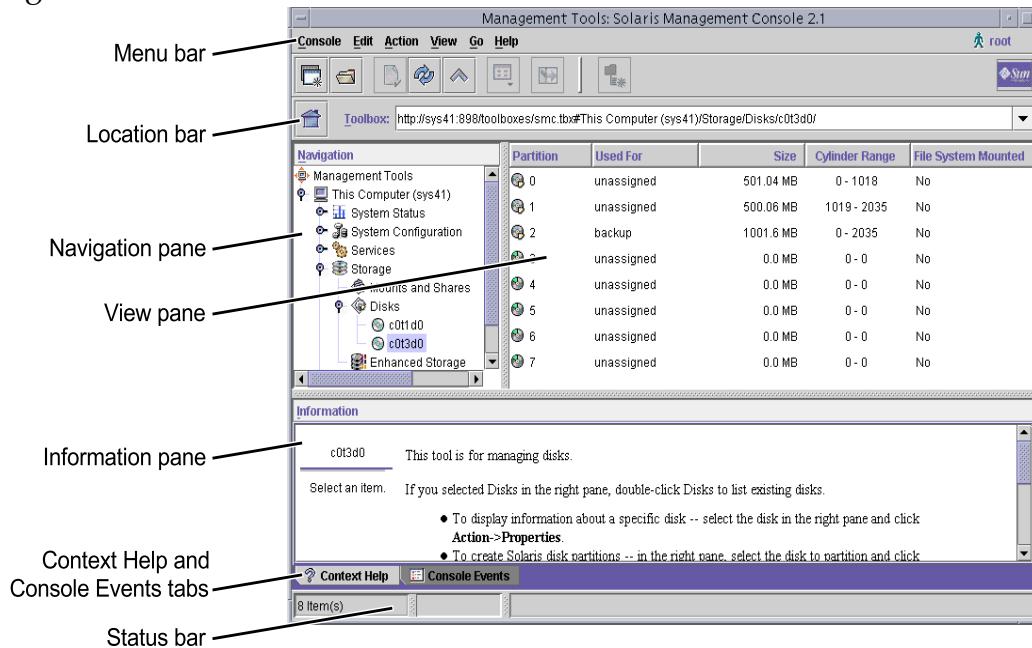


Figure 3-15 Solaris Management Console Overview

Note – The Location bar does not appear by default when you first launch the Solaris Management Console. Click View on the Menu bar, select the Show option, and select the Location option to display the Location bar.



Navigation Pane

The Navigation pane works like a frame in a web page. Clicking an item in the Navigation pane determines what appears in the View pane. The turner icon is displayed to the left of items that represent a group of items. Click the icon or the item to expand or collapse the group.

The Navigation pane is displayed or not displayed, depending on the Show setting in the View menu. Click View on the Menu bar, select the Show option, and select or deselect the Navigation option.

View Pane

The View pane displays the contents of the node selected in the Navigation pane. The contents could be a folder or a tool.

If the node selected in the Navigation pane is a folder, the View pane displays the contents of that folder.

If the node selected is a simple tool, such as the Process tool, the View pane displays a list of current processes. If the node selected is a complex tool, such as User Manager, the View pane displays additional tools, such as the tools for user accounts and email accounts. Select one of the additional tools, such as the user accounts node, and the View pane displays the contents of the tool.

Information Pane

The Information pane at the bottom of the Solaris Management Console window displays either context help for the object selected in the Navigation pane or a list of events and alarms for all Solaris Management Console events.

The Context Help tab and Console Events tab determine what is shown in the Information pane. Click the Context Help tab to display context help for the object selected. Click the Console Events tab to display a list of events and alarms for all Console events.

The Information pane is displayed or not displayed, depending on the Show setting in the View menu. Click View on the Menu bar, select the Show option, and select or deselect the Information option.

Location Bar

The Location bar, beneath the tool bar in the Solaris Management Console window, displays a Home Toolbox icon and a Toolbox field. Click the Home Toolbox icon to open the home toolbox. The Toolbox field indicates the current toolbox and the item currently selected in the toolbox. Click the button to the right of the Toolbox field to display a pull-down menu of recent toolboxes visited. Select a toolbox from the pull-down menu to open that toolbox.

The Location bar is displayed or not displayed, depending on the Show setting in the View menu. Click View on the Menu bar, select the Show option, and select or deselect the Location option.

Status Bar

The Status bar, located across the bottom of the Solaris Management Console window, displays three panes. The left pane of the Status bar indicates the number of nodes directly subordinate to the node selected in the Navigation pane. The center pane of the Status bar indicates Console activity. A moving bar inside the center pane functions as an activity indicator when Console activity occurs. The right pane of the Status bar provides progress information during some Console tasks.

The Status Bar is displayed or not displayed, depending on the Show setting in the View menu. Click View on the Menu bar, select the Show option, and select or deselect the Status bar option.

Partitioning a Disk by Using the Solaris Management Console Disks Manager Tool

The following section describes how to partition a disk by using the Solaris Management Console Disks Manager Tool (from this point on, referred to as the Disks Tool).

Partitioning the Disk Using the Disks Tool

To partition a disk by using the Disks Tool, you must first locate the Storage folder within the Navigation pane. The Storage folder consists of the Mounts and Shares folder, the Disks Tool, and the Enhanced Storage tools.

Use the Disks Tool to perform the following tasks:

- Display information about a specific disk
- Create Solaris OS disk partitions
- List partitions
- Copy the layout of one disk to another disk of the same type
- Change the disk's label

Partitioning a Disk by Using the Solaris Management Console Disks Manager Tool

Perform the following steps to partition a disk by using Disks Tool:

1. Click Storage and then the Disks Tool. The Log In: User Name window appears, prompting you to enter the root password.

Figure 3-16 shows the Log In: User Name window.

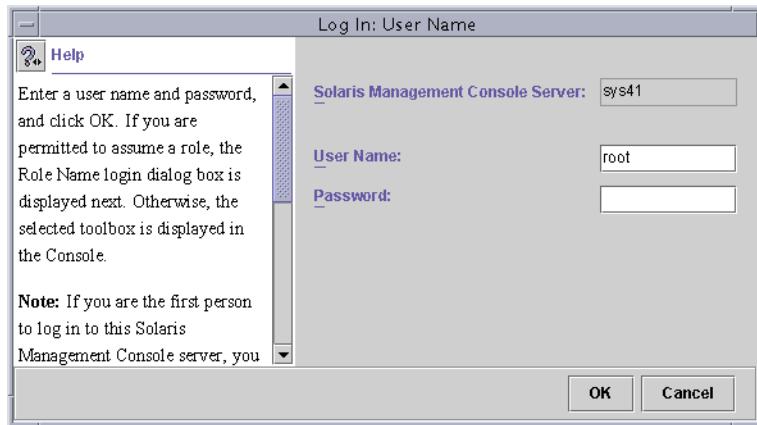


Figure 3-16 Log In: User Name Window

Figure 3-17 shows the Solaris Management Console after you have opened the Storage folder and then the Disks Tool. The figure shows a system with two disks.

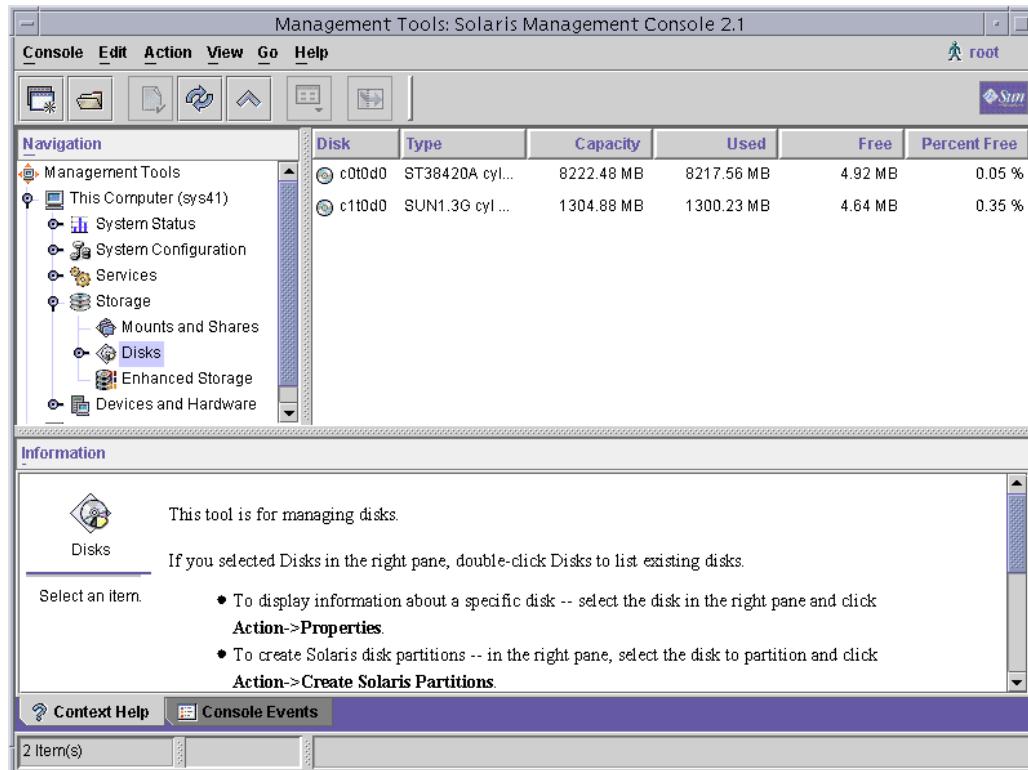


Figure 3-17 Management Tools: Solaris Management Console Window

2. Click to select a specific disk. Then click the Action menu on the Menu bar.

The Action menu displays a list of functions that this window performs.

3. To display a graphical representation of a disk's partitioning, select the Properties option from the Action menu.

Figure 3-18 shows a 1.3 Gbyte drive.

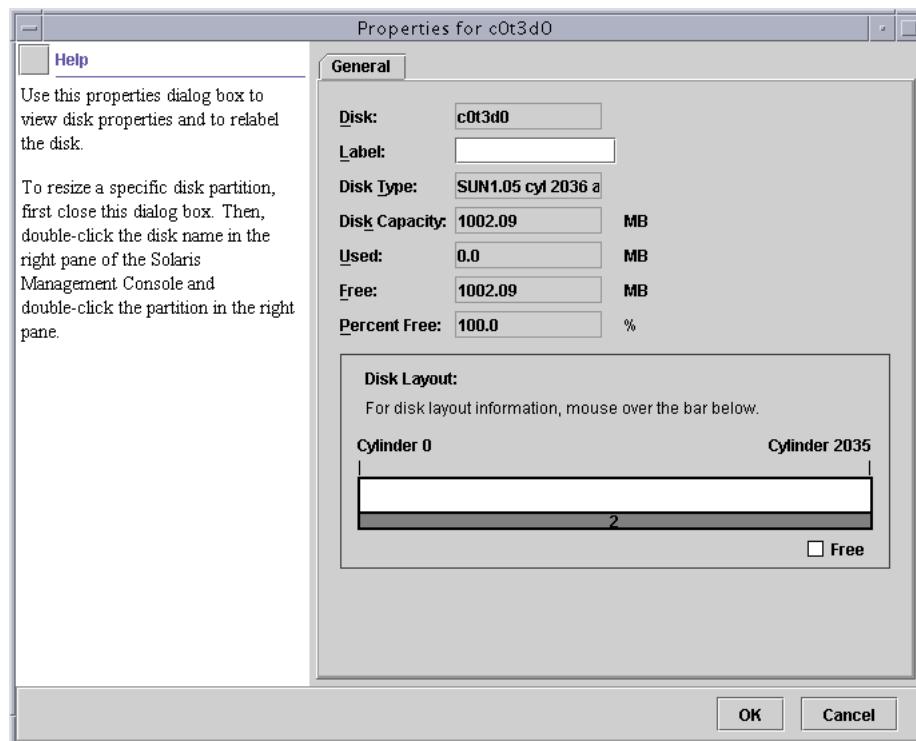


Figure 3-18 Properties Window

Basic disk information, including size, address, and available free space, is reported. Move the cursor over any partition on the Disk Layout bar to see the size and geometry of the partition slice in a pop-up window.

4. To create a new partition map on a disk, select the Create Solaris Partitions option from the Action menu.

Figure 3-19 shows the first window that you use to create partitions on a disk. This window prompts you to choose between creating custom-sized partitions and creating equal-sized partitions. In the figure, Create Custom-Sized Partitions is selected.

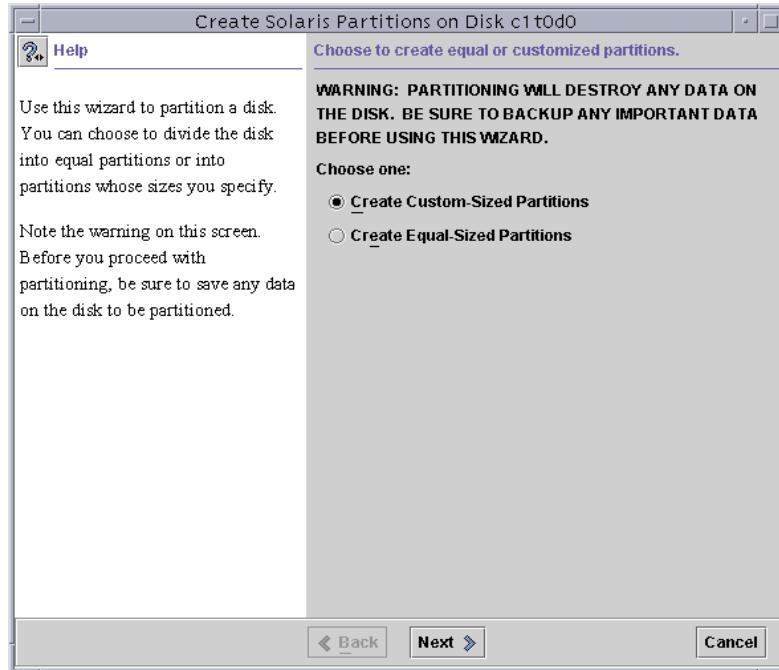


Figure 3-19 Create Solaris Partitions on Disk Window

5. Click Next after choosing how to divide the disk.

Figure 3-20 shows the next window that you use to create partitions on a disk. You are prompted to select the number of partitions. You can select up to seven partitions. You can also create some of them as zero-length partitions.

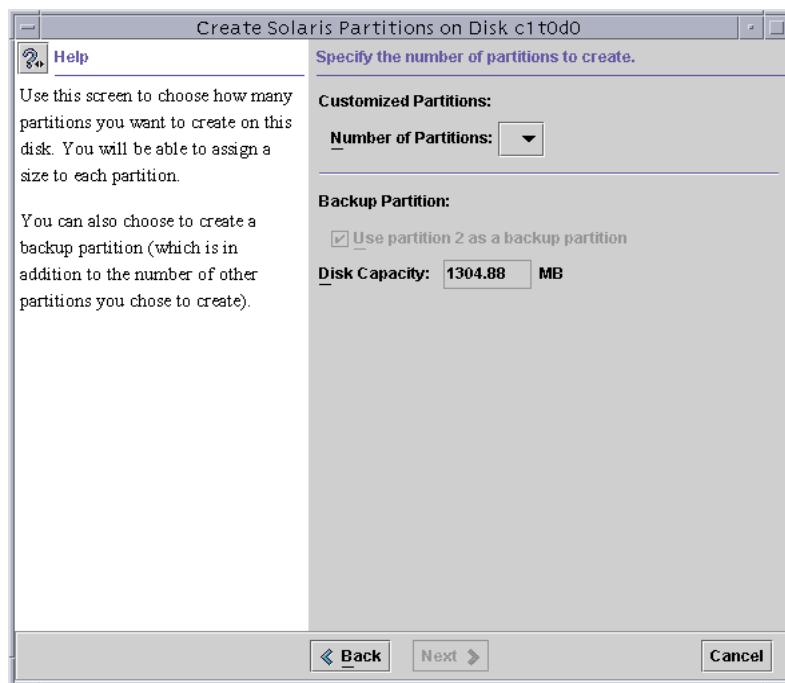


Figure 3-20 Create Solaris Partitions on Disk Window – Specify Number of Partitions

6. After selecting the number of partitions, click Next.

Figure 3-21 shows the window that enables you to display each partition. When a partition is displayed, the size of the partition is also displayed. You can choose to display the size of the partition in either a percentage of the disk space or the total number of megabytes, and you can adjust the size of each partition. The disk layout bar graphically represents the disk partitions. Place the cursor over the bar to view the amount of space that remains to be partitioned.

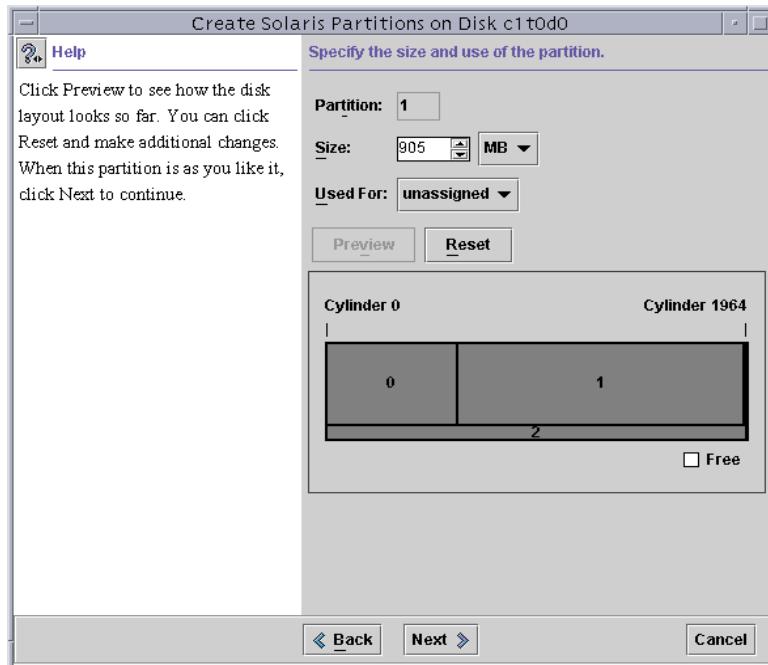


Figure 3-21 Create Solaris Partitions on Disk Window – Specify Size and Use of Partitions

7. Use this window to adjust the size of each partition to the desired size. Click Next when you have finished sizing the partitions.

Figure 3-22 shows the window that allows you to specify the partitions on which to create file systems.

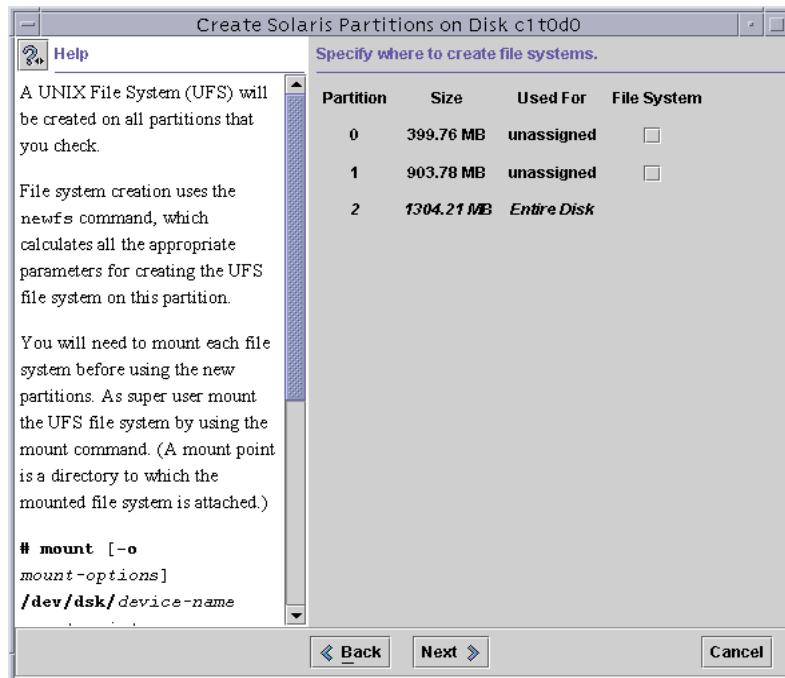


Figure 3-22 Create Solaris Partitions on Disk Window – Specify Where to Create File Systems

Partitioning a Disk by Using the Solaris Management Console Disks Manager Tool

8. In the Create Solaris Partitions on Disk window, check the box under the file system that corresponds to each partition you want to use. Click Next when you are finished making your selections.

Figure 3-23 displays a list of the disk partitions you have created.

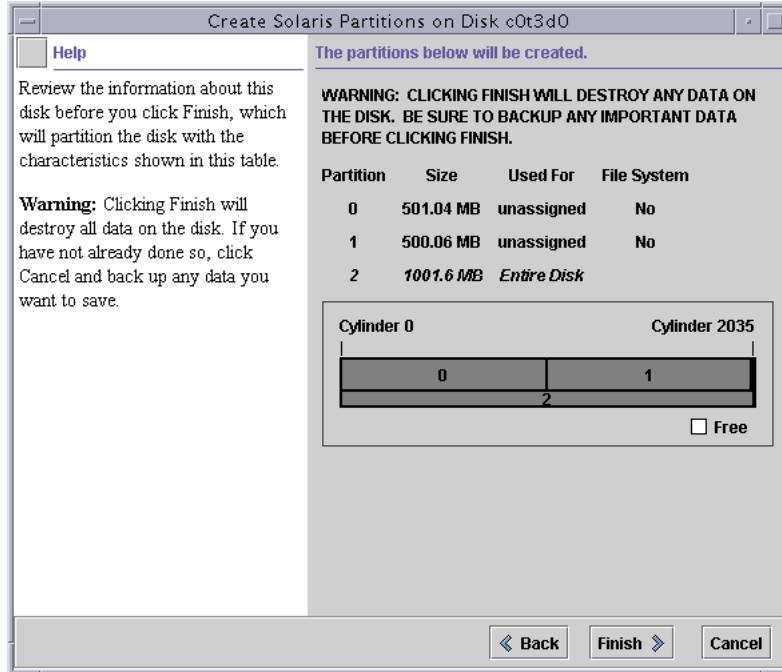


Figure 3-23 Create Solaris Partitions on Disk Window – Confirmation

9. If you are satisfied with the partitions, click Finish. The new partitioning is written, and the newfs utility runs on the partitions you selected to create a new file system.

Figure 3-24 displays the disks window of the Solaris Management Console after you have completed partitioning the disk. The created partitions are displayed in the Management Tools: Solaris Management Console window.

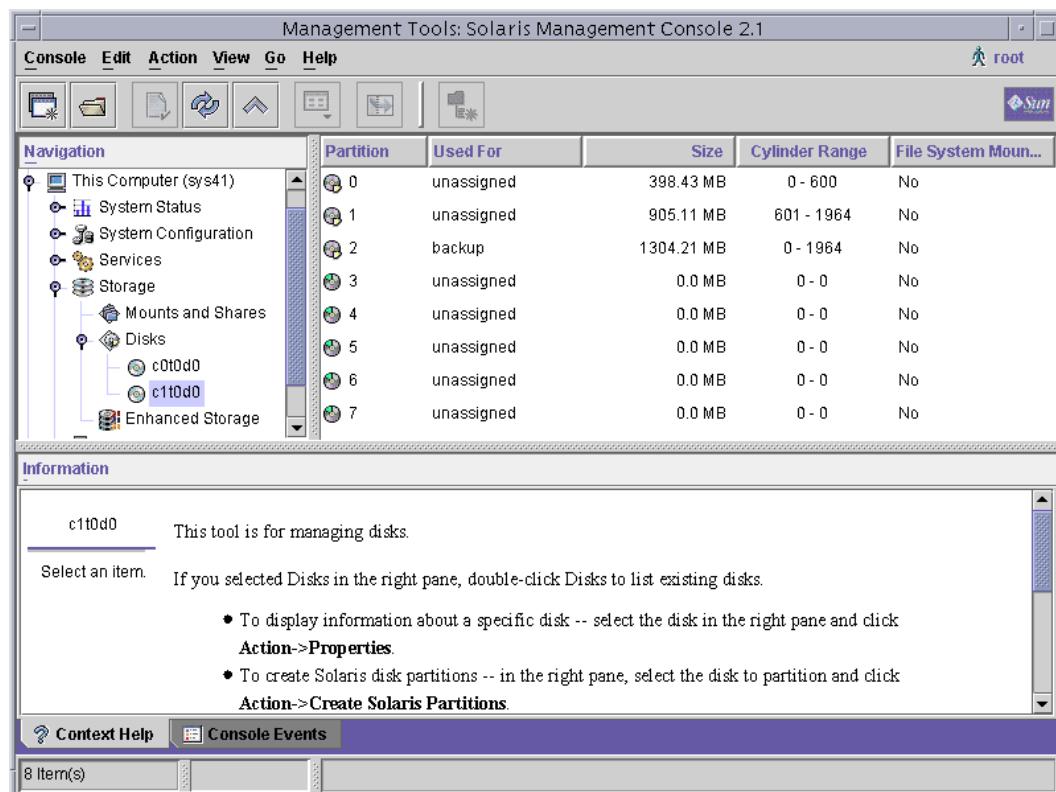


Figure 3-24 Management Tools: Solaris Management Console Window – Partitioning Completed

Performing the Exercises

You have the option to complete either of these labs. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab is guided. Although each step describes what you should do, you must determine which commands (and options) to input.

There are only two levels of this lab due to the nature of working within the Solaris Management Console GUI. Should you require assistance with any of the steps, consult the help functionality from the Solaris Management Console.

Exercise: Working With the Solaris Management Console (Level 1)

In this exercise, you complete the following tasks:

- Launch the Solaris Management Console Disks Manager Tool
- Partition the second drive of your system to match the boot drive

Preparation

This exercise requires a system with at least two disks, one of which is available for the student to re-partition.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fnl.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Launch the Solaris Management Console, and choose the Disks Tool from the Storage folder.
- Authenticate as the root user by typing the root password.
- View information about your boot drive from the Disks Tool, and make note of it.
- On your spare hard drive, make four equal sized partitions on Slices 0, 1, 3, and 4.

Exercise: Working With the Solaris Management Console (Level 2)

In this exercise, you complete the following tasks:

- Launch the Solaris Management Console Disks Manager Tool
- Partition the second drive of your system to match the boot drive

Preparation

This exercise requires a system with at least two disks, one of which is available for the student to re-partition.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

Complete the following tasks:

- Launch the Solaris Management Console, and choose the Disks Tool from the Storage folder.
- Authenticate as the `root` user by typing the `root` password.
- View information about your boot drive from the Disks Tool, and make note of it.
- On your spare hard drive, make four equal sized partitions on Slices 0, 1, 3, and 4.

Tasks

Complete the following steps:

1. Launch the Solaris Management Console from the command line or by using Application Manager.
2. Open the Disks Tool.
3. Select your boot drive from the Disks Tool, and record the partition information listed.
4. Select your spare drive from the Disks Tool. Select Create Solaris Partitions from the Action menu.
5. Choose Create Equal-Sized Partitions, and click Next.
6. Specify Number of Partitions as 4. Click Next.
7. Verify that you have four equal-sized partitions on Slices 0, 1, 3, and 4. Click Next.
8. Check the box beside Slice 4 to create a File System, and click Next.
9. After reviewing your choices and verifying that they are correct, click Finish.

The Solaris Management Console window refreshes, and you should see the four equal-sized partitions listed in the View Pane.

10. Exit from the Solaris Management Console.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 4

Managing Solaris OS File Systems

Objectives

Upon completion of this module, you should be able to:

- Describe Solaris OS file systems
- Create a new `ufs` file system
- Check the file system by using the `fsck` command
- Resolve file system inconsistencies
- Monitor file system use

The course map in Figure 4-1 map shows how this module fits into the current instructional goal.

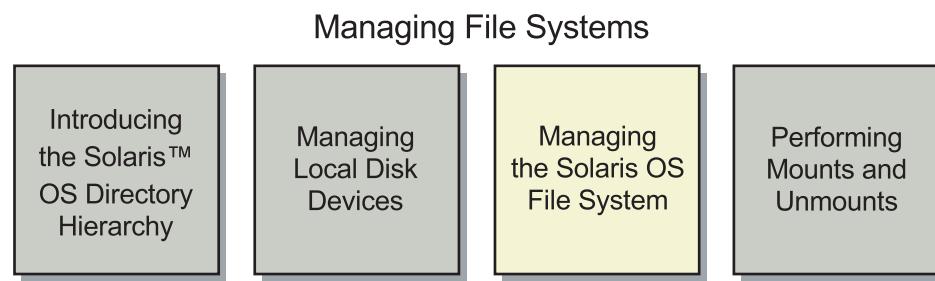


Figure 4-1 Course Map

Introducing Solaris OS File Systems

A file system is a collection of files and directories that make up a structured set of information. The Solaris OS supports three different types of file systems:

- Disk-based file systems
- Distributed file systems
- Pseudo file systems

Disk-based File Systems

Disk-based file systems are found on hard disks, CD-ROMs, diskettes, and DVDs. The following are examples of disk-based file systems:

- ufs – The UNIX file system in the Solaris OS is based on the Berkeley fast file system. Enhancements in the Solaris 10 OS allow the ufs to grow to multiple terabytes in size.
- hsfs – The High Sierra file system is a special-purpose file system developed for use on CD-ROM media.
- pcfs – The PC file system is a UNIX implementation of the disk operating system (DOS) file allocation table (FAT32) file system. The pcfs file system allows the Solaris OS to access PC-DOS formatted file systems. Users can use UNIX commands for direct read and write access to PC-DOS files.
- udfs – The Universal Disk Format file system is used for optical storage targeted at DVD and CD-ROM media. The UDF file system allows universal data exchange and supports read and write operations.

Distributed File Systems

Distributed file systems provide network access to file system resources.

- NFS – The network file system allows users to share files among many types of systems on the network. The NFS file system makes part of a file system on one system appear as though it were part of the local directory tree.

Pseudo File Systems

Pseudo file systems are memory based. These file systems provide for better system performance, in addition to providing access to kernel information and facilities. Pseudo file systems include:

- tmpfs – The temporary file system stores files in memory, which avoids the overhead of writing to a disk-based file system. The tmpfs file system is created and destroyed every time the system is rebooted.
- swapfs – The swap file system is used by the kernel to manage swap space on disks.
- fdfs – The file descriptor file system provides explicit names for opening files by using file descriptors (for example, /dev/fd/0, /dev/fd/1, /dev/fd/2) in the /dev/fd directory.
- procfs – The process file system contains a list of active processes in the /proc directory. The processes are listed by process number. Information in this directory is used by commands, such as the ps command.
- mntfs – The mount file system provides read-only information from the kernel about locally mounted file systems.
- objfs – The kernel object file system. This file system is used by the kernel to store details relating to the modules currently loaded by the kernel. The object file system is used for the /system/object directory.
- devfs – The device file system is used to manage the namespace of all devices on the system. This file system is used for the /devices directory.
- ctfs – The contract file system is associated with the /system/contract directory. This is used by the Service Management Facility to track the processes which compose a service, so that a failure in a part of a multi-process service can be identified as a failure of that service.

Creating a New ufs File System

This section describes the ufs file system in the Solaris OS.

Viewing the Solaris OS ufs File System

The user views the ufs file system differently than the operating system does in the Solaris OS. To a user, a file system appears as a collection of files and directories used to store and organize data for access by the system and its users. To the operating system, a file system is a collection of control structures and data blocks that occupy the space defined by a partition, which allow for data storage and management.

The Solaris OS stores data in a logical file hierarchy often consisting of several file systems. This file hierarchy is referred to as the Solaris directory hierarchy.

Figure 4-2 shows the Solaris OS hierarchy beginning with the / (root) directory.

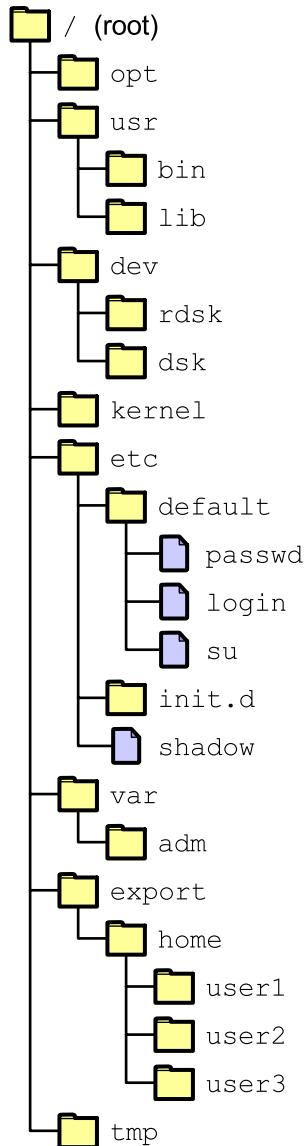


Figure 4-2 Solaris OS Directory Hierarchy

Note – Figure 4-2 is not a complete representation of a Solaris OS directory hierarchy.



Creating a New ufs File System

A ufs file system is created on a disk slice before it is used in the Solaris OS. Creating a ufs file system on a disk slice enables the Solaris OS to store UNIX directories and files.

Figure 4-3 shows how the ufs file systems are located on various disk slices.

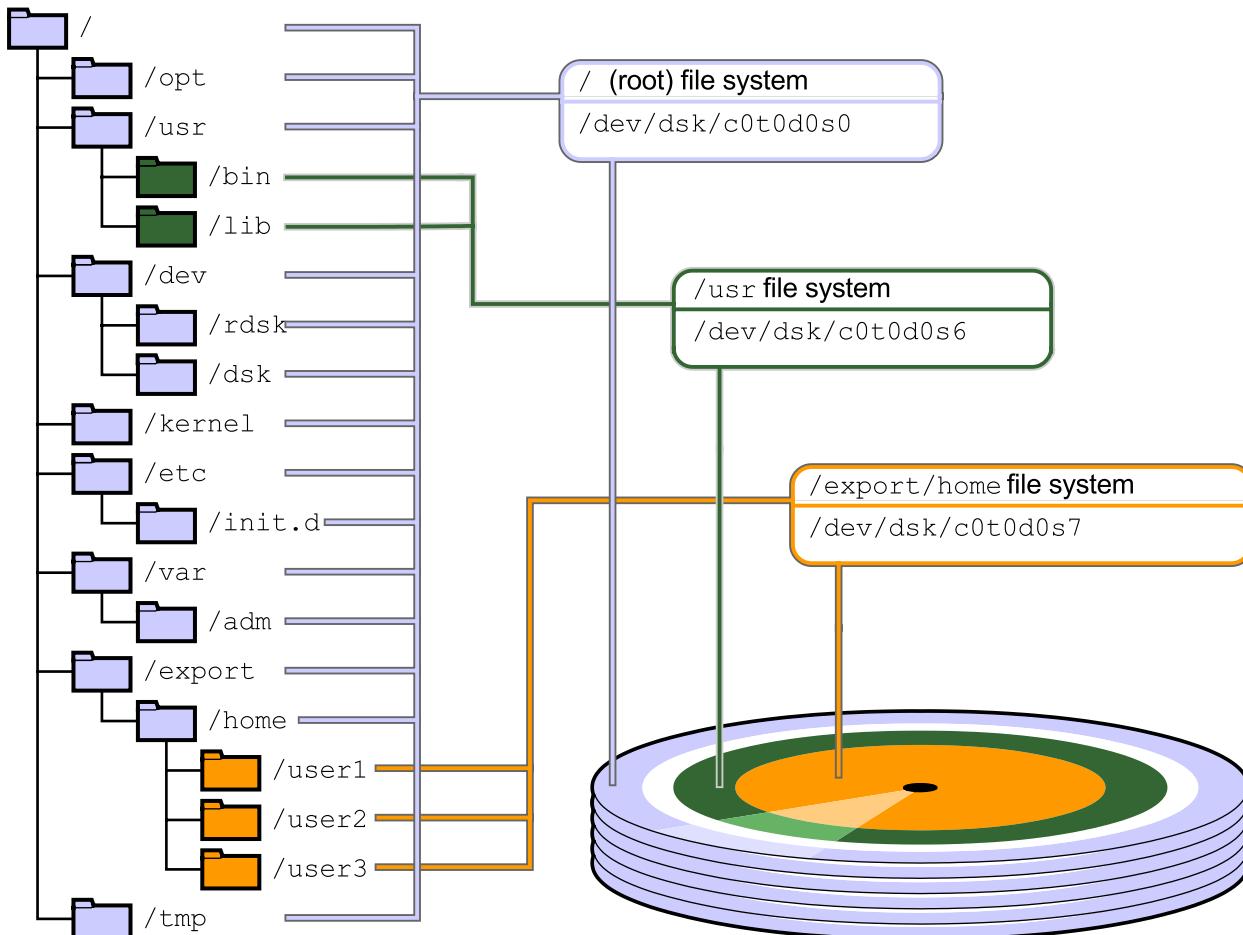


Figure 4-3 Solaris ufs File Systems Residing on Disk Slices

The Solaris OS ufs file system contains the following basic support structures.

Disk Label (VTOC)

The disk label (VTOC) contains the partition table for the disk. The VTOC resides in the first disk sector (512-byte blocks). Only the first disk slice contains a VTOC, although file systems created on any slice reserve the first sector to allow for a VTOC.

Boot Block

The bootstrap program (bootblk) resides in the 15 disk sectors (Sectors 1–15) that follow the VTOC. Only the / (root) file system has an active boot block. However, space is allocated for a boot block at the beginning of each file system.

Primary Superblock

The superblock resides in the 16 disk sectors (Sectors 16–31) that follow the boot block. The superblock is a table of information that describes the file system, including:

- The number of data blocks
- The number of cylinder groups
- The size of a data block and fragment
- A description of the hardware, derived from the label
- The name of the mount point
- File system state flag: clean, stable, active, logging, or unknown

Backup Superblocks

When the file system is created, backup copies of the superblock are created beginning at sector 32. This replication protects the critical data in the superblock against catastrophic loss.

Cylinder Groups

Each file system is divided into cylinder groups with a minimum default size of 16 cylinders per group. Cylinder groups improve disk access.

The file system constantly optimizes disk performance by attempting to place a file's data into a single cylinder group, which reduces the distance a head has to travel to access the file's data. The file system stores large files across several cylinder groups, if needed.

Cylinder Group Blocks

The cylinder group block is a table in each cylinder group that describes the cylinder group, including:

- The number of inodes
- The number of data blocks in the cylinder group
- The number of directories
- Free blocks, free inodes, and free fragments in the cylinder group
- The free block map
- The used inode map

Figure 4-4 shows a series of cylinder groups in a ufs file system.

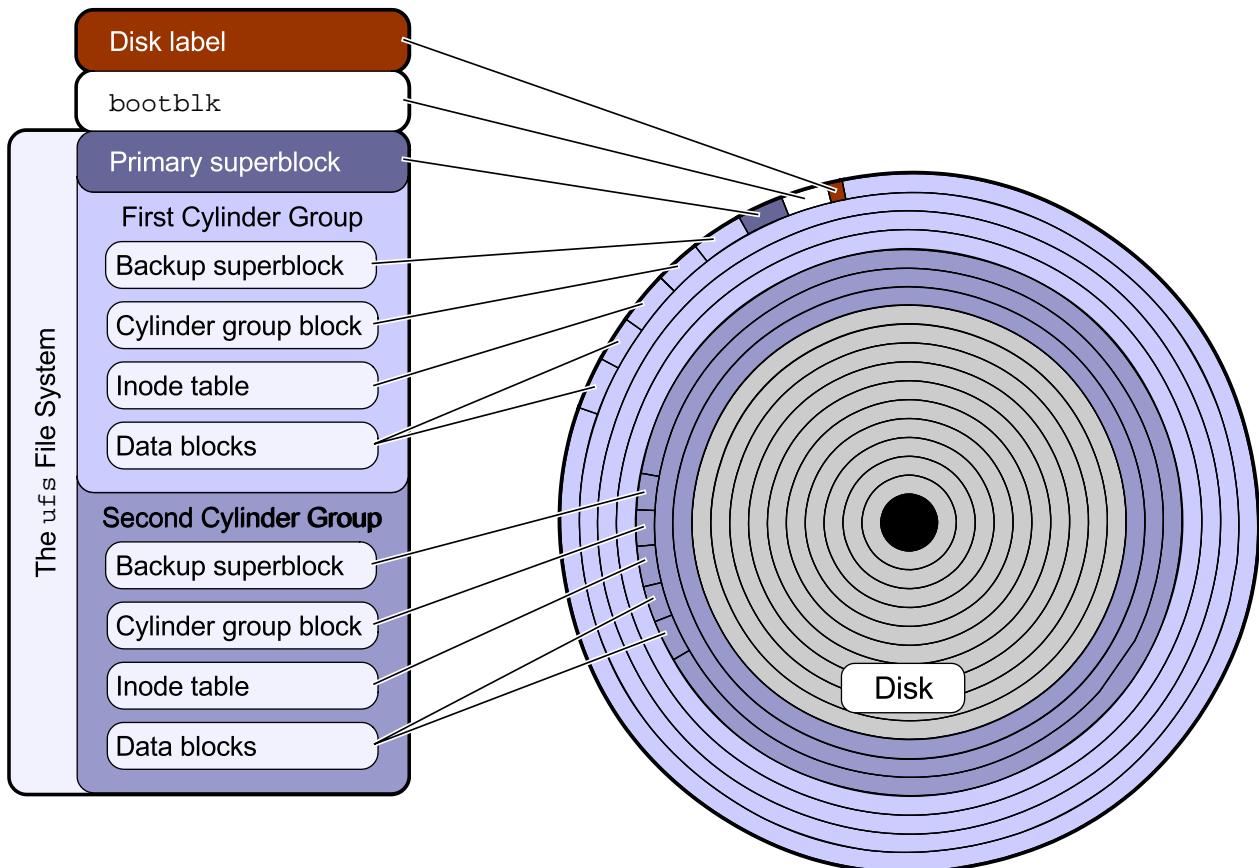


Figure 4-4 Solaris ufs File System Structure

The ufs Inode

An inode contains the following information about a file:

- The type of file and the access modes
- The user identification (UID) and group identification (GID) numbers of the file's owner and group
- The size of the file
- The link count
- The time the file was last accessed and modified and the inode changed
- The total number of data blocks used by or allocated to the file
- Two types of pointers: direct pointers and indirect pointers

Creating a New ufs File System

Figure 4-5 shows some of the information contained in an inode.

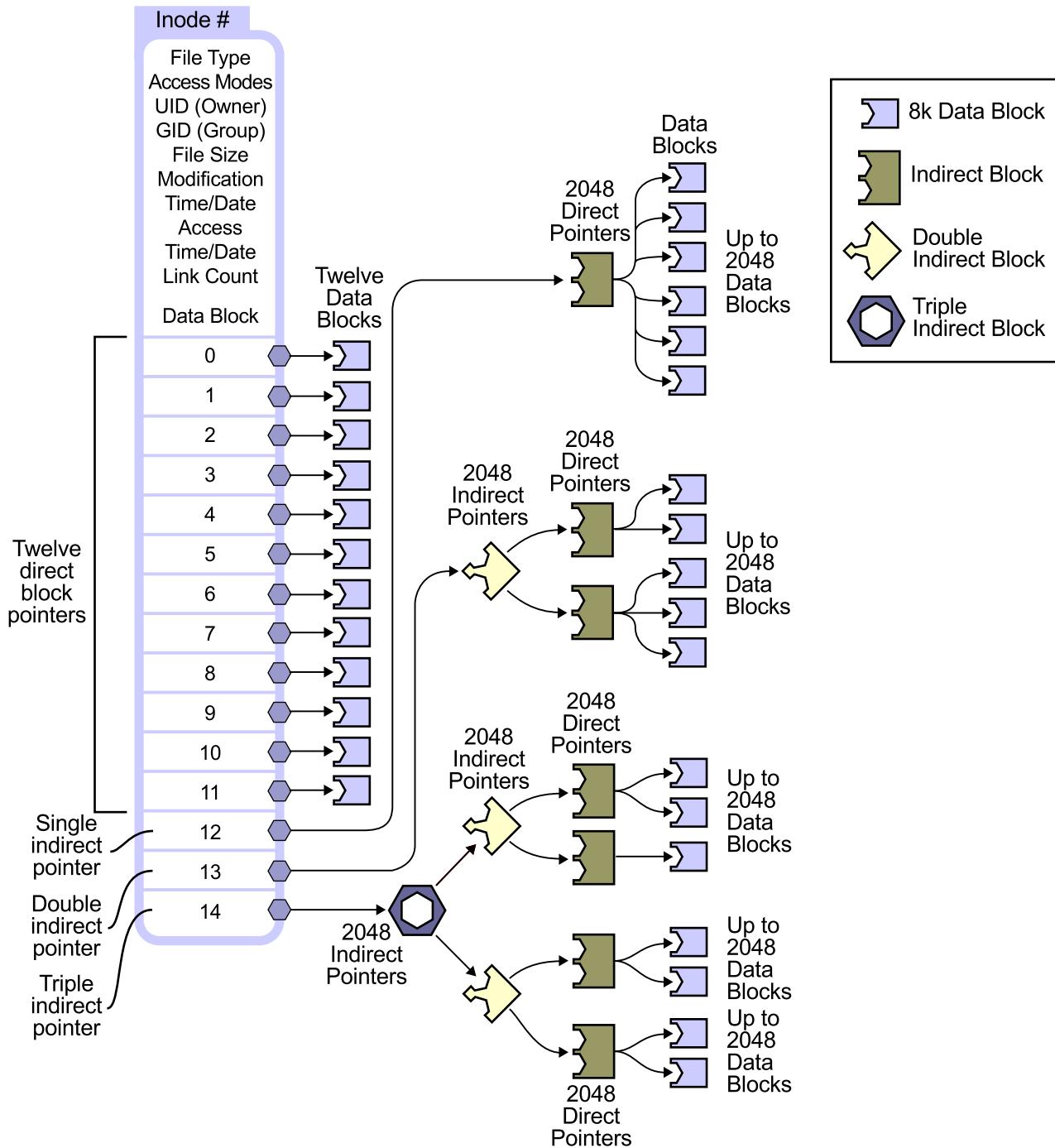


Figure 4-5 Structure of a ufs Inode



Note – To view some of the information contained in a file or directory inode, use the `ls -l` command. To view the inode number assigned to the file or directory, use the `ls -i` command.

Direct Pointers

Inside the inode there are 12 direct pointers, which contain addresses for the file's first 12 data blocks. The 12 direct pointers can each reference 8-Kbyte data blocks for a file that is up to 96 Kbytes.

Indirect Pointers

The three types of indirect pointers within an inode are:

- Single indirect pointer – Refers to a file system block that contains pointers to data blocks. This file system block contains 2048 additional addresses of 8-Kbyte data blocks, which can point to an additional 16 Mbytes of data.
- Double indirect pointer – Refers to a file system block that contains single indirect pointers. Each indirect pointer refers to a file system block that contains the data block pointers. Double indirect pointers point to an additional 32 Gbytes of data.
- Triple indirect pointer – Can reference up to an additional 64 Tbytes of data.

Data Blocks

The remaining space allocated to the ufs file system holds data blocks. Data blocks are allocated, by default, in 8-Kbyte logical block sizes. The blocks are further divided into 1-Kbyte fragments. For a regular file, the data blocks contain the contents of the file. For a directory, the data blocks contain entries that associate the inode numbers and the file names of the files and directories contained in that directory.

Within a file system, those blocks that are currently not being used as files, directories, indirect address blocks, or storage blocks are marked as free in the cylinder group map. This map also keeps track of fragments to prevent disk performance from degrading.

Fragmentation

Fragmentation is the method used by the ufs file system to allocate disk space efficiently. Files less than 96 Kbytes in size are stored using fragmentation.

By default, data blocks can be divided into eight fragments of 1024 bytes each. Fragments store files and pieces of files smaller than 8192 bytes. For files larger than 96 Kbytes, fragments are never allocated and full blocks are exclusively used.

Figure 4-6 shows a fragment in a data block.

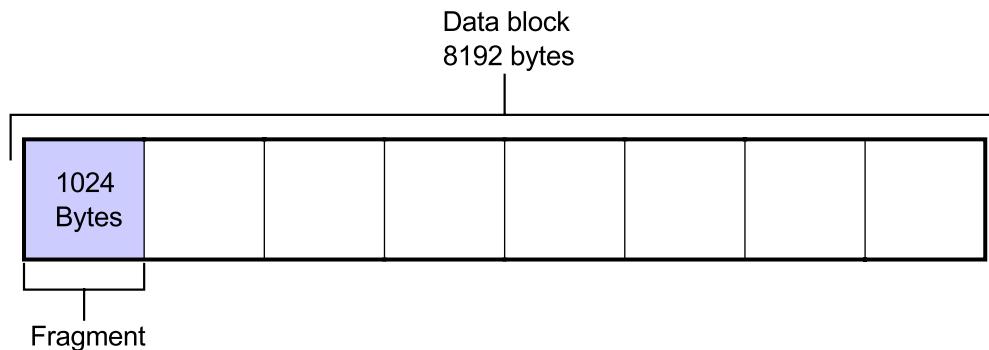


Figure 4-6 Divided Data Block

If a file contained in a fragment grows and requires more space, it is allocated one or more additional fragments in the same data block.

Figure 4-7 shows the contents of two different files stored in fragments in the same data block.

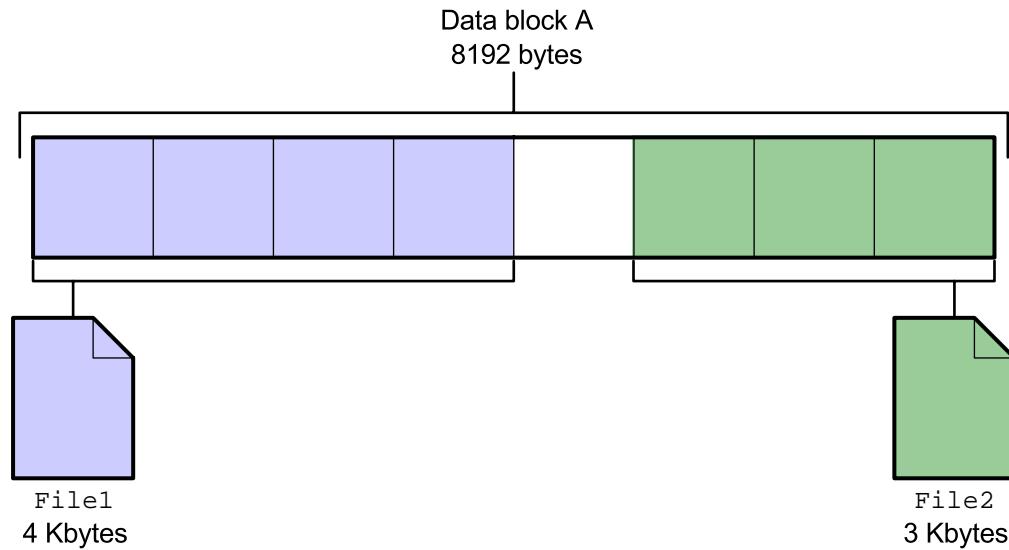


Figure 4-7 Two Files Stored in One Data Block

For example, if File1 requires more space than is currently available in the shared data block, the entire contents of that expanding file are moved by the ufs file system into a free data block. This requirement by the ufs file system assures that all of a file's fragments are contained in a whole data block. The ufs file system does not allow fragments of the same file to be stored in two different data blocks.

Using the newfs Command

To use the disk to store directories or files, a file system must be created on every disk partition. As the root user, you can construct a ufs file system on a disk slice by using the newfs command.

The newfs command is an easy-to-use front-end to the mkfs command, which you use to create file systems. The newfs command is located in the /usr/sbin directory.

Caution – Creating a new file system is destructive. The mkfs and newfs commands overwrite data that resides on the selected disk slice.



To create a ufs file system, by using the newfs command, perform the following steps:

1. As the root user, create a file system on a slice of a newly partitioned disk by entering the command:

```
# newfs /dev/rdsck/c1t3d0s7
```

2. The newfs command asks for confirmation before continuing. Verify that the correct disk slice on the correct disk is selected. To proceed, type y, to terminate the process, type n.

```
newfs: construct a new file system /dev/rdsck/c1t3d0s7: (y/n)? y
```

The newfs command displays information about the new file system being created.

```
/dev/rdsck/c1t3d0s7: 6295022 sectors in 1753 cylinders of 27 tracks, 133
sectors 3073.7MB in 110 cyl groups (16 c/g, 28.05MB/g, 3392 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 57632, 115232, 172832, 230432, 288032, 345632, 403232, 460832,
 518432, 5746208, 5803808, 5861408, 5919008, 5976608, 6034208, 6091808,
 6149408, 6207008, 6264608,
```

#

The first line printed by the newfs command describes the basic disk geometry. The second line describes the ufs file system created in this slice. The third and remaining lines list the beginning sector locations of the backup superblocks.

The newfs -i command is used to specify the density of the number of bytes per inode in the file system. To create more inodes, a smaller number should be given.



Note – This process also creates a `lost+found` directory for the `ufs` file system, which is a directory that is used by the file system check and repair (`fsck`) utility.

3. Repeat Steps 1 and 2 for every disk slice on any newly partitioned disk that needs to contain a file system.

The `newfs` command reserves between 1 and 10 percent of the file system space, depending on the size of the file system, for maintenance. This free space, referred to as `minfree`, specifies the amount of space on the slice that is reserved or held back from regular users. You can use the `newfs -m %free` command to preset the percentage of free space when you create a new file system.

```
# fstyp -v /dev/dsk/c0t1d0s6 |head
(output omitted for brevity)
minfree 10% maxbpg 2048 optim time
# newfs -m 2 /dev/dsk/c0t1d0s6
newfs: construct a new file system /dev/rdsk/c0t1d0s6: (y/n)? y
(output omitted for brevity)
# fstyp -v /dev/dsk/c0t1d0s6 |head
(output omitted for brevity)
minfree 2% maxbpg 2048 optim time
```

To show the value of `minfree` on a file system, use the `fstyp` command.

Creating a New ufs File System

The following command shows the minfree value for the file system on the c0t0d0s0 device.

```
# fstyp -v /dev/rdsck/c0t0d0s0 | head
ufs
magic 11954  format dynamic time      Fri Oct 22 10:09:11 2004
sblkno 16      cbblkno 24      iblkno 32      dblkno 456
sbsize 5120    cgsize 5120    cgoffset 72    cgmask 0xffffffe0
ncg   110      size   3147511 blocks 3099093
bsize 8192    shift  13      mask  0xfffffe000
fsize 1024    shift  10      mask  0xfffffc00
frag   8       shift  3       fsbtodb 1
minfree 2% maxbpg 2048 optim  time
maxcontig 128 rotdelay 0ms    rps   120
```

To change the minimum percentage value of free space on an existing file system, you can use the tunefs -m %free command.

The following command changes the minimum percentage of free space on the /dev/rdsck/c0t0d0s0 device to 1 percent.

```
# tunefs -m 1 /dev/rdsck/c0t0d0s0
minimum percentage of free space changes from 10% to 1%
```

Checking the File System by Using the `fsck` Command

A file system can become damaged if it is corrupted from a power failure, a software error in the kernel, a hardware failure, or an improper shutdown of the system. The file system check program, `fsck`, checks the data consistency of a file system and attempts to correct or repair any inconsistencies or damage found.



Caution – Never run the `fsck` command on a mounted file system. This could leave the file system in an unusable state. It could also delete data. The `/` (root), `/usr`, and `/var` file systems should have the `fsck` command run on them while in single-user mode.

Every time you boot a system, the operating system determines which file systems the `fsck` command should check. The `fsck` command checks and repairs any problems encountered in file systems before they are mounted.



Note – The status of a file system's state flag determines whether the file system needs to be scanned by the `fsck` command. When the state flag is "clean," "stable," or "logging," file system scans are not run.

Data Inconsistencies Checked by the `fsck` Command

The `fsck` command makes several passes through a file system. During each pass, the `fsck` command checks for several types of file system inconsistencies.

Superblock Consistency

The file system superblock is checked for inconsistencies involving such parameters as file system size, free block count, and free inode count.

Cylinder Group Block Consistency

The `fsck` command checks any unallocated data blocks claimed by inodes, the unallocated data block count, and the unallocated inode count.

Inode Consistency

The fsck command checks for the allocation state of inodes, as well as the type, the link count, duplicate blocks (blocks already claimed by another inode), bad blocks, the inode size, and the block count for each inode. Any unreferenced inode with a nonzero link count is linked to the file system's lost+found directory.

Data Block Consistency

The fsck command cannot check ordinary data blocks, but it can check directory data blocks. In directory data blocks, the fsck command checks for inodes that point to unallocated blocks, unallocated blocks tagged as in use, allocated blocks tagged as free (incorrect inodes for . and ..) and directories not connected to the file system. These directories are linked back to the file system in its lost+found directory.

The lost+found Directory

The fsck command puts files and directories that are allocated but unreferenced in the lost+found directory located in that file system. The inode number of each file is assigned as the file name. If the lost+found directory does not exist, the fsck command creates it. If not enough space exists in the lost+found directory, the fsck command increases the directory's size.

Noninteractive Mode

During a normal system boot, the fsck command operates in noninteractive mode, which is often referred to as preen, or silent mode. In this mode, the fsck command addresses only minor inconsistency problems that can be corrected. If a more serious inconsistency is found and a decision has to be made, the fsck program terminates and requests the root password to enter single-user mode. Execute the fsck command in interactive mode to continue.

Interactive Mode

In interactive mode, the `fsck` command lists each problem it encounters, followed by a suggested corrective action in the form of a question that requires a yes or no response.

The following example shows how the `fsck` command displays a message that asks if you want to correct the block count.

```
# fsck /dev/rdsk/c0t0d0s7
** /dev/rdsk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
INCORRECT BLOCK COUNT I=743 (5 should be 2)
CORRECT?
```

If you respond with yes, the `fsck` command applies the corrective action and moves on. If you respond with no, the `fsck` command repeats the message about the original problem and suggests corrective action. It does not fix the inconsistency until you respond yes.

The following examples demonstrate how you as the system's `root` user can run the `fsck` command to check the integrity of file systems.

- To check a single unmounted file system, perform the command:

```
# fsck /dev/rdsk/c0t0d0s7
```

This is the only way to check a file system that has not been entered in the `/etc/vfstab` file.

- To check a file system using the mount point directory name as listed in the `/etc/vfstab` file, perform the command:

```
# fsck /export/home
```

In the following example, the `fsck` command checks and repairs the file system with the force (`f`) and preen (`p`) options.

```
# fsck -o f,p /dev/rdsk/c0t0d0s7
/dev/rdsk/c0t0d0s7: 77 files, 9621 used, 46089 free
/dev/rdsk/c0t0d0s7: (4 frags, 57 blocks, 0.0% fragmentation)
```

The `f` option of the `fsck` command forces a file system check, regardless of the state of the file system's superblock state flag.

The `p` option checks and fixes the file system noninteractively (preen). The program exits immediately if a problem requiring intervention is found.

Resolving File System Inconsistencies

If problems are located in a file system, you are alerted by the `fsck` utility. Some of the more common file system errors that require interactive intervention are:

- Allocated unreferenced file
- Inconsistent link count
- Free block count corruption
- Superblock corruption

Reconnecting an Allocated Unreferenced File

If the `fsck` command discovers an inode that is allocated but unreferenced or not linked in any directory, the command sends a message that asks you if you want to reconnect the inode.

```
** Phase 3 - Check Connectivity
UNREF FILE I=788 OWNER=root MODE=100644
SIZE=19994 MTIME=Oct 18 10:49 2004
RECONNECT? y
```

A yes response causes the `fsck` command to save the file to the `lost+found` directory. The `fsck` command references the inode number.

To determine the type of file moved to the `lost+found` directory by the `fsck` command, perform the following steps:

1. List the contents of the file system's `lost+found` directory.

```
# ls /export/home/lost+found
#788
```

2. Determine the file type by using the `file` command.

```
# file /export/home/lost+found/#788
/export/home/lost+found/#788: ascii text
```

3. To view the contents of an ASCII text file, use the more or cat command. To view the contents of a binary file, use the strings command. If the file is associated with an application, such as a word processing document, use the application to view the contents of the file.

```
# cat /export/home/lost+found/#788
```

4. If the file is intact and you know where it belongs, you can copy the file back to its original location in the file system.

```
# cp /export/home/lost+found/#788 /export/home/user1/report
```

Adjusting a Link Counter

If the fsck program discovers that the value of a directory inode link counter and the actual number of directory links are inconsistent, the command displays a message that asks you if you want to adjust the counter.

```
** Phase 4 - Check Reference Counts
LINK COUNT DIR I=2 OWNER=root MODE=40755
SIZE=512 MTIME=Oct 18 15:59 2004 COUNT 4 SHOULD BE 3
ADJUST? y
```

In the example, a **y** (yes) response causes the fsck command to correct the directory inode link counter from 4 to 3.

During this phase, you might also be asked to clear or remove a link.

```
BAD/DUP type I=200 OWNER=root MODE=40755
SIZE=512 MTIME=Mar 14 08:03 2004
CLEAR? y
```

Salvaging the Free List

If the fsck utility discovers that the unallocated block count and the free block number listed in the superblock are inconsistent, the fsck command displays a message that asks if you want to salvage the free block count by rectifying it with the unallocated block count.

```
** Phase 5 - Check Cyl groups
CG 0: BAD MAGIC NUMBER
FREE BLK COUNT(S) WRONG IN SUPERBLK
SALVAGE? y
```

In the example, a **y** (yes) response causes the fsck command to update the information in the file system superblock.

Using Backup Superblocks

Superblock corruption can cause a file system to be unmountable. A file system is unusable when the message such as “Can’t mount *file_system_name*” or “*device_name* is not this fstype” appears.

```
Can't mount /dev/dsk/c0t0d0s7
```

This message can appear during a system boot or when you are manually mounting the file system.

If the `fsck` command fails because of a corrupted superblock, you see an error message that tells you to execute the `fsck` command using a superblock backup to recover the file system. Execute the `fsck` command with the `-o` option and with the `b` flag followed by a backup superblock number. Every file system has an alternative backup superblock at block number 32, which can be used with the `fsck` command to repair the primary superblock.

The following command uses a backup superblock.

```
# fsck -o b=32 /dev/rdsck/c0t0d0s7
Alternate super block location: 32.
** /dev/rdsck/c0t0d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 9 used, 5174880 free (16 frags, 646858 blocks, 0.0%
fragmentation)
#
```

The `fsck` utility compares the information in the backup superblock with the actual file system and attempts to rebuild the primary superblock. However, if the first backup superblock is part of the file system that was damaged, it might be unusable. Select another backup superblock to continue the `fsck` command.

To list the locations of all the alternative backup superblocks in the file system, run the `newfs -N` command.



Caution – This method works if the underlying file system was built using the newfs default parameters. If the file system was not built with these defaults, execute the newfs -N command, using the same parameters originally used, to generate identical superblock locations.

Use the -N option to view the file system parameters that you could use to create a new file system without actually creating the file system. A portion of the output is a list of the locations of all the alternative backup superblocks that can be used with the fsck -o b=# command.

```
# newfs -N /dev/rdsk/c1t3d0s7
/dev/rdsk/c1t3d0s7:      6295022 sectors in 1753 cylinders of 27 tracks,
133 sectors
      3073.7MB in 110 cyl groups (16 c/g, 28.05MB/g, 3392 i/g)
super-block backups (for fsck -F ufs -o b=##) at:
  32, 57632, 115232, 172832, 230432, 288032, 345632, 403232, 460832,
  518432, 5746208, 5803808, 5861408, 5919008, 5976608, 6034208, 6091808,
  6149408, 6207008, 6264608,
#
```

The -T option allows the file system to be a multi-Terabyte file system. You can view the file system parameters using this option without actually creating the file system.

```
# newfs -N -T /dev/rdsk/c1t3d0s7
Warning: cylinder groups must have a multiple of 16 cylinders with the
given parameters
Rounded cgsize up to 176
Warning: 15 sector(s) in last cylinder unallocated
/dev/rdsk/c1t3d0s7:      6295008 sectors in 1753 cylinders of 27 tracks,
133 sectors
      3073.7MB in 11 cyl groups (160 c/g, 280.55MB/g, 320 i/g)
super-block backups (for fsck -F ufs -o b=##) at:
  32, 574736, 1149440, 1724144, 2298848, 2873552, 3448256, 4022960,
  4597664, 5172368, 5747072,
```

Resolving File System Inconsistencies

You can use any other alternative superblock number in the list with the fsck command.

```
# fsck -o b=535952 /dev/rdsk/c0t0d0s7
Alternate super block location: 518432
** /dev/rdsk/c0t0d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
7 files, 14 used, 279825 free (17 frags, 347891 blocks, 0.0% fragmentation)
*****FILE SYSTEM WAS MODIFIED*****
#
#
```

Monitoring File System Use

An important activity of a system administrator is to monitor file system use on a regular basis. There are three useful commands available for this task:

- df – Displays the number of free disk blocks
- du – Summarizes disk use
- quot – Summarizes file system ownership

Using the df Command

Use the df command to display the amount of disk space used in file systems. This command lists the amount of used and available space and the amount of the file system's total capacity being used.

The format for the df command is:

```
df -option resource
```

Table 4-1 lists some of the more common options used with the df command.

Table 4-1 Partial Listing of Options for the df Command

Option	Description
-a	Reports on all file systems, including those with entries in the /etc/mnttab file for which the ignore option is set
-b	Prints the total number of Kbytes free
-e	Prints only the number of files free
-k	Displays disk allocation in Kbytes
-h	Acts like the -k option, except that sizes are in a more readable format, for example, 14K, 234M, 2.7G, or 3.0T
-l	Reports on local file systems only
-F FSType	Specifies the file system type on which to operate. This is intended for use on unmounted file systems.

Monitoring File System Use

To display the capacity of file systems, perform the command:

```
# df -k
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/dsk/c0t0d0s0	849518	113485	676567	15%	/
/devices	0	0	0	0%	/devices
ctfs	0	0	0	0%	/system/contract
proc	0	0	0	0%	/proc
mnttab	0	0	0	0%	/etc/mnttab
swap	596608	360	596248	1%	/etc/svc/volatile
objfs	0	0	0	0%	/system/object
/dev/dsk/c0t0d0s6	3007086	2317518	629427	79%	/usr
fd	0	0	0	0%	/dev/fd
/dev/dsk/c0t0d0s3	1785654	92508	1639577	6%	/var
swap	596352	104	596248	1%	/var/run
swap	596568	320	596248	1%	/tmp
/dev/dsk/c0t0d0s7	424239	1046	380770	1%	/export/home
/dev/dsk/c1t3d0s7	3099093	3097	3065006	1%	/data

The same file system displayed with the -h option would appear in human-readable format.

```
# df -h
```

Filesystem	size	used	avail	capacity	Mounted on
/dev/dsk/c0t0d0s0	830M	111M	661M	15%	/
/devices	0K	0K	0K	0%	/devices
ctfs	0K	0K	0K	0%	/system/contract
proc	0K	0K	0K	0%	/proc
mnttab	0K	0K	0K	0%	/etc/mnttab
swap	583M	360K	582M	1%	/etc/svc/volatile
objfs	0K	0K	0K	0%	/system/object
/dev/dsk/c0t0d0s6	2.9G	2.2G	615M	79%	/usr
fd	0K	0K	0K	0%	/dev/fd
/dev/dsk/c0t0d0s3	1.7G	90M	1.6G	6%	/var
swap	582M	104K	582M	1%	/var/run
swap	583M	320K	582M	1%	/tmp
/dev/dsk/c0t0d0s7	414M	1.0M	372M	1%	/export/home
/dev/dsk/c1t3d0s7	3.0G	3.0M	2.9G	1%	/data

Table 4-2 defines the fields displayed by the df -k command.

Table 4-2 Fields for the df -k Command

Field	Definition
Filesystem	The mounted file system
kbytes	The size of the file system in Kbytes (1024 bytes)
used	The number of Kbytes used
avail	The number of Kbytes available
capacity	The percentage of file system capacity used
Mounted on	The mount point

The amount of space that is reported as used and avail is typically less than the amount of total space in the file system. A fraction of space, from 1 to 10 percent, is reserved in each file system as the minfree value.

When all of the reported space on the file system is in use, the file system capacity is displayed as 100 percent. Regular users receive the message “File System Full” and cannot continue working. The reserved space is available to the root user, who can then delete or back up files in the file system.

The df -k command can be used with the device as the resource to show available space on the device:

```
# df -k /dev/dsk/c0t1d0s6
Filesystem          kbytes   used   avail capacity  Mounted on
/dev/dsk/c0t1d0s6     17153338      9  16810225      0%
#
```

Note – This command does not work on a partition without a file system, but does work on a partition with an unmounted file system.



Using the du Command

Use the du command to display the number of disk blocks used by directories and files. Each disk block consists of 512 bytes.

The format for the du command is:

```
du -options directory
```

Table 4-3 describes the options for the du command.

Table 4-3 Options for the du Command

Option	Description
-k	Displays disk use in Kbytes.
-s	Displays only the summary in 512-byte blocks. Using the s and k options together shows the summary in Kbytes.
-a	Displays the number of blocks used by all files in addition to directories within the specified directory hierarchy.

To display disk usage in kilobytes, perform the command:

```
# cd /opt
# du -k
3      ./SUNWits/Graphics-sw/xil/lib
4      ./SUNWits/Graphics-sw/xil
5      ./SUNWits/Graphics-sw
6      ./SUNWits
15     ./SUNWmlib/lib/sparcv8
15     ./SUNWmlib/lib/sparcv8plus
15     ./SUNWmlib/lib/sparcv8plus+vis
15     ./SUNWmlib/lib/sparcv8plus+vis2
15     ./SUNWmlib/lib/sparcv9
15     ./SUNWmlib/lib/sparcv9+vis
15     ./SUNWmlib/lib/sparcv9+vis2
120    ./SUNWmlib/lib
24     ./SUNWmlib/include
146    ./SUNWmlib
376    ./SUNWrvc/bin
10     ./SUNWrvc/examples/rtvc_capture_movie
24     ./SUNWrvc/examples/rtvc_display
68     ./SUNWrvc/examples/rtvc_video_conference
25     ./SUNWrvc/examples/test
```

```

128      ./SUNWrtvc/examples
7        ./SUNWrtvc/man/man1
19       ./SUNWrtvc/man/man3
28       ./SUNWrtvc/man
533      ./SUNWrtvc
686      .

```

To display disk usage in human readable form, perform the command:

```
# du -h /opt |more
3K   /opt/SUNWits/Graphics-sw/xil/lib
4K   /opt/SUNWits/Graphics-sw/xil
5K   /opt/SUNWits/Graphics-sw
6K   /opt/SUNWits
24K  /opt/SUNWmlib/include
15K  /opt/SUNWmlib/lib/sparcv8
15K  /opt/SUNWmlib/lib/sparcv8plus
15K  /opt/SUNWmlib/lib/sparcv8plus+vis
15K  /opt/SUNWmlib/lib/sparcv8plus+vis2
15K  /opt/SUNWmlib/lib/sparcv9
15K  /opt/SUNWmlib/lib/sparcv9+vis
15K  /opt/SUNWmlib/lib/sparcv9+vis2
120K /opt/SUNWmlib/lib
146K /opt/SUNWmlib
376K /opt/SUNWrtvc/bin
10K  /opt/SUNWrtvc/examples/rtvc_capture_movie
24K  /opt/SUNWrtvc/examples/rtvc_display
68K  /opt/SUNWrtvc/examples/rtvc_video_conference
25K  /opt/SUNWrtvc/examples/test
128K /opt/SUNWrtvc/examples
    7K  /opt/SUNWrtvc/man/man1
    19K /opt/SUNWrtvc/man/man3

```

To display disk usage including files, perform the command:

```
# du -ak /opt
1      /opt/SUNWits/Graphics-sw/xil/lib/libxil.so
1      /opt/SUNWits/Graphics-sw/xil/lib/libxil.so.1
3      /opt/SUNWits/Graphics-sw/xil/lib
4      /opt/SUNWits/Graphics-sw/xil
(output removed for brevity)
19     /opt/SUNWrtvc/man/man3
1      /opt/SUNWrtvc/man/windex
28     /opt/SUNWrtvc/man
533    /opt/SUNWrtvc
686    /opt

```

To display only a summary of disk usage, perform the command:

```
# du -sk /opt  
686      /opt
```

Using the quot Command

Use the quot command to display how much disk space, in kilobytes, is being used by users.

The format for the quot command is:

```
quot -options filesystem
```

Table 4-4 describes the options for the quot command.

Table 4-4 Options for the quot Command

Option	Description
-a	Reports on all mounted file systems
-f	Includes the number of files

To display disk space being used by users on all mounted file systems, perform the command:

```
# quot -af
/dev/rdsk/c0t0d0s0 (/):
112410 5246 root
 31     12 uucp
 11     11 lp
   1      1 adm
/dev/rdsk/c0t0d0s6 (/usr):
2313692 102415 root
 806     15 uucp
 11      4 bin
   1      1 adm
...
...
```

The columns represent kilobytes used, number of files, and owner, respectively.

To display a count of the number of files and space owned by each user for a specific file system, enter the following:

```
# quot -f /dev/dsk/c0t0d0s7
/dev/rdsk/c0t0d0s7 (/export/home):
 9      2 root
15     35 sue
51     51 paul
23     25 jeff
```

Using the Solaris Management Console Usage Tool

The Solaris Management Console Usage Tool provides a graphical display of the available space for all mounted file systems.

To use the Solaris Management Console storage Usage Tool, launch the Solaris Management Console by typing **smc&** at a command line, or select it from the Application Manager Window. To locate the Usage Tool, select This Computer, then select Storage, then select Mounts and Shares on the Solaris Management Console.

Monitoring File System Use

Figure 4-8 shows the Management Tools: Solaris Management Console window with the disk usage information.

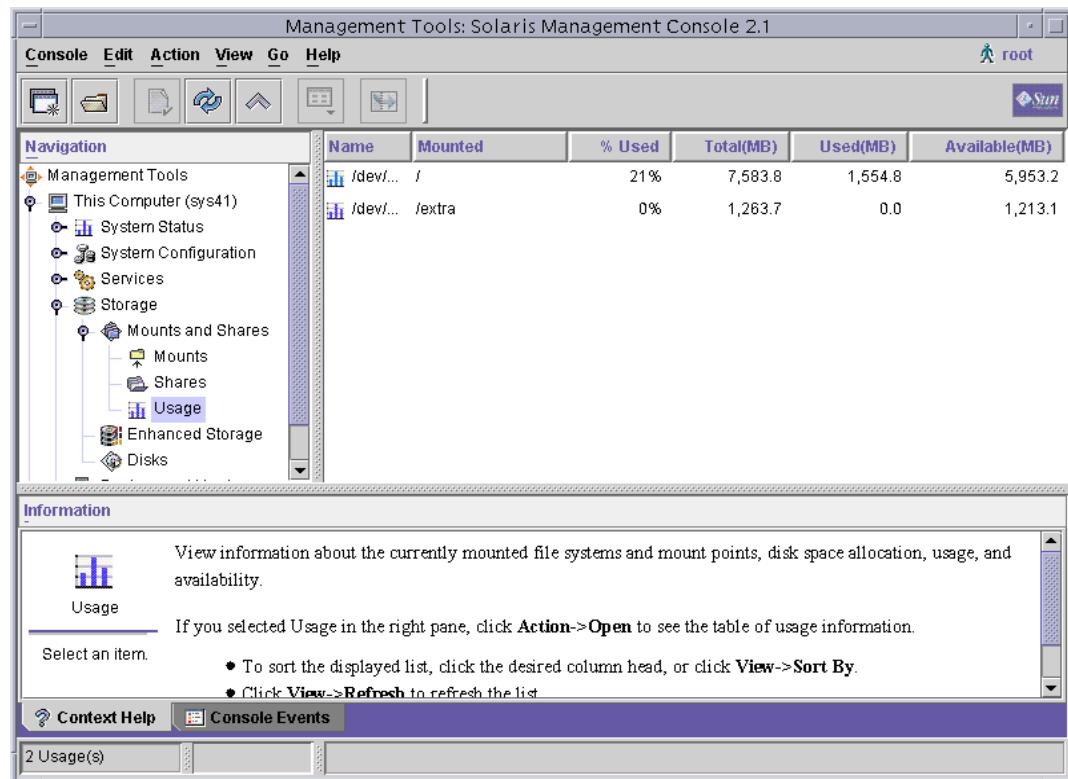


Figure 4-8 Management Tools: Solaris Management Console Window

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Creating and Maintaining ufs File Systems (Level 1)

In this exercise, you complete the following tasks:

- Create ufs file systems
- Calculate and adjust minfree values
- Destroy the superblock on an unused file system and repair it using an alternative

Preparation

This exercise requires an unused disk divided into four slices. Slices 0, 1, and 3 are equal in size, and Slice 4 takes up the remaining space on the disk. If it is necessary to partition the disk, this exercise requires an understanding of how to use the `format` utility. Refer to the lecture notes as necessary to perform the steps.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Perform the following tasks:

- Find a disk that is not in use and that is partitioned as specified in the preceding preparation description. If necessary, partition a disk accordingly. Create a new file system on Slice 0. Create a file system on Slice 1 with an inode ratio of 1 per 16,384 bytes of data space.

Compare how quickly the newfs command makes the file systems. For both file systems, record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group. How do the file systems differ?

(Steps 1–6 in the Level 2 lab)

- Display the number of Kbytes used, the number available, and the number allocated to both file systems. Record these values. Which file system has more available space and why? For each file system, calculate how much larger the Kbytes value is than the sum of the used and available values, and express the result as a percentage. Use the fstyp command to verify the result.

(Steps 7–8 in the Level 2 lab)

- Adjust the minfree value up or down by 3 percent. Record the message that your command displays. Verify the change made by using the tunefs command.

(Steps 9–10 in the Level 2 lab)

- Create new file systems on Slices 3 and 4 of your spare disk.

(Step 11 in the Level 2 lab)

- Check the file system on Slice 3 with the fsck command, and record if it reports any errors. Use the dd command from Step 13 in the Level 2 lab to destroy the primary superblock of the new file system. Run the fsck command, and see if you get an error. Use the fsck command and the backup superblock found at Sector 32 to repair the file system and main superblock. Verify the repair by running the fsck command again.

(Steps 12–16 in the Level 2 lab)

Exercise: Creating and Maintaining ufs File Systems (Level 2)

In this exercise, you complete the following tasks:

- Create ufs file systems
- Calculate and adjust minfree values
- Destroy the superblock on an unused file system and repair it using an alternative

Preparation

This exercise requires an unused disk, divided into four slices. Slices 0, 1, and 3 are equal in size, and Slice 4 takes up the remaining space on the disk. If it is necessary to partition this disk, this exercise requires an understanding of how to use the `format` utility. Refer to the lecture notes as necessary to perform the steps.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Find a disk that is not in use and that is partitioned as specified in the preceding preparation description. If necessary, partition a disk accordingly. Create a new file system on Slice 0. Create a file system on Slice 1 with an inode ratio of 1 per 16,384 bytes of data space. Compare how quickly the `newfs` command makes the file systems. For both file systems, record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group. How do the file systems differ?
- Display the number of Kbytes used, the number available, and the number allocated to both file systems. Record these values. Which file system has more available space and why? For each file system, calculate how much larger the Kbytes value is than the sum of the used and available values, and express the result as a percentage. Use the `fstyp` command to verify the result.
- Adjust the `minfree` value up or down by 3 percent. Record the message that your command displays. Verify the change made by using the `tunefs` command.
- Create new file systems on Slices 3 and 4 of your spare disk.
- Check the file system on Slice 3 with the `fsck` command, and record if it reports any errors. Use the `dd` command from Step 13 in the Level 2 lab to destroy the primary superblock of the new file system. Run the `fsck` command, and see if you get an error. Use the `fsck` command and the backup superblock found at Sector 32 to repair the file system and main superblock. Verify the repair by running the `fsck` command again.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change the directory to /dev/rdsk.
2. To find a spare disk, use the ls command to display a list of possible disks and the prtvtoc command to display the VTOC for each disk you find. Examine the partition list as well as the Mount Directory field that the prtvtoc command displays. Disks that are not in use have no mount directory listed. Record the name of the unused disk.

Unused disk:

 **Note** – This procedure works for the classroom environment. A disk that does not show mounted slices in the Mount Directory field of the prtvtoc output is not necessarily unused.

3. If a spare disk exists but it is not divided into four slices, use the format utility to partition the disk. Make three slices exactly the same size (approximately 25 percent of the total disk space each), and use the fourth partition for the remainder of the available space. Exit from the format utility when you are finished. You can also use the Solaris Management Console to partition the drive.
4. Use the newfs command without options to create a new file system on Slice 0 on the spare disk. Observe how quickly the newfs command creates cylinder groups on this slice. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

Cylinder groups:

Cylinders per group:

Inodes per group:

5. Use the newfs command to create a new file system on Slice 1 on the spare disk. Use the -i option to create one inode per 16,384 bytes of data space. Observe how quickly the newfs command creates cylinder groups on this slice. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

Cylinder groups:

Cylinders per group:

Inodes per group:

6. According to the statistics you have gathered, how do the file systems on Slices 0 and 1 differ?
7. Use the `df` command to display statistics for the file systems on Slices 0 and 1 that you used in the previous steps. Record the values listed in the `kbytes`, `used`, and `avail` columns.

Which file system has the larger amount of available data space and why?

8. For each file system, add the `used` and `avail` values, and compare the sum to the `kbytes` value. Expressed as a percentage, how much larger is the `kbytes` value than the sum of `used` and `avail`? This percentage should approximately match the `minfree` value.

Use the `fstyp` command to verify your result.

9. Use the `tunefs` command to change the `minfree` value for the file system on Slice 0 of the spare disk. If the current `minfree` value is greater than 5 percent, reduce it by 3 percent. If it is less than or equal to 5 percent, add 3 percent.

What message does the `tunefs` command display?

10. Use the `df -k` command to verify that the `minfree` value has changed. Record the values listed in the `kbytes`, `used`, and `avail` columns.

Which value has changed from the information you gathered in Step 7?

11. Create new file systems on Slices 3 and 4 of your spare disk.
 12. Run the `fsck` command interactively to check the new file system previously created on Slice 3 of the spare disk.
- Did the `fsck` command report errors?
13. Use the `dd` command to destroy the main superblock of the file system on Slice 3.

```
# dd if=/dev/zero of=/dev/rdsck/c1t0d0s3 count=32 bs=512
```

14. Run the `fsck` command interactively to check the new file system. Did the `fsck` command report errors? If so, what corrective action does the `fsck` command suggest?
15. Run the `fsck` command, and specify an alternative superblock. Block 32 is always one of the alternatives available.
16. Run the `fsck` command again to verify that the file system was repaired.

Exercise: Creating and Maintaining ufs File Systems (Level 3)

In this exercise, you complete the following tasks:

- Create ufs file systems
- Calculate and adjust minfree values
- Destroy the superblock on an unused file system and repair it using an alternative

Preparation

This exercise requires an unused disk, divided into four slices. Slices 0, 1, and 3 are equal in size, and Slice 4 takes up the remaining space on the disk. If it is necessary to partition this disk, this exercise requires an understanding of how to use the `format` utility. Refer to the lecture notes as necessary to perform the steps.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Find a disk that is not in use and that is partitioned as specified in the preceding preparation description. If necessary, partition a disk accordingly. Create a new file system on Slice 0. Create a file system on Slice 1 with an inode ratio of 1 per 16,384 bytes of data space. Compare how quickly the newfs command makes the file systems. For both file systems, record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group. How do the file systems differ?
- Display the number of Kbytes used, the number available, and the number allocated to both file systems. Record these values. Which file system has more available space and why? For each file system, calculate how much larger the Kbytes value is than the sum of the used and available values, and express the result as a percentage. Use the fstyp command to verify the result.
- Adjust the minfree value up or down by 3 percent. Record the message that your command displays. Verify the change made by using the tunefs command.
- Create new file systems on Slices 3 and 4 of your spare disk.
- Check the file system on Slice 3 with the fsck command, and record if it reports any errors. Use the dd command from Step 13 in the Level 2 lab to destroy the primary superblock of the new file system. Run the fsck command, and see if you get an error. Use the fsck command and the backup superblock found at Sector 32 to repair the file system and main superblock. Verify the repair by running the fsck command again.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change the directory to /dev/rdsk.

```
# cd /dev/rdsk
```

2. To find a spare disk, use the ls command to display a list of possible disks and the prtvtoc command to display the VTOC for each disk you find. Examine the partition list as well as the Mount Directory field that the prtvtoc command displays. Disks that are not in use have no mount directory listed. Record the name of the unused disk.

```
# ls *s2  
# prtvtoc /dev/rdsk/c1t0d0s2
```

Unused disk:

 **Note** – This procedure works for the classroom environment. A disk that does not show mounted slices in the Mount Directory field of the prtvtoc output is not necessarily unused.

3. If a spare disk exists, but it is not divided into four slices, use the format utility to partition the disk. Make three slices exactly the same size (approximately 25 percent of the total disk space each), and use the fourth partition for the remainder of the available space. You can also use the Solaris Management Console to partition the drive. Exit from the format utility when you are finished.

Example of the partition table:

Part	Tag	Flag	Cylinders	Size	Blocks
0	alternates	wm	0 - 1168	2.00GB	(1169/0/0) 4197879
1	alternates	wm	1169 - 2337	2.00GB	(1169/0/0) 4197879
2	backup	wm	0 - 4923	8.43GB	(4924/0/0) 17682084
3	alternates	wm	2338 - 3506	2.00GB	(1169/0/0) 4197879
4	alternates	wm	3507 - 4922	2.42GB	(1416/0/0) 5084856

4. Use the newfs command without options to create a new file system on Slice 0 on the spare disk. Observe how quickly the newfs command creates cylinder groups on this slice. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group, for example, your spare disk may be a different device.

```
# newfs /dev/rdsk/c1t0d0s0
```

Cylinder groups:

Cylinders per group:

Inodes per group:

5. Use the newfs command to create a new file system on Slice 1 on the spare disk. Use the -i option to create one inode per 16,384 bytes of data space. Observe how quickly the newfs command creates cylinder groups on this slice. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

```
# newfs -i 16384 /dev/rdsk/c1t0d0s1
```

Cylinder groups:

Cylinders per group:

Inodes per group:

6. According to the statistics you have gathered, how do the file systems on Slices 0 and 1 differ?

The number of inodes per group is less on File System 1 than on File System 0.

7. Use the df command to display statistics for the file systems on Slices 0 and 1 that you used in the previous steps, for example:

```
# df -k /dev/dsk/c1t0d0s0
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/dsk/c1t0d0s0	8705501	9	8618436	0%	

```
# df -k /dev/dsk/c1t0d0s1
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/dsk/c1t0d0s1	8769565	9	8681796	0%	

Record the values listed in the kbytes, used, and avail columns.

Which file system has the larger amount of available data space and why?

File System 1 has the larger amount of available data space because it holds fewer inode records.

Exercise: Creating and Maintaining ufs File Systems (Level 3)

8. For each file system, add the used and avail values, and compare the sum to the kbytes value. Expressed as a percentage, how much larger is the kbytes value than the sum of used and avail? This percentage should approximately match the minfree value.

Use the `fstyp -v /dev/rdsck/c#t#d#s# | head` command to verify your result.

To calculate the percentage difference between the sum of used and avail and the kbytes value, perform the following:

- a. *Add the values listed as used and avail, for example:*

$$9 + 1926799 = 1926808$$

- b. *Divide the sum of used and avail by the kbytes value, for example:*

$$1926808 / 1986439 = 0.969981$$

- c. *Multiply the result of Step b by 100, for example:*

$$0.969981 * 100 = 96.9981$$

- d. *Subtract the result of Step c from 100, for example:*

$$100 - 96.9981 = 3.0019$$

- e. *Round the result of Step d to the nearest whole number, for example:*

$$3.0019 = 3 \text{ percent}$$

9. Use the `tunefs -m # /dev/rdsck/c#t#d#s#` command to change the minfree value for the file system on Slice 0 of the spare disk. If the current minfree value is greater than 5 percent, reduce it by 3 percent. If it is less than or equal to 5 percent, add 3 percent, for example:

```
# tunefs -m 4 /dev/rdsck/c1t0d0s0
minimum percentage of free space changes from 1% to 4%
```

What message does the tunefs command display?

The minimum percentage of free space changes from x percent to x percent.

10. Use the `df -k` command to verify that the `minfree` value has changed. Record the values listed in the `kbytes`, `used`, and `avail` columns, for example:

```
# df -k /dev/dsk/c1t0d0s0
Filesystem          kbytes   used   avail capacity  Mounted on
/dev/dsk/c1t0d0s0    8705501      9  8357271      0%
```

Which value has changed from the information you gathered in Step 7?

The avail column changes but not the kbytes or used columns.

11. Create new file systems on Slices 3 and 4 of your spare disk, for example:

```
# newfs /dev/rdsk/c1t0d0s3
# newfs /dev/rdsk/c1t0d0s4
```

12. Run the `fsck` command interactively to check the new file system previously created on Slice 3 of the spare disk.

```
# fsck /dev/rdsk/c1t0d0s3
```

Did the `fsck` command report errors?

No.

13. Use the `dd` command to destroy the main superblock of the file system on Slice 3.

```
# dd if=/dev/zero of=/dev/rdsk/c1t0d0s3 count=32 bs=512
```

14. Run the `fsck` command interactively to check the new file system.

```
# fsck /dev/rdsk/c1t0d0s3
```

Did the `fsck` command report errors? If so, what corrective action does the `fsck` command suggest?

The fsck command indicates that the magic number in the superblock is wrong and suggests repairing it by using an alternative superblock, for example:

```
** /dev/rdsk/c1t0d0s3
BAD SUPER BLOCK: MAGIC NUMBER WRONG
USE AN ALTERNATE SUPER-BLOCK TO SUPPLY NEEDED INFORMATION;
e.g. fsck [-F ufs] -o b=# [special ...]
where # is the alternate super block. SEE fsck_ufs(1M).
```

Exercise: Creating and Maintaining ufs File Systems (Level 3)

15. Run the fsck command, and specify an alternative superblock. Block 32 is always one of the alternatives available.

```
# fsck -o b=32 /dev/rdsk/c1t0d0s3
```

16. Run the fsck command again to verify that the file system was repaired.

```
# fsck /dev/rdsk/c1t0d0s3
```

This time the fsck command output does not report that the file system was modified.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 5

Performing Mounts and Unmounts

Objectives

Upon completion of this module, you should be able to:

- Identify mounting basics
- Perform mounts
- Perform unmounts
- Access a mounted diskette, CD-ROM, or DVD
- Restrict access to a mounted diskette, CD-ROM, or DVD
- Access a diskette, CD-ROM, or DVD without Volume Management (vold)

The course map in Figure 5-1 shows how this module fits into the current instructional goal.

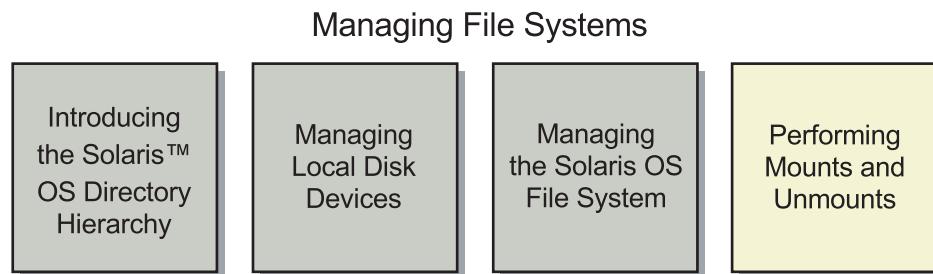


Figure 5-1 Course Map

Working With Mounting Basics

In the Solaris OS, you use the mounting process to attach individual file systems to their mount points on the directory hierarchy. This action makes a file system accessible to the system and to the users.

You use the unmounting process to detach a file system from its mount point in the directory hierarchy. This action makes a file system unavailable to the system or users.

After you have created a file system by using the newfs command, you must attach it to the Solaris OS directory hierarchy at a mount point. A mount point is a directory that is the point of connection for a file system. File systems are commonly referred to by the names of their mount points, for example, the / (root) file system or the /usr file system.

Figure 5-2 shows how the directory hierarchy spans from one file system to the next.

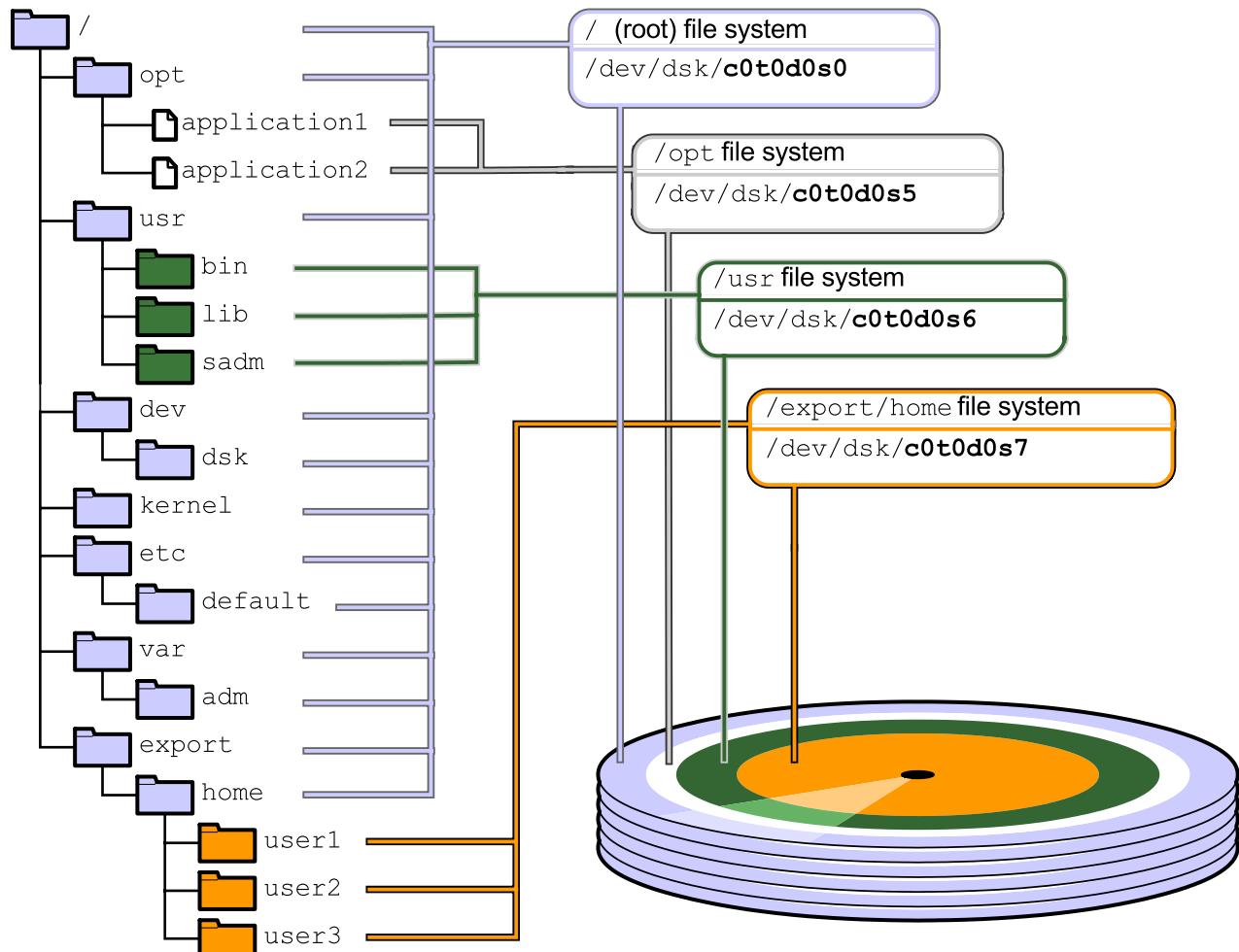


Figure 5-2 File Systems and Mount Points

File systems do not contain their own mount point directories.

Determining Which File Systems Are Currently Mounted

You can determine which file systems are currently mounted by using the mount command or the df command.

The df command displays the amount of disk space occupied by mounted or unmounted file systems and, depending on the options used, displays both locally mounted and virtual file system information.

The mount command, which is located in the /usr/sbin directory, maintains a table of currently mounted file systems in the /etc/mnttab file. When the mount command is used without arguments, it lists all of the mounted file systems in the /etc/mnttab directory. When used with only a partial argument list, the command searches the /etc/vfstab file for an entry that supplies the missing arguments.

Note – Options to the mount command are discussed later in this module.



Note – While system administrators typically use the /usr/sbin/mount command, the system boot scripts use the /sbin/mount command.



Mounting a File System Automatically

The Solaris OS provides several methods for automating file system mounts.

The Solaris OS creates a default /etc/vfstab file during software installation, based on your selections. However, you can edit the /etc/vfstab file whenever file system entries need to be added or modified.

Note – The automounter can mount network file systems on demand.



Introducing the Virtual File System Table: /etc/vfstab

The /etc/vfstab file lists all the file systems to be automatically mounted at system boot time, with the exception of the /etc/mnttab and /var/run file systems.

The file format includes seven fields per line entry. By default, a tab separates each field, but any whitespace can be used for separators. The dash (-) character is used as a placeholder for fields when text arguments are not appropriate. Commented lines begin with the (#) symbol.



Note – Because the default is to use tabs to separate the fields in the /etc/vfstab file, the fields often do not line up under their respective headings. This can lead to some confusion when you are viewing this file in a terminal window.

An example of a /etc/vfstab file follows:

```
# more /etc/vfstab
#device      device        mount          FS      fsck    mount
mount
#to mount    to fsck       point         type    pass   at boot
options
#
fd      -      /dev/fd fd      -      no      -
/proc    -      /proc     proc      -      no      -
/dev/dsk/c0t0d0s1      -      -      swap    -      no      -
/dev/dsk/c0t0d0s0      /dev/rdsk/c0t0d0s0  /      ufs     1      no      -
/dev/dsk/c0t0d0s6      /dev/rdsk/c0t0d0s6  /usr    ufs     1      no      -
/dev/dsk/c0t0d0s3      /dev/rdsk/c0t0d0s3  /var    ufs     1      no      -
/dev/dsk/c0t0d0s7      /dev/rdsk/c0t0d0s7  /export/home ufs    2 yes   -
/devices    -      /devices    devfs   -      no      -
ctfs      -      /system/contract  ctfss   -      no      -
objfs    -      /system/object   objfs   -      no      -
swap      -      /tmp      tmpfs   -      yes   -
#
```

To add a line entry, you need the following information:

device to mount	The device to be mounted. For example, a local ufs file system /dev/dsk/c#t#d#s#, or a pseudo file system /proc.
device to fsck	The raw or <i>character</i> device checked by the file system check program (fsck) if applicable. Pseudo and distributed file systems have a dash (-) in this field.
mount point	The name of the directory that serves as the attach mount point in the Solaris OS directory hierarchy.
FS type	The type of file system to be mounted.
fsck pass	The pass number used by the fsck command to decide whether to check a file system. When the field contains a (-), the file system is not checked. When the field contains a zero, UFS file systems are not checked, however, non-UFS file systems are checked. When the field contains a value greater than zero, the file system is always checked. All file systems with a value of 1 in this field are checked one at a time in the order they appear in the vfstab file. When the fsck command is run on multiple UFS file systems that have fsck pass values greater than 1 and the preen option (-o p) is used, the fsck command automatically checks the file systems on different disks in parallel to maximize efficiency. Otherwise, the value of the pass number does not have any effect.
mount at boot	Enter yes to enable the mountall command to mount the file systems at boot time. Enter no to prevent a file system mount at boot time.



Note – For / (root), /usr, and /var (if it is a separate file system) file systems, the mount at boot field value is specified as no. The kernel mounts these file systems as part of the boot sequence before the mountall command is run. SMF mounts the file systems as specified under the /lib/svc/method directory beginning with fs-.

mount options	A comma-separated list of options passed to the mount command. A dash (-) indicates the use of default mount options.
---------------	-----------------------------------------------------------------------------------------------------------------------

Introducing the /etc/mnttab File

The /etc/mnttab file is an `mntfs` file system that provides read-only information directly from the kernel about mounted file systems on the local host.

Each time a file system is mounted, the `mount` command adds an entry to this file. Whenever a file system is unmounted, its entry is removed from the /etc/mnttab file.

Device Name	The name of the device that is mounted at the mount point. This block device is where the file system is physically located.
Mount Point	The mount point or directory name where the file system is to be attached within the / (root) file system (for example, /usr, /opt).
Mount Options	The list of mount options in effect for the file system.
<code>dev=number</code>	The major and minor device number of the mounted file system.
Date and Time Mounted	The date and time that the file system was mounted to the directory hierarchy.

The /var/run file system is a `tmpfs` mounted file system in the Solaris OS. It is the repository for temporary operating system files that are not needed across system reboots in this Solaris OS release. It is mounted as a pseudo file system rather than a disk-based file system.

The /var/run directory requires no administration. For security reasons, it is owned by the root user.

The /tmp directory continues to be a `tmpfs` mounted file system in the Solaris OS. It is the repository for temporary user and application files that are not needed across system reboots. It is a pseudo file system rather than a disk-based file system.

The following examples show two ways to display currently mounted file systems.

```
# more /etc/mnttab
/dev/dsk/c0t0d0s0      /      ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=2200008 1098604644
/devices      /devices      devfs      dev=4a80000      1098604620
ctfs      /system/contract      ctf      dev=4ac0001      1098604620
proc      /proc      proc      dev=4b00000      1098604620
mnttab      /etc/mnttab      mntfs      dev=4b40001      1098604620
swap      /etc/svc/volatile      tmpfs      xattr,dev=4b80001      1098604620
objefs      /system/object      objfs      dev=4bc0001      1098604620
/dev/dsk/c0t0d0s6      /usr      ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=220000e 1098604645
fd      /dev/fd fd      rw,dev=4d40001 1098604645
/dev/dsk/c0t0d0s3      /var      ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=220000b 1098604647
swap      /var/run      tmpfs      xattr,dev=4b80002      1098604647
swap      /tmp      tmpfs      xattr,dev=4b80003      1098604647
/dev/dsk/c0t0d0s7      /export/home      ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=220000f 1098604661
-hosts      /net      autoofs      nosuid,indirect,ignore,nobrowse,dev=4dc0001
1098604678
auto_home      /home      autoofs      indirect,ignore,nobrowse,dev=4dc0002
1098604678
sys-01:vold(pid491)      /vol      nfs      ignore,noquota,dev=4e00001
1098604701

# mount
/ on /dev/dsk/c0t0d0s0
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000
8 on Sun Oct 24 08:57:24 2004
/devices on /devices read/write/setuid/devices/dev=4a80000 on Sun Oct 24
08:57:00 2004
/system/contract on ctf      read/write/setuid/devices/dev=4ac0001 on Sun Oct 24
08:57:00 2004
/proc on proc read/write/setuid/devices/dev=4b00000 on Sun Oct 24 08:57:00 2004
/etc/mnttab on mnttab read/write/setuid/devices/dev=4b40001 on Sun Oct 24
08:57:00 2004
/etc/svc/volatile on swap read/write/setuid/devices/xattr/dev=4b80001 on Sun Oct
24 08:57:00 2004
/system/object on objfs read/write/setuid/devices/dev=4bc0001 on Sun Oct 24
08:57:00 2004
/usr on /dev/dsk/c0t0d0s6
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000
e on Sun Oct 24 08:57:25 2004
/dev/fd on fd read/write/setuid/devices/dev=4d40001 on Sun Oct 24 08:57:25 2004
/var on /dev/dsk/c0t0d0s3
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000
b on Sun Oct 24 08:57:27 2004
```

Working With Mounting Basics

```
/var/run on swap read/write/setuid/devices/xattr/dev=4b80002 on Sun Oct 24  
08:57:27 2004  
/tmp on swap read/write/setuid/devices/xattr/dev=4b80003 on Sun Oct 24 08:57:27  
2004  
/export/home on /dev/dsk/c0t0d0s7  
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000  
f on Sun Oct 24 08:57:41 2004
```

Performing Mounts

You can mount file systems manually by running the `mount` command, or the system can automatically mount file systems at boot time after consulting the `/etc/vfstab` file.

Mounting a Local File System Manually

The `mount` command not only lists which file systems are currently mounted, it also provides you with a method for mounting file systems.

Default Behavior of the `mount` Command

To mount a local file system manually, you need to know the name of the device where the file system resides and its mount point path name. Perform the command:

```
# mount /dev/dsk/c0t0d0s7 /export/home
```

In this example, the default action mounts the file system with the following options: `read/write`, `setuid`, `intr`, `logging`, `largefiles`, `xattr`, and `onerror`.

The following list explains the default options for the `mount` command.

<code>read/write</code>	Indicates whether reads and writes are allowed on the file system.
<code>setuid</code>	Permits the execution of <code>setuid</code> programs in the file system.
<code>intr/nointr</code>	Allows and forbids keyboard interrupts to kill a process that is waiting for an operation on a locked file system.
<code>logging</code>	Indicates that logging is enabled for the <code>ufs</code> file system. This is the default for the Solaris 10 OS.
<code>largefiles</code>	Allows for the creation of files larger than 2 Gbytes. A file system mounted with this option can contain files larger than 2 Gbytes.
<code>xattr</code>	Supports extended attributes not found in standard UNIX filesystems.



Note – Due to file system overhead, the largest file size that can be created is approximately 1 Tbyte. The data capacity of a 1 Tbyte file system is approximately 1 Tbyte minus 0.5% overhead and the recommended 1% free space.

`onerror=action` Specifies the action that the ufs file system should take to recover from an internal inconsistency on a file system. An action can be specified as:

`panic` — Causes a forced system shutdown. This is the default.

`lock` — Applies a file system lock to the file system.

`umount` — Forcibly unmounts the file system.

The /etc/vfstab file provides you with another important feature. Because the /etc/vfstab file contains the mapping between the mount point and the actual device name, the root user can manually mount a file system specifying only the mount point on the command line.

```
# mount /export/home
```

Using the `mount` Command Options

When you are using mount options on the command line, remember that the options are preceded by the `-o` flag. When you are using multiple options, enter them as a comma-separated list following the `-o` flag.

```
mount -o option,option,... device_name mount_point
```



Note – Mount options are described in detail in the man page for the `mount_ufs` command.

Some options used to mount local file systems include: `ro`, `nosetuid`, `noatime`, `nolargefiles`, and `nologging`.

- `ro` – Mounts the file system as read-only.

The following is an example using this option on the command line:

```
# mount -o ro /dev/dsk/c0t0d0s7 /export/home
```

- `nosuid` – Prohibits the execution of `setuid` programs in the file system. This does not restrict the creation of `setuid` programs.

The following example shows the use of multiple options on the command line:

```
# mount -o ro,nosuid /dev/dsk/c0t0d0s7 /export/home
```

- `noatime` – Suppresses the time-last-accessed modification on inodes, which reduces disk activity on a file system where access times are not important. Specifying this option generally improves file access times and boosts overall performance, for example:

```
# mount -o noatime /dev/dsk/c0t0d0s7 /export/home
```

- `nolargefiles` – Prevents a file system that contains one or more “large files” from being mounted, for example:

```
# mount -o nolargefiles /dev/dsk/c0t0d0s7 /export/home
```

Use of the `nolargefiles` option fails if the file system to be mounted contains a large file or did contain a large file at one time.

If the file system currently contains a large file and the root user needs to mount it with this option, then the large file must be located and moved or removed from the file system. Then you must execute the `fsck` command manually to update the superblock information.

The mount also fails if the file system at one time contained a large file, even though it was moved or removed. You must execute the `fsck` command to clear the old information and allow the file system to be mounted.

Mounting All File Systems Manually

The `/etc/vfstab` file is read by the `/usr/sbin/mountall` command during the system boot sequence and mounts all file systems that have a yes in the mount at boot field.

The root user can use the `mountall` command to mount manually every file system in the `/etc/vfstab` file that has a yes in the mount at boot field, for example:

```
# mountall
```

To mount only the local file systems listed in the `/etc/vfstab` file, execute:

```
# mountall -l
```

During the boot sequence, the `fsck` utility checks each local file system in the `/etc/vfstab` file that has a device to `fsck` entry and an `fsck` pass number greater than 0. The utility determines if the file system is in a usable state to be safely mounted.

If the `fsck` utility determines that the file system is in an unusable state (for example, corrupted), the `fsck` utility repairs it before the mount is attempted. The system attempts to mount any local file systems that have a - (dash) or 0 (zero) entry in the `fsck` pass field without checking the file system itself.

Mounting a New File System

To add a new disk to the system, prepare the disk to hold a file system, and mount the file system, perform these general steps:

1. Set up the disk hardware, which might include setting address switches and connecting cables.
2. Perform a reconfiguration boot or run the `devfsadm` utility to add support for the new disk.
3. Use the `format` utility to partition the disk into one or more slices.
4. Create a new file system on one slice by using the `newfs` command.
5. Create a mount point for the file system by using the `mkdir` command to create a new directory in the / (root) file system.

```
# mkdir /data
# mount /dev/dsk/c1t3d0s7 /data
# mount
```

6. Mount the new file system manually by using the `mount` command.
7. Use the `mount` command to determine if the file system is mounted.

(Some output is omitted.)

```
/data on /dev/dsk/c1t3d0s7
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev
=800027 on Sun Oct 24 11:55:34 2004
```

8. Edit the `/etc/vfstab` file to add a line entry for the new file system.

```
# vi /etc/vfstab
fd      -      /dev/fd  fd      -      no      -
/proc   -      /proc    proc   -      no      -
/dev/dsk/c0t0d0s1   -      -      swap   -      no      -
/dev/dsk/c0t0d0s0   /dev/rdsk/c0t0d0s0   /      ufs    1      no      -
/dev/dsk/c0t0d0s6   /dev/rdsk/c0t0d0s6   /usr   ufs    1      no      -
/dev/dsk/c0t0d0s3   /dev/rdsk/c0t0d0s3   /var   ufs    1      no      -
/dev/dsk/c0t0d0s7   /dev/rdsk/c0t0d0s7   /export/home ufs   2      yes     -
/devices          -      /devices      devfs   -      no      -
ctfs       -      /system/contract  ctfss   -      no      -
objfs      -      /system/object   objfss  -      no      -
swap       -      /tmp      tmpfs   -      yes     -
/dev/dsk/c1t3d0s7   /dev/rdsk/c1t3d0s7   /data   ufs    2      yes     -
#
```

The file system automatically mounts whenever the system boots.

Mounting Different Types of File Systems

Different file system types have unique properties that affect how the mount command functions.

By default, the mount command assumes it is mounting a ufs-type file system. However, when you are mounting a different type of file system, you might have to specify its type on the command line.

You use the `-F` option with the mount command to specify the type of file system mounted. The file system type must be determinable from the `/etc/vfstab`, `/etc/default/fs`, or `/etc/dfs/fstypes` files.

Determining a File System's Type

Because the mount commands need the file system type to function properly, the file system type must be explicitly specified or determined by searching the following files:

- The `/etc/vfstab` file for the `FS type` field
- The `/etc/default/fs` file for a local file system type
- The `/etc/dfs/fstypes` file for a remote file system type

If the file system's type has not been explicitly specified on the command line using the mount `-F FStype` option, the mount command examines the `/etc/vfstab` file to determine the file system's type. The mount command makes this determination by using the file system's block device name, raw device name, or mount point directory name.

If the mount command cannot determine the file system's type by searching the `/etc/vfstab` file, the mount command uses the default file system type specified in either the `/etc/default/fs` file or the `/etc/dfs/fstypes` file, depending on whether the file system is local or remote.

The default local file system type is specified in the `/etc/default/fs` file by the line entry `LOCAL=fstype`.

`LOCAL=ufs`

The first line entry in the /etc/dfs/fstypes file determines the default remote file system type.

```
nfs NFS Utilities  
autofs AUTOFS Utilities  
cachefs CACHEFS Utilities
```

Using the `fstyp` Command

You can also use the `fstyp` command with the raw device name of the disk slice to determine a file system's type.

```
# fstyp /dev/rdsk/c0t0d0s7  
ufs
```

Specifying an `hsfs` File System Type

To mount a file system that resides on a CD-ROM when the Volume Management (vold) services (vold) are stopped, as the `root` user, perform the command:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
```

In this example, the file system type is `hsfs`, the file system resides on disk slice `/dev/dsk/c0t6d0s0`, and the mount point `/cdrom` is a preexisting directory in the Solaris OS.

Specifying a `pcfs` File System Type

To mount a file system that resides on a diskette when the Volume Management (vold) services are stopped, perform the commands:

```
# mkdir /pcfs  
# mount -F pcfs /dev/diskette /pcfs
```

In this example, the file system type is `pcfs`. This file system resides on the device `/dev/diskette`, and the mount point is `/pcfs`.

Performing Unmounts

A file system is commonly unmounted if it needs to be checked and repaired by the `fsck` command, or if it needs to be backed up completely.

Unmounting a File System

Some file system administration tasks cannot be performed on mounted file systems.

To unmount a file system to prepare it for system maintenance, use the `umount` command.

Unmounting a file system by using the `umount` command removes it from the file system mount point and deletes its entry from the `/etc/mnttab` file.

Note – Notify users before unmounting a file system that they are currently accessing.



To unmount a file system manually by using the directory mount point, perform the command:

```
# umount /export/home
```

To unmount a file system manually by using the logical disk device name, perform the command:

```
# umount /dev/dsk/c0t0d0s7
```

Unmounting All File Systems

The `/etc/mnttab` file is read by the `/usr/sbin/umountall` command during the system shutdown sequence or when `umountall` is invoked from the command line. The `umountall` unmounts all file systems specified in the `vfstab` file except `/` (root), `/usr`, `/proc`, `/dev/fd`, `/var`, `/var/run`, and `/tmp`.

Run the `umountall` command as the root user when you want to unmount manually all the file systems listed in the `/etc/mnttab` file, for example:

```
# umountall
```

To unmount only the local file systems listed in the `/etc/mnttab` file, perform the command:

```
# umountall -l
```

To verify that a file system or a number of file systems have been unmounted, invoke the `mount` command and check the output.

Unmounting a Busy File System

Any file system that is busy is not available for unmounting. Both the `umount` and `umountall` commands display the error message:

```
umount: file_system_name busy
```

A file system is considered to be busy if one of the following conditions exists:

- A program is accessing a file or directory in the file system
- A user is accessing a directory or file in the file system
- A program has a file open in that file system
- The file is being shared

There are two methods to make a file system available for unmounting if it is busy:

- `fuser` command – Lists all of the processes that are accessing the file system and kills them if necessary
- `umount -f` command – Forces the unmount of a file system



Note – The fuser command displays the process IDs of all processes currently using the specified file system. Each process ID is followed by a letter code. These letter codes are described in the man page for this command.

Using the `fuser` Command

To stop all processes that are currently accessing a file system, follow these steps:

1. As the root user, list all of the processes that are accessing the file system. Use the following command to identify which processes need to be terminated.

```
# fuser -cu mount_point
```

This command displays the name of the file system and the user login name for each process currently active in the file system.

2. Kill all processes accessing the file system.

```
# fuser -ck mount_point
```

A SIGKILL message is sent to each process that is using the file system.

3. Verify that there are no processes accessing the file system.

```
# fuser -c mount_point
```

4. Unmount the file system.

```
# umount mount_point
```

Using the `umount -f` Command

As the root user, you can unmount a file system, even if it is busy, by using the `-f` (force) option with the `umount` command. The following is the format for this command:

```
umount -f mount_point
```

The file system is unmounted even if it contains open files. A forced unmount can result in loss of data and in zombie processes that are left running on the system. However, it is particularly useful for unmounting a shared file system if the remote file server is nonfunctional.

Repairing Important Files if Boot Fails

The following procedure describes how to boot from the Solaris OS software CD-ROM or DVD to edit a misconfigured /etc/vfstab file.

1. Insert the Solaris 10 OS Software 1 of 4 CD-ROM into the CD-ROM drive.
2. Execute a single-user boot from the CD-ROM or DVD.

```
ok boot cdrom -s
Boot device: /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f File and args -s
SunOS Release 5.10 Generic 64 bit
Copyright 1983-2004 by Sun Microsystems, Inc. All rights reserved.
Booting to milestone "milestone/single-user:default"
Configuring /dev and /devices
Use is subject to license terms
Using RPC Bootparams for network configuration information.
Skipping interface hme0
-
INIT: SINGLE USER MODE
#
```



Note – Performing a single-user boot operation from this software CD-ROM creates an *in-memory* copy of the /root file system, which supports your ability to perform administrative tasks.

3. Use the fsck command on the faulty / (root) partition to check and repair any potential problems in the file system and make the device writable.

```
# fsck /dev/rdsck/c0t0d0s0
```

4. If the fsck command is successful, mount the / (root) file system on the /a directory to gain access to the file system on disk.

```
# mount /dev/dsk/c0t0d0s0 /a
```

5. Set and export the TERM variable, which enables the vi editor to work properly.

```
# TERM=sun
# export TERM
```

6. Edit the /etc/vfstab file, and correct any problems. Then exit the file.

```
# vi /a/etc/vfstab
:wq!
```

Performing Unmounts

7. Unmount the file system.

```
# cd /
# umount /a
```

8. Reboot the system.

```
# init 6
```

Accessing Mounted Diskettes, CD-ROMs or DVDs

To provide access to file systems on diskettes and CD-ROMs, the Solaris OS provides users a standard interface referred to as Volume Management (vold).



Note – The Solaris 10 OS includes support for additional removable media such as DVDs, Jaz drives, and Zip drives. For more information on using these devices, see the resources available on the Solaris 10 Documentation CD or visit <http://docs.sun.com> to access online documentation.

Volume Management vold provides two major benefits:

- It automatically mounts removable media for both the `root` user and non-root users.
- It can give other systems on the network automatic access to any removable media currently inserted in the local system.

The Volume Management (vold) service is controlled by the `/usr/sbin/vold` daemon. On a default install, this service is always running on the system so that it can automatically manage diskettes and CD-ROMs for regular users.

Volume Management (vold) features automatic detection of CD-ROMs. However, it does not detect the presence of a diskette that has been inserted in the drive until the `volcheck` command is run. This command instructs the vold daemon to check the diskette drive for any inserted media. Volume Management (vold) can mount `ufs`, `pcfs`, `hsfs`, and `udfs` file systems.

Using Volume Management (vold)

To make working with diskettes and CD-ROMs simple for your users, each device is easy to mount and mounts at an easy-to-remember location.

If the vold daemon detects that the mounted device contains a file system, then the device is mounted at the directory location.

Table 5-1 lists the directory locations of mounted devices that contain file systems.

Table 5-1 Directory Locations

Media Device	Access File Systems On
First diskette drive	/floppy/floppy0
First CD-ROM or DVD drive	/cdrom/cdrom0
First Jaz drive	/rmdisk/jaz0
First Zip drive	/rmdrive/zip0
First PCMCIA card	/pcmem0

If the vold daemon detects that the mounted device does not contain a file system, the device is accessible through a path.

Table 5-2 lists the paths for mounted devices that do not contain file systems.

Table 5-2 Paths for Accessing Devices

Media Device	Access Raw Device On
First diskette drive	/vol/dev/aliases/floppy0
First CD-ROM or DVD drive	/vol/dev/aliases/cdrom0
First Jaz drive	/vol/dev/aliases/jaz0
First Zip drive	/vol/dev/aliases/zip0
First PCMCIA card	/vol/dev/aliases/pcmem0

When Volume Management (`vold`) is running on the system, a regular user can easily access a diskette or CD-ROM by following these basic steps:

1. Insert the media.
2. For diskettes only, enter the `volcheck` command.
3. Use the `cd` command to change to the directory of the mounted volume.
4. Work with files on the media.
5. Use the `cd` command to leave the directory structure of the mounted volume.
6. Eject the media.

Table 5-3 shows the configuration files used by Volume Management (`vold`).

Table 5-3 Volume Management (`vold`) Configuration Files

File	Description
<code>/etc/vold.conf</code>	The Volume Management (<code>vold</code>) configuration file. This file defines items, such as what action should be taken when media is inserted or ejected, which devices are managed by Volume Management (<code>vold</code>), and which file system types are unsafe to eject.
<code>/etc/rmmount.conf</code>	The <code>rmmount</code> command configuration file. The <code>rmmount</code> command is a removable media mounter that is executed by the Volume Management (<code>vold</code>) daemon whenever a CD-ROM or diskette is inserted.

Restricting Access to Mounted Diskettes, CD-ROMs, or DVDs

To restrict regular users from accessing diskettes or CD-ROMs on the system, you can, as the `root` user, terminate the Volume Management (`vold`) service.

Stopping Volume Management (`vold`)

To stop Volume Management (`vold`) from running on a system temporarily, as the `root` user, perform the command:

```
# /etc/init.d/volmgt stop
```

To restart the Volume Management (`vold`) service, as the `root` user, perform the command:

```
# /etc/init.d/volmgt start
```

Troubleshooting Volume Management (`vold`) Problems

If a CD-ROM fails to eject from the drive, as the `root` user, attempt to stop Volume Management (`vold`). If this is unsuccessful, kill the `vold` daemon.

```
# /etc/init.d/volmgt stop
```

or as a last resort:

```
# pkill -9 vold
```

Push the button on the system to eject the CD-ROM. The CD-ROM tray ejects. Remove the CD-ROM, and leave the tray out. Then restart the Volume Management (`vold`) service.

```
# /etc/init.d/volmgt start
```

Wait a few seconds, and then push the CD-ROM tray back into the drive.

Accessing a Diskette, CD-ROM, or DVD Without Volume Management (vold)

When Volume Management (vold) is not running, only the root user can mount and access a diskette or CD-ROM. Follow these steps:

1. Insert the media device.
2. Become the root user.
3. Create a mount point, if necessary.
4. Determine the file system type.
5. Mount the device by using the mount options listed in the following sections.
6. Work with files on the media device.
7. Unmount the media device.
8. Eject the media device.
9. Exit the root session.

Using the mount Command

To mount a file system that resides on a CD-ROM when the Volume Management (vold) services are stopped, as the root user, perform the command:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
```

In this example, the file system type is hsfs, the file system resides on disk slice /dev/dsk/c0t6d0s0, and the mount point /cdrom is a preexisting directory in the Solaris OS.

To mount a file system that resides on a diskette when the Volume Management (vold) services are stopped, as the root user, perform the command:

```
# mkdir /pcfs  
# mount -F pcfs /dev/diskette /pcfs
```

In this example, the file system type is pcfs. This file system resides on the /dev/diskette device, and the mount point used is /pcfs.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Mounting File Systems (Level 1)

In this exercise, you complete the following tasks:

- Create mount points
- Mount file systems
- Specify mount options

Preparation

This exercise requires a spare disk that contains four unmounted ufs file systems on Slices 0, 1, 3, and 4. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Record the default mount options that are used by the / (root) file system mounted on your system. Mount the file system found on Slice 4 of your spare disk on the /morespace directory. Verify the mount options applied to the /morespace file system.
(Steps 1–3 in the Level 2 lab)
- Create a new file in the /morespace file system that contains one line of text. Record the modify time for this file. Use the ls command to display the last access time for this file. Record the time value. Wait one minute, and then display the file content. Again check and record the last access time for this file.
(Steps 4–7 in the Level 2 lab)

Exercise: Mounting File Systems (Level 1)

- Unmount the `/morespace` file system. Remount the same file system as `/morespace`, and use the `noatime` mount option. Again display the content of your text file. Check and record the last access time for it. Add a line to the `/etc/vfstab` file that mounts the `/morespace` file system when the system reboots. Reboot the system by using the `reboot` command, and verify that the `/morespace` file system is mounted.

(Steps 8–11 in the Level 2 lab)

- Mount the file system on Slice 0 as `/dir0`. Mount the file system on Slice 1 as `/dir0/dir1`. In a second terminal window, change to the `/dir0/dir1` directory. In the original terminal window, try to unmount the `/dir0` directory. Record the error messages. Attempt to forcibly unmount the `/dir0` directory. Record the result. Attempt to use the `pwd` command in the second terminal window. Record what happens.

(Steps 12–17 in the Level 2 lab)

Exercise: Mounting File Systems (Level 2)

In this exercise, you complete the following tasks:

- Create mount points
- Mount file systems
- Specify mount options

Preparation

This exercise requires a spare disk that contains four unmounted ufs file systems on Slices 0, 1, 3, and 4. Refer to the lecture notes as necessary to perform the tasks listed. Be aware, that the path name of your disk device might differ from that used in these exercise notes. Be sure to use the device path name appropriate to your system.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Record the default mount options that are used by the / (root) file system mounted on your system. Mount the file system found on Slice 4 of your spare disk on the /morespace directory. Verify the mount options applied to the /morespace file system.
- Create a new file in the /morespace file system that contains one line of text. Record the modify time for this file. Use the ls command to display the last access time for this file. Record the time value. Wait one minute, and then display the file content. Again check and record the last access time for this file.

- Unmount the `/morespace` file system. Remount the same file system as `/morespace`, and use the `noatime` mount option. Again display the content of your text file. Check and record the last access time for it. Add a line to the `/etc/vfstab` file that mounts the `/morespace` file system when the system reboots. Reboot the system using the `reboot` command, and verify that the `/morespace` file system is mounted.
- Mount the file system on Slice 0 as `/dir0`. Mount the file system on Slice 1 as `/dir0/dir1`. In a second terminal window, change to the `/dir0/dir1` directory. In the original terminal window, try to unmount the `/dir0/dir1` directory. Record the error messages. Attempt to forcibly unmount the `/dir0/dir1` directory. Record the result. Attempt to use the `pwd` command in the second terminal window. Record what happens.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. Use the `mount` command to list the file systems that are currently mounted on your system. What are the default mount options applied to the `/` (root) file system?
2. Create the directory `/morespace` to use as the mount point.
3. Mount the file system on Slice 4 of your spare disk to the `/morespace` directory. Record the default mount options that were applied to this mount.
4. Change to the `/morespace` directory, and create a new file that has one line of content.
5. Display a long listing for this file, and record the time value it reports. This time value represents when the file was last modified.
6. Add the `-u` option to the `ls` command to show when the file was last accessed. This time value is updated whenever you read the file.
7. Wait one minute or more, and then use the `cat` command to display the file. Again check and record the access time. The access time should differ from the access time indicated in the previous step.
8. Change to the `/` (root) directory. Unmount the `/morespace` file system. Remount the same file system to the `/morespace` directory, but add the option that prevents update of access time values. Verify that the options to the mount were applied.

9. Return to the /morespace file system, and use the cat command to display your test file. Again check and record the access time. The access time should match the access time that existed prior to your unmounting and mounting the /morespace file system.
10. Add a line to the /etc/vfstab file to make the mount for the /morespace file system happen when you boot the system.
11. Reboot your system. Log in as the root user, and open a terminal window. Use the mount command to verify that the /morespace file system is mounted.
12. Create a directory called /dir0. Mount the file system that resides on Slice 0 of your spare disk as /dir0.
13. Create a directory called /dir0/dir1. Mount the file system that resides on Slice 1 of your spare disk as /dir0/dir1.
14. Open a second terminal window. In this new window, change the directory to /dir0/dir1.
15. In your original terminal window, attempt to unmount the file system mounted below the /dir0/dir1 directory. Which message is displayed? Does the file system unmount?



Note – To discover why you could not unmount the file system, use the fuser -cu /dir0/dir1 command. The fuser command should show the process ID of the shell.

16. In your original terminal window, again attempt to unmount the file system mounted below the /dir0/dir1 directory. Add the -f option to the umount command. Which message is displayed? Does the file system unmount?
17. In the second terminal window, attempt to determine your current working directory. Which message is displayed? Change the directory to / (root), and verify that the pwd command works.

Exercise: Mounting File Systems (Level 3)

In this exercise, you complete the following tasks:

- Create mount points
- Mount file systems
- Specify mount options

Preparation

This exercise requires a spare disk that contains four unmounted ufs file systems on Slices 0, 1, 3, and 4. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Record the default mount options that are used by the / (root) file system mounted on your system. Mount the file system found on Slice 4 of your spare disk on the /morespace directory. Verify the mount options applied to the /morespace file system.
- Create a new file in the /morespace file system that contains one line of text. Record the modify time for this file. Use the ls command to display the last access time for this file. Record the time value. Wait one minute, and then display the file content. Again check and record the last access time for this file.

- Unmount the /morespace file system. Remount the same file system as /morespace, and use the noatime mount option. Again display the content of your text file. Check and record the last access time for it. Add a line to the /etc/vfstab file that mounts the /morespace file system when the system reboots. Reboot the system using the reboot command, and verify that the /morespace file system is mounted.
- Mount the file system on Slice 0 as /dir0. Mount the file system on Slice 1 as /dir0/dir1. In a second terminal window, change to the /dir0/dir1 directory. In the original terminal window, try to unmount the /dir0/dir1 directory. Record the error messages. Attempt to forcibly unmount the /dir0/dir1 directory. Record the result. Attempt to use the pwd command in the second terminal window. Record what happens.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Use the mount command to list the file systems that are currently mounted on your system. What are the default mount options applied to the / (root) file system?

```
# mount
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/
dev=2200000
```

Your dev= number depends on the architecture of your system.

2. Create the directory /morespace to use as the mount point.

```
# mkdir /morespace
```

3. Mount the file system on Slice 4 of your spare disk to the /morespace directory. Record the default mount options that were applied to this mount.

```
# mount /dev/dsk/c1t0d0s4 /morespace
```

```
# mount
```

```
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/
dev=80001c
```

4. Change to the /morespace directory, and create a new file that has one line of content.

```
# cd /morespace
```

```
# echo "Some Text" > testfile
```

```
#
```

Exercise: Mounting File Systems (Level 3)

5. Display a long listing for this file, and record the time value it reports. This time value represents when the file was last modified.

```
# ls -l
```

6. Add the `-u` option to the `ls` command to show when the file was last accessed. This time value is updated whenever you read the file.

```
# ls -lu
```

7. Wait one minute or more, and then use the `cat` command to display the file. Again check and record the access time. The access time should differ from the access time indicated in the previous step.

```
# cat testfile
```

```
This is a test
```

```
# ls -lu
```

8. Change to the `/` (root) directory. Unmount the `/morespace` file system. Remount the same file system to the `/morespace` directory, but add the option that prevents update of access time values. Verify that the options to the mount were applied.

```
# cd /
```

```
# umount /morespace
```

```
# mount -o noatime /dev/dsk/c1t0d0s4 /morespace
```

```
# mount
```

9. Return to the `/morespace` file system, and use the `cat` command to display your test file. Again check and record the access time. The access time should match the access time that existed prior to your unmounting and mounting the `/morespace` file system.

```
# cd /morespace
```

```
# cat testfile
```

```
This is a test
```

```
# ls -lu
```

10. Add a line to the `/etc/vfstab` file to make the mount for the `/morespace` file system happen when you boot the system.

```
/dev/dsk/c1t0d0s4 /dev/rdsck/c1t0d0s4 /morespace ufs 2 yes noatime
```

11. Reboot your system. Log in as the root user, and open a terminal window. Use the `mount` command to verify that the `/morespace` file system is mounted.

```
# init 6
```

```
(reboot messages & login prompts)
```

```
# mount
```

12. Create a directory called `/dir0`. Mount the file system that resides on Slice 0 of your spare disk as `/dir0`.

```
# mkdir /dir0
```

```
# mount /dev/dsk/c1t0d0s0 /dir0
```

13. Create a directory called /dir0/dir1. Mount the file system that resides on Slice 1 of your spare disk as /dir0/dir1.

```
# mkdir /dir0/dir1
# mount /dev/dsk/c1t0d0s1 /dir0/dir1
```

14. Open a second terminal window. In this new window, change the directory to /dir0.

```
# cd /dir0
```

15. In your original terminal window, attempt to unmount the file system mounted below the /dir0 directory. What message is displayed? Does the file system unmount?

```
# umount /dev/dsk/c1t0d0s0
# mount
umount: /dir0 busy
```

The file system does not unmount.



Note – To discover why you could not unmount the file system, use the fuser -cu /dir0 command. The fuser command should show the process ID of the shell.

16. In your original terminal window, again attempt to unmount the file system mounted below the /dir0/dir1 directory. Add the -f option to the umount command. Which message is displayed? Does the file system unmount?

```
# umount -f /dir0
# mount
```

No messages are displayed. The file system unmounts.

17. In the second terminal window, attempt to determine your current working directory. Which message is displayed? Change the directory to / (root), and verify that the pwd command works.

```
# pwd
Cannot determine current directory.
```



Note – You may get a different error if you are using another shell. Users of the BASH shell see the correct directory.

```
# cd /
# pwd
/
```

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 6

Performing Solaris 10 OS Package Administration

Objectives

Upon completion of this module, you should be able to:

- Describe the fundamentals of package administration
- Administer packages using the command-line interface

The course map in Figure 6-1 shows how this module fits into the current instructional goal.

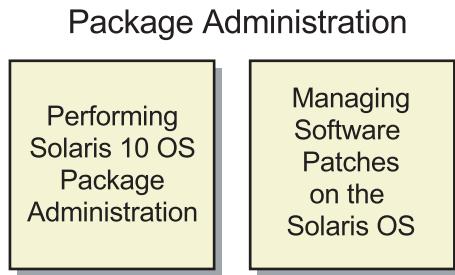


Figure 6-1 Course Map

Introducing the Fundamentals of Package Administration

Software package administration adds software to systems and removes software from systems. Sun and its third-party vendors deliver software products to users in software packages.

Software Packages

The term package refers to the method of distributing software products and installing them in systems. In its simplest form, a package is a collection of files and directories.

Note – The Solaris OS software installation process installs all the required software packages automatically, based on the software group configuration choice.



Software packages contain:

- Files that describe the package and the amount of disk space required for installation
- Compressed software files to be installed on the system
- Optional scripts that run when the package is added or removed

The /var/sadm/install/contents File

The /var/sadm/install/contents file is a complete record of all the software packages installed on the local system disk. It references every file and directory belonging to every software package and shows the configuration of each product installed. To list the contents of the /var/sadm/install/contents file, perform the command:

```
# more /var/sadm/install/contents
(output edited for brevity)
/bin=./usr/bin s none SUNWcsr
/dev d none 0755 root sys SUNWcsr SUNWcsd
/dev/allkmem=../devices/pseudo/mm@0:allkmem s none SUNWcsd
/dev/arp=../devices/pseudo/arp@0:arp s none SUNWcsd
/etc/ftpd/ftpusers e ftpusers 0644 root sys 198 16387 1094222536 SUNWftpr
/etc/passwd e passwd 0644 root sys 580 48298 1094222123 SUNWcsr
```

The `pkgadd` command updates the `/var/sadm/install/contents` file each time new packages are installed.

The `pkgrm` command uses the `/var/sadm/install/contents` file to determine where the files for a software package are located on the system. When a package is removed from the system, the `pkgrm` command updates the `/var/sadm/install/contents` file.

To determine if a particular file was installed on the system disk and to find the directory in which it is located, use the `pkgchk` command with either the full or partial path name of the command you want to report on. For example, to verify that the `showrev` command is installed on the system disk, perform the command:

```
# pkgchk -l -P showrev
Pathname: /usr/bin/showrev
Type: regular file
Expected mode: 0755
Expected owner: root
Expected group: sys
Expected file size (bytes): 29980
Expected sum(1) of contents: 57864
Expected last modification: Dec 14 06:17:58 AM 2004
Referenced by the following packages:
    SUNWadmc
Current status: installed

Pathname: /usr/share/man/man1m/showrev.1m
Type: regular file
Expected mode: 0644
Expected owner: root
Expected group: root
Expected file size (bytes): 3507
Expected sum(1) of contents: 35841
Expected last modification: Dec 10 10:42:54 PM 2004
Referenced by the following packages:
    SUNWman
Current status: installed
```

Package Formats

Solaris OS packages can be in one of two formats:

- File system (or Directory) format
- Data stream format

Packages delivered in file system format consist of multiple files and directories. Packages delivered in data stream format consist of a single file only.

File System Format

An example of a package (SUNWrsc) in file system format:

```
# ls -ld SUNWrsc
drwxr-xr-x  3 root      root          377 Dec 13 15:55 SUNWrsc
# cd SUNWrsc
# ls -l
total 3280
drwxr-xr-x  2 root      other         249 Dec 13 15:55 install
-rw-r--r--  1 root      other        372 May  3 2003 pkginfo
-rw-r--r--  1 root      other       3667 May  3 2003 pkgmap
-rw-r--r--  1 root      other    1648011 May  3 2003 reloc.cpio.Z
#
```

The package consists of a directory that matches the package name, and other files and directories including the `pkginfo` and `pkgmap` files.

Data Stream Format

An example of a package in data stream format:

```
# ls -l SUNWrsc.pkg
-rw-r--r-- 1 root      root      1658880 Dec 13 15:59 SUNWrsc.pkg
# file SUNWrsc.pkg
SUNWrsc.pkg:package datastream
# head SUNWrsc.pkg
# PaCkAgE DaTaStReAm
SUNWrsc 1 3266
# end of header
SUNW_PRODVERS=2.2.1
SUNW_PKGVERS=1.0
PKG=SUNWrsc
NAME=Remote System Control
DESC=Sun Remote System Control system software
ARCH=sparc
VENDOR=Sun Microsystems, Inc.
#
```

Packages downloaded from the Internet are most often in data stream format.

Administering Packages From the Command Line

From the command line, you can translate, add, remove, check the state of, and display information about packages.

The command-line tools for translating packages, viewing software, adding software, and removing software from a workstation after the Solaris OS software is installed on a system include:

pkgtrans	Translates packages from one format to another
pkgadd	Installs software packages to the system
pkgrm	Removes a package from the system
pkginfo	Displays software package information
pkgchk	Checks package installation state

Translating Package Formats

Use the `pkgtrans` command to translate a package from file system format to data stream format, or from data stream format to file system format. The command syntax for the `pkgtrans` command is:

```
# pkgtrans file_or_dir_path file_or_dir_path [ package_name ... ]
```

For example, to translate a package from file system format in `/var/tmp` to data stream format, use:

```
# pkgtrans /var/tmp /tmp/SUNWrsc.pkg SUNWrsc
Transferring <SUNWrsc> package instance
```

The first argument above is the directory where the file system format package is stored. The second argument is the package data stream file. The third argument is the package to translate.

If a package name is not given, the `pkgtrans` command provides a list of all packages in the directory, and prompts the user for the packages to translate.

Note – Students need to insert the appropriate Solaris 10 OS Software CD-ROM or Solaris 10 OS Software DVD to demonstrate the steps described in this module.



Displaying Information About Installed Software Packages

Use the `pkginfo` command to display information about the software packages installed on the local system's disk. The `/var/sadm/pkg` directory maintains a record of all installed packages.

For example, to display information about software packages installed on the local system's disk, perform the command:

```
# pkginfo | more
<some output omitted>
system      SUNWaccr  System Accounting, (Root)
system      SUNWaccu  System Accounting, (Usr)
system      SUNWaclg  Apache Common Logging
system      SUNWadmap System administration applications
system      SUNWadmc  System administration core libraries
system      SUNWadmfw System & Network Administration Framework
system      SUNWadmj  Admin/Install Java Extension Libraries
system      SUNWadmr  System & Network Administration Root
ALE         SUNWciu8  Simplified Chinese (EUC) iconv modules for UTF-8
CTL         SUNWctpls Portable layout services for Complex Text Layout
support
```

The column on the left displays the package category, such as application, system, Complex Text Layout (CTL), or Alternate Language Environment (ALE). A CTL language is any language which stores text differently than it is displayed. An ALE is an alternate language, different from the basic Solaris OS languages.

The center column displays the software package name. If it begins with `SUNW`, it is a Sun Microsystems product. Otherwise, it represents a third-party package.

The column on the right displays a brief description of the software product.

Displaying Information for All Packages

To display all the available information about the software packages, use the **pkginfo** command with the **-l** option.

For example, to view additional information about each software package installed on the local systems hard drive, perform the command:

```
# pkginfo -l | more  
(output omitted)
```

Displaying Information for a Specific Package

To display the information for a specific software package, specify its name on the command line.

For example, to view the information for the SUNWman software package, perform the command:

```
# pkginfo -l SUNWman  
  
PKGINST: SUNWman  
NAME: On-Line Manual Pages  
CATEGORY: system  
ARCH: sparc  
VERSION: 43.0,REV=67.0  
BASEDIR: /usr  
VENDOR: Sun Microsystems, Inc.  
DESC: System Reference Manual Pages  
PSTAMP: 2004.09.01.17.00  
INSTDATE: Sep 24 2004 12:32  
HOTLINE: Please contact your local service provider  
STATUS: completely installed  
FILES: 11383 installed pathnames  
      8 shared pathnames  
      97 directories  
      119848 blocks used (approx)
```

The last line identifies the size of the package. The number of blocks used defines how much space is needed on the disk to install the package.



Note – A block is a 512-byte disk block.

To determine how many packages are currently installed on disk, perform the command:

```
# pkginfo | wc -l  
657
```

Displaying Information for Software Packages

To view information about packages that are located on the Solaris 10 OS Software 1 CD-ROM, perform the command:

```
pkginfo -d /cdrom/cdrom0/s0/Solaris_10/Product |more
```

The software groups located on Solaris 10 OS Software 1 CD-ROM are Reduced Networking Core System Support and Core System Support.

To view information about packages that are located on any of the remaining Solaris 10 Software CD-ROMs or on the Solaris 10 OS Software DVD, perform the command:

```
pkginfo -d /cdrom/cdrom0/Solaris_10/Product |more
```

The software groups located on Solaris 10 OS Software 2, 3, and 4 CD-ROMs are the End User System Support, Developer System Support, Entire Distribution, and Entire Distribution Plus OEM Support software groups.

Adding a Software Package

When you add a software package, the `pkgadd` command copies the files from the installation media to the local system's disk and executes scripts to uncompress files. By default, the `pkgadd` command requires confirmation during the package add process.

For example, to transfer the `SUNWvts` software package from a CD-ROM and install it on the system, perform the commands:

```
# cd /cdrom/cdrom0/Solaris_10/ExtraValue/CoBundled/SunVTS_6.0/Packages  
# pkgadd -d . SUNWvts
```

```
Processing package instance < SUNWvts > from
</cdrom/sol_10_sparc_4/Solaris_10/ExtraValue/CoBundled/SunVTS_6.0/Package
s>
```

```
SunVTS Framework(sparc) 6.0,REV=2004.08.18.12.00
Copyright 2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Using </opt> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
```

This package contains scripts which will be executed with super-user permission during the process of installing this package.

```
Do you want to continue with the installation of < SUNWvts > [y,n,?] y
```

```
Installing SunVTS Framework as < SUNWvts >
```

```
## Installing part 1 of 1.
9213 blocks
```

Installation of < SUNWvts > was successful.

To install all packages in a data stream format package, perform the command:

```
# pkgadd -d /tmp/SUNWrsc.pkg all
```

```
Processing package instance < SUNWrsc > from </tmp/SUNWrsc.pkg>
```

```
Remote System Control(sparc) 2.2.1,REV=2002.02.11
Copyright 2001 Sun Microsystems, Inc. All rights reserved.
Using </> as the package base directory.
## Processing package information.
## Processing system information.
15 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
```

```
Installing Remote System Control as < SUNWrsc >
```

```
## Installing part 1 of 1.
```

```
10499 blocks
```

```
Installation of <SUNWrsc> was successful.  
#
```

Packages in data stream format can also be added from a web server using the following command:

```
# pkgadd -d http://instructor/packages/SUNWrsc.pkg all  
  
## Downloading...  
.....25%.....50%.....75%.....100%  
## Download Complete
```

```
Processing package instance <SUNWrsc> from  
<http://instructor/packages/SUNWrsc.pkg>
```

```
Remote System Control(sparc) 2.2.1,REV=2002.02.11  
Copyright 2001 Sun Microsystems, Inc. All rights reserved.  
Using </> as the package base directory.  
## Processing package information.  
## Processing system information.  
    15 package pathnames are already properly installed.  
## Verifying disk space requirements.  
## Checking for conflicts with packages already installed.  
## Checking for setuid/setgid programs.
```

```
Installing Remote System Control as <SUNWrsc>
```

```
## Installing part 1 of 1.  
10499 blocks
```

```
Installation of <SUNWrsc> was successful.  
#
```

Checking a Package Installation

The **pkgchk** command checks to determine if a package has been completely installed on the system. The **pkgchk** command also checks the path name, the file size and checksum, and the file attributes of a package. If the **pkgchk** command does not display a message, it indicates the package was installed successfully and that no changes have been made to any files or directories in the package.

The following example checks the contents and attributes of the **SUNWladm** software package currently installed on the system.

```
# pkgchk SUNWladm  
#
```

To list the files contained in a software package, use the **-v** option.

For example, to list the files in the **SUNWladm** software package, perform the command:

```
# pkgchk -v SUNWladm  
/usr  
/usr/sadm  
/usr/sadm/lib  
/usr/sadm/lib/localeadm  
/usr/sadm/lib/localeadm/Locale_config_S10.txt  
/usr/sadm/lib/localeadm/admin  
/usr/sbin  
/usr/sbin/localeadm
```

To determine if the contents and attributes of a file have changed because it was installed with its software package, use the **-p** option.

For example, to check the **/etc/shadow** file, perform the command:

```
# pkgchk -p /etc/shadow  
ERROR: /etc/shadow  
      modtime <09/03/04 03:35:24 PM> expected <09/30/04 08:06:14 PM> actual  
      file size <296> expected <309> actual  
      file cksum <20180> expected <21288> actual
```

The differences in `modtime`, `file size`, and `checksum` indicate that the original `/etc/shadow` file has changed in size because the initial Solaris OS software installation.

The `-l` option with the `pkgchk` command lists information about selected files that make up a package.

For example, to list information about the `/usr/bin/showrev` file, perform the command:

```
# pkgchk -l -p /usr/bin/showrev
Pathname: /usr/bin/showrev
Type: regular file
Expected mode: 0755
Expected owner: root
Expected group: sys
Expected file size (bytes): 29656
Expected sum(1) of contents: 31261
Expected last modification: Sep 02 09:21:11 2004
Referenced by the following packages:
    SUNWadmc
Current status: installed
```

If the `-p` option is used, the full path must be typed for the `pkgchk` command to return information about the file. If the `-P` option is used, a partial path name can be supplied.

For example, the `pkgchk` command does not return any information if the `/usr/bin/` path is removed from the previous example.

```
# pkgchk -l -p showrev
#
```

Removing a Software Package

The pkgrm command removes a software package from the system and deletes all of the files associated with that package, unless other packages share those files.

By default, the pkgrm command requires confirmation to continue removing a package and issues a message to warn about possible package dependencies. If package dependencies do exist, the command again requires confirmation to continue with the package removal process.

The following command removes the SUNWapchr software package from the system.



Caution – Be cautious of the dependency warnings you receive when removing a package. The system allows you to remove these packages even though they may be required by a different package.

```
# pkgrm SUNWapchr
```

The following package is currently installed:

```
SUNWapchr      Apache Web Server (root)
                (sparc) 11.10.0,REV=2004.08.20.02.37
```

```
Do you want to remove this package? [y,n,?,q] y
```

```
## Removing installed package instance <SUNWapchr>
## Verifying package dependencies.
```

WARNING:

```
The <SUNWapchu> package depends on the package
currently being removed.
```

WARNING:

```
The <SUNWapchd> package depends on the package
currently being removed.
```

WARNING:

```
The <SUNWippllr> package depends on the package
currently being removed.
```

WARNING:

```
The <SUNWserweb> package depends on the package
currently being removed.
```

Dependency checking failed.

```
Do you want to continue with the removal of this package [y,n,?,q] y
```

```
## Processing package information.
```

```
## Removing pathnames in class <initrd>
```

```
/etc/rcS.d/K16apache  
/etc/rc3.d/S50apache  
/etc/rc2.d/K16apache  
  
(output omitted for brevity)  
  
/etc/apache/httpd.conf-example  
/etc/apache/README.Solaris  
/etc/apache <shared pathname not removed>  
/etc <shared pathname not removed>  
## Updating system information.  
  
Removal of <SUNWapchr> was successful.
```



Note – A file shared by two or more packages displays the message *filename <shared pathname not removed>*. The message is removed only when the file is no longer shared.

Adding Packages by Using a Spool Directory

For convenience, copy frequently installed software packages from the Solaris 10 Software CD-ROMs or Solaris 10 Software DVD to a spool directory on the system.

The default installation directory for packages that have been spooled, but not installed, is */var/spool/pkg*. The *pkgadd* command, by default, looks in the */var/spool/pkg* directory for any packages specified on the command line.

Copying packages from the CD-ROM or DVD into a spool directory is not the same as installing the packages on disk.

To copy a package from the Solaris 10 OS Software CD-ROM into the */var/spool/pkg* directory, perform the command:

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_10/Product -s spool SUNWauda  
Transferring <SUNWauda> package instance
```

The *-s* option with the keyword *spool* copies the package into the */var/spool/pkg* directory by default.

To verify that the package exists in the spool directory, perform the command:

```
# ls -al /var/spool/pkg
total 6
drwxrwxrwt 3 root bin 512 Oct 1 14:26 .
drwxr-xr-x 12 root bin 512 Sep 30 20:03 ..
drwxrwxr-x 5 root root 512 Oct 1 14:26 SUNWauda
```

To add the package from the spool area, perform the following:

```
# pkgadd SUNWauda
```

(Some output is omitted.)

To remove software packages from a spool directory, use the pkgrm command with the -s option.

```
# pkgrm -s spool SUNWauda
```

The following package is currently spooled:

```
SUNWauda      Audio Applications
(sparc) 11.10.0,REV=2004.09.03.08.15
```

Do you want to remove this package? [y,n,?,q] **y**

Removing spooled package instance <SUNWauda>

If alternative spooling directories exist, specify which directory to use by adding a directory path to the -s option.

For example, to select the /export/pkg directory, perform the commands:

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_10/Product -s /export/pkg SUNWauda
# pkgrm -s /export/pkg SUNWauda
```

Streaming One or More Packages

Packages can be individually or collectively packaged into a data stream file format. The data stream file can then be made available as a shared network file or from a web page.

Worked Example:

To create a data streamed package, perform the following commands:

```
# cd /cdrom/cdrom0/s0/Solaris*
# pkgtrans -s Product /var/tmp/stream.pkg SUNWzlib SUNWftpr SUNWftpu
Transferring <SUNWzlib> package instance
Transferring <SUNWftpr> package instance
Transferring <SUNWftpu> package instance

# file /var/tmp/stream.pkg
/var/tmp/stream.pkg:      package datastream

# head -5 /var/tmp/stream.pkg
# PaCkAgE DaTaStReAm
SUNWzlib 1 186
SUNWftpr 1 70
SUNWftpu 1 300
# end of header

# pkgadd -d /var/tmp/stream.pkg
```

The following packages are available:

- | | | |
|---|----------|--------------------------------------|
| 1 | SUNWftpr | FTP Server, (Root) |
| | | (sparc) 11.10.0,REV=2004.12.11.01.30 |
| 2 | SUNWftpu | FTP Server, (Usr) |
| | | (sparc) 11.10.0,REV=2004.12.11.01.30 |
| 3 | SUNWzlib | The Zip compression library |
| | | (sparc) 11.10.0,REV=2004.12.10.05.25 |

Select package(s) you wish to process (or 'all' to process all packages). (default: all) [?,??,q]: q

Reviewing Package Administration

This section details the package administration tasks.

Table 6-1 summarizes the commands used for package administration.

Table 6-1 Package Administration Commands

Command Name	Description
pkginfo	Lists packages installed on the system or available on distribution media
pkgadd	Installs packages
pkgrm	Removes packages
pkgchk	Verifies the attributes of the path names that belong to packages

Table 6-2 summarizes the files and directories used in package administration.

Table 6-2 Package Administration Files and Directories

File or Directory	Description
/var/sadm/install/contents	A software package map of the entire system
/opt/ <i>pkgname</i>	The preferred location for the installation of unbundled packages
/opt/ <i>pkgname</i> /bin or /opt/bin	The preferred location for the executable files of unbundled packages
/var/opt/ <i>pkgname</i> or /etc/opt/ <i>pkgname</i>	The preferred location for log files of unbundled packages

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Manipulating Software Packages (Level 1)

In this exercise, you use package-related commands to remove, install, and spool packages.

Preparation

Locate the Solaris 10 Software CD-ROMs or Solaris 10 OS Software DVD. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

For this particular exercise, a Solaris 10 OS DVD is installed in the RLDC systems.

Tasks

Complete the following tasks:

- Find the names of packages installed on your system that relate to manuals. List and record the status of, the install date of, the number of files used by, and the number of blocks used by the SUNWman package. Obtain the same information from the spooled SUNWman package on the correct Solaris 10 OS Software CD-ROM or Solaris 10 OS Software DVD. Remove and reinstall the SUNWman package.
(Steps 1–6 in the Level 2 lab)
- Remove the SUNWdoc package from the system. Attempt to access the online man pages. Spool the SUNWdoc package from the correct Solaris 10 OS Software CD-ROM or Solaris 10 OS Software DVD into the default spool area. Verify the presence of this package in the spool area. Add the SUNWdoc package to the system. Remove the SUNWdoc package from the spool area.
(Steps 7–15 in the Level 2 lab)

Exercise: Manipulating Software Packages (Level 2)

In this exercise, you use package-related commands to remove, install, and spool packages.

Preparation

Locate the Solaris 10 Software CD-ROMs or Solaris 10 OS Software DVD. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

For this particular exercise, a Solaris 10 OS DVD is installed in the RLDC systems.

Task Summary

In this exercise, you accomplish the following:

- Find the names of packages installed on your system that relate to manuals. List and record the status of, the install date of, the number of files used by, and the number of blocks used by the SUNWman package. Obtain the same information from the spooled SUNWman package on the correct Solaris 10 OS Software CD-ROM or Solaris 10 OS Software DVD. Remove and reinstall the SUNWman package.
- Remove the SUNWdoc package from the system. Attempt to access the online man pages. Spool the SUNWdoc package from the correct Solaris 10 OS software CD-ROM or Solaris 10 OS Software DVD into the default spool area. Verify the presence of this package in the spool area. Add the SUNWdoc package to the system. Remove the SUNWdoc package from the spool area.

Tasks

Complete the following steps:

1. Insert the Solaris 10 Software 4 of 4 CD-ROM SPARC Platform Edition or the Solaris 10 Software DVD into the drive.
2. Use the `pkginfo` command to search for packages currently on your system that are related to manuals.
Which packages were listed?
3. Display a long-format listing of the information for the `SUNWman` package installed on your system. What is listed for the status of, the install date of, the number of files used by, and the number of blocks used by this package?
4. Display a long-format listing of the information for the `SUNWman` package on the Solaris 10 OS Software 4 of 4 CD-ROM or Solaris 10 OS Software DVD. Obtain the same information as in the previous step.

Note – Steps 5 and 6 take several minutes to perform.



5. Remove the `SUNWman` package from your system, and verify that it has been removed by trying to access the manual pages.
6. Reinstall the `SUNWman` package from the Solaris 10 OS Software 4 of 4 CD-ROM or Solaris 10 OS Software DVD. Respond `y` to questions asked by the `pkgadd` command. Verify that the manual pages work.
7. Remove the `SUNWdoc` package from your system and answer yes to the remove questions.
8. Are there any package dependencies related to removing this package?
9. If using CD-ROMs, eject the Solaris 10 Software 4 of 4 CD-ROM, and insert the Solaris 10 Software 2 of 4 CD-ROM.
10. Use the `pkgadd` command to spool the `SUNWdoc` package into the default spool area.
11. Use the `pkginfo` command with the appropriate options to verify the presence of the `SUNWdoc` package in the default spool area.
12. Install the `SUNWdoc` package. Observe the messages, and verify that the package is installed from the `/var/spool/pkg` directory.
13. Remove the `SUNWdoc` package from the default spool area.

14. Verify that the SUNWdoc package no longer exists in the spool area and that it is installed on your system.
15. Eject the Solaris 10 Software 4 of 4 CD-ROM or DVD.

Exercise: Manipulating Software Packages (Level 3)

In this exercise, you use package-related commands to remove, install, and spool packages.

Preparation

Locate the Solaris 10 Software CD-ROMs or Solaris 10 OS Software DVD. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

For this particular exercise, a Solaris 10 OS DVD is installed in the RLDC systems.

Task Summary

In this exercise, you accomplish the following:

- Find the names of packages installed on your system that relate to manuals. List and record the status of, the install date of, the number of files used by, and the number of blocks used by the SUNWman package. Obtain the same information from the spooled SUNWman package on the correct Solaris 10 OS Software CD-ROM or Solaris 10 OS Software DVD. Remove and reinstall the SUNWman package.
- Remove the SUNWdoc package from the system. Attempt to access the online man pages. Spool the SUNWdoc package from the correct Solaris 10 OS software CD-ROM or Solaris 10 OS Software DVD into the default spool area. Verify the presence of this package in the spool area. Add the SUNWdoc package to the system. Remove the SUNWdoc package from the spool area.

Tasks and Solutions

Complete the following steps:

1. Insert the Solaris 10 Software 4 of 4 CD-ROM SPARC Platform Edition or the Solaris 10 Software DVD into the drive.
2. Use the `pkginfo` command to search for packages currently on your system that are related to manuals.

```
# pkginfo | grep -i manual
```

Which packages were listed?

`SUNWman`, `SUNWmfman`, `SUNWopenssl-man`, `SUNWperl584man`,
`SUNWp15m`, and `SUNWt1tkm`

These packages contain the online manual pages, CDE motif manuals, Secure-Shell manual, Perl Reference manual pages for two versions of Perl, and ToolTalk™ software manual pages, respectively.

3. Display a long-format listing of the information for the `SUNWman` package installed on your system.

```
# pkginfo -l SUNWman
```

What is listed for the status of, the install date of, the number of files used by, and the number of blocks used by this package?

Status: Completely installed

Install date: Should match the date and time when you installed Solaris OS on your system

Number of files: xxxx installed path names, x shared directories, xx directories

Number of blocks: xxxxx

Exercise: Manipulating Software Packages (Level 3)

4. Display a long-format listing of the information for the SUNWman package on the Solaris 10 OS Software 4 of 4 CD-ROM or the Solaris 10 OS Software DVD. Obtain the same information as in the previous step.

```
# pkginfo -d /cdrom/cdrom0/Solaris_10/Product -l SUNWman
```

<i>Status:</i>	<i>Spooled</i>
<i>Install date:</i>	<i>No install date indicated</i>
<i>Number of files:</i>	<i>xxxx spooled path names, xx directories, x package information files</i>
<i>Number of blocks:</i>	<i>xxxxx</i>

5. Remove the SUNWman package from your system, and verify that it has been removed by trying to access the manual pages.

```
# pkgrm SUNWman
```

```
# pkginfo SUNWman
```

```
ERROR: information for "SUNWman" was not found
```

```
# man ls
```

```
No manual entry for ls.
```

6. Reinstall the SUNWman package from the Solaris 10 OS Software 2 of 4 CD-ROM or Solaris 10 OS Software DVD. Respond **y** to questions asked by the pkgadd command. Verify that the manual pages work.

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product SUNWman
```

```
# man ls
```

The manual page for ls appears.

7. Check the package and then remove the SUNWdoc package from your system.

```
# pkginfo SUNWdoc
```

```
system      SUNWdoc Documentation Tools
```

```
# pkgrm SUNWdoc
```

```
The following package is currently installed:
```

```
  SUNWdoc  Documentation Tools  
    (sparc) 11.10.0,REV=2004.12.11.01.30
```

```
Do you want to remove this package? [y,n,?,q] y
```

```
## Removing installed package instance <SUNWdoc>
```

```
This package contains scripts which will be executed with super-user
```

permission during the process of removing this package.

```
Do you want to continue with the removal of this package [y,n,?,q] y
## Verifying package <SUNWdoc> dependencies in global zone
WARNING:
```

The <SUNWuium> package depends on the package currently being removed.

(output removed for brevity)

```
Do you want to continue with the removal of this package [y,n,?,q] y
## Processing package information.
```

```
## Executing preremove script.
```

```
## Removing pathnames in class <none>
```

```
/usr/share/man <shared pathname not removed>
```

```
/usr/share/lib/tmac/vgrind
```

```
/usr/share/lib/tmac/v
```

(output removed for brevity)

```
## Updating system information.
```

Removal of <SUNWdoc> was successful.

8. Answer yes to questions from the pkgrm command.
9. Are there any package dependencies related to removing this package?

Yes there are. They are five other packages dependent on the SUNWdoc package.

10. If using CD-ROMs, ensure that the Solaris 10 OS Software 2 of 4 CD-ROM is inserted in the CD-drive.
11. Use the pkgadd command to spool the SUNWdoc package into the default spool area.

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product -s spool SUNWdoc
```

12. Use the pkginfo command with the appropriate options to verify the presence of the SUNWdoc package in the default spool area.

```
# pkginfo -d spool SUNWdoc
```

```
system SUNWdoc Documentation Tools
```

```
# pkginfo -d /var/spool/pkg -l SUNWdoc
```

```
PKGINST: SUNWdoc
```

(further output omitted)

13. Install the SUNWdoc package. Observe the messages, and verify that the package is installed from the /var/spool/pkg directory.

```
# pkgadd SUNWdoc
```

Processing package instance <SUNWdoc> from

</var/spool/pkg>

(further output omitted)

Exercise: Manipulating Software Packages (Level 3)

14. Remove the SUNWdoc package from the default spool area.

```
# pkgrm -s spool SUNWdoc
```

15. Verify that the SUNWdoc package no longer exists in the spool area and that it is installed on your system.

```
# pkginfo -d spool SUNWdoc
```

```
ERROR: information for "SUNWdoc" was not found
```

```
# pkginfo -l SUNWdoc
```

```
PKGINST: SUNWdoc
```

```
(further output omitted)
```

16. Eject the Solaris 10 OS Software 2 of 4 CD-ROM or Solaris 10 OS Software DVD.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 7

Managing Software Patches on the Solaris 10 OS

Objectives

Upon completion of this module, you should be able to:

- Describe the fundamentals of patch administration
- Install and remove patches

The course map in Figure 7-1 shows how this module fits into the current instructional goal.

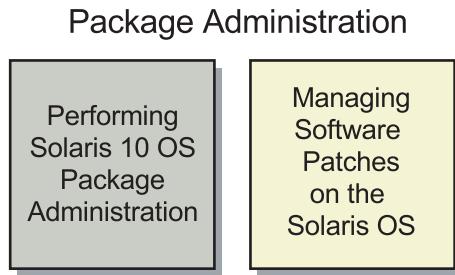


Figure 7-1 Course Map

Preparing for Patch Administration

The administration of patches involves installing or removing Solaris OS patches from a running Solaris OS.

Introducing Solaris OS Patches

A patch contains a collection of files and directories. This collection replaces existing files and directories that prevent proper execution of the software. Some patches contain product enhancements.

The Solaris OS patch types include:

- Standard patches – Patches that fix specific problems with the Solaris OS and other Sun hardware and software products.
- Recommended patches – Solaris OS patches that fix problems that might occur on a large percentage of systems. These include recommended security patches.
- Firmware and PROM patches.
- Patch clusters – A group of standard, recommended, security, or Y2K patches that have been bundled into a single archive for easy downloading and installation.

A patch is distributed as a directory that is identified by a unique number. The number assigned to a patch includes the patch base code first, a hyphen, and a number that represents the patch revision number. For example, a patch directory named 105050-01, indicates that 105050 is the patch number and 01 is the revision number.

The Solaris 10 OS patches are in zip format, for example, 105050-01.zip.

Accessing Patch Documents

Prior to installing patches on your system, you should review the patch documents available through the World Wide Web, patch update CD-ROMs, or anonymous FTP.

To access patch documents through the World Wide Web, go to:

<http://sunsolve.sun.com>

Click Worldwide for a list of alternative sites by geographic areas.

Anonymous FTP access to patch documents is available from sunsolve.sun.com. Use your complete email address as a password. After the connection is complete, the publicly available patch documents are located in the /patchroot/all_unsigned and the /patchroot/all_signed directories.

Table 7-1 shows important summary documents that list all recommended patches for the Solaris OS.

Table 7-1 Patch Documents and Files

Patch Document	Contents
Solaris10.PatchReport	A summary of all patches for the Solaris 10 OS release
10_Recommended README	Instructions for how to install the recommended patch cluster for the Solaris 10 OS, as well as any important notes or warnings, special installation instructions, and usually a note to reboot the system

When you are reviewing patch documentation, start with the Patch Report document first. This report is divided into categories that include information about all patches for a Solaris OS release.



Note – Not all patches available from Sun Microsystems must be installed. Care should be taken to study the README documents for each patch, and then decide on each patch before it is applied to a system.

Checking Patch Levels

Before installing operating system patches, you should know about patches that have been previously installed on a system.

The showrev command and the patchadd command provide useful information about currently installed patches.

```
# showrev -p  
Patch: 106793-01 Obsoletes: Requires: Incompatibles: Packages: SUNWhea  
...  
# patchadd -p  
Patch: 106793-01 Obsoletes: Requires: Incompatibles: Packages: SUNWhea  
...
```



Note – Command output is the same for the patchadd -p and showrev -p commands; however, the patchadd command takes longer to display patch information. The showrev command is a binary, and the patchadd command is a script.

Historical information about all patches that are currently installed on a system and that can be uninstalled using the patchrm command is stored in the /var/sadm/patch directory.

The following command lists the contents of the /var/sadm/patch directory.

```
# ls /var/sadm/patch  
107558-05 107594-04 107630-01 107663-01 107683-01 107696-01  
107817-01 107582-01 107612-06 107640-03
```



Caution – Deleting files from the /var/sadm directory to make more space is a Solution Center call generator. The only way to correct the problems that occur is to restore the deleted files from backup tapes or to reload the Solaris OS.



Note – It is important to ensure that sufficient space has been allocated for the /var file system. There must be sufficient space for the /var/sadm directory to grow as new software packages and patches are installed on the system.

Obtaining Patches

Sun customers who have a maintenance contract have access to the SunSolveSM program's database of patches and patch information, technical white papers, the Symptom and Resolution database, and more. These are available using the World Wide Web.

Sun customers without maintenance contracts have access to a subset of the patches available through the SunSolve program. These patches are available at no charge and include important security and bug fix patches.

To access patches through the World Wide Web, use the following Universal Resource Locators (URLs):

`http://sunsolve.sun.com` – United States
`http://sunsolve.sun.com.au` – Australia
`http://sunsolve.sun.fr` – France
`http://sunsolve.sun.de` – Germany
`http://sunsolve.sun.co.jp` – Japan
`http://sunsolve.sun.se` – Sweden
`http://sunsolve.sun.ch` – Switzerland
`http://sunsolve.sun.co.uk` – United Kingdom

The comprehensive set of patches and patch information is available to contract customers through the button labeled Login. The customer's assigned SunService program password is required to access this database.

To access patches using FTP, use the `ftp` command to connect to:

`sunsolve.sun.com`

The `ftp` utility has many commands; however, only a few are necessary for moving files from system to system. You can locate and copy patches to the local system with a few basic FTP commands.

The following example shows the procedure for changing to the `/var/tmp` directory on the local system, connecting to the remote FTP site, locating a patch and its `README` file in the `/pub/patches` directory, and transferring both files to the local system's directory.



Note – The default mode for an `ftp` connection is binary mode in Solaris 10 OS. The default mode for an `ftp` connection in Solaris 8 or earlier versions is American Standard Code for Information Interchange (ASCII) mode. You use the `bin` command to set the FTP transfer mode to binary mode to transfer binary, image, or a non-text files in these earlier versions of the OS.

```
# cd /var/tmp
# ftp sunsolve.sun.com
Connected to sunsolve.sun.com.
(output omitted)
Name (sunsolve:usera): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:yourpassword
(output omitted)
ftp> bin
200 Type set to I.
ftp> cd /patchroot/reports
ftp> get public_patch_report
(output omitted)
ftp> cd /patchroot/clusters
ftp> get 10_SunAlert_Patch_Cluster.README
(output omitted)
ftp> cd /patchroot/current_unsigned
ftp> mget 112605*
mget 112605-01.zip? y
(output omitted)
mget 112605.readme? y
ftp> bye
```



Note – To disable interactive prompting during multiple (`mget`) file transfers, you can begin a session using `ftp -i sitename` or use the `prompt` command at the `ftp>` prompt.

Preparing Patches for Installation

When patches are downloaded to the local system, you must place the patches in a temporary directory to prepare them for installation. The directory commonly used is the `/var/tmp` directory.

One of the common reasons for patch installation failure is directory permission or ownership problems. The /var/tmp directory is open to all and eliminates any of these types of problems.

The Solaris 7, Solaris 8, Solaris 9, and Solaris 10 OS patches are in zip format, for example, 105050-01.zip.

Use the unzip command to unpack the patch files.

```
# /usr/bin/unzip 105050-01.zip
```

Earlier versions of the Solaris OS used compressed tar files in a tar.Z format, for example, 101010-01.tar.Z.

Use the `zcat` command to uncompress the patch files and the `tar` command to create the patch directories.

```
# /usr/bin/zcat 105050-01.tar.Z | tar xvf -
```

Patch Contents

Figure 7-2 shows the contents of a patch directory after it is extracted from the `.zip` file.

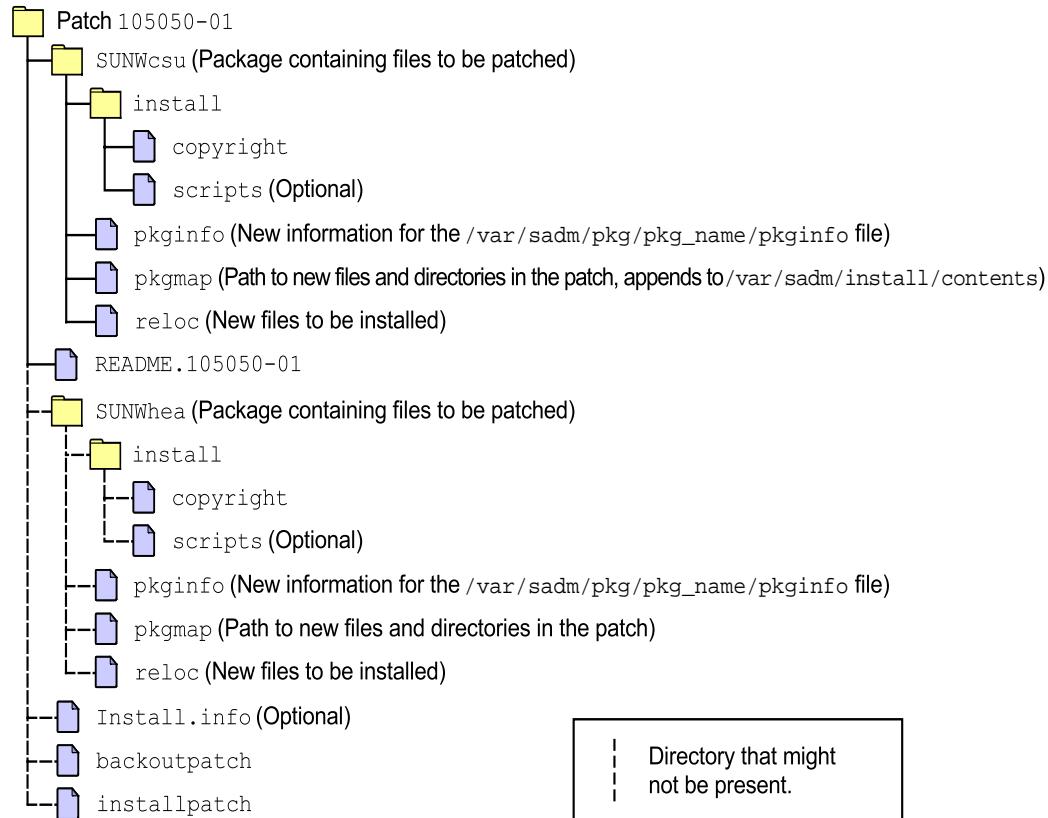


Figure 7-2 An Extracted Patch Directory

Installing and Removing Patches

The two most common commands for managing patches are:

- `patchadd` – Installs uncompressed patches to the Solaris OS
- `patchrm` – Removes patches installed on the Solaris OS

Additionally, you install cluster patches by using the `install_cluster` command. You can also manage patches through the Solaris Management Console.

Installing a Patch

When a patch is installed, the `patchadd` command calls the `pkgadd` command to install the patch packages.

The following example shows the procedure for patch installation. This example assumes that the patch to be installed exists in the `/var/tmp` directory and has been unzipped or uncompressed for installation.

```
# cd /var/tmp
# patchadd 105050-01
Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)
Installing patch packages...
Patch number 105050-01 has been successfully installed.
See /var/sadm/patch/105050-01/log for details.
Patch packages installed:
  SUNWhea
```

Figure 7-3 shows those components of the `/var/sadm` directory that are updated during the installation of patch 105050-01.

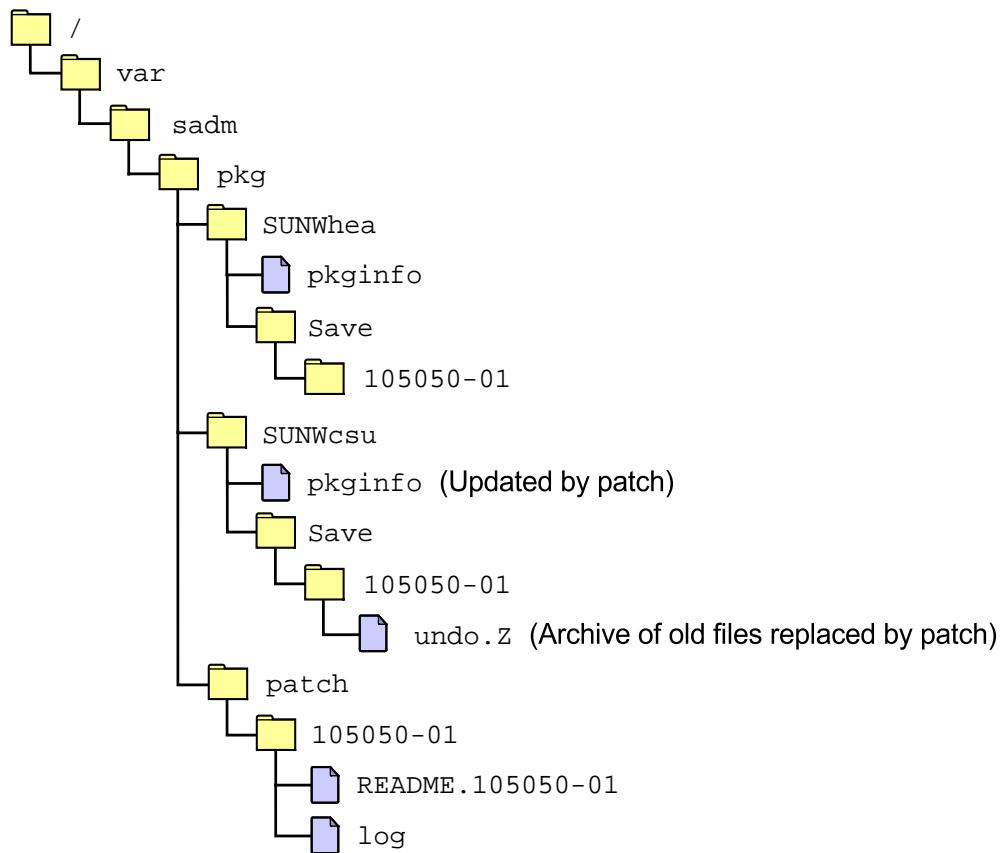


Figure 7-3 Updated `/var/sadm` Directories



Caution – Patches can be added using the `-d` (nosave) option to save space. When this option is used, `patchadd` does not save copies of the files being updated or replaced, and therefore cannot be backed out.

Removing a Patch

When you remove a patch, the `patchrm` command restores all files that were modified or replaced by that patch, unless:

- The patch was installed with the `patchadd -d` option (which instructs the `patchadd` command not to save copies of files being updated or replaced)
- The patch is required by another patch
- The patch has been obsoleted by a later patch

The `patchrm` command calls the `pkgadd` utility to restore packages that were saved during the initial patch installation.

The following example shows how to remove a patch by using the `patchrm` command.

```
# patchrm 105050-01
Checking installed packages and patches...
Backing out patch 105050-01...
Patch 105050-01 has been backed out.
#
```

Installing Patch Clusters

The patch cluster provides a selected set of patches for a designated Solaris OS level and is conveniently wrapped for one-step installation. Patch clusters are usually a set of recommended and security patches.

You should not install cluster patches on systems with limited disk space.

By default, the cluster installation procedure saves the base objects being patched. Prior to installing the patches, the cluster installation script first determines if enough system disk space is available in the `/var/sadm/pkg` directory to save the base packages and terminates if not enough space is available.



Caution – You can override the save feature by using the `-nosave` option when you are executing the cluster installation script. If you use the `-nosave` option, you cannot back out these patches if the need arises.

You can remove individual patches that were installed by the patch cluster by using the `patchrm` command. The `README` file is located in the specific patch directory under the `/var/sadm/patch` directory after the patch has been installed.

To install a patch cluster, perform the following steps:

1. Be sure the patch cluster has been unzipped or uncompressed and extracted if the cluster was received as a `.tar.Z` file.
2. Decide on which method to use to install the cluster—the recommended default `save` option or the `-nosave` option.
3. Change to the directory that contains the patch cluster. Read the `CLUSTER_README` file, which contains information about the bundled set of patches, including:
 - Cluster description
 - Patches included
 - Important notes and warnings
 - Save and backout options
 - Special install instructions
 - Special patch circumstances
 - Any notices and other recommendations

Then run the `install_cluster` script.

```
# cd 10_Recommended  
# ./install_cluster
```

The installation appears as follows:

```
Patch cluster install script for Solaris 10 Recommended
```

```
*WARNING* SYSTEMS WITH LIMITED DISK SPACE SHOULD *NOT* INSTALL PATCHES:
```

(Other disk space warning messages omitted.)

```
Are you ready to continue with install? [y/n]:y  
Determining if sufficient save space exists...  
Sufficient save space exists, continuing...  
Installing patches located in /tmp/10_Recommended  
Using patch_order file for patch installation sequence  
Installing 113319-01...
```

(Other patch messages omitted.)

The following patches were not able to be installed:

```
112875-01  
113023-01
```

For more installation messages refer to the installation logfile:
`/var/sadm/install_data/Solaris_10_Recommended_log`

Use '/usr/bin/showrev -p' to verify installed patch-ids.
Refer to individual patch README files for more patch detail.
Rebooting the system is usually necessary after installation
#

4. Read each individual patch README file to determine if any additional steps are required to fully install any individual patch.
5. Check the log file if more detail is needed.

Reviewing the log provides information about why the patches listed above were not able to be installed:

```
# more /var/sadm/install_data/Solaris_10_Recommended_log
*** Install Solaris 10 Recommended begins Mon Oct 18 14:47:11 BST 2004 ***
*** PATCHDIR = /tmp/10_Recommended ***
(output omitted)
Installing 112875-01...
```

Checking installed patches...
Patch 112875-01 has already been applied.
See patchadd(1M) for instructions.

Installing 113023-01...

Checking installed patches...
One or more patch packages included in
113023-01 are not installed on this system.
(output omitted)
#

6. Reboot the system for all patches to take effect.

The **smpatch** Utility

The **smpatch** utility program allows you to download, apply, and remove patches on a single system or on multiple systems.

The system on which you run Sun Patch Manager must be running at least Solaris 8 OS and have the Developer Software Support Group installed. If your system runs Solaris 8 OS or Solaris 9 OS, it must also have the Sun Patch Manager 2.0 software installed. If your system runs Solaris 10 OS and has the Developer Software Support Group installed, the Sun Patch Manager 2.0 software is included.

The **smpatch** command can also be used to download the required patches for your systems from the Sun patch server URL at:
<https://updateserver.sun.com/solaris/>. The default location for downloaded patches is the /var/sadm/spool directory.

The values used by the **smpatch** command can be displayed using the following command:

```
# smpatch get -L patchpro.patch.source patchpro.download.directory  
https://updateserver.sun.com/solaris/  
/var/sadm/spool
```

All **smpatch** commands must be issued on the command line. To obtain patches from the Sun patch server, your system must be configured to access the Internet.

The **smpatch** command can analyze the patch requirements for a system and automatically patch that system with all appropriate patches.

For further details, refer to **man smpatch**.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Maintaining Patches (Level 1)

In this exercise, you transfer a patch from a classroom server, apply the patch, and then remove it.

Preparation

Your instructor should provide directions for accessing a patch on a server that is available to systems in the classroom. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Create a directory to hold patches. Use the `ftp` command to transfer a patch from a classroom server into the directory you create. Unzip the patch. Verify that no patch has been applied to your system. Verify that the `/var/sadm/patch` directory is empty.
- Read the `README` file associated with the patch to verify which Solaris OS release is appropriate for the patch. Add the patch, and verify that it is installed in the `/var/sadm/patch` directory. View the log file for this patch.
- Remove the patch you just installed, and verify that it is no longer applied to the system.

Exercise: Maintaining Patches (Level 2)

In this exercise, you transfer a patch from a classroom server, apply the patch, and then remove it.

Preparation

Your instructor should provide directions for accessing a patch on a server that is available to systems in the classroom. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Create a directory to hold patches. Use the `ftp` command to transfer a patch from a classroom server into the directory you create. Unzip the patch. Verify that no patch has been applied to your system. Verify that the `/var/sadm/patch` directory is empty.
- Read the `README` file associated with the patch to verify which Solaris OS release is appropriate for the patch. Add the patch, and verify that it is installed in the `/var/sadm/patch` directory. View the log file for this patch.
- Remove the patch you just installed, and verify that it is no longer applied to the system.

Tasks

Complete the following steps:

1. Create a directory to hold patches. Use the binary transfer mode of the `ftp` command to transfer a patch from a classroom server into the directory you created. Your instructor should provide information about where to find a patch on the server. Close your `ftp` connection when you are finished.



Note – The default mode for an `ftp` connection is binary mode in Solaris 10 OS. The default mode for an `ftp` connection in Solaris 8 or earlier versions is ASCII mode. You use the `bin` command to set the FTP transfer mode to binary mode to transfer binary, image, or non-text files in these earlier versions of the OS.

2. Use the `unzip` command to extract the patch from the zip archive.
3. Use the `patchadd` command to determine if any patches are currently installed on your system.
4. Verify that the `/var/sadm/patch` directory is empty.
5. Read the `README` file that is associated with the patch you unzipped. Verify the Solaris OS release for which the patch is required.
Solaris OS release:
 6. Add the patch.
 7. Verify that the patch is installed. What are the packages that the patch affects?
 8. Examine the patch installation log.
 9. Remove the patch you just installed. Verify that the patch is no longer installed.

Exercise: Maintaining Patches (Level 3)

In this exercise, you transfer a patch from a classroom server, apply the patch, and then remove it.

Preparation

Your instructor should provide directions for accessing a patch on a server that is available to systems in the classroom. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Create a directory to hold patches. Use the `ftp` command to transfer a patch from a classroom server into the directory you create. Unzip the patch. Verify that no patch has been applied to your system. Verify that the `/var/sadm/patch` directory is empty.
- Read the `README` file associated with the patch to verify which Solaris OS release is appropriate for the patch. Add the patch, and verify that it is installed in the `/var/sadm/patch` directory. View the log file for this patch.
- Remove the patch you just installed, and verify that it is no longer applied to the system.

Tasks and Solutions

Complete the following steps:



Note – The default mode for an `ftp` connection is binary mode in Solaris 10 OS. The default mode for an `ftp` connection in Solaris 8 OS or earlier versions is ASCII mode. You use the `bin` command to set the FTP transfer mode to binary mode to transfer binary, image, or non-text files in these earlier versions of the OS.

1. Create a directory to hold patches. Use the binary transfer mode of the `ftp` command to transfer a patch from a classroom server into the directory you created. Your instructor should provide information about where to find a patch on the server. Close your `ftp` connection when you are finished. For example:

```
# cd /var/tmp  
# ftp instructor  
(connection and login messages)  
ftp> cd /export/patches  
ftp> get 112875-01.zip  
(ftp messages)  
ftp> bye  
221 Goodbye.  
#
```

2. Use the `unzip` command to extract the patch from the zip archive, for example:

```
# unzip 112875-01.zip  
Archive: 112875-01.zip  
  creating: 112875-01/  
  inflating: 112875-01/.diPatch  
  inflating: 112875-01/patchinfo  
  creating: 112875-01/SUNWrcmds/  
  inflating: 112875-01/SUNWrcmds/pkgmap  
  inflating: 112875-01/SUNWrcmds/pkginfo  
  creating: 112875-01/SUNWrcmds/install/  
  inflating: 112875-01/SUNWrcmds/install/checkinstall  
  inflating: 112875-01/SUNWrcmds/install/copyright  
  inflating: 112875-01/SUNWrcmds/install/i.none  
  inflating: 112875-01/SUNWrcmds/install/patch_checkinstall  
  inflating: 112875-01/SUNWrcmds/install/patch_postinstall  
  inflating: 112875-01/SUNWrcmds/install/postinstall  
  inflating: 112875-01/SUNWrcmds/install/preinstall  
  creating: 112875-01/SUNWrcmds/reloc/
```

```

creating: 112875-01/SUNWrcmds/reloc/usr/
creating: 112875-01/SUNWrcmds/reloc/usr/lib/
creating: 112875-01/SUNWrcmds/reloc/usr/lib/netsvc/
creating: 112875-01/SUNWrcmds/reloc/usr/lib/netsvc/rwall/
inflating: 112875-01/SUNWrcmds/reloc/usr/lib/netsvc/rwall/rpc.rwalld
inflating: 112875-01/README.112875-01
#

```

3. Use the patchadd command to determine if any patches are currently installed on your system.

```
# patchadd -p
```

The patchadd command should display a message.

4. Verify that the /var/sadm/patch directory is empty.

```
# ls /var/sadm/patch
#
```

5. Read the README file that is associated with the patch you unzipped. Verify the Solaris OS release for which the patch is required.

```
# more 112875-01/README.112875-01
```

Patch-ID# 112875-01

Keywords: security rpc.rwalld string

Synopsis: SunOS 5.10: patch /usr/lib/netsvc/rwall/rpc.rwalld

Date: Jun/21/2004

(output omitted)

6. Add the patch.

```
# patchadd 112875-01
```

Checking installed patches...

Verifying sufficient filesystem capacity (dry run method)...

Installing patch packages...

Patch number 112875-01 has been successfully installed.

See /var/sadm/patch/112875-01/log for details

Patch packages installed:

SUNWrcmds

```
#
```

7. Verify that the patch is installed. What are the packages that the patch affects?

```
# patchadd -p
```

Patch: 112875-01 Obsoletes: Requires: Incompatibles: Packages: SUNWrcmds

Exercise: Maintaining Patches (Level 3)

8. Examine the patch installation log file.

```
# cd /var/sadm/patch/112875-01  
# more log  
(output omitted)  
Installation of <SUNWrcmds> was successful.
```

9. Remove the patch you just installed. Verify that the patch is no longer installed.

```
# cd  
# patchrm 112875-01  
Checking installed patches...  
Backing out patch 112875-01...  
Patch 112875-01 has been backed out.  
# patchadd -p
```

The patchadd -p command should not contain any reference to 112875-01.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 8

Executing Boot PROM Commands

Objectives

Upon completion of this module, you should be able to:

- Identify boot programmable read-only memory (PROM) fundamentals
- Use basic boot PROM commands
- Identify the system's boot device
- Create and remove custom device aliases
- View and change nonvolatile random access memory (NVRAM) parameters from the shell
- Interrupt an unresponsive system

The course map in Figure 8-1 shows how this module fits into the current instructional goal.

Performing System Boot Procedures

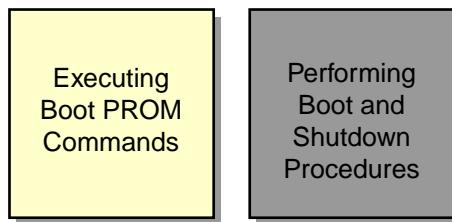


Figure 8-1 Course Map

Introducing Boot PROM Fundamentals

All Sun systems have resident boot PROM firmware that provides basic hardware testing and initialization prior to booting. The boot PROM also enables you to boot from a wide range of devices. In addition, there is a user interface that provides several important functions.

The Sun boot PROM has access to a standard set of generic device drivers. The system needs these drivers to access and control the buses and the boot device to boot the system properly.

All versions of the OpenBoot™ architecture allow a third-party board to identify itself and load its own plug-in device driver. Each device identifies its type and furnishes its plug-in device driver when requested by the OpenBoot PROM during the system hardware configuration phase of the boot process.

Figure 8-2 shows the identification process.

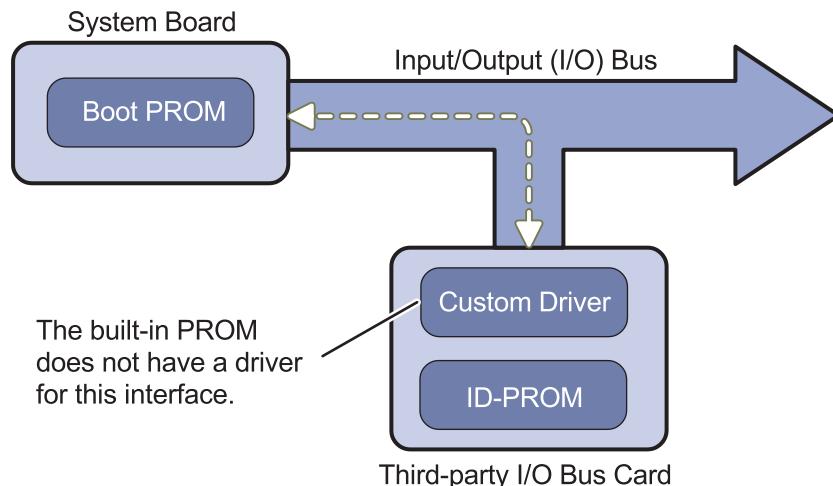


Figure 8-2 Third-Party Device Identification Process

Goal of the OpenBoot™ Architecture Standard

The overall goal of the Institute of Electrical and Electronics Engineers (IEEE) standard #1275 for the OpenBoot architecture is to provide the capabilities to do the following:

- Test and initialize system hardware
- Determine the system's hardware configuration
- Boot the operating system
- Provide an interactive interface for configuration, testing, and debugging
- Enable the use of third-party devices

Boot PROM

Each Sun SPARC system has a boot PROM chip. This 1-Mbyte chip is typically located on the same board as the central processing unit (CPU). Boot PROM chips are usually found in a pluggable socket on older systems. As of the 3.x PROM, they are permanently soldered to the main system board.

The Ultra™ workstations use a reprogrammable boot PROM called a flash PROM (FPROM). The FPROM allows you to load new boot program data into the PROM by using software, instead of having to replace the chip.

Desktop systems have a write-protect jumper that must be moved before you can write to the PROM. You have to move the jumper because the default position is write-protect. Refer to the *Sun Flash PROM Guide for Workstations and Workgroup Servers - Standalone Version* part number 802-3233-27, for the jumper location on your system.



Caution – Many systems have the jumper under an installed frame buffer or other removable card. Be careful when removing or replacing this card.

The main functions of the boot PROM are to test the system hardware and to boot the operating system. The boot PROM firmware is referred to as the *monitor* program.

The boot PROM firmware controls the operation of the system before the operating system has been booted and the kernel is available. The boot PROM also provides the user with a user interface and firmware utility commands, known as the FORTH command set. Commands include the boot commands, diagnostics commands, and commands to modify the default configuration.



Note – The boot PROM does not work with the Solaris OS file systems or files. It handles mainly hardware devices. The OS works with and is dependent on firmware, but firmware is independent of the OS.

To determine which revision of OpenBoot PROM is running on the system, perform either the command:

```
# /usr/platform/'uname -m'/sbin/prtdiag -v
```

or

```
# prtconf -v
```

System Configuration Information

Another important element in each Sun system is the system configuration information. The system configuration information includes the following:

- The Ethernet or MAC address, such as 8:0:20:5d:6f:9e
- The system host ID, such as 805d6f9e
- User-configurable parameters which have been modified from the default settings

The user-configurable parameters are known as NVRAM variables, or EEPROM parameters. They allow an administrator to control things such as the default boot device, the level of Power-on self-test (POST), and so on.

Depending on the system, one of three different components store the system configuration information:

- NVRAM chip
- Serial Electronically Erasable Programmable Read Only Memory (SEEPROM) chip
- System Configuration Card (SCC)

NVRAM Chip

Older systems contain a removable NVRAM chip, normally located on the main system board. In addition to the system configuration information, the NVRAM chip contains an integrated lithium battery which provides battery backup for the configuration information and also provides the system's time-of-day (TOD) function.

SEEPROM Chip

Most newer systems contain a non-removable SEEPROM chip, normally located on the main system board. SEEPROM chips do not require a battery to maintain the system configuration information.

System Configuration Card

Some newer systems contain a removable System Configuration Card which holds the system configuration information. It is inserted into the System Configuration Card Reader.

Figure 8-3 shows the basic elements of the Boot PROM and NVRAM

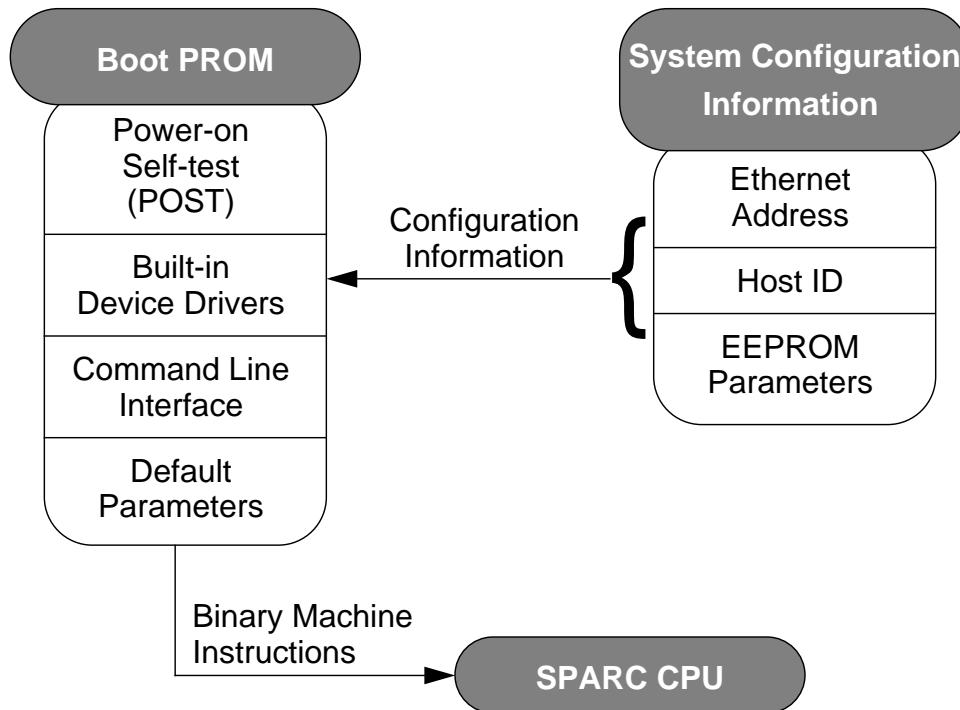


Figure 8-3 Basic Elements of the Boot PROM and NVRAM POST

When a system's power is turned on, a low-level POST is initiated. This low-level POST code is stored in the boot PROM and is designed to test the most basic functions of the system hardware.

At the successful completion of the low-level POST phase, the boot PROM firmware takes control and performs the following initialization sequence:

- Probes the memory and then the CPU
- Probes bus devices, interprets their drivers, and builds a device tree
- Installs the console

After the boot PROM initializes the system, the banner displays on the console. The system checks parameters stored in the boot PROM and NVRAM to determine if and how to boot the operating system.

One of the first tests that POST runs is to check to determine if a keyboard is connected to the system and if a Stop-key option is present.

The Stop-key is located on the left side of the keyboard. To enable various diagnostic modes, hold down the Stop-key simultaneously with another key. The Stop-key sequences have an effect on the OpenBoot PROM and define how POST runs when a system's power is turned on. The following is a list of the Stop-key sequences:

- Stop-D key sequence – Hold down the Stop and D keys simultaneously while system power is turned on, and the firmware automatically switches to diagnostic mode. This mode runs more extensive POST diagnostics on the system hardware. The OpenBoot PROM variable diag-switch? is set to true.

See Figure 8-4 to show the effect of the variable diag-switch?.

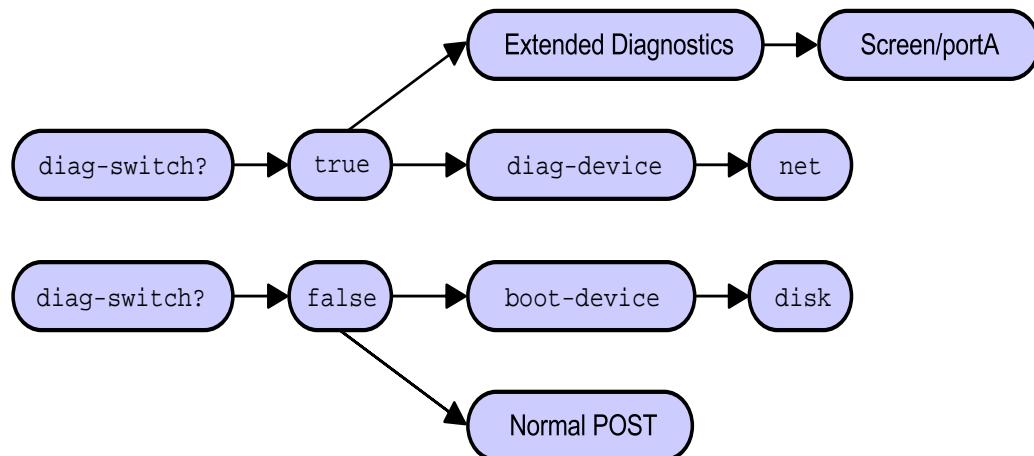


Figure 8-4 Post Diagnostics

Note – The Stop-D key sequence is not available on a serial port terminal.



- Stop-N key sequence – Hold down the Stop and N keys simultaneously while the system power is turned on to set the NVRAM parameters to the default values. When you see the light emitting diodes (LEDs) on the keyboard begin to flash, you can release the keys, and the system should continue to boot.

Incorrect NVRAM settings can cause system boot failure. For example, during a flash PROM download, if a power failure occurs, some of the contents of the NVRAM can become unusable.

If the system does not boot and you suspect that the NVRAM parameters are set incorrectly, the parameters can easily be changed to the default values.



Caution – Where possible, capture non-default NVRAM values before using the Stop-N key sequence.

Describing Abort Sequences

As a system administrator, you might want to abort a running system with a key sequence.

- Stop-A key sequence – Hold down the Stop and A keys simultaneously to interrupt any program that is running at the time these keys are pressed and to put the system into the command entry mode for the OpenBoot PROM. The system presents an ok prompt for the user, which signifies it is ready to accept OpenBoot PROM commands.



Caution – The Stop-A key sequence, as a method for getting to the ok prompt, is not recommended unless there is absolutely no alternative. The Stop-A key sequence can cause Solaris OS file system corruption which can be difficult to repair.

Disabling the Abort Sequence

As a system administrator, you might want to disable the abort key sequence on a system to prevent possible corruption of a file system or to provide tighter security.

To disable the abort key sequence, edit the `/etc/default/kbd` file. Inside the file, the statement `KEYBOARD_ABORT=disable` is commented out. Remove the comment from in front of the value, save the file, and execute the command `kbd -i`. When you have completed these steps, the system allows Stop-A key sequence only during the boot process.

You can also configure the system to change the keyboard abort sequence to an alternate keystroke. Review the man page for the `kbd` command for more information.

Displaying POST to the Serial Port

As the system administrator, you can attach a terminal to the serial port of a system to capture a far greater amount of information from the POST output.

When the power is turned on, POST looks for a keyboard. If there is no keyboard present, POST diverts system output to serial port A.

POST runs more extensive tests when the system is in diagnostic mode with the PROM parameter diag-switch? set to true.

Be sure to attach the correct type of null modem cable for your system type to serial port A.

Some systems require a serial port cable, as shown in Figure 8-5.

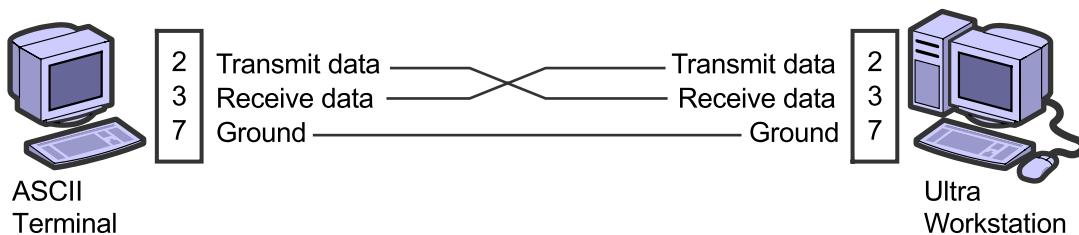


Figure 8-5 Serial Port Connection to a Sun Workstation

The following example is the POST output from a Sun Blade™ 1000:

```
@(#) 4.0 Version 29 created 2000/07/12 16:46
Clearing TLBs Done
Reset: 0000.0000.0000.0010 SPOR
Loading Configuration
Membase: 0000.0000.0000.0000
MemSize: 0000.0000.2000.0000
Init CPU arrays Done
Init E$ tags Done
Setup TLB Done
MMUs ON
Block Scrubbing Done
Copy Done
PC = 0000.07ff.f000.3138
Decompressing Done
Size = 0000.0000.0006.e3b0
ttya initialized
Start Reason: Soft Reset
System Reset: (SPOR)
```

Introducing Boot PROM Fundamentals

```
Probing gptwo at 0,0 SUNW,UltraSPARC-III (750 MHz @ 5:1, 8 MB)
    memory-controller
Probing gptwo at 1,0 Nothing there
Probing gptwo at 8,0 pci pci upa ppm
Loading Support Packages: kbd-translator
Loading onboard drivers: ebus flashprom bbc ppm i2c dimm-fru dimm-fru
    dimm-fru dimm-fru dimm-fru dimm-fru dimm-fru nvram idprom
    i2c cpu-fru temperature fan-control card-reader motherboard-fru
Memory Configuration:
Segment @ Base:      0  Size: 512 MB (2-Way)
Probing /upa@8,480000 Device 0,0 Nothing there
Probing /upa@8,480000 Device 1,0 Nothing there
Probing /pci@8,600000 Device 4  SUNW,qlc fp disk
Probing /pci@8,600000 Device 1  Nothing there
Probing /pci@8,700000 Device 5  network firewire usb
dev-descrip
next-add
node made
Probing /pci@8,700000 Device 6  scsi disk tape scsi disk tape
Probing /pci@8,700000 Device 1  Nothing there
Probing /pci@8,700000 Device 2  Nothing there

(UltraSPARC-III) , Keyboard Present
OpenBoot 4.0, 512 MB memory installed, Serial #12134217.
Ethernet address 8:0:20:b9:27:49, Host ID: 80b92749.
```

Using Basic Boot PROM Commands

The boot PROM monitor provides a user interface for invoking OpenBoot commands.



Note – The ok prompt indicates that the Solaris OS is currently not running.

Table 8-1 shows some of the commands typically entered at the ok prompt.

Table 8-1 Typical Commands Used at the ok Prompt

Command	Description
banner	Displays the power-on banner
boot	Boots the system
help	Lists the main help categories
printenv	Displays all parameters' current and default values
setenv	Sets the specified NVRAM parameter to some value
reset-all	Resets the entire system; similar to a power cycle
set-defaults	Resets all parameter values to the factory defaults
sifting <i>text</i>	Displays the FORTH commands containing <i>text</i>
.registers	Displays the contents of the registers
probe-scsi	Identifies the devices on the internal Small Computer System Interface (SCSI) bus
probe-scsi-all	Identifies the devices on all SCSI buses
probe-ide	Identifies devices on the internal integrated device electronics (IDE) bus
probe-fcal-all	Identifies devices on all Fibre Channel loops
show-devs	Displays the entire device tree
dealias	Identifies the current boot device alias for the system

Table 8-1 Typical Commands Used at the **ok** Prompt (Continued)

<code>nvalias</code>	Creates a new device alias name
<code>nvunalias</code>	Removes a device alias name
<code>show-disks</code>	Displays and allows a selection of device paths for the disks to be used for <code>nvalias</code>
<code>sync</code>	Manually attempts to flush memory and synchronize file systems
<code>test</code>	Runs self-tests on specified devices

Identifying the System Boot PROM Version

The `banner` command lists useful information about the system, such as the model name, the boot PROM version number (for example, 1.x, 2.x, 3.x, 4.x, or 5.x), the amount of memory, the Ethernet address, and the host ID.

The following example shows output from the `banner` command.

```
ok banner
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 360MHz), Keyboard Present
OpenBoot 3.31, 128 MB (50 ns) memory installed, Serial #11888271.
Ethernet address 8:0:20:b5:66:8f, Host ID: 80b5668f.
```

Booting the System

Use the `boot` command to boot the Solaris OS from the `ok` prompt. This command has several options available for booting the system in different situations.

The format for the `boot` command is:

```
boot device_name -options
```

Enter the `boot` command at the `ok` prompt to boot the system to multiuser mode automatically.

```
ok boot
```

The following list describes some of the options for the boot command:

- **-s** – Boots the system to a single-user mode and asks the user for the root password.

ok **boot -s**

- **cdrom -s** – Boots the system to single user mode from a CD-ROM or a DVD.

ok **boot cdrom -s**

- **-a** – Boots the system interactively. Use this option if an alternative file needs to be executed during boot. The boot program asks for the following information.

ok **boot -a**

Enter filename [kernel/sparcv9/unix]:

Enter default directory for modules [/platform/SUNW,UltraAX-i2/kernel /platform/sun4u/kernel /kernel /usr/kernel]:

Name of system file [etc/system]:

SunOS Release 5.10 Version s10 64-bit

Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.

Use is subject to license terms.

root filesystem type [ufs]:

Enter physical name of root device

[/pci@1f,0/pci@1/scsi@8/disk@0,0:a]:

- **-r** – Performs a reconfiguration boot. Use this option to find a newly attached device and to create new device entries in the /devices and /dev directories. It also updates the /etc/path_to_inst file.

ok **boot -r**

- **-v** – Boots the system while displaying more detailed device information to the console. Use this option to troubleshoot problems during the boot process. You can use this option with other options.

ok **boot -v**

ok **boot -rv**

ok **boot -sv**

Accessing More Detailed Information

You use the **help** command to obtain help on the main categories in the OpenBoot firmware.

The following is an example of the **help** output from an Ultra 5 workstation that is running OpenBoot PROM version 3.31:

```
ok help
Enter 'help command-name' or 'help category-name' for more help
(Use ONLY the first word of a category description)
Examples: help system -or- help nvramrc
Categories:
boot (Load and execute a program)
nvramrc (Store user defined commands)
system configuration variables (NVRAM variables)
command line editing
editor (nvramrc editor)
resume execution
devaliases (Device aliases)
diag (Diagnostics commands)
ioredirect (I/O redirection commands)
misc (Miscellaneous commands)
ok
```

The **help** command listing provides a number of other keywords that you can use to view further details.

For example, to view specific information for one of the main categories listed in the preceding example, perform one of the following commands:

```
ok help boot

ok help nvramrc

ok help diag

ok help misc
```

Listing NVRAM Parameters

You use the `printenv` command to list all the NVRAM parameters. If the parameter can be modified, the `printenv` command displays its default setting and current setting.

The following example shows output from the `printenv` command.

```
ok printenv
Variable Name          Value           Default Value
tpe-link-test?         true            true
scsi-initiator-id     7                7
keyboard-click?       false           false
keymap
ttyb-rts-dtr-off      false           false
ttyb-ignore-cd         true            true
ttya-rts-dtr-off      false           false
ttya-ignore-cd         true            true
ttyb-mode              9600,8,n,1,-   9600,8,n,1,-
ttya-mode              9600,8,n,1,-   9600,8,n,1,-
pcia-probe-list        1,2,3,4        1,2,3,4
pcib-probe-list        1,2,3          1,2,3
mfg-mode               off             off
diag-level             max             max
#power-cycles          273
output-device          screen          screen
input-device            keyboard         keyboard
boot-command            boot            boot
auto-boot?              true            true
diag-device             net             net
boot-device             disk net        disk net
local-mac-address?     false           false
screen-#columns         80              80
screen-#rows             34              34
use-nvramrc?           false           false
nvramrc                dealias pgx24 /pcilf,0 ...
security-mode           none
security-password
security-#badlogins     0
diag-switch?            false           false
ok
```

You can also use the **printenv** command to display a single parameter and its values.

For example, to display only the **boot-device** parameter, perform the command:

```
ok printenv boot-device
boot-device = disk net
```

The possible values of the **boot-device** parameter include **disk**, **net**, and **cdrom**.

Note – Some OpenBoot PROM parameters, such as **auto-boot?**, end in a question mark. If an OpenBoot PROM parameter ends in a question mark, the parameter value is typically either **true** or **false**.



Changing NVRAM Parameters

You use the **setenv** command to change the current values assigned to NVRAM parameters.

If the **auto-boot?** parameter is set to **true**, the system boots automatically. If it is set to **false**, the system stops at the **ok** prompt.

The following example changes the **auto-boot?** parameter from its default setting of **true** to the value of **false**.

```
ok printenv auto-boot?
auto-boot? = true
ok
ok setenv auto-boot? false
auto-boot? = false
```

The **reset-all** command halts the system, clears all buffers and registers, and performs a software simulated power-off/power-on of the system.

```
ok reset-all
Resetting ...
```

Note – The **reset-all** command, combined with the **auto-boot? = false** setting clears system registers, which is required on a system with a PROM 3.x or higher before you can use the **probe** command or perform other tests.



Restoring Default NVRAM Parameters

You use the `set-defaults` command to reset all NVRAM parameters to their default values. It affects only parameters that have assigned default values.

```
ok set-defaults
Setting NVRAM parameters to default values.
ok
```

To reset a specific parameter to its default value, use the `set-default` command followed by the parameter name.

```
ok set-default parameter-name
```

For example, to reset the `diag-level` parameter, perform the command:

```
ok set-default diag-level
```

Displaying Devices Connected to the Bus

To identify the peripheral devices currently connected to the system, such as disks, tape drives, or CD-ROMs, use the `probe` command.

To identify the various probe commands that are available with your system, use the `sifting` command. The `sifting` command is useful for finding OpenBoot PROM commands when you do not know the exact command syntax.

For example, to find the probe commands available, perform the command:

```
ok sifting probe

(f006c954) probe-all          (f006c5a0) probe-all    (f006c378) probe-ide
(f006c1e8) probe-pci-slot     (f006bc8c) probe-scsi
(f006bd78) probe-scsi-all     (f0060fe8) probe-pci
(output truncated)
```

The most common probe commands are the `probe-scsi` command, the `probe-scsi-all` command, and the `probe-ide` command.



Caution – The following warning message might be displayed if you invoke the probe commands on Sun systems that contain a boot PROM that is version 3.x and above.

Using Basic Boot PROM Commands

This command may hang the system if a Stop-A or halt command has been executed. Please type reset-all to reset the system before executing this command.

Do you wish to continue? (y/n) **n**

If any portion of the Solaris OS has loaded into memory when the system has been aborted, the probe commands can hang the system. To avoid having your system hang, perform the commands:

```
ok setenv auto-boot? false  
ok reset-all
```

One method you can use to tell if the system might hang during a probe command is to use the .registers command.

```
ok .registers  
      Normal      Alternate      MMU      Vector  
0:          0            0            0            0  
1:          0            0            0            0  
2:          0            0            0            0  
3:          0            0            0            0  
4:          0            0            0            0  
(output edited for brevity)  
%PC  0  %nPC  0  
%TBA 0  %CCR  0  XCC:nzvc    ICC:nzvc
```

The preceding output shows that the registers are all empty, with values of 0 (zero). Should the registers hold values other than 0, the probe command would most likely hang the system.

The probe-scsi Command

The probe-scsi command identifies the peripheral devices attached to the on-board SCSI controller. The probe-scsi command identifies such peripheral devices as disks, tape drives, or CD-ROMs by their target addresses.

```
ok probe-scsi  
Target 1  
Unit 0 Disk      FUJITSU MAB3045S SUN4.2G17059825M62990  
Target 3  
Unit 0 Disk      IBM      DDRS34560SUN4.2GS98E99255C5917  
      (C) Copyright IBM Corp.  
      1997. All rights reserved.  
Target 6  
Unit 0 Removable Read Only device SONY CDROM
```

The **probe-scsi-all** Command

The **probe-scsi-all** command identifies the peripheral devices that are attached to the on-board SCSI controller and all peripheral devices attached to separate SBus or PCI SCSI controllers.

```
ok probe-scsi-all
/pci@1f,0/pci@1/pci@1/SUNW,isptwo@4
Target 3
Unit 0    Disk FUJITSU MAB3045S SUN4.2G1907
Target 4
Unit 0    Removable Tape EXABYTE EXB-8505SMBANSH20090
```

The **probe-ide** Command

The **probe-ide** command identifies disks and CD-ROMs that are attached to the on-board IDE controller. This command displays the device number of the internal device.

```
ok probe-ide
Device 0          ( Primary Master )
                  ATA Model : ST 38420A (DISK)
Device 1          ( Primary Slave )
                  Not Present
Device 2          ( Secondary Master )
                  Removable ATAPI Model : CRD-8322B (CD-ROM)
Device 3          ( Secondary Slave )
                  Not Present
```

Identifying the System's Boot Device

Sun hardware uses the concept of a device tree to organize devices that are attached to the system.

Figure 8-6 shows the organizational structure of a device tree for an Ultra 5 or an Ultra 10 workstation.

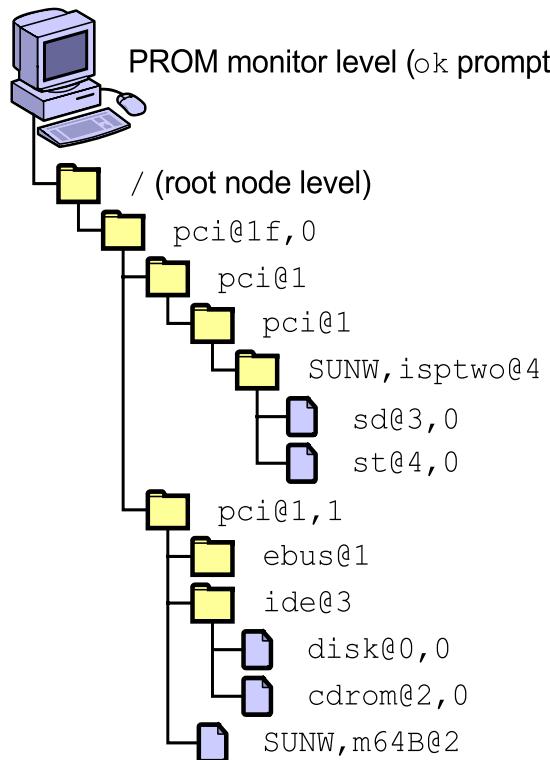


Figure 8-6 Partial Device Tree for an Ultra 5 or Ultra 10 Workstation

Note – In Figure 8-6, some license has been taken in naming these directories to simplify the illustration.



The OpenBoot firmware builds the device tree from information gathered at the POST. This device tree is loaded into memory and is used by the kernel during the boot process to identify all configured devices. A boot -r synchronizes the Solaris 10 OS device tree with the OBP device tree.

The top of the device tree is the root device node. Following the root device node is a bus nexus node. Connected to a bus nexus node is a leaf node, typically a controller for an attached device.

In Figure 8-6, the `disk@0,0` device is the IDE device for the hard disk drive, and the `cdrom@2,0` device is the IDE device for the CD-ROM drive. Both are attached to the IDE controller `ide@3`. Similarly, the `sd@3,0` device is the SCSI disk device and the `st@4,0` device is the SCSI tape device. Both are attached to the PCI-based SCSI controller `SUNW,isptwo@4`.

The paths built in the device tree by the OpenBoot firmware vary depending on the type of system and its device configuration.

Figure 8-7 shows a sample disk device path on an Ultra workstation with a PCI bus.

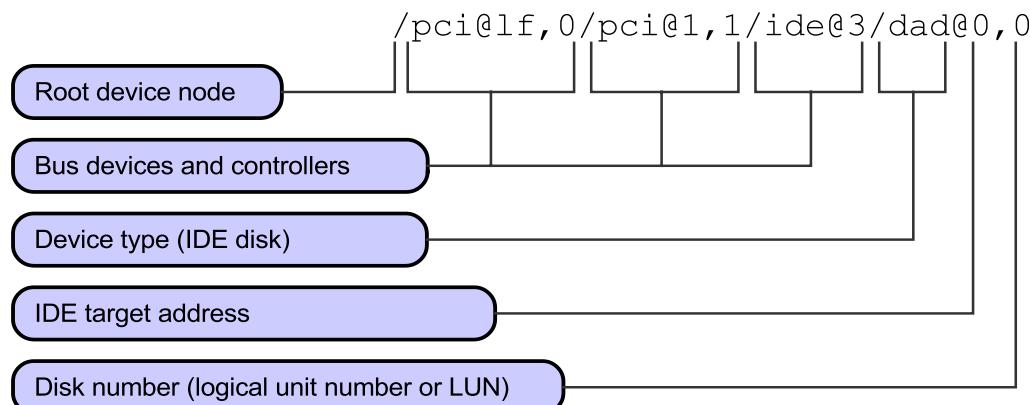


Figure 8-7 Disk Device Path – Ultra Workstation With a PCI IDE Bus

Figure 8-8 shows a sample disk device path on an Ultra workstation with a PCI-SCSI bus.

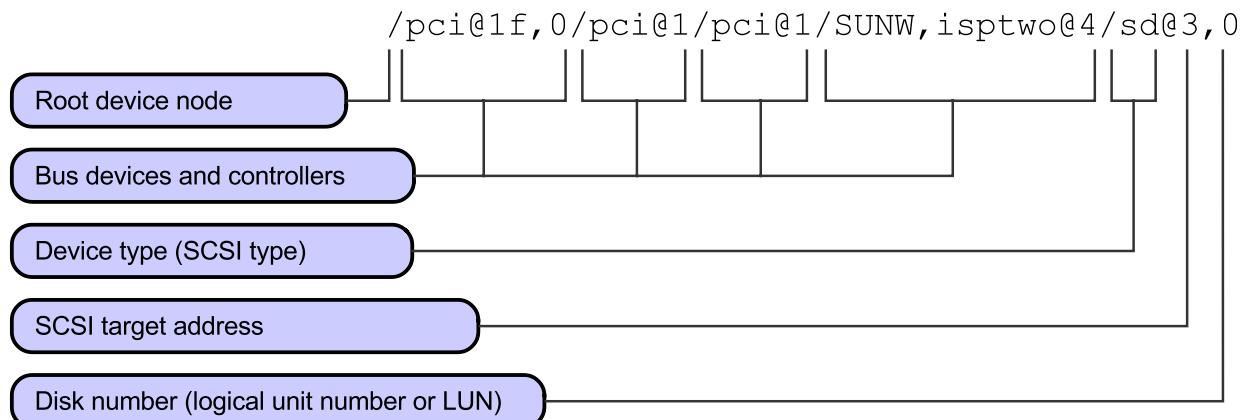


Figure 8-8 Disk Device Path – Ultra Workstation With a PCI-SCSI Bus

The show-devs Command

You use the `show-devs` command to view the entire device tree.

The following example shows output from the `show-devs` command.

```
ok show-devs
/SUNW,UltraSPARC-IIi@0,0
/pci@1f,0
/virtual-memory
/memory@0,10000000
/pci@1f,0/pci@1
/pci@1f,0/pci@1,1
/pci@1f,0/pci@1,1/ide@3
/pci@1f,0/pci@1,1/SUNW,m64B@2
/pci@1f,0/pci@1,1/network@1,1
/pci@1f,0/pci@1,1/ebus@1
/pci@1f,0/pci@1,1/ide@3/cdrom
/pci@1f,0/pci@1,1/ide@3/disk
/pci@1f,0/pci@1,1/ebus@1/SUNW,CS4231@14,200000
/pci@1f,0/pci@1,1/ebus@1/flashprom@10,0
/pci@1f,0/pci@1,1/ebus@1/eeprom@14,0
/pci@1f,0/pci@1/pci@1
/pci@1f,0/pci@1/pci@1/SUNW,isptwo@4
(output truncated)
ok
```



Note – In addition to the show-devs command, use the following additional OpenBoot PROM commands to view specific device information: show-ttys, show-displays, show-nets, show-disks, and show-tapes.

The devalias Command

To identify the current boot device alias for the system, use the devalias command.

The following example shows output from the devalias command.

```
ok devalias
screen          /pci@1f,0/pci@1,1/SUNW,m64B@2
net             /pci@1f,0/pci@1,1/network@1,1
cdrom           /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f
disk            /pci@1f,0/pci@1,1/ide@3/disk@0,0
disk3           /pci@1f,0/pci@1,1/ide@3/disk@3,0
disk2           /pci@1f,0/pci@1,1/ide@3/disk@2,0
disk1           /pci@1f,0/pci@1,1/ide@3/disk@1,0
disk0           /pci@1f,0/pci@1,1/ide@3/disk@0,0
ide              /pci@1f,0/pci@1,1/ide@3
floppy          /pci@1f,0/pci@1,1/ebus@1/fdthree
ttyb            /pci@1f,0/pci@1,1/ebus@1/se:b
ttya            /pci@1f,0/pci@1,1/ebus@1/se:a
keyboard!       /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8:forcemode
keyboard        /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse           /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
name            aliases
```

The left side of the command output lists the device alias names, and the right side of the output lists the physical address of each device.

Predefined device aliases are built into the OpenBoot PROM firmware, and they are easier to remember and use than the physical device addresses. The disk device alias identifies the default boot device for the system.

The boot-device parameter sets the system's boot device in the NVRAM. By default, the boot-device parameter is set to disk net. You can view the system's boot device parameter through commands from the ok prompt.

To boot the system from the default device, perform the boot command:

```
ok boot
```

Creating and Removing Custom Device Aliases

A portion of the NVRAM called NVRAMRC contains registers to hold custom parameters and is also reserved for storing new device alias names. External devices do not, by default, have built-in device aliases associated with them.

The NVRAMRC is affected by the commands `nvalias` and `nvunalias`, as well as the parameter `use-nvramrc?`.

The `nvalias` Command

You use the `nvalias` command to create a new device alias name to access a newly attached external device. The command format is:

```
nvalias aliasname device_path
```

The effect of the `nvalias` command is to store the following command line in the NVRAMRC:

```
devalias aliasname device_path
```

The following example shows how to add a new boot device alias, called `mydisk`, and boot the system from this new boot device alias.

 **Note** – A shortcut provided with the `show-disks` command enables you to select a device and use the Control-Y keys to copy the device path onto the command line.

The example uses the `show-disks` command to select the device path for the disk being used. It then uses the `nvalias` command to create a new device alias called `mydisk`.

```
ok show-disks
a) /pci@1f,0/pci@1/scsi@1,1/disk
b) /pci@1f,0/pci@1/scsi@1/disk
c) /pci@1f,0/pci@1,1/ide@3/cdrom
d) /pci@1f,0/pci@1,1/ide@3/disk
e) /pci@1f,0/pci@1,1/ebus@1/fdthree@14,3023f0
q) NO SELECTION
Enter Selection, q to quit: d
/pci@1f,0/pci@1,1/ide@3/disk has been selected.
Type ^Y (Control-Y) to insert it in the command line.
e.g. ok nvalias mydev ^Y
      for creating devalias mydev for
```

```
/pci@1f,0/pci@1,1/ide@3/disk
ok nvalias mydisk ^y
```

To paste the device path selected, press Control-Y on the command line.

```
ok nvalias mydisk /pci@1f,0/pci@1,1/ide@3/disk
```



Note – When the device path has been pasted on the command line (by the Control-Y keys), the target number and logical unit number (LUN) must be added for the disk device, for example, sd@0,0 or disk@0,0. If the boot slice on the disk device to be used is not slice 0, the slice letter must also be added.

```
ok nvalias mydisk /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
```

Set the boot-device parameter to the new value, in this case mydisk, and boot the system.

```
ok setenv boot-device mydisk
boot-device = mydisk
ok boot
```

The nvunalias Command

You use the nvunalias command to remove an alias name.

To remove a custom device alias name, use the following command format:

```
ok nvunalias aliasname
```



Note – The nvunalias command is the single exception to the rule that changes to NVRAM occur immediately and do not require a reset-all command.

In the example, you would use the nvunalias command to delete the alias name mydisk from NVRAMRC and use the setenv command to set the boot-device parameter to disk.

```
ok nvunalias mydisk
ok setenv boot-device disk
boot-device = disk
ok reset-all
Resetting ...
```

Viewing and Changing NVRAM Parameters From the OS

Use the `/usr/sbin/eeprom` command to view and to change the NVRAM parameters while the Solaris OS is running.

Using the `eeprom` Command

Be aware of the following guidelines when using the `eeprom` command:

- Only the root user can change the value of a parameter.
- You must enclose parameters with a trailing question mark in single quotation marks (single quotes) when the command is executed in the C shell.

The following examples use the `eeprom` command to view and change NVRAM parameters.

- To list all of the parameters with their current values, perform the command:

```
# eeprom
```

- To list a single parameter and its value, in this case, the `boot-device` parameter, perform the command:

```
# eeprom boot-device
```

```
boot-device=disk
```

```
#
```

- To change the value of the default boot device to `disk2`, perform the command:

```
# eeprom boot-device=disk2
```

```
#
```

- To change the value of the `auto-boot?` parameter, perform the command:

```
# eeprom auto-boot?=true
```

```
#
```

Interrupting an Unresponsive System

When a system freezes or stops responding to the keyboard, you might have to interrupt it. When you interrupt the system, all active processes stop immediately, and the processor services the OpenBoot PROM exclusively. It does not allow you to flush memory or to synchronize file systems.

Aborting an Unresponsive System

To abort or interrupt an unresponsive system:

1. Attempt a remote login on the unresponsive system to locate and kill the offending process.
2. Attempt to reboot the unresponsive system gracefully.
3. Hold down the Stop-A key sequence on the keyboard of the unresponsive system. The system is placed at the ok prompt.

Note – If an ASCII terminal is being used as the system console, use the Break sequence keys.



4. Manually synchronize the file systems by using the OpenBoot PROM sync command.

ok sync

This command causes the system to panic, synchronize the file systems, perform a crash dump of memory, and then reboot the system.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using the OpenBoot PROM Commands (Level 1)

In this exercise, you use the OpenBoot PROM and Solaris OS commands to perform the tasks described in this module.

Preparation

Refer to the lecture notes as necessary to perform the following tasks and answer the questions listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Shut down the system to run level 0, and gather information about your system. Find out the following:
 - OpenBoot PROM revision
 - Megabytes of installed memory
 - System type
 - NVRAM serial number
 - Ethernet address
 - Host ID
- Set the auto-boot? parameter to false.
(Steps 1–11 in the Level 2 lab)

Exercise: Using the OpenBoot PROM Commands (Level 1)

- Create a new device alias called `mydisk` that uses the same device as the `disk` device alias. Verify the contents of the `nvrampc` file, and verify how to set the `use-nvrampc?` parameter.
(Steps 12–17 in the Level 2 lab)
- Boot the system using the new alias. As the `root` user, use the `eeprom` command to list all parameters. Set the `boot-device` parameter to the `mydisk` device alias.
(Steps 18–22 in the Level 2 lab)
- Shut down the system to run level 0, and verify the change you made by using the `printenv` command. Remove the `mydisk` device alias. Reset the `boot-device` parameter to its default value, and boot the system.
(Steps 23–31 in the Level 2 lab)

Exercise: Using the OpenBoot PROM Commands (Level 2)

In this exercise, you use the OpenBoot PROM and Solaris OS commands to perform the tasks described in this module.

Preparation

Refer to the lecture notes as necessary to perform the following tasks and answer the questions listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Shut down the system to run level 0, and use the following commands to set parameters and gather basic information about your system.
`banner`
`set-defaults`
`help`
`help file`
`printenv`
`setenv`
`reset-all`
`probe-scsi`
`probe-scsi-all`
`probe-ide`
- Set the `auto-boot?` parameter to `false`.

Exercise: Using the OpenBoot PROM Commands (Level 2)

- Create a new device alias called `mydisk` that uses the same device as the `disk` device alias. Verify the contents of the `nvrampc` file, and verify how to set the `use-nvrampc?` parameter.
- Boot the system using the new alias. As the `root` user, use the `eeprom` command to list all parameters. Set the `boot-device` parameter to the `mydisk` device alias.
- Shut down the system to run level 0, and verify the change you made by using the `printenv` command. Remove the `mydisk` device alias. Reset the `boot-device` parameter to its default value, and boot the system.

Tasks

Complete the following steps:

1. If the Solaris OS is currently running, log in as the `root` user, and halt your system.
2. When the `ok` prompt appears, use the `help` command to display the list of help topics.
3. Use the `help` command to display information about the `boot` command.

What does the `help` command list for `boot`?
4. Use the `banner` command to obtain the following information:
 - OpenBoot PROM revision:
 - Megabytes of installed memory:
 - System type:
 - NVRAM serial number:
 - Ethernet address:
 - Host ID:
5. Use the `printenv` command to display the list of OpenBoot PROM parameters. Record the current values for the following parameters:
 - `output-device`
 - `input-device`
 - `auto-boot?`
 - `boot-device`
6. Prevent the system from booting automatically after you use the `reset-all` command by setting the `auto-boot?` parameter to `false`.
7. Use the `reset-all` command to verify that the new `auto-boot?` value is in effect. The system should remain at the `ok` prompt after the `reset-all` command completes.
8. Use the `probe-scsi`, `probe-scsi-all`, and `probe-ide` commands to display the list of disk devices attached to your system. Not all of these commands are present on all systems.
9. What are the main differences that you see in the information that these commands display?

10. List the target number and device type (disk, tape, or CD-ROM) of all the devices shown by the `probe-scsi`, `probe-scsi-all`, and `probe-ide` commands.

11. Verify that your default boot-device is set to disk net.

12. Use the `devalias` command to display the full device path for the disk alias.

Record the path name reported:

13. Use the `show-disks` command to select the device path that relates to the disk recorded in Step 13, and use the `nvalias` command to create a new device alias called `mydisk`. Set the `mydisk` alias to the path and disk name you recorded in Step 13.

Remember to use the Control-Y key sequence to paste the disk path into your `nvalias` command. You must manually complete the path to specify the disk you want to use.

14. Verify that the new alias is correctly set.

15. Use the `printenv` command to display the contents of the `nvramrc` file.

What command does the `nvramrc` file contain that creates the `mydisk` alias?

16. Use the `printenv` command to display the setting of the `use-nvramrc?` parameter.

What is the current setting of the `use-nvramrc?` parameter?

17. Boot your system using the `mydisk` alias.

18. Log in as the `root` user on your system. Open a new terminal window.

19. Use the `eeprom` command to list all NVRAM parameters.

20. Use the `eeprom` command to list the setting of the `boot-device` parameter.

21. Use the `eeprom` command to set the `boot-device` parameter to the alias `mydisk`.

22. Bring your system to run level 0.

23. Verify that the `eeprom` command set the `boot-device` parameter to the alias `mydisk`.

24. Set the `boot-device` parameter to its default value, and verify the setting.

25. Use the `nvunalias` command to remove the alias `mydisk`.

26. Verify that the `mydisk` alias is no longer in the `nvrampc` file.
27. Use the `dealias` command to see if the `mydisk` alias has been removed from the list of device aliases.
Has it?
28. Run the `reset-all` command, and then check again if the `mydisk` alias has been removed from the list of device aliases.
(If your system reboots, interrupt the reboot with a Stop-A key sequence.)
Has it?
29. Set the OpenBoot PROM parameters back to their default values, and boot the system from the default device.
30. Log in as the `root` user.

Exercise: Using the OpenBoot PROM Commands (Level 3)

In this exercise, you use the OpenBoot PROM and Solaris OS commands to perform the tasks described in this module.

Preparation

Refer to the lecture notes as necessary to perform the following tasks and answer the questions listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Shut down the system to run level 0, and use the following commands to set parameters and gather basic information about your system.
banner
set-defaults
help
help file
printenv
setenv
reset-all
probe-scsi
probe-scsi-all
probe-ide
- Set the auto-boot? parameter to false.

- Create a new device alias called `mydisk` that uses the same device as the `disk` device alias. Verify the contents of the `nvrampc` file, and verify how to set the `use-nvrampc?` parameter.
- Boot the system using the new alias. As the root user, use the `eeprom` command to list all parameters. Set the `boot-device` parameter to the `mydisk` device alias.
- Shut down the system to run level 0, and verify the change you made by using the `printenv` command. Remove the `mydisk` device alias. Reset the `boot-device` parameter to its default value, and boot the system.

Tasks and Solutions

Complete the following steps:

1. If the Solaris OS is currently running, log in as the `root` user, and halt your system.

```
# init 0
```

2. When the `ok` prompt appears, use the `help` command to display the list of help topics.

```
ok help
```

3. Use the `help` command to display information about the `boot` command.

```
ok help boot
```

What does the `help` command list for `boot`?

`boot` – *Default boot (values specified in NVRAM variables).*

`boot <network-device>:[dhcp,][server-ip], [boot-filename], [client-ip], [router-ip], [boot-retries], [tftp-retries], [subnet-mask], [boot-arguments].`

4. Use the `banner` command to obtain the following information:

OpenBoot PROM revision:

Megabytes of installed memory:

System type:

NVRAM serial number:

Ethernet address:

Host ID:

Each system presents its own unique information.

5. Use the `printenv` command to display the list of OpenBoot PROM parameters. Record the current values for the following parameters:

```
ok printenv
```

<code>output-device</code>	– screen
<code>input-device</code>	– keyboard
<code>auto-boot?</code>	– true
<code>boot-device</code>	– disk net

6. Prevent the system from booting automatically after using the `reset-all` command by setting the `auto-boot?` parameter to `false`.

```
ok setenv auto-boot? false
```

7. Use the `reset-all` command to verify that the new auto-boot? value is in effect. The system should remain at the `ok` prompt after the `reset-all` command completes.

`ok reset-all`

8. Use the `probe-scsi`, `probe-scsi-all`, and `probe-ide` commands to display the list of disk devices attached to your system. Not all of these commands are present on all systems.

`ok probe-scsi`

`ok probe-scsi-all`

`ok probe-ide`

9. What are the main differences that you see in the information that these commands display?

The `probe-scsi-all` command lists all devices on all SCSI chains and their full device paths. The `probe-scsi` command only lists devices on the built-in SCSI chain and does not list the full device paths. The `probe-ide` command reports the list of IDE devices attached to the system.

10. List the target number and device type (disk, tape, or CD-ROM) of all the devices shown by the `probe-scsi`, `probe-scsi-all`, and `probe-ide` commands.

Each system presents its own unique information.

11. Verify that your default boot-device is set to disk net.

`ok printenv boot-device`

12. Use the `devalias` command to display the full device path for the disk alias.

`ok devalias disk`

Record the path name reported:

This differs from system to system. On an Ultra 5 workstation, the alias is defined as follows:

`/pci@1f,0/pci@1,1/ide@3/disk@0,0`

13. Use the `show-disks` command to select the device path that relates to the disk recorded in Step 13, and use the `nvalias` command to create a new device alias called `mydisk`. Set the `mydisk` alias to the path and disk name you recorded in Step 13.

Remember to use the Control-Y key sequence to paste the disk path into your `nvalias` command. You must manually complete the path to specify the disk you want to use.

`ok show-disks`

(select one of the disks from the list)

`ok nvalias mydisk pathname#@#,#`

Exercise: Using the OpenBoot PROM Commands (Level 3)

14. Verify that the new alias is correctly set.

```
ok devalias mydisk
```

15. Use the **printenv** command to display the contents of the **nvramrc** file.

```
ok printenv nvramrc
```

What command does the **nvramrc** file contain that creates the **mydisk** alias?

Systems differ according to the disk devices they use. An Ultra 5 workstation would report the following:

```
devalias mydisk /pci@1f,0/pci@1,1/ide@3/disk@0,0
```

16. Use the **printenv** command to display the setting of the **use-nvramrc?** parameter.

```
ok printenv use-nvramrc?
```

What is the current setting of the **use-nvramrc?** parameter?
true

17. Boot your system using the **mydisk** alias.

```
ok boot mydisk
```

18. Log in as the **root** user on your system. Open a new terminal window.
19. Use the **eeprom** command to list all NVRAM parameters.

```
# eeprom
```

20. Use the **eeprom** command to list the setting of the **boot-device** parameter.

```
# eeprom boot-device
```

21. Use the **eeprom** command to set the **boot-device** parameter to the alias **mydisk**.

```
# eeprom boot-device=mydisk
```

22. Bring your system to run level 0.

```
# init 0
```

23. Verify that the **eeprom** command set the **boot-device** parameter to the alias **mydisk**.

```
ok printenv boot-device
```

24. Set the **boot-device** parameter to its default value, and verify the setting.

```
ok set-default boot-device
```

```
ok printenv boot-device
```

25. Use the nvunalias command to remove the alias mydisk.

```
ok nvunalias mydisk
```

26. Verify that the mydisk alias is no longer in the nvramrc file.

```
ok printenv nvramrc
```

27. Use the devalias command to see if the mydisk alias has been removed from the list of device aliases.

```
ok devalias mydisk
```

Has it?

No

28. Run the reset-all command, and then check again if the mydisk alias has been removed from the list of device aliases.

```
ok reset-all
```

(If your system reboots, interrupt the reboot with a Stop-A key sequence.)

```
ok devalias mydisk
```

Has it?

Yes

29. Set the OpenBoot PROM parameters back to their default values, and boot the system from the default device.

```
ok set-defaults
```

```
ok printenv
```

```
ok reset-all
```

30. Log in as the root user.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 9

Performing Boot and Shutdown Procedures

Objectives

Upon completion of this module, you should be able to:

- Describe the features of the Service Management Facility
- Identify run level fundamentals
- Compare run levels and SMF milestones
- Identify the phases of the boot process
- Use SMF administrative commands
- Control boot processes
- Perform system shutdown procedures

The course map in Figure 9-1 shows how this module fits into the current instructional goal.

Performing System Boot Procedures

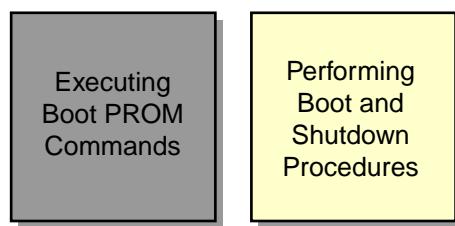


Figure 9-1 Course Map

The Service Management Facility (SMF)

The SMF provides a centralized configuration structure for managing system services and the interaction of a service with other services. The SMF includes the following:

- A mechanism to establish and formalize dependency relationships between services.
- Information on procedures to start, stop, and restart services.
- A centralized repository for information on startup behavior and service status.
- A structured mechanism for Fault Management of system services.
- Detailed information about misconfigured services such as an explanation of why a service is not running.
- Individual log files for each service.

SMF Service

A service can be described as an entity which provides a resource or list of capabilities to applications and other services, both local and remote. A service is not necessarily a running process, such as a web server. A service can also be the software state of a device, such as a configured network device, or a mounted file system.

A system can have more than one occurrence of a service running. For example, a system can have more than one configured network interface, or more than one mounted file system.

Service Identifiers

Each instance of a service within SMF has a name which is referred to as a “Service Identifier.” This service identifier is in the form of a Fault Management Resource Identifier or FMRI. The FMRI indicates the type of service or category, and the name and instance of the service.

The service categories include the following:

- application
- device
- legacy
- milestone
- network
- platform
- site
- system

An example of an FMRI for a service instance is:

```
svc:/system/filesystem/root:default
```

Where:

- The prefix `svc` indicates that this service is managed by SMF
- The category of the service is `system`
- The service itself is a `filesystem`
- The instance of the service is the `root` file system
- The word `default` identifies the first, in this case only, instance of the service

Another example of an FMRI for a service is:

```
lrc:/etc/rc3.d/S90samba
```

Where:

- The prefix `lrc` (Legacy Run Control) indicates that this service currently is not managed by SMF
- The pathname `/etc/rc3.d` refers to the directory `/etc/rc3.d` where there is a script used to manage this service
- The name of the script is `S90samba`

Listing Service Information

Service instance names and the state of the service can be listed using the svcs command.

```
# svcs
STATE      STIME     FMRI
legacy_run Feb_10   lrc:/etc/rc2_d/S10lu
legacy_run Feb_10   lrc:/etc/rc2_d/S20sysetup
legacy_run Feb_10   lrc:/etc/rc2_d/S90wbem
legacy_run Feb_10   lrc:/etc/rc2_d/S99dtlogin
legacy_run Feb_10   lrc:/etc/rc3_d/S81volmgt
(output removed)
online      Feb_10   svc:/system/system-log:default
online      Feb_10   svc:/system/fmd:default
online      Feb_10   svc:/system/console-login:default
online      Feb_10   svc:/network/smtp:sendmail
online      Feb_10   svc:/milestone/multi-user:default
online      Feb_10   svc:/milestone/multi-user-server:default
online      Feb_10   svc:/system/zones:default
offline     Feb_10   svc:/application/print/ipp-listener:default
offline     Feb_10   svc:/application/print/rfc1179:default
maintenance 10:24:15 svc:/network/rpc/spray:default
```

Service States

The svcs command can be used to list service identifiers and the state of the service instance. A service can be either enabled or disabled. Service states can include the following:

- **online**
The service instance is enabled and has successfully started.
- **offline**
The service instance is enabled, but the service is not yet running or available to run.
- **disabled**
The service instance is not enabled and is not running.
- **legacy_run**
The legacy service is not managed by SMF, but the service can be observed. This state is only used by legacy services.
- **uninitialized**
This state is the initial state for all services before their configuration has been read.
- **maintenance**
The service instance has encountered an error that must be resolved by the administrator.
- **degraded**
The service instance is enabled, but is running at a limited capacity.

Milestones

A milestone is a special type of service which is made up of a defined set of other services.

A milestone can be regarded as a system state to reach. This system state requires a defined set of services to be running. These services depend on other services being available. Hence, there is a hierarchy of dependency relationships. This is one of the core features managed by SMF. Currently there are seven milestones.

- single-user
- multi-user
- multi-user-server
- network
- name-services
- sysconfig
- devices

Figure 9-2 shows the relationship between a milestone and services.

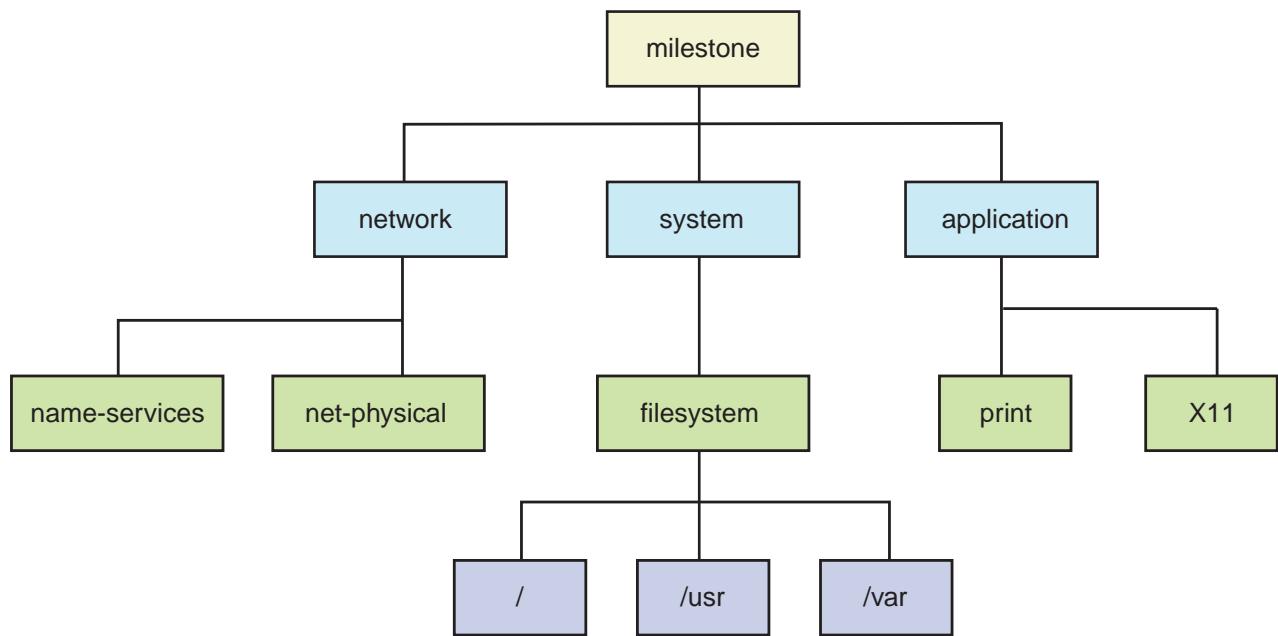


Figure 9-2 SMF Milestone and Services

Figure 9-3 shows an example of the dependency relationships.

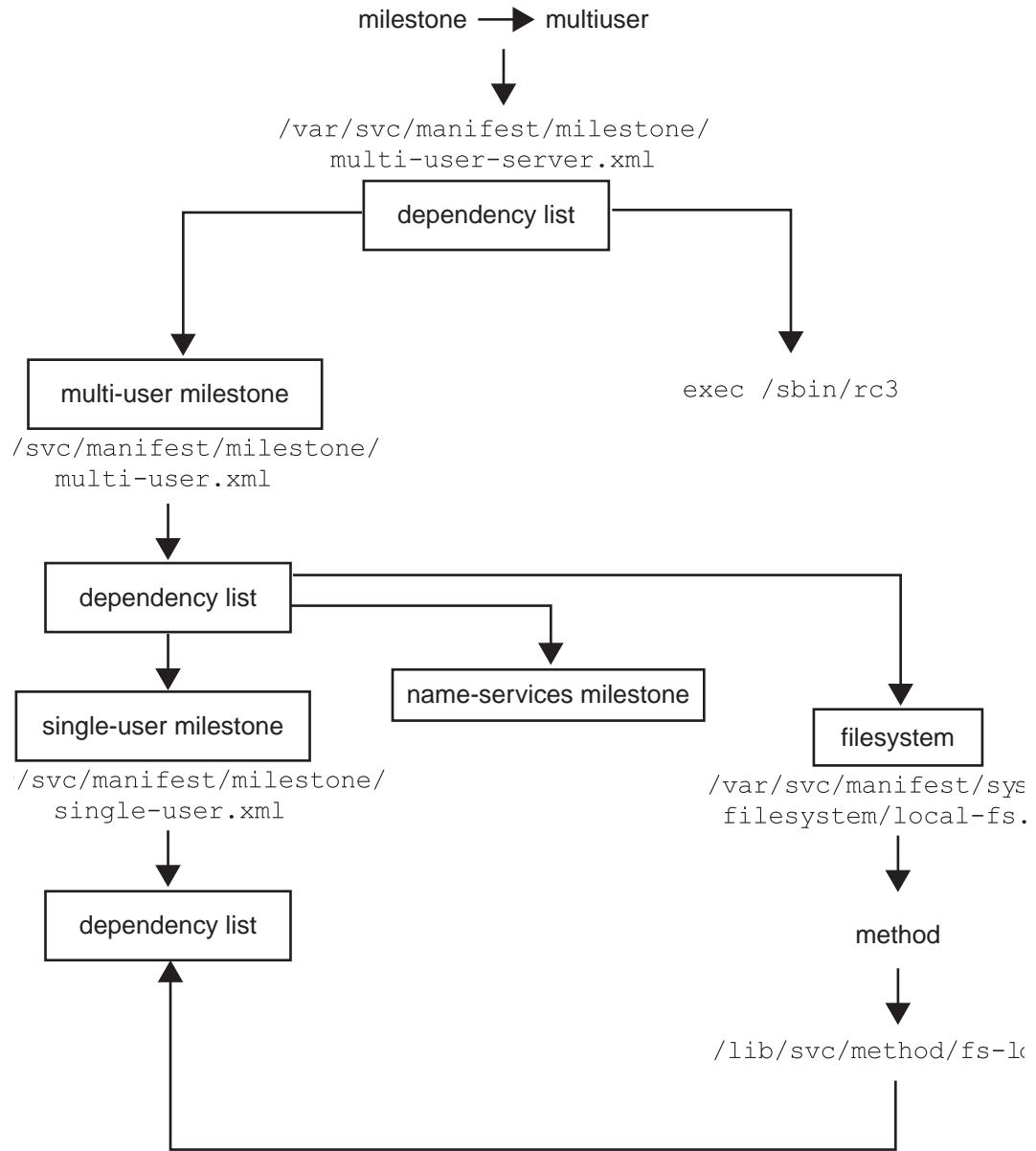


Figure 9-3 SMF Dependency Relationships

To determine the current milestones

```
# svcs | grep milestone
online      9:58:42 svc:/milestone/name-services:default
online      9:58:53 svc:/milestone/network:default
online      9:58:54 svc:/milestone/devices:default
online      9:59:09 svc:/milestone/single-user:default
online      9:59:13 svc:/milestone/sysconfig:default
online      9:59:42 svc:/milestone/multi-user:default
online      9:59:51 svc:/milestone/multi-user-server:default
```

The svc.startd Daemon

The svc.startd daemon is the daemon which is responsible for maintaining the system services. It is the svc.startd daemon which ensures that the system boots to the appropriate milestone. If no milestone is specified at boot up, svc.startd boots to the built-in milestone “all” which includes all the system services.

Currently the milestones that can be used at boot time are the following:

- none
- single-user
- multi-user
- multi-user-server
- all

In order to boot the system to a specific milestone, the **-m** option is passed to the boot command from OBP.

```
ok> boot -m milestone=single-user
```

The svc.startd daemon can be referred to as the master restarter daemon because it is responsible for ensuring the correct running, starting, and restarting of system services. The svc.startd daemon can obtain information about services from the repository.

The svc.startd daemon is able to delegate responsibility for services to other delegated restarter daemons for example, the inetd daemon.

The Service Configuration Repository

The repository database stores information about the state of each service instance. It also stores configuration information about the services and system. The repository is distributed among local memory and local disk-based files. The disk-based database is /etc/svc/repository.db.

This file can only be manipulated using the SMF interface utilities svccfg and svcprop.

The repository is managed by the `svc.configd` daemon. The `svc.configd` daemon backs up the repository before applying any changes issued by the SMF commands and utilities. These backup copies of the repository ensure that fallback is possible.

A corrupt repository prevents the system from booting. A corrupt repository can be repaired by booting the system to single-user, and running the command:

```
# /lib/svc/bin/restore_repository
```

and following the instructions.

Identifying Legacy Run Level Fundamentals

A run level is a system state, represented by a digit or letter, that defines the services and resources that are currently available to users.

Table 9-1 shows the eight run levels found in the Solaris OS.

Table 9-1 Solaris OS Run Levels

Run Level	Milestone	Function
0		System is running the PROM monitor.
s or S	single-user	Solaris OS single-user mode with critical file systems mounted and accessible.
1		The system is running in a single-user administrative state with access to all available file systems.
2	multi-user	The system is supporting multiuser operations. Multiple users can access the system. All system daemons are running except for the Network File System (NFS) server and some other network resource server related daemons.
3	multi-user-server	The system is supporting multiuser operations and has NFS resource sharing and other network resource servers available.
4		This level is currently not implemented.
5		A transitional run level in which the Solaris OS is shut down and the system is powered off.
6		A transitional run level in which the Solaris OS is shut down and the system reboots to the default run level.

Determining a System's Current Run Level

To determine the current run level of a system, use the `who -r` command.

Figure 9-4 shows output from the command.

```
# who -r
```

Current run level	. run level 3
Date and time of the last run level change	Jun 9 08:30
Current run level	3 0 S
Number of times at this run level since the last reboot	
Previous run level	

Figure 9-4 The System's Current Run Level

Changing Run Levels

Run levels are sometimes referred to as init states because the `init` command can be used to transition between run levels. The `init` command passes the required run level to `svc.startd`.

You can use the `init` command to manually initiate run-level transitions. You can also change run levels with the `shutdown`, `halt`, `reboot`, and `poweroff` commands. In addition, the `svcadm` command, can be used to change the run level that a system will boot to, by selecting the milestone to achieve.



Note – Prior to Solaris 10 OS, the init daemon was responsible for starting and stopping system services. Starting with Solaris 10 OS, this role is now the responsibility of the svc.startd daemon. The init daemon initializes stream modules, configures socket transport providers, sets up the system for a correct response to a power fail shutdown, and starts the svc.startd daemon.

Identifying the Phases of the Boot Process

In general, when a system is powered on, the PROM monitor runs a POST procedure that checks the hardware and memory on the system. If no errors are found, and the auto-boot? parameter is set to true, the system begins the boot process.

The entire boot process is described by five distinct phases:

- The boot PROM phase
- The boot programs phase
- The kernel initialization phase
- The init phase
- The svc.startd phase

Figure 9-5 shows the phases of the boot process.

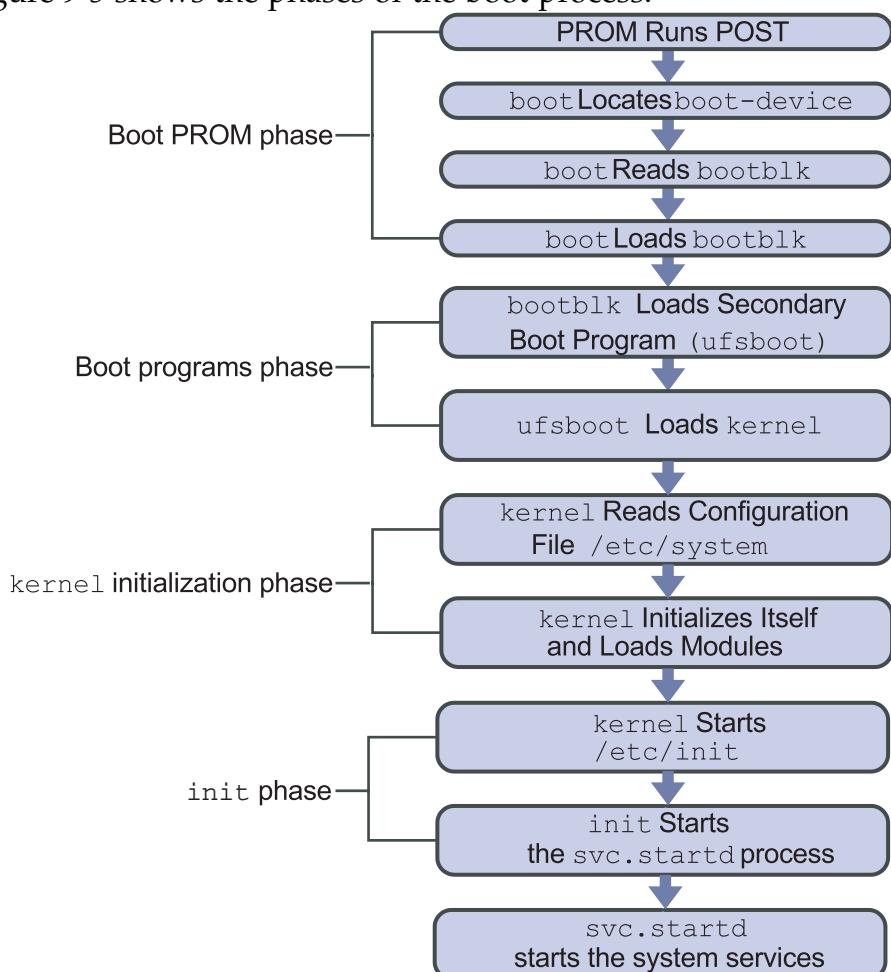


Figure 9-5 Phases of the Boot Process

Boot PROM Phase

The boot PROM performs the following steps during the first part of the boot sequence:

- The PROM runs the POST.

The boot PROM firmware runs the POST to verify the system's hardware and memory. It then begins its boot sequence upon successful completion of the self-test diagnostics.
- The PROM displays the system identification banner.

The model type, processor type and speed, keyboard status, PROM revision number, amount of installed random access memory (RAM), NVRAM serial number, Ethernet address, and host ID are displayed.
- The boot PROM determines the boot device by reading the PROM parameter boot-device.
- The boot PROM reads the disk label located at Sector 0 on the default boot device.
- The boot PROM finds the boot program from the default boot device programmed into the PROM.

The boot PROM program reads a system's primary boot program called bootblk (located at Sectors 1 through 15) that contains a UNIX file system (UFS) file system reader. (The bootblk program is placed on the disk by the installboot command during system installation.)

The boot command loads the bootblk program from its location on the boot device into memory.

Boot Programs Phase

The following describes the boot programs phase:

- The bootblk program loads the secondary boot program, ufsboot, from the boot device into memory.

The path to ufsboot is recorded in the bootblk program, which is installed by the installboot command.

- The ufsboot program locates and loads the appropriate two-part kernel.

The core of the kernel is two pieces of static code called genunix and unix, where genunix is the platform-independent generic kernel file and unix is the platform-specific kernel file.

When ufsboot loads these two files into memory, they are combined to form the running kernel.

On a system running in 64-bit mode, the two-part kernel is located in the directory:

```
/platform/`uname -m'/kernel/sparcv9
```

Note – Solaris 10 for SPARC only runs on 64-bit systems.



Note – To determine the platform name (for example, the system hardware class), type the `uname -m` command. For example, when you type this command on an Ultra 10 workstation, the console displays `sun4u`.

The kernel Initialization Phase

The following describes the kernel initialization phase:

- The kernel reads its configuration file, called `/etc/system`.
- The kernel initializes itself and begins loading modules.

The kernel uses the `ufsboot` command to load the files. When it has loaded enough modules to mount the `/` (root) file system, it unmaps the `ufsboot` program and continues.

- The kernel starts the `/etc/init` daemon.



Note – The /etc/init file is a symbolic link to /sbin/init.

The SunOS™ kernel is a small static core, consisting of genunix and unix and many dynamically loadable kernel modules.

Modules can consist of device drivers, binary files to support file systems, and streams, as well as other module types used for specific tasks within the system.

The modules that make up the kernel typically reside in the directories /kernel and /usr/kernel. Platform-dependent modules reside in the /platform/`uname -m`/kernel and /platform/`uname -i`/kernel directories.

The following describes the types of module subdirectories contained in the /kernel, /usr/kernel, /platform/`uname -m`/kernel, or /platform/`uname -i`/kernel directories:

- drv/sparcv9 – Device drivers
- exec/sparcv9 – Executable file formats
- fs/sparcv9 – File system types, for example, ufs, nfs, and proc
- misc/sparcv9 – Miscellaneous modules, for example, usb
- sched/sparcv9 – Scheduling classes (process execution scheduling)
- strmmod/sparcv9 – Streams modules (generalized connection between users and device drivers)
- sys/sparcv9 – System calls (defined interfaces for applications to use)

The /kernel/drv/sparcv9 directory contains all of the device drivers that are used for system boot. The /usr/kernel/drv/sparcv9 directory is used for all other device drivers.

Modules are loaded automatically as needed either at boot time or on demand, if requested by an application. When a module is no longer in use, it might be unloaded on the basis that the memory it uses is needed for another task.

After the boot process is complete, device drivers are loaded when devices, such as tape devices, are accessed. This process is called autoconfiguration because some kernel driver modules are loaded automatically when needed.

Upon initial or reconfiguration boot, the system does a self-test and checks for all devices that are attached.

The advantage of this dynamic kernel arrangement is that the overall size of the kernel is smaller, which makes more efficient use of memory and allows for simpler modification and tuning. Figure 9-6 shows this arrangement.

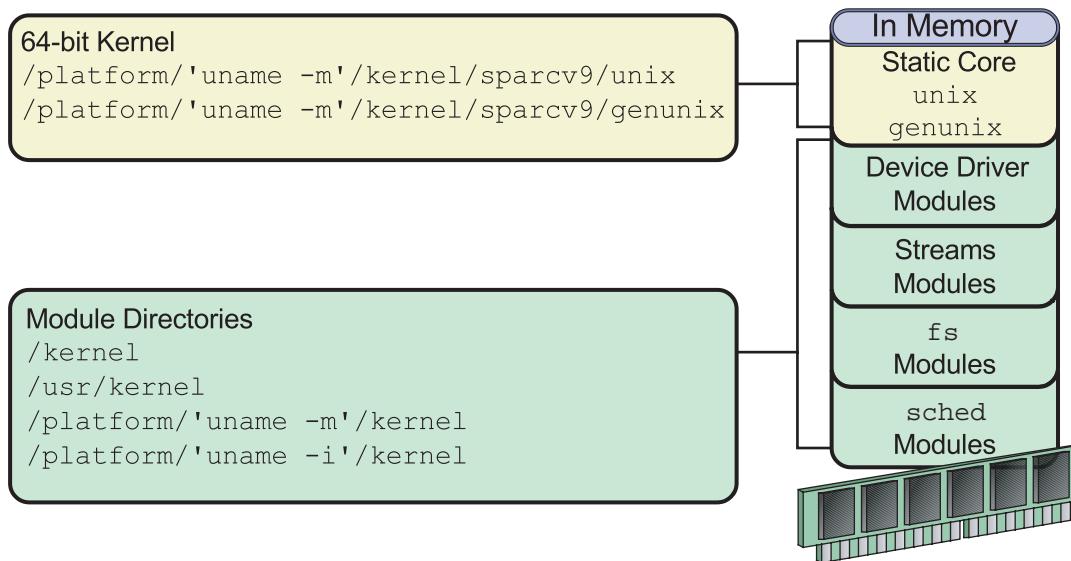


Figure 9-6 Dynamic kernel Arrangement

Note – The sparcv9 CPU is the type of CPU that supports 64-bit processing.



The /etc/system File and Kernel Configuration



Caution – The Solaris OS builds the kernel based upon the size of the system (memory, CPUs, and so on). In almost all cases, the performance of the default kernel that is built is quite adequate to handle most day to day activities on the system. Any modifications should be made with extreme caution.

The `/etc/system` file is the control file for modifying which modules and parameters are to be loaded by the kernel at boot time. By default, all lines in this file are commented out.

Modifying the kernel's behavior (or configuration) requires editing the /etc/system file. Altering this file allows you to modify the kernel's treatment of loadable modules as well as to modify kernel parameters for some performance tuning.

The ufsboot program contains a list of default loadable kernel modules that are loaded at boot time. However, you can override this list by modifying the /etc/system file to control which modules, as well as which parameters, are loaded.

All changes to this file take effect after a reboot.

The /etc/system file can explicitly control:

- The search path for default kernel modules to be loaded at boot time
- The root file system type and device
- The modules that are excluded from loading automatically at boot time
- The modules to be forcibly loaded at boot time, rather than at first access
- The new values to override the default kernel parameter values

Note – Command lines must be 80 characters or less in length, and comment lines must begin with an asterisk (*) and end with a newline character.



The /etc/system file is divided into five distinct sections:

- moddir:
Sets the search path for default loadable kernel modules. You can list together multiple directories to search, delimited either by blank spaces or colons. If the module is not found in the first directory, the second directory is searched, and so on.

- root device and root file system configuration:
Sets the root file system type to the listed value. The default is rootfs:ufs.

Sets the root device. The default is the physical path name of the device on which the boot program resides. The physical path name is platform dependent and configuration dependent. The following is an example path:

```
rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a
```

- exclude:
Does not allow the loadable kernel modules to be loaded during kernel initialization, for example:

```
exclude: sys/shmsys
```

- forceload:
Forces the kernel modules to be loaded during kernel initialization, for example:

```
forceload: drv/vx
```

The default action is to load a kernel module automatically when its services are first accessed during runtime by a user or an application.

- set:
Changes kernel parameters to modify the operation of the system, for example:

```
set maxusers=40
```

Editing the /etc/system File



Caution – Before you edit the /etc/system file, you should make a backup copy. If you enter incorrect values in this file, the system might not be able to boot.

The following example shows how to copy the original /etc/system file to a backup file and then edit the /etc/system file.

```
# cp /etc/system /etc/system.orig  
# vi /etc/system
```

If a boot process fails because of an unusable /etc/system file, issue the interactive boot command: boot -a. When you are requested to enter the name of the system file, type in the name of your backup system file, or, alternatively, enter /dev/null for a null configuration file.

```
ok boot -a  
Enter filename [kernel/sparcv9/unix]: <Return>  
Enter default directory for modules [/platform...]: <Return>  
Name of system file [etc/system]: etc/system.orig - or - /dev/null  
root filesystem type [ufs]: <Return>  
Enter physical name of root device [/...]: <Return>  
(further boot messages omitted)
```

The init Phase

The next to the last phase of the boot process is the init phase. During this phase, the init daemon starts the svc.startd daemon that is responsible for starting and stopping services as requested. The /sbin/init phase uses information stored in the /etc/inittab file.

The /etc/inittab File

Each line in the /etc/inittab file contains the following four fields:

id:rstate:action:process

Table 9-2 describes the fields in an inittab entry.

Table 9-2 Fields in the inittab File

Field	Description
id	Two character identifier for the entry
rstate	Run levels to which this entry applies
action	Defines how the process listed should be run For a description of the action keywords see <code>man inittab</code>
process	Defines the command to execute

The following example shows the default inittab file installed with the Solaris 10 OS. The lines of output are described after the example:

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
sp::sysinit:/sbin/soconfig -f /etc/sock2path
smf::sysinit:/lib/svc/bin/svc.startd>/dev/msglog 2<>/dev/msglog
</dev/console
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog
2<>/dev/msglog
```

 **Note** – Message output from rc scripts is directed to the /dev/msglog file. Prior to the Solaris 8 OS, all of these messages were written to the /dev/console file. The /dev/msglog file is used for message output collection from system startup or background applications.

Table 9-3 shows an explanation for each action keyword.

Table 9-3 The action Field Keywords

Keyword	Explanation
sysinit	Executes the process before the init process tries to access the console, for example, the console login prompt. The init process waits for completion of the process before it continues to read the inittab file.
powerfail	Executes the process only if the init process receives a power fail signal.

Information about additional action keywords is available in the inittab man page. The following describes each of the lines in the /etc/inittab file in order:

1. Initializes the STREAMS modules used for communication services.
2. Configures the socket transport providers for network connections.
3. Initializes the svc.startd daemon for SMF.
4. Describes a power fail shutdown.

The svc.startd Daemon

The `svc.startd` daemon is the master process starter and restarter for SMF. This daemon takes on the role of starting the appropriate processes for the achieved run level. This was previously the responsibility of the `init` process.

The `svc.startd` daemon uses information in the repository to determine the required milestone and then starts to process the manifests located in the `/var/svc/manifest` directory.

The `/var/svc/manifest/milestone` directory contains Extensible Markup Language (XML) files which describe the dependencies for this milestone. Recall that a milestone is made up of multiple SMF services.

Files in the `/var/svc/manifest/milestone` directory:

- `single-user.xml`
- `multi-user.xml`
- `multi-user-server.xml`
- `network.xml`
- `name-services.xml`
- `sysconfig.xml`

These `.xml` files might refer to other `.xml` files in subdirectories below `/var/svc/manifest` that contain commands to run, such as:

- `/sbin/rc2`
- `/lib/svc/method/fs-local`

Controlling Legacy Boot Processes

The Solaris OS provides a series of Legacy scripts to stop and start processes typically associated with run levels or milestones.

The /sbin Directory

Each run level has an associated script located in the /sbin directory, with some scripts hard-linked to each other.

The scripts are executed by the svc.startd daemon to set up variables, test conditions, and make calls to other scripts that start and stop processes for that run level.

The rc0, rc5, and rc6 scripts are hard-linked to each other. Notice that each script is assigned the same inode number. The following is an example of the hard links:

```
# ls -li /sbin/rc*
2317 -rwxr--r-- 3 root sys 1983 Dec 22 18:06 rc0
2318 -rwxr--r-- 1 root sys 2242 Dec 22 18:06 rc1
2319 -rwxr--r-- 1 root sys 2536 Dec 22 18:06 rc2
2320 -rwxr--r-- 1 root sys 2567 Dec 22 18:06 rc3
2317 -rwxr--r-- 3 root sys 1983 Dec 22 18:06 rc5
2317 -rwxr--r-- 3 root sys 1983 Dec 22 18:06 rc6
2321 -rwxr--r-- 1 root sys 5125 Dec 22 18:06 rcS
```

The Solaris OS provides the same series of rc scripts in the /etc directory for backward compatibility. These scripts are symbolic link files to the rc scripts in the /sbin directory. The following example shows this connection:

```
# ls -l /etc/rc?
ls -l rc?
lrwxrwxrwx 1 root root 11 Oct 12 17:15 rc0 -> ../sbin/rc0
lrwxrwxrwx 1 root root 11 Oct 12 17:15 rc1 -> ../sbin/rc1
lrwxrwxrwx 1 root root 11 Oct 12 17:15 rc2 -> ../sbin/rc2
lrwxrwxrwx 1 root root 11 Oct 12 17:15 rc3 -> ../sbin/rc3
lrwxrwxrwx 1 root root 11 Oct 12 17:15 rc5 -> ../sbin/rc5
lrwxrwxrwx 1 root root 11 Oct 12 17:15 rc6 -> ../sbin/rc6
lrwxrwxrwx 1 root root 11 Oct 12 17:15 rcS -> ../sbin/rcS
rcm:
total 2
drwxr-xr-x 2 root sys 512 Oct 12 17:18 scripts
```

Table 9-4 summarizes the tasks performed by each of the /sbin scripts.

Table 9-4 Run Control Scripts and Their Functions

Script	Function
/sbin/rc0	Runs the /etc/rc0.d/K* scripts and the /etc/rc0.d/S* scripts to stop system services and daemons. Start scripts should only perform fast system cleanup functions.
/sbin/rc1	Runs the /etc/rc1.d/S* scripts to perform the following tasks: <ul style="list-style-type: none"> • Stops system services and daemons • Terminates certain running application processes • Unmounts all remote file systems
/sbin/rc2	Runs the /etc/rc2.d/K* scripts and the /etc/rc2.d/S* scripts to start certain application daemons.
/sbin/rc3	Runs the /etc/rc3.d/K* scripts and the /etc/rc3.d/S* scripts to start certain application daemons. Note: The K scripts are not normally present in the /etc/rc3.d directory, although if they were present, they would be run.
/sbin/rc5 /sbin/rc6	Runs the /etc/rc0.d/K* scripts and then the /etc/rc0.d/S* scripts to perform the following tasks: <ul style="list-style-type: none"> • Stops system services and daemons • Starts scripts that should only perform fast system cleanup functions
/sbin/rcS	Runs the /etc/rcS.d scripts to bring up the system to run level S, and establish a minimal network.

The /etc/rc#.d Directories

The /etc/rc#.d directories contain scripts that start and stop system processes for that run level.

Figure 9-7 shows an example of /etc/rc#.d directories.

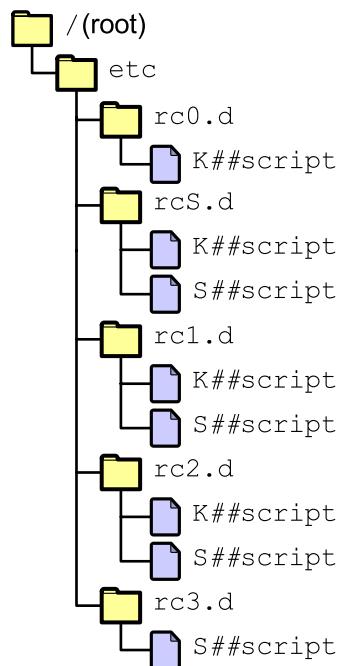


Figure 9-7

The /etc/rc#.d directories, for example /etc/rc2.d, contain scripts to start and stop processes for run level 2. The following output shows a partial list of these scripts.

```
# ls -l /etc/rc2.d
total 130
-rwxr--r-- 6 root      sys          324 Dec 10 11:26 K03samba
-rwxr--r-- 5 root      sys          824 May 27 2004 K05appserv
(some output omitted)
-rwxr--r-- 5 root      sys         2329 Dec 11 08:52 S94ncalodg
-rwxr--r-- 2 root      sys          733 Dec 11 08:54 S98deallocate
-rwxr--r-- 5 root      sys         1023 Dec 11 08:53 S99audit
-rwxr--r-- 5 root      sys         2804 Dec  7 20:52 S99dtlogin
```

Start Run Control Scripts

The `/etc/rc#.d` start scripts are always run in the sort order shown by the `ls` command. The files that begin with S are run to start a system process. These scripts are called by the appropriate `/sbin/rc#` and this script passes the argument “start” to them if their names do not end in `.sh`. There are no arguments passed to `.sh` scripts. These files have names in the form of:

`S##name-of-script`

For example, the `S99dtlogin` script starts the graphical login process.

Stop Run Control Scripts

The `/etc/rc#.d` stop scripts (also referred to as the kill scripts) are always run in the sort order shown by the `ls` command. The files that begin with K are run to stop or kill a system process. These scripts are called by the appropriate `/sbin/rc#`, and this script passes the argument “stop” to them if their names do not end in `.sh`.

These files have names in the form of:

`K##name-of-script`

For example, the `K03samba` script stops the Samba server.

Note – File names that begin with a lowercase k or s are ignored, and are not executed. To disable a script, rename it with the appropriate lowercase letter.



The /etc/init.d Directory

Run control scripts are located in the /etc/init.d directory.

The run control script /etc/init.d/samba is hard-linked to the corresponding run control script /etc/rc3.d/S90samba, as shown by the ls commands:

```
# cd /etc/init.d  
# ls -i samba  
4715 samba  
  
# cd /etc/rc3.d  
# ls -i S90samba  
4715 samba
```

You can stop a process or start a process without changing the system's run level.

For example, to stop and restart the samba file and print sharing service, run the following command with a start or stop argument:

```
# /etc/init.d/samba start  
  
# /etc/init.d/samba stop
```

Stopping and Starting Services Using SMF Commands

List Services With the svcs Command

The `svcs` command is used to monitor SMF services. It is useful for examining the status of services, and for following the dependency relationship between services.

```
# svcs
STATE      STIME      FMRI
legacy_run 13:45:11  lrc:/etc/rcS_d/S29wrsmcfg
legacy_run 13:45:37  lrc:/etc/rc2_d/S10lu
legacy_run 13:45:38  lrc:/etc/rc2_d/S20syssetup
legacy_run 13:45:38  lrc:/etc/rc2_d/S40llc2
legacy_run 13:45:38  lrc:/etc/rc2_d/S42ncakmod
legacy_run 13:45:39  lrc:/etc/rc2_d/S47pppd
(output omitted)
online     13:45:36  svc:/network/smtp:sendmail
online     13:45:38  svc:/network/ssh:default
online     13:45:38  svc:/system/fmd:default
online     13:45:38  svc:/application/print/server:default
online     13:45:39  svc:/application/print/rfc1179:default
online     13:45:41  svc:/application/print/ipp-listener:default
online     13:45:45  svc:/milestone/multi-user:default
online     13:45:53  svc:/milestone/multi-user-server:default
online     13:45:54  svc:/system/zones:default
online     8:46:25   svc:/system/filesystem/local:default
online     8:46:26   svc:/network/inetd:default
online     8:46:32   svc:/network/rpc/meta:tcp
online     8:46:32   svc:/system/mdmonitor:default
online     8:46:38   svc:/milestone/multi-user:default
online     13:14:35  svc:/network/telnet:default
maintenance 8:46:21  svc:/network/rpc/keyserv:default
```

The `svcs` command can also be used to examine the status of a specific service instance. For example:

```
# svcs svc:/system/console-login:default
STATE      STIME      FMRI
online     14:38:27  svc:/system/console-login:default
#
```

It is also possible to examine the dependency relationships of services using the svcs command by using the -d and the -D options. The -d option shows what other services the named service is dependent on. The -D option shows what other services depend on the named service.

The following example shows what the service dependencies are for the filesystem/local:default service instance:

```
# svcs -d svc:/system/filesystem/local:default
STATE      STIME      FMRI
online     14:38:15  svc:/system/filesystem/minimal:default
online     14:38:26  svc:/milestone/single-user:default
#
#
```

The following example shows what services the multi-user:default milestone depends on, or requires to run:

```
# svcs -d milestone/multi-user:default
STATE      STIME      FMRI
online     13:44:53  svc:/milestone/name-services:default
online     13:45:12  svc:/milestone/single-user:default
online     13:45:13  svc:/system/filesystem/local:default
online     13:45:15  svc:/network/rpc/bind:default
online     13:45:16  svc:/milestone/sysconfig:default
online     13:45:17  svc:/system/utmp:default
online     13:45:19  svc:/network/inetd:default
online     13:45:31  svc:/network/nfs/client:default
online     13:45:34  svc:/system/system-log:default
online     13:45:36  svc:/network/smtp:sendmail
#
#
```

The following example shows what other services depend on the system/filesystem/local service:

```
# svcs -D svc:/system/filesystem/local
STATE          STIME      FMRI
disabled       13:44:50  svc:/network/inetd-upgrade:default
disabled       13:44:51  svc:/network/nfs/server:default
disabled       13:45:10  svc:/application/management/webmin:default
disabled       13:45:12  svc:/application/gdm2-login:default
online         13:45:14  svc:/system/sysidtool:net
online         13:45:14  svc:/system/cron:default
online         13:45:16  svc:/system/sysidtool:system
online         13:45:16  svc:/network/nfs/status:default
online         13:45:17  svc:/system/sac:default
online         13:45:19  svc:/network/inetd:default
online         13:45:21  svc:/application/font/fc-cache:default
online         13:45:34  svc:/system/filesystem/autofs:default
online         13:45:34  svc:/system/system-log:default
online         13:45:35  svc:/system/dumpadm:default
online         13:45:36  svc:/network/smtp:sendmail
online         13:45:38  svc:/network/ssh:default
online         13:45:38  svc:/application/print/server:default
online         13:45:45  svc:/milestone/multi-user:default
#
#
```

Changing Service States Using the svcadm Command

The `svcadm` command can be used to change the state of services managed by SMF. For example, to verify the status of the `cron` service:

```
# pgrep -fl cron
 180 /usr/sbin/cron
#
# svcs cron
STATE          STIME      FMRI
online          14:38:30  svc:/system/cron:default

# svcadm -v disable system/cron:default
svc:/system/cron:default disabled.

# svcs cron
STATE          STIME      FMRI
disabled        20:35:25  svc:/system/cron:default

# pgrep -fl cron
#
# svcadm -v enable system/cron:default
svc:/system/cron:default enabled.

# svcs cron
STATE          STIME      FMRI
online          20:35:59  svc:/system/cron:default
#
# pgrep -fl cron
 180 /usr/sbin/cron
#
```

Disabling the `cron` service with the `svcadm -v disable` command disables the service permanently until it is enabled from the command line. You can disable the service temporarily until the next reboot by using the `-t` option.

```
# svcadm -v disable -t system/cron:default
svc:/system/cron:default temporarily disabled.
```

Using svcs to Determine Why Services are Not Running

The svcs command can also be used to troubleshoot why services are not running.

```
# svcs -x cron
svc:/system/cron:default (clock daemon (cron))
  State: disabled since Fri Feb 25 15:05:47 2005
Reason: Temporarily disabled by an administrator.
  See: http://sun.com/msg/SMF-8000-1S
  See: cron(1M)
  See: crontab(1)
  See: /var/svc/log/system-cron:default.log
Impact: This service is not running.
```

The cron service has been temporarily disabled by the administrator.

Further information about the service can be found in the /var/svc/log/system-cron:default.log log file, and at <http://sun.com/msg/SMF-8000-1S>.

Manipulating Services That Are Not Managed by SMF

If the FMRI prefix for a service is `lrc`, then that service is not currently managed by SMF. In order to start and stop the service without changing run levels, the script associated with that service has to be run manually. The script to run a legacy service, non-SMF managed, is in `/etc/init.d`.

For example, to stop and start the `vold` volume manager daemon:

```
# svcs | grep vol
legacy_run      14:38:57 lrc:/etc/rc3_d/S81volmgt

# pgrep -lf vold
480 /usr/sbin/vold

# ls /etc/init.d/volmgt
/etc/init.d/volmgt

# pgrep -lf vold
#
# /etc/init.d/volmgt start
volume management starting.

# pgrep -lf vold
1070 /usr/sbin/vold

# svcs | grep vol
legacy_run      14:38:57 lrc:/etc/rc3_d/S81volmgt
#
```

Creating New Service Scripts

You can create new scripts to start and stop additional processes or services to customize a system.

For example, to eliminate the requirement for a manual start of a database server, you could create a script to start the database server automatically after the appropriate network services have started.

You could then create another script to terminate this service and shut down the database server before the network services are stopped.

The correct procedure is to incorporate the new service into the SMF. This procedure can be quite complex. The general steps required are detailed in the following list:

- Determine the process for starting and stopping your service.
- Establish a name for the service, and the category this service falls into.
- Determine whether your service runs multiple instances.
- Identify any dependency relationships between this service and any other services.
- If a script is required to start and stop the process, create the script and place it in a local directory such as `/usr/local/svc/method`.
- Create a service manifest file for your service. This file describes the service and any dependency relationships. Service manifests are pulled into the repository either by using the `svccfg` command or at boot time.
- Incorporate the script into the SMF using the `svccfg` utility.

The following displays an example:

```
# vi /usr/local/svc/method/newservice
#!/sbin/sh
#
# Copyright 2004 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# ident "@(#)newservice 1.14      04/08/30 SMI"

case "$1" in
'start')
    /usr/bin/newservice &
;;
'stop')
    /usr/bin/pkill -x -u 0 newservice
;;
*)
    echo "Usage: $0 { start | stop }"
;;
esac
exit 0

# chmod 544 /usr/local/svc/method/newservice

# cd /var/svc/manifest/site
# vi newservice.xml
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM
"/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
    Copyright 2004 Sun Microsystems, Inc. All rights reserved.
    Use is subject to license terms.

    ident      "@(#)newservice.xml      1.2      04/09/13 SMI"
-->

<service_bundle type='manifest' name='OPTnew:newservice'>

<service
    name='site/newservice'
    type='service'
    version='1'>

    <single_instance/>
```

Identifying the Phases of the Boot Process

```
<dependency
    name='usr'
    type='service'
    grouping='require_all'
    restart_on='none'>
    <service_fmri value='svc:/system/filesystem/local' />
</dependency>

<dependent
    name='newservice'
    grouping='require_all'
    restart_on='none'>
    <service_fmri value='svc:/milestone/multi-user' />
</dependent>

<exec_method
    type='method'
    name='start'
    exec='/lib/svc/method/newservice start'
    timeout_seconds='30' />

<exec_method
    type='method'
    name='stop'
    exec='/lib/svc/method/newservice stop'
    timeout_seconds='30' />

<property_group name='startd' type='framework'>
    <propval name='duration' type='astring' value='transient' />
</property_group>

<instance name='default' enabled='true' />

<stability value='Unstable' />

<template>
    <common_name>
        <loctext xml:lang='C'>
            New service
        </loctext>
    </common_name>
</template>
</service>
```

```
</service_bundle>
```

The following describes the entries in the file:

- Standard header.

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM
"/usr/share/lib/xml/dtd/service_bundle.dtd.1">
```

- Comment section.

```
<!--
```

```
Copyright 2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
```

```
ident      "@(#)newservice.xml      1.2      04/09/13 SMI"
-->
```

- The name of the service. The type (manifest) indicates a simple service rather than a milestone, the package providing the service, and the service name.

```
<service_bundle type='manifest' name='OPTnew:newservice'>
```

- Service category, type, name, and version.

```
<service
```

```
    name='site/newservice'
    type='service'
    version='1'>
```

- Whether multiple instances of the service will run.

```
<single_instance/>
```

- The service model to use. The entry shows that the service will be started by svc.startd. Transient services are started once and not restarted.

```
<property_group name='startd' type='framework'>
```

```
    <propval name='duration' type='astring' value='transient'
```

```
/>
```

```
    </property_group>
```

- How the service is started and stopped.

```
<exec_method
```

```
    type='method'
    name='start'
    exec='/lib/svc/method/newservice start'
```

```
        timeout_seconds='30' />

<exec_method
    type='method'
    name='stop'
    exec='/lib/svc/method/newservice stop'
    timeout_seconds='30' />
    • Define any dependencies for this service. The first entry states that
      the newservice requires the filesystem/local service.

<dependency
    name='usr'
    type='service'
    grouping='require_all'
    restart_on='none'>
    <service_fmri value='svc:/system/filesystem/local' />
</dependency>
    • The second entry makes sure that your service is associated with the
      multi-user milestone and that the multi-user milestone requires this
      service.

<dependent
    name='newservice'
    grouping='require_all'
    restart_on='none'>
    <service_fmri value='svc:/milestone/multi-user' />
</dependent>

    • Creating the instance.

<instance name='default' enabled='true' />

    <stability value='Unstable' />
    • Creating information to describe the service.

<template>
    <common_name>
        <loctext xml:lang='C'>
            New service
        </loctext>
    </common_name>
</template>
```

The new service (newservice) now needs to be imported into SMF.

This is done by running the svccfg utility:

```
# svccfg import /var svc/manifest/site/newservice.xml
```

After the service has been imported into SMF it should be visible using the svcs command.

```
# svcs newservice
STATE          STIME      FMRI
online         8:43:45  svc:/site/newservice:default
#
#
```

It should also be possible to manipulate the service using svcadm.

```
# svcadm -v disable site/newservice
site/newservice disabled.
# svcs newservice
STATE          STIME      FMRI
disabled       9:11:38  svc:/site/newservice:default
# svcadm -v enable site/newservice
site/newservice enabled.
# svcs newservice
STATE          STIME      FMRI
online         9:11:54  svc:/site/newservice:default
#
#
```

Finally, you can observe that the multiuser milestone requires the newservice in order to complete its requirements.

```
# svcs -d milestone/multi-user:default
STATE          STIME      FMRI
disabled       8:43:16  svc:/platform/sun4u/sf880drd:default
online         8:43:16  svc:/milestone/name-services:default
online         8:43:33  svc:/system/rmtmpfiles:default
online         8:43:42  svc:/network/rpc/bind:default
online         8:43:46  svc:/milestone/single-user:default
online         8:43:46  svc:/system/utmp:default
online         8:43:47  svc:/system/system-log:default
online         8:43:47  svc:/system/system-log:default
online         8:43:49  svc:/system/filesystem/local:default
online         8:44:01  svc:/system/mdmonitor:default
online         9:11:54  svc:/site/newservice:default
#
#
```

Adding Scripts to Start and Stop Services Not Managed by SMF

To add run control scripts to start and stop a service not managed by SMF, create the script in the /etc/init.d directory and create links in the appropriate /etc/rc#.d directory for the run level in which the service is to be started and stopped.

The following procedure describes how to add a run control script:

1. Create the script in the /etc/init.d directory.

```
# vi /etc/init.d/filename
# chmod 744 /etc/init.d/filename
# chgrp sys /etc/init.d/filename
```

2. Create links to the appropriate /etc/init.d directory.

```
# cd /etc/init.d
# ln filename /etc/rc#.d/S##filename
# ln filename /etc/rc#.d/K##filename
```

For instance, you might link a file in /etc/init.d called database to a file called /etc/rc3.d/S99database, with a corresponding stop script called /etc/rc3.d/K99database.

3. Use the ls command to verify that the script has links in the appropriate directories.

```
# ls -li /etc/init.d/filename
# ls -li /etc/rc#.d/S##filename
# ls -li /etc/rc#.d/K##filename
```

4. Test the *filename* by performing the following commands:

```
# /etc/init.d/filename start
```

Figure 9-8 shows the run-level transitions that occur during the process of a system boot or shut down.

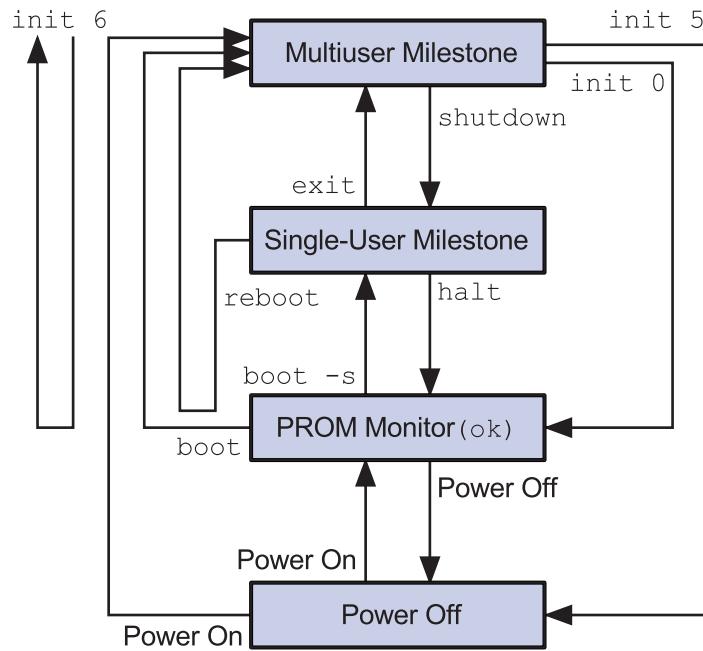


Figure 9-8 Run-Level Transitions

Performing System Shutdown Procedures

You can shut down the Solaris OS to perform administration tasks or maintenance activities if you are anticipating a power outage or if you need to move the system to a new location.

The Solaris OS requires a clean and orderly shutdown, which stops processes, writes data in memory to disks, and unmounts file systems.

Of course, the type of work you need to do while the system is shut down determines how the system is shut down and which command you use.

The following describes the different types of system shutdowns.

- Shut down the system to single-user mode
- Shut down the system to stop the Solaris OS, and display the `ok` prompt
- Shut down the system and turn off power
- Shut down the system and automatically reboot to multiuser mode

The commands available to the root user for doing these types of system shutdown procedures include:

- `/sbin/init` (using run levels S, 0, 1, 5, or 6)
- `/usr/sbin/shutdown` (using run levels S, 0, 1, 5, or 6)

Note – The `init` command accepts more arguments than those listed here. These arguments are not listed here because they are outside of the topic of system shutdown procedures.



The /usr/sbin/init Command

You use the `init` command to shut down, power off, or reboot a system in a clean and orderly manner. It informs the `svc.startd` daemon of the change in `runlevel`. `svc.startd`, achieves the appropriate milestone, and ultimately executes the `rc0` kill scripts. However, this command does not warn logged-in users that the system is being shut down, and there is no grace period.

To shut down the system to single-user mode, use either run level S or 1.

```
# init s
```

To shut down the system to stop the Solaris OS and display the `ok` prompt, perform the command:

```
# init 0
```

To shut down the system and turn its power off, perform the command:

```
# init 5
```

To shut down the system and then reboot to multiuser mode, perform the command:

```
# init 6
```

The /usr/sbin/shutdown Command

The `shutdown` command is a script that invokes the `init` daemon to shut down, power off, or reboot the system. It executes the `rc0` kill scripts to shut down processes and applications gracefully. But unlike the `init` command, the `shutdown` command does the following:

- Notifies all logged-in users that the system is being shut down
- Delays the shutdown for 60 seconds by default
- Enables you to include an optional descriptive message to inform your users of what will transpire

The command format for the `shutdown` command is:

```
shutdown -y -g grace-period -i init-state
optional message
```

Performing System Shutdown Procedures

The **-y** option pre-answers the final shutdown confirmation question so that the command runs without your intervention.

The **-g grace-period** allows you to change the number of seconds from the 60-second default.

The **-i init-state** specifies the run level that the system is to attain. By default, system state S is used.

 **Note** – If the shutdown command displays the error message: “shutdown: ‘i’ - unknown flag,” it indicates that the shell has located and executed the /usr/ucb/shutdown command. Reissue the command using its full path (for example, /usr/sbin/shutdown), or set the PATH variable to ensure /usr/sbin comes before /usr/ucb.

To shut down the system to single-user mode, enter the shutdown command without options.

```
# shutdown
```

To shut down the system to stop the Solaris OS, and display the ok prompt, perform the command:

```
# shutdown -i0
```

To shut down the system and turn off its power automatically, perform the command:

```
# shutdown -i5
```

To shut down the system and then reboot to multiuser mode, perform the command:

```
# shutdown -i6
```

The **-i** option can be used with other command options. For example, to shut down the system and then reboot to multiuser mode, answer yes to the questions presented, provide a grace period of two minutes, and provide a message to the users, perform the command:

```
# shutdown -y -g120 -i6 "The system is being rebooted"
```

“Ungraceful” Shutdown Commands

The following commands perform an immediate system shutdown. They do not execute the `rc0` kill scripts. They do not notify logged-in users, and there is no grace period.

```
# halt
# poweroff
# reboot
```



Caution – These commands should be used with extreme caution, and only when there is no other alternative.

Setting the Default Boot-time Milestone

The `svcadm` command can be used to control the milestone the `svc.startd` daemon meets on boot. The default milestone if one is not specified is “all” which is an abstract milestone where all system services are started.

To ensure that the `svc.startd` daemon meets the requirements of the multi-user-server milestone on the next reboot, use the following command before rebooting:

```
# svcadm -v milestone -d multi-user-server:default
```

Valid options for default boot level using the `svcadm` command include the following:

- all
- none
- svc:/milestone/single-user:default
- svc:/milestone/multi-user:default
- svc:/milestone/multi-user-server:default

The Service Repository Database

A database is saved in the `/etc/svc` directory that contains details of the available services and their settings. The `/lib/svc/bin/restore_repository` utility can be used to repair or restore a corrupt repository.

To see how the repository database is used, perform the following steps:

1. `cd /lib/svc/bin`
2. `./restore_repository`

Repository Restore utility

See <http://sun.com/msg/SMF-8000-MY> for more information on the use of this script to restore backup copies of the `smf(5)` repository.

If there are any problems which need human intervention, this script will give instructions and then exit back to your shell.

Note that upon full completion of this script, the system will be rebooted using `reboot(1M)`, which will interrupt any active services.

The following backups of `/etc/svc/repository.db` exist, from oldest to newest:

```
manifest_import-20050221_112255
manifest_import-20050221_144358
boot-20050223_100423
boot-20050223_211258
boot-20050224_095929
boot-20050225_134532
```

The backups are named based on their type and the time what they were taken.

Backups beginning with "boot" are made before the first change is made to the repository after system boot. Backups beginning with "manifest_import" are made after `svc:/system/manifest-import:default` finishes its processing.

The time of backup is given in `YYYYMMDD_HHMMSS` format.

Please enter one of:

- 1) boot, for the most recent post-boot backup
- 2) manifest_import, for the most recent manifest_import backup.
- 3) a specific backup repository from the above list
- 4) -seed-, the initial starting repository. (All customizations will be lost.)
- 5) -quit-, to cancel.

Enter response [boot]: manifest_import-20050221_144358

After confirmation, the following steps will be taken:

```
svc.startd(1M) and svc.configd(1M) will be quiesced, if running.  
/etc/svc/repository.db  
  -- renamed --> /etc/svc/repository.db_old_20050225_163816  
/etc/svc/repository-manifest_import-20050221_144358  
  -- copied --> /etc/svc/repository.db  
and the system will be rebooted with reboot(1M).
```

Proceed [yes/no]? no

Exiting...

#

In the above example, the alternate repository.db file naming convention is YYYYMMDD_number, therefore the repository.db_old_20050225_163816 file was saved on February 25, 2005.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Controlling the Boot Process (Level 1)

In this exercise, you create a new startup script, make changes in the /etc/system file, and observe their effects.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Your instructor should provide you with instructions on how to obtain the following three files used during this exercise.

- A script called banner-smf
- A script called banner-rc
- A file called banner-smf.xml

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, lab files are found in the /export/home/student directory.

Tasks

Complete the following tasks:



Note – Before copying any files out of your student directory, make sure that all files have executable permission set.

- Create the `/usr/local/svc/method` directory.
- Copy the `banner-smf` script to the `/usr/local/svc/method` directory.
- Copy the `banner-smf.xml` file to the `/var svc/manifest/site` directory.
- Import the new service into the SMF repository.
- Verify that the `banner-smf` service can be enabled and disabled.
- Reboot the system and verify that the `banner-smf` service is started.
- Remove the `banner-smf` service from the system.
- In the `/etc/rc2.d` directory, create a hard link to the `/etc/init.d/banner-rc` file, called `S22banner`. In the `/etc/rcS.d` directory, create a hard link to the `/etc/init.d/banner-rc` file called `K99banner`.
(Steps 1–5 in the Level 2 lab)
- Reboot the system, and verify that `S22banner` runs. Shut down the system to run level S, and verify that `K99banner` runs. Change back to run level 3.
- Make a backup copy of the `/etc/system` file. Check if any instances of the `st` driver are loaded. Modify the `/etc/system` file to force-load the `st` driver. Reboot the system, and verify that `st` driver instances are loaded.
(Steps 6–10 in the Level 2 lab)
- Edit the `/etc/system` file to exclude the boot disk driver for your system (either `dad` or `sd`). Shut down the system to run level 0, and attempt to boot it. Make note of what happens. Interactively boot your system, and return it to an operational state.
(Steps 11–14 in the Level 2 lab)

Exercise: Controlling the Boot Process (Level 2)

In this exercise, you create a new startup script, make changes in the /etc/system file, and observe their effects.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Your instructor should provide you with instructions on how to obtain the following three files used during this exercise.

- A script called banner-smf
- A script called banner-rc
- A file called banner-smf.xml

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, lab files are found in the /export/home/student directory.

Task Summary

In this exercise, you accomplish the following:

- Create the /usr/local/svc/method directory.
- Copy the banner-smf script to the /usr/local/svc/method directory.
- Copy the banner-smf.xml file to the /var/svc/manifest/site directory.

Exercise: Controlling the Boot Process (Level 2)

- Import the new service into the SMF repository.
- Verify that the banner-smf service can be enabled and disabled.
- Reboot the system and verify that the banner service is started.
- Remove the banner-smf service from the system.
- In the /etc/rc2.d directory, create a hard link to the /etc/init.d/banner-rc file, called S22banner. In the /etc/rcS.d directory, create a hard link to the /etc/init.d/banner-rc file called K99banner.
- Reboot the system, and verify that S22banner runs. Shut down the system to run level S, and verify that K99banner runs. Change back to run level 3.
- Make a backup copy of the /etc/system file. Check if any instances of the st driver are loaded. Modify the /etc/system file to force-load the st driver. Reboot the system, and verify that st driver instances are loaded.
- Edit the /etc/system file to exclude the boot disk driver for your system (either dad or sd). Shut down the system to run level 0, and attempt to boot it. Make note of what happens. Boot the system using the -a option of the boot command. Use your backup of the /etc/system file as required. Replace the /etc/system file with your backup when finished, and reboot the system.

Tasks



Note – Before copying any files out of your student directory, make sure that all files have executable permission set.

Complete the following steps:

1. Copy the banner-smf file to the /usr/local/svc/method directory.
2. Copy the banner-smf.xml file to the /var/svc/manifest/site directory.
3. Import the new banner-smf service into the SMF repository.
4. Verify that the service is now part of SMF.
5. Check that the new service (banner-smf) works. Open a console window to view the output of the banner service.
6. Run the command to disable the service and view the output in the console window.

7. Run the command to enable the service and view the output in the console window.
8. Reboot the system and observe the output displayed as the banner service is started.
9. Remove the banner-smf service from the system.
10. Log in as the root user, and open a terminal window. Change the directory to /etc/init.d. Make sure that the banner-rc script your instructor provided you with is present and executable.
11. Verify that the script runs with both the start and stop arguments.
12. Change the directory to /etc/rc2.d. Create a hard link called S22banner that points to the same data as the /etc/init.d/banner-rc file.
13. Change the directory to the /etc/rcS.d directory. Create a hard link called K99banner that points to the /etc/init.d/banner-rc file.
14. Reboot the system, and watch for the output of the script you just installed.
Does the startup message from S22banner appear?
15. Log in as the root user, and open a terminal window. Use the init command to change to run level S.
Does the shutdown message from K99banner appear?
16. Type the password for the root user to log in at the command line. Change to run level 3.
17. Log in as the root user, and open a terminal window. Change the directory to /etc.
18. Make a backup copy of the /etc/system file, and name the backup file system.orig.

Exercise: Controlling the Boot Process (Level 2)

19. If your system uses a SCSI tape device, perform the following:
 - a. Log in as the `root` user, and open a terminal window. Use the `prtconf` command to list instances of the `st` driver currently loaded.
How many instances are reported?
 - b. Edit the `/etc/system` file so that it includes the following line:
`forceload: drv/st`
Then reboot the system.
 - c. Log in as `root`, and open a terminal window. Again list instances of the `st` driver currently loaded.
How many instances are reported?
20. Edit the `/etc/system` file so that it excludes the main disk driver for your system.
On systems using SCSI disks, add the following:
`exclude: drv/sd`
On systems using IDE disks, add the following:
`exclude: drv/dad`
21. Shut down the system to run level 0, and then attempt to boot it again.
What happened?
22. Use the `boot -a` command to boot the system, and supply the name of your backup file called `etc/system.orig` (note there is *not* a leading slash to the `etc`). Press the Return key to accept the default values for all other boot parameters. For example:

```
ok boot -a
Enter filename [kernel/sparcv9/unix]: <Return>
Enter default directory for modules [/platform...]: <Return>
Name of system file [etc/system]: etc/system.orig
root filesystem type [ufs]: <Return>
Enter physical name of root device [/...]: <Return>
```

23. Log in as the `root` user, and open a terminal window. Copy the `/etc/system.orig` file to the `/etc/system` file. Reboot the system.

Exercise: Controlling the Boot Process (Level 3)

In this exercise, you create a new startup script, make changes in the /etc/system file, and observe their effects.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Your instructor should provide you with instructions on how to obtain the following three files that will be used during this exercise.

- A script called banner-smf
- A script called banner-rc
- A file called banner-smf.xml

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, lab files are found in the /export/home/student directory.

Task Summary

In this exercise, you accomplish the following:

- Create the /usr/local/svc/method directory.
- Copy the banner-smf script to the /usr/local/svc/method directory.
- Copy the banner-smf.xml file to the /var/svc/manifest/site directory.

- Import the new service into the SMF repository.
- Verify that the banner-smf service can be enabled and disabled.
- Reboot the system and verify that the banner service is started.
- Remove the banner-smf service from the system.
- In the /etc/rc2.d directory, create a hard link to the /etc/init.d/banner-rc file, called S22banner. In the /etc/rcS.d directory, create a hard link to the /etc/init.d/banner-rc file called K99banner.
- Reboot the system, and verify that S22banner runs. Shut down the system to run level S, and verify that K99banner runs. Change back to run level 3.
- Make a backup copy of the /etc/system file. Check if any instances of the st driver are loaded. Modify the /etc/system file to force-load the st driver. Reboot the system, and verify that st driver instances are loaded.
- Edit the /etc/system file to exclude the boot disk driver for your system (either dad or sd). Shut down the system to run level 0, and attempt to boot it. Make note of what happens. Boot the system using the -a option of the boot command. Use your backup of the /etc/system file as required. Replace the /etc/system file with your backup when finished, and reboot the system.

Tasks and Solutions



Note – Before copying any files out of your student directory, make sure that all files have executable permission set.

Complete the following steps:

1. Create the /usr/local/svc/method directory.

```
# mkdir -p /usr/local/svc/method
```
2. Copy the banner-smf file to the /usr/local/svc/method directory.

```
# cp banner-smf /usr/local/svc/method
```
3. Copy the banner-smf.xml file to the /var/svc/manifest/site directory.

```
# cp banner-smf.xml /var/svc/manifest/site
```
4. Import the new banner service into the SMF repository.

```
# svccfg import /var svc/manifest/site/banner-smf.xml
```

5. Verify that the service is now part of SMF.

```
# svcs site/banner-smf
```

STATE	STIME	FMRI
online	9:11:54	svc:/site/banner-smf:default

```
#
```

6. Check that the new service banner-smf works. Open a console window used to view the output of the banner service.

```
# dtterm -C &
```

7. Run the command to disable the service and view the output in the console window.

```
# svcadm -v disable site/banner-smf
```

```
# svcs site/banner-smf
```

STATE	STIME	FMRI
disabled	9:11:54	svc:/site/banner-smf:default

8. Run the command to enable the service and view the output in the console window.

```
# svcadm -v enable site/banner-smf
```

```
svc:/site/banner-smf:default enabled.
```

```
# svcs site/banner-smf
```

STATE	STIME	FMRI
online	9:11:54	svc:/site/banner-smf:default

9. Shut down the system and boot it with the -m verbose option. Observe the output displayed as the banner service is started.

```
# init 0
```

```
ok boot -m verbose
```

Does the startup message from banner-smf appear?

Yes.

10. Remove the banner-smf service from the system.

```
# svcadm -v disable site/banner-smf
```

```
# svccfg delete -f svc:/site/banner-smf:default
```

```
# rm /var svc/manifest/site/banner-smf.xml
```

11. Make sure that the banner-rc script your instructor provided you is present and executable in /etc/init.d.

```
# cp /export/home/student/banner-rc /etc/init.d
```

```
# cd /etc/init.d
```

```
# chmod 755 banner-rc
```

```
# ls -l banner-rc
```

Exercise: Controlling the Boot Process (Level 3)

12. Make the banner script executable, and verify that it runs with both the start and stop arguments.

```
# ./banner-rc start  
# ./banner-rc stop
```

13. Change the directory to the /etc/rc2.d directory. Create a hard link called S22banner that points to the same data as the /etc/init.d/banner file.

```
# cd /etc/rc2.d  
# ln /etc/init.d/banner-rc S22banner
```

14. Change the directory to the /etc/rcS.d directory. Create a hard link called K99banner that points to the same data as the /etc/init.d/banner file.

```
# cd /etc/rcS.d  
# ln /etc/init.d/banner-rc K99banner
```

15. Reboot the system, and watch for the output of the script you just installed.

```
# init 6
```

Does the startup message from S22banner appear?

Yes.

16. Log in as the root user, and open a terminal window. Use the init command to change to run level S.

```
# init s
```

Does the shutdown message from K99banner appear?

Yes.

17. Type the password for the root user to log in at the command line. Change to run level 3 by typing Control-D.

```
# cntl-d
```

18. Log in as the root user, and open a terminal window. Change the directory to /etc.

```
# cd /etc
```

19. Make a backup copy of the /etc/system file, and name the backup file system.orig.

```
# cp system system.orig
```

20. If your system uses a SCSI tape device, perform the following:

- a. Log in as the root user, and open a terminal window. Use the prtconf command to list instances of the st driver currently loaded.

```
# prtconf | grep "st, instance"
```

How many instances are reported?

None

- b. Edit the /etc/system file so that it includes the following line:

```
forceload: drv/st
```

Then reboot the system.

```
# init 6
```

- c. Log in as root, and open a terminal window. Again list instances of the st driver currently loaded.

```
# prtconf | grep "st, instance"
```

How many instances are reported?

The number varies depending on how many SCSI controllers are present. You should see Instances 0 through 6 for a system with one controller.

21. Edit the /etc/system file so that it excludes the main disk driver for your system.

On systems using SCSI disks, add the following:

```
exclude: drv/sd
```

On systems using IDE disks, add the following:

```
exclude: drv/dad
```

22. Shut down the system to run level 0, and then attempt to boot it again.

```
# shutdown -y -i0 -g0
```

(shutdown messages)

```
ok boot
```

What happened?

The system is unable to boot. Excluding this driver prevents you from using the boot disk so long as you use the same /etc/system file. You must boot using the -a option to be able to supply an alternative file for the /etc/system file.

Exercise: Controlling the Boot Process (Level 3)

23. Use the `boot -a` command to boot the system, and supply the name of your backup file called `etc/system.orig` (Note there is *not* a leading slash to the `etc`). Press Return to accept the default values for all other boot parameters. For example:

```
ok boot -a  
Enter filename [kernel/sparcv9/unix]: <Return>  
Enter default directory for modules [/platform...]: <Return>  
Name of system file [etc/system]: etc/system.orig  
root filesystem type [ufs]: <Return>  
Enter physical name of root device [/...]: <Return>
```

24. Log in as the root user, and open a terminal window. Copy the `/etc/system.orig` file to the `/etc/system` file. Reboot the system.

```
# cd /etc  
# cp system.orig system  
# init 6
```

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 10

Performing User Administration

Objectives

Upon completion of this module, you should be able to:

- Describe user administration fundamentals
- Manage user accounts
- Manage initialization files

The course map in Figure 10-1 shows how this module fits into the current instructional goal.

Performing User and Security Administration

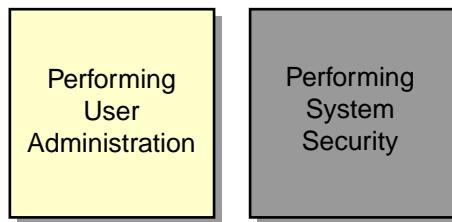


Figure 10-1 Course Map

Introducing User Administration

An important system administration task is setting up user accounts for each user who requires system access. Each user needs a unique account name, a user identification (UID) number, a home directory, and a login shell. You also have to determine which groups a user may access.

Main Components of a User Account

The following is a list of the main components of a user account:

- User name – A unique name that a user enters to log in to a system. The user name is also called the login name.
- Password – A combination of up to 256 letters, numbers, or special characters that a user enters with the login name to gain access to a system.
- UID number – A user account's unique numerical identification within the system.
- Group identification (GID) number – A unique numerical identification of the group to which the user belongs.

Note – You can add a user to predefined groups listed in the /etc/group file.

- 
- Comment – Information that identifies the user. A comment generally contains the full name of the user and optional information, such as a phone number or a location.
 - User's home directory – A directory into which the user is placed after login. The directory is provided to the user to store and create files.
 - User's login shell – The user's work environment is set up by the initialization files that are defined by the user's login shell.

System Files That Store User Account Information

The Solaris 10 OS stores user account and group entry information in the following system files:

- /etc/passwd
- /etc/shadow
- /etc/group

Authorized system users have login account entries in the /etc/passwd file.

The /etc/shadow file is a separate file that contains the encrypted passwords. To further control user passwords, you can enforce password aging. This information is also maintained in the /etc/shadow file.

The /etc/group file defines the default system group entries. You use this file to create new group entries or modify existing group entries on the system.

The /etc/passwd File

Due to the critical nature of the /etc/passwd file, you should refrain from editing this file directly. Instead, you should use the Solaris™ Management Console or command-line tools to maintain the file.

The following is an example of an /etc/passwd file that contains the default system account entries.

```
root:x:0:0:Super-User:/sbin/sh
daemon:x:1:1:::
bin:x:2:2::/usr/bin:
sys:x:3:3:::
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webservd:x:80:80:WebServer Reserved UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002>No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
```

Each entry in the /etc/passwd file contains seven fields. A colon separates each field. The following is the format for an entry:

loginID:x:UID:GID:comment :home_directory:login_shell

Table 10-1 defines the requirements for each of the seven fields.

Table 10-1 Fields in the /etc/passwd File

Field	Description
<i>loginID</i>	<p>Represents the user's login name. It should be unique to each user. The field should contain a string of no more than eight letters (A-Z, a-z) and numbers (0-9). The first character should be a letter, and at least one character should be lowercase.</p> <p>Note – Even though some programs allow a maximum of 32 characters, as well as user names that contain periods (.), underscores (_), and hyphens (-), this practice is not recommended and might cause problems with other programs.</p>
<i>x</i>	Represents a placeholder for the user's encrypted password, which is kept in the /etc/shadow file.
<i>UID</i>	<p>Contains the UID number used by the system to identify the user. UID numbers for users range from 100 to 60000. Values 0 through 99 are reserved for system accounts. UID number 60001 is reserved for the nobody account. UID number 60002 is reserved for the noaccess account. While duplicate UID numbers are allowed, they should be avoided unless absolutely required by a program.</p> <p>Note – The maximum value for a UID is 2147483647. However, the UIDs over 60000 do not have full utility and are incompatible with some Solaris OS features. Avoid using UIDs over 60000 so as to be compatible with earlier versions of the operating system.</p>

Table 10-1 Fields in the /etc/passwd File (Continued)

Field	Description
<i>GID</i>	Contains the GID number used by the system to identify the user's primary group. GID numbers for users range from 100 to 60000. (Those between 0 and 99 are reserved for system accounts.)
<i>comment</i>	Typically contains the user's full name.
<i>home_directory</i>	Contains the full path name to the user's home directory.
<i>login_shell</i>	Defines the user's login shell. There are six possible login shells in the Solaris OS: the Bourne shell, the Korn shell, the C shell, the Z shell, the BASH shell, and the TC shell.

Table 10-2 shows the default system account data for entries in the /etc/passwd file.

Table 10-2 Default System Account Entries

User Name	User ID	Description
root	0	The root account that has access to the entire system. It has almost no restrictions and overrides all other logins, protections, and permissions.
daemon	1	The system daemon account that is associated with routine system tasks.
bin	2	The administrative daemon account that is associated with running system binary files.
sys	3	The administrative daemon account that is associated with system logging or updating files in temporary directories.
adm	4	The administrative daemon account that is associated with system logging.
lp	71	The line printer (lp) daemon account.
uucp	5	The daemon account associated with UNIX®-to-UNIX Copy Protocol (UUCP) functions.
nuucp	6	The account that is used by remote systems to log in to the host and start file transfers using uucp.

Table 10-2 Default System Account Entries (Continued)

User Name	User ID	Description
smmssp	25	The sendmail message submission daemon account.
listen	37	The network listener daemon account.
gdm	50	Gnome Display Manager daemon.
webservd	80	Account reserved for WebServer access.
nobody	60001	The anonymous user account that is assigned by a Network File System (NFS) server when an unauthorized root user makes a request. The nobody user account is assigned to software processes that do not need any special permissions.
noaccess	60002	The account assigned to a user or a process that needs access to a system through some application instead of through a system login procedure.
nobody4	65534	The anonymous user account that is the SunOS™ 4.X software version of the nobody account



Note – The nobody account secures NFS resources. When a user is logged in as root on an NFS client and attempts to access a remote file resource, the UID number changes from 0 to the UID of nobody (60001)

The /etc/shadow File

Due to the critical nature of the /etc/shadow file, you should refrain from editing it directly. Instead, maintain the fields of the file by using the Solaris Management Console or command-line tools. Only the root user can read the /etc/shadow file.

The following is an example /etc/shadow file that contains initial system account entries.

```
root:rJrdhjNWQQHoY:6445:::::::  
daemon:NP:6445:::::::  
bin:NP:6445:::::::  
sys:NP:6445:::::::  
adm:NP:6445:::::::  
lp:NP:6445:::::::
```

```

uucp:NP:6445::::::
nuucp:NP:6445::::::
smmsp:NP:6445::::::
listen:*LK*:::::::
gdm:*LK*:::::::
webservd:*LK*:::::::
nobody:*LK*:6445::::::
noaccess:*LK*:6445::::::
nobody4:*LK*:6445::::::

```

Each entry in the /etc/shadow file contains nine fields. A colon separates each field.

Following is the format of an entry:

loginID:password:lastchg:min:max:warn:inactive:expire:

Table 10-3 defines the requirements for each of the eight fields.

Table 10-3 Fields in the /etc/shadow File

Field	Description
<i>loginID</i>	The user's login name.
<i>password</i>	A 13-character encrypted password. The string *LK* indicates a locked account, and the string NP indicates no valid password. Passwords must be constructed to meet the following requirements: Each password must be at least six characters and contain at least two alphabetic characters and at least one numeric or special character. It cannot be the same as the login ID or the reverse of the login ID.
<i>lastchg</i>	The number of days between January 1, 1970, and the last password modification date.
<i>min</i>	The minimum number of days required between password changes.
<i>max</i>	The maximum number of days the password is valid before the user is prompted to enter a new password at login.

Table 10-3 Fields in the /etc/shadow File (Continued)

<i>warn</i>	The number of days the user is warned before the password expires.
<i>inactive</i>	The number of inactive days allowed for the user before the user's account is locked.
<i>expire</i>	The date (given as number of days since January 1, 1970) when the user account expires. After the date is exceeded, the user can no longer log in.
<i>flag</i>	To track failed logins. The count is in low order four bits. The remainder is reserved for future use, set to zero.

The /etc/group File

Each user belongs to a group that is referred to as the user's primary group. The GID number, located in the user's account entry within the /etc/passwd file, specifies the user's primary group.

Each user can also belong to up to 15 additional groups, known as secondary groups. In the /etc/group file, you can add users to group entries, thus establishing the user's secondary group affiliations.

The following is an example of the default entries in an /etc/group file:

```
root::0:  
other::1:root  
bin::2:root,daemon  
sys::3:root,bin,adm  
adm::4:root,daemon  
uucp::5:root  
mail::6:root  
tty::7:root,adm  
lp::8:root,adm  
nuucp::9:root  
staff::10:  
daemon::12:root  
sysadmin::14:  
smmsp::25:
```

```

gdm:::50:
webservd:::80:
nobody:::60001:
noaccess:::60002:
nogroup::65534::
```

Each line entry in the /etc/group file contains four fields. A colon character separates each field. The following is the format for an entry:

groupname:group-password:GID:username-list

Table 10-4 defines the requirements for each of the four fields.

Table 10-4 Fields in the /etc/group File

Field	Description
<i>groupname</i>	Contains the name assigned to the group. Group names contain up to a maximum of eight characters.
<i>group-password</i>	Usually contains an empty field or an asterisk. This is a relic of earlier versions of UNIX. Caution – A group-password is a security hole because it might allow an unauthorized user who is not a member of the group but who knows the group password, to enter the group.
	Note – The newgrp command changes a user's primary group association within the shell environment from which it is executed. If this new, active group has a password and the user is not a listed member in that group, the user must enter the password before the newgrp command can continue.
<i>GID</i>	Contains the group's GID number. It is unique on the local system and should be unique across the organization. Numbers 0 to 99, 60001, 60002 and 65534 are reserved for system group entries. User-defined groups range from 100 to 60000.

Table 10-4 Fields in the /etc/group File (Continued)

Field	Description
<i>username-list</i>	<p>Contains a comma-separated list of user names that represent the user's secondary group memberships. By default, each user can belong to a maximum of 15 secondary groups.</p> <hr/> <p>Note – The maximum number of groups is set by the kernel parameter called <code>ngroups_max</code>. You can set this parameter in the /etc/system file to allow for a maximum of 32 groups. Not all applications will be able to reference group memberships greater than 16. NFS is a notable example.</p>

The /etc/default/passwd File

Set values for the following parameters in the /etc/default/passwd file to control properties for all users' passwords on the system:

- `MAXWEEKS` – Sets the maximum time period (in weeks) that the password is valid.
- `MINWEEKS` – Sets the minimum time period before the password can be changed.
- `PASSLENGTH` – Sets the minimum number of characters for a password. Valid entries are 6, 7, and 8.
- `WARNWEEKS` – Sets the time period prior to a password's expiration to warn the user that the password will expire.

Note – The `WARNWEEKS` value does not exist by default in the /etc/default/passwd file, but it can be added.



The password aging parameters `MAXWEEKS`, `MINWEEKS`, and `WARNWEEKS` are default values. If set in the /etc/shadow file, the parameters in that file override those in the /etc/default/passwd file for individual users.

The Solaris 10 OS release introduces a number of new controls for password management. These controls are configured by setting values in the /etc/default/passwd file. These controls are commented out by default.

- NAMECHECK=NO – Sets the password controls to verify that the user is not using the login name as a component of the password.
- HISTORY=0 – Forces the passwd program to log up to 26 changes to the user's password. This prevents the user from reusing the same password for 26 changes. If the HISTORY value is set to another number other than zero (0), and then set back to zero, it causes the password log for a user to be removed on the next password change.
- DICTIONLIST= – Causes the passwd program to perform dictionary word lookups.
- DICTIONDBDIR=/var/passwd – The location of the dictionary where the generated dictionary databases reside. This directory must be created manually.

Note – To pre-build the dictionary database, refer to the man page for mkpwdict(1M).



Complexity of the password can be controlled using the following parameters:

```
#MINDIFF=3
#MINALPHA=2
#MINNONALPHA=1
#MINUPPER=0
#MINLOWER=0
#MAXREPEATS=0
#MINSPECIAL=0
#MINDIGIT=0
#WHITESPACE=YES
```

By default, all of the above parameters are commented out.



Note – By forcing greater complexity of password structure, you may inadvertently cause the users to write down their passwords as they may be too difficult for the user to remember. When setting a password change policy, you must not underestimate the problems that too much complexity may cause.

Password Management

The Solaris 10 OS has new security enhancements. The `pam_unix_auth` module implements account locking for local users. Account locking is enabled by the `LOCK_AFTER_RETRIES` tunable parameter in `/etc/security/policy.conf` and the `lock_after-retries` key in `/etc/user_attr`.

The `LOCK_AFTER_RETRIES=YES|NO` parameter specifies whether a local account is locked after the number of failed login attempts for a user is equal to, or exceeds the allowed number of retries. The number of retries is defined by `RETRIES` in `/etc/default/login`.

Note – These files are discussed in greater detail in:
SA-202-S10; Advanced Administration for Solaris 10 OS.



The `passwd` command has two new options, `-N` and `-u`. The `-N` option creates a password entry for a non-login account. This option is useful for accounts that should not be logged in to, but must run cron jobs. The `-u` option unlocks a previously locked account. The `passwd -N username` command sets the password field in `/etc/shadow` to `NP` which is an unmatchable password. This effectively disables the account from logging in.

For more information, see the `passwd(1)` man page.

The following example shows how to prevent a user from reusing too many previous passwords.

```
# vi /etc/default/passwd  
(output omitted)
```

Locate the line called `#HISTORY=0`, and remove the comment from the beginning of the line. Modify the number to 3, so the line shows as `HISTORY=3`. Write and quit the file. As a regular user, log in and attempt to change your password a number of times, using different passwords and then one of the previous passwords.

```
# telnet localhost  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
login: testuser  
Password: 123pass
```

```
$ passwd
passwd: Changing password for testuser
Enter existing login password: 123pass
New Password: pass123
Re-enter new Password: pass123
passwd: password successfully changed for testuser
$ passwd
passwd: Changing password for testuser
Enter existing login password: pass123
New Password: 123pass
passwd: Password in history list.
Please try again
New Password: newpas1
Re-enter new Password: newpas1
passwd: password successfully changed for testuser
$
```

By uncommenting the HISTORY= line in the /etc/default/passwd file, prior password history is checked. By changing the value to 3, the number of prior password changes to keep and check when a user changes passwords is set to three.

Managing User Accounts

Each of the following sections present two sets of command-line tools for managing user accounts: the command-line tools used in the Solaris OS versions prior to the Solaris 10 OS, and the new set of command-line tools developed for the Solaris 10 OS.

Introducing Command-Line Tools

The Solaris OS provides these command-line tools, defined as follows:

- `useradd` – Adds a new user account on the local system
- `usermod` – Modifies a user's account on the local system
- `userdel` – Deletes a user's account from the local system
- `groupadd` – Adds a new group entry to the system
- `groupmod` – Modifies a group entry on the system
- `groupdel` – Deletes a group entry from the system

In addition to these standard command-line tools, the Solaris 9 and 10 OS has a set of command-line tools that accomplish the same tasks. They are the `smuser` and `smgroup` commands.

The `smuser` command enables you to manage one or more users on the system with the following set of subcommands:

- `add` – Adds a new user account
- `modify` – Modifies a user's account
- `delete` – Deletes a user's account
- `list` – Lists one or more user entries

The `smuser` and `smgroup` commands interact with naming services, can use autohome functionality, and are better suited for remote management.

Note – The `smuser` and `smgroup` commands are the command-line interface equivalent to the Solaris Management Console range of operation, and allow you to perform Solaris Management Console actions in scripts. Therefore, the `smuser` and `smgroup` commands have numerous subcommands and options designed to function across domains and multiple systems. This module describes only the basic commands.



The `smgroup` command enables you to manage one or more groups on the system with the following set of subcommands:

- `add` – Adds a new group entry
- `modify` – Modifies a group entry
- `delete` – Deletes a group entry
- `list` – Lists one or more group entries

Any subcommand to add, modify, list, or delete users with the `smuser` and `smgroup` commands requires authentication with the Solaris Management Console server and requires the initialization of the Solaris Management Console. For example, the following is the command format for the `smuser` command:

```
/usr/sadm/bin/smuser subcommand [auth_args] -- [subcommand_args]
```

The authorization arguments are all optional. However, if you do not specify the authorization argument, the system might prompt you for additional information, such as a password for authentication purposes.

The `--` option separates the subcommand-specific options from the authorization arguments. The `--` option must be entered even if an authorization argument is not specified because it must precede the subcommand arguments.

The subcommand arguments are quite numerous. For a complete listing of the subcommands, refer to the `smuser` man page. It is important to note that descriptions and other arguments that contain white space must be enclosed in double quotation marks.

Creating a User Account

Use the `useradd` or `smuser add` command to add new user accounts to the local system. These commands add an entry for a new user into the `/etc/passwd` and `/etc/shadow` files.

These commands also automatically copy all the initialization files from the `/etc/skel` directory to the user's new home directory.

The useradd Command Format and Options

The following is the command format for the useradd command:

```
useradd [ -u uid ][ -g gid ][ -G gid [,gid,.. ] ]
[ -d dir ][ -m ][ -s shell ][ -c comment ] loginname
```

Table 10-5 shows the options for the useradd command.

Table 10-5 Options for the useradd Command

Option	Definition
-u <i>uid</i>	Sets the UID number for the new user
-g <i>gid</i>	Defines the new user's primary group
-G <i>gid</i>	Defines the new user's secondary group memberships
-d <i>dir</i>	Defines the full path name for the user's home directory
-m	Creates the user's home directory if it does not already exist
-s <i>shell</i>	Defines the full path name for the shell program of the user's login shell
-c <i>comment</i>	Specifies any comment, such as the user's full name and location
<i>loginname</i>	Defines the user's login name for the user account
-D	Displays the defaults that are applied to the useradd command

The following example uses the useradd command to create an account for a user named newuser1. It assigns 100 as the UID number, adds the user to the group other, creates a home directory in the /export/home directory, and sets /bin/ksh as the login shell for the user account.

```
# useradd -u 100 -g other -d /export/home/newuser1 -m -s /bin/ksh -c
"Regular User Account" newuser1
64 blocks
#
```

The useradd command has a preset range of default values. These values can be displayed using the useradd -D command. When this command has been used for the first time, the useradd command generates a file called /var/sadm/defadduser that contains the default values. If the contents of this file are amended, the new contents become the default values for the next time the useradd command is used.

```
# ls -l /usr/sadm/defadduser
/usr/sadm/defadduser: No such file or directory
# useradd -D
group=other,1 project=default,3 basedir=/home
skel=/etc/skel shell=/bin/sh inactive=0
expire= auths= profiles= roles= limitpriv=
defaultpriv= lock_after_retries=
# ls -l /usr/sadm/defadduser
-rw-r--r-- 1 root      root          286 Oct 17 09:04
/usr/sadm/defadduser
# cat /usr/sadm/defadduser
# Default values for useradd.  Changed Sun Oct 17 09:04:27 2004
defgroup=1
defgname=other
defparent=/home
defskel=/etc/skel
defshell=/bin/sh
definact=0
defexpire=
defauthorization=
defrole=
defprofile=
defproj=3
defprojname=default
deflimitpriv=
defdefaultpriv=
deflock_after_retries=
```

User accounts are locked by default when added with the useradd command. This can be verified by viewing the contents of the /etc/shadow file:

```
# grep 'newuser1' /etc/shadow
newuser1:*LK*:12708:::::::
```

By convention, a user's login name is also the user's home directory name.

You use the `passwd` command to create a password for the new account.

```
# passwd newuser1
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for newuser1
```

This password setting can be verified by viewing the contents of the `/etc/shadow` file:

```
# grep 'newuser1' /etc/shadow
newuser1:M0/jolfmSbYio:12708:::::::
```

The `smuser add` Command Format and Options

The following is the command format for the `smuser add` command:

```
smuser add [auth_args] -- [subcommand_args]
```

Table 10-6 shows some of the most common subcommand arguments for the `smuser add` command.

Table 10-6 Subcommand Arguments for the `smuser add` Command

Subcommand Argument	Definition
<code>-c comment</code>	A short description of the login, typically the user's name. This string can be up to 256 characters.
<code>-d directory</code>	Specifies the home directory of the new user and is limited to 1024 characters.
<code>-g group</code>	Specifies the new user's primary group membership.
Subcommand Argument	Definition
<code>-G group</code>	Specifies the user's secondary group membership.
<code>-n login</code>	Specifies the user's login name.

Table 10-6 Subcommand Arguments for the smuser add Command
(Continued)

-s shell	Specifies the full path name of the user's login shell.
-u uid	Specifies the user ID of the user you want to add. If you do not specify this option, the system assigns the next available unique UID greater than 100.
-x autohome=Y/N	Sets the home directory to automount if set to Y.

The following example uses the smuser add command to create an account for a user named newuser2. It designates the login name as newuser2, assigns the UID number 500, adds the user to the group other, creates a home directory in the /export/home directory, and sets /bin/ksh as the login shell for the user account.



Note – The -x autohome=N option to the smuser command adds the user without automounting the user's home directory. See the man page for automount for more information.

```
# /usr/sadm/bin/smuser add -- -n newuser2 -u 500 -g other -d
/export/home/newuser2 -c "Regular User Account 2" -s /bin/ksh -x
autohome=N
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02
was successful.
```

Users are added without a password by default with the smuser command. This can be verified by viewing the appropriate entry in the /etc/shadow file:

```
# grep 'newuser2' /etc/shadow
newuser2::12708:::::::
```

Use the `passwd` command to create a new password for the user.

```
# passwd newuser2
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for newuser2
```

Confirm that the password change has been applied by viewing the entry for that user in the `/etc/shadow` file:

```
# grep 'newuser2' /etc/shadow
newuser2:*LK*:12708:::::::
```

Modifying a User Account

Use the `usermod` or `smuser` modify command to modify a user's login account on the system.

The `usermod` Command Format and Options

The following is the command format for the `usermod` command:

```
usermod [ -u uid [ -o ] ] [ -g gid ] [ -G gid [ , gid . . . ] ]
[ -d dir ] [ -m ] [ -s shell ] [ -c comment ]
[ -l newlogname] loginname
```

In general, the options for the `usermod` command function the same as those for the `useradd` command.

Table 10-7 shows the key options to the `usermod` command.

Table 10-7 Key Options for the `usermod` Command

Option	Definition
<code>-o</code>	Allows a UID to be duplicated.
<code>-m</code>	Moves the user's home directory to the new location specified with the <code>-d</code> option.
<code>-l newlogname</code>	Changes a user's login name for the specified user account.
<code>-f inactive</code>	Sets the number of inactive days that are allowed on a user account. If the account is not logged in to for the specified number of days, it is locked.

Table 10-7 Key Options for the usermod Command (Continued)

<code>-e expire</code>	Sets an expiration date on the user account. Specifies the date (<i>mm/dd/yy</i>) on which a user can no longer log in and access the account. After that date, the account is locked.
<code>loginname</code>	Identifies the user's login name for the current user account.

The following example changes the login name and home directory for newuser1 to usera.

```
# usermod -m -d /export/home/usera -l usera newuser1
```

The smuser modify Command Format and Options

The following is the command format for the smuser modify command:

```
smuser modify [auth_args] -- [subcommand_args]
```

In general, the options for the smuser modify command function the same as for the smuser add command. Refer to the *smuser(1M)* man page for additional options.

Table 10-8 shows the options for the smuser modify command.

Table 10-8 Options for the smuser modify Command

Option	Definition
<code>-n login</code>	Specifies the user's login name
<code>-N login</code>	Specifies the user's new login name

The following example changes the login name and home directory for newuser2 to userb.

```
# /usr/sadm/bin/smuser modify -- -n newuser2 -N userb -d
/export/home/userb
```

Authenticating as user: root

```
Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02 was
successful.
```

Deleting a User Account

Use the `userdel` command or `smuser delete` command to delete a user's login account from the system.

The following is the command format for the `userdel` command:

```
userdel -r login
```

The `userdel` command also removes the user's home directory and all of its contents if you request it to do so. Use the `-r` option to remove the user's home directory from the local file system. This directory must exist.

The following example removes the login account for a user named `usera`.

```
# userdel usera
```

To request that both the user's account and home directory be removed from the system at the same time, perform the command:

```
# userdel -r usera
```

 **Note** – This command does not remove all files owned by the user, just the home directory. The system administrator should run a `find` command to locate all files owned by the user to be backed up or removed.

The `smuser delete` Command Format and Options

The following is the command format for the `smuser delete` command:

```
smuser delete [auth_args] -- [subcommand_args]
```

The following example removes the `userb` account from the system:

```
# /usr/sadm/bin/smuser delete -- -n userb
```

```
Authenticating as user: root
```

```
Type /? for help, pressing <enter> accepts the default denoted by [ ]  
Please enter a string value for: password :: Enter_The_root_Password  
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02  
Login to sys-02 as user root was successful.
```

```
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02 was  
successful.
```



Note – Unlike the userdel command, the smuser delete command has no -r equivalent option for deleting the home directory. The user's home directory must be deleted manually.

Creating a Group Entry

As the root user, you create new group entries on the local system by using the groupadd or smgroup add command. These commands add an entry for the new group into the /etc/group file. Like the smuser command, the smgroup add command uses the same subcommands and authentication arguments derived from the Solaris Management Console.

The groupadd Command Format and Options

The following is the command format for the groupadd command:

```
groupadd [ -g gid [ -o ] ] groupname
```

Table 10-9 shows the options for the groupadd command.

Table 10-9 Options for the groupadd Command

Option	Description
-g <i>gid</i>	Assigns the GID number for the new group
-o	Allows the GID number to be duplicated

The following example uses the groupadd command to create the new group class1 on the local system:

```
# groupadd -g 301 class1
```

The smgroup add Command Format and Options

The following is the command format for the smgroup add command:

```
/usr/sadm/bin/smgroup subcommand [auth_args] -- [subcommand_args]
```

Table 10-10 shows the options for the smgroup add command.

Table 10-10 Options for the smgroup add Command

Option	Description
<code>-g <i>gid</i></code>	Specifies the GID number for the new group
<code>-m <i>group_member</i></code>	Specifies the new members to add to the group
<code>-n <i>group_name</i></code>	Specifies the name of the new group

The following example uses the smgroup add command to create a new group called workgroup with a GID of 123, and to add usera to the group:

```
# /usr/sadm/bin/smgroup add -- -n workgroup -g 123 -m usera
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
was successful.
```

Modifying a Group Entry

You can use the following commands to modify a group entry:

- The groupmod command
- The smgroup modify command

The groupmod Command Format and Options

The following is the command format for the groupmod command:

```
groupmod [ -g gid [ -o ] ] [ -n name ] groupname
```

Table 10-11 defines the options for the groupmod command:

Table 10-11 Options for the groupmod Command

Options	Description
<code>-g gid</code>	Specifies the new GID number for the group
<code>-o</code>	Allows the GID number to be duplicated
<code>-n name</code>	Specifies the new name for the group

The following example changes the class1 account group GID number to 400:

```
# groupmod -g 400 class1
```

The smgroup modify Command Format and Options

The following is the command format for the smgroup modify command:

```
/usr/sadm/bin/smgroup subcommand [auth_args] -- [subcommand_args]
```

Table 10-12 shows the options for the smgroup modify command.

Table 10-12 Options for the smgroup modify Command

Option	Description
<code>-n name</code>	Specifies the name of the group you want to modify
<code>-m new_member</code>	Specifies the new members to add to the group
<code>-N new_group</code>	Specifies the new group name

The following example changes the group workgroup to schoolgroup:

```
# /usr/sadm/bin/smgroup modify -- -n workgroup -N schoolgroup
Authenticating as user: root
```

```
Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
was successful.
```

Deleting a Group Entry

Use the `groupdel` or `smgroup delete` commands to delete a group entry from the `/etc/group` file on the system.

The `groupdel` Command Format

The following is the command format for the `groupdel` command:

```
groupdel groupname
```

The following example removes the group entry `class1` from the local system:

```
# groupdel class1
```

The `smgroup delete` Command Format and Options

The following is the command format for the `smgroup delete` command:

```
/usr/sadm/bin/smgroup subcommand [auth_args] -- [subcommand_args]
```

You can use the `-n group_name` option with the `smgroup delete` command to specify the name of the group you want to delete.

The following example deletes the group entry `schoolgroup` from the local system:

```
# /usr/sadm/bin/smgroup delete -- -n schoolgroup
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
was successful.
```

Using the Solaris Management Console Users Tool

The Solaris Management Console Users Tool is a graphical user interface (GUI) that provides access to Solaris OS system administration tools. You can use it for adding, removing, and modifying user and group entries. The following sections contain a demonstration.

Start the Solaris Management Console by typing **smc&** on the command line or by clicking the SMC icon under the Tools submenu. After the “Welcome to Solaris Management Console” message appears, click This Computer to open the Solaris Management Console window. See Figure 10-2.

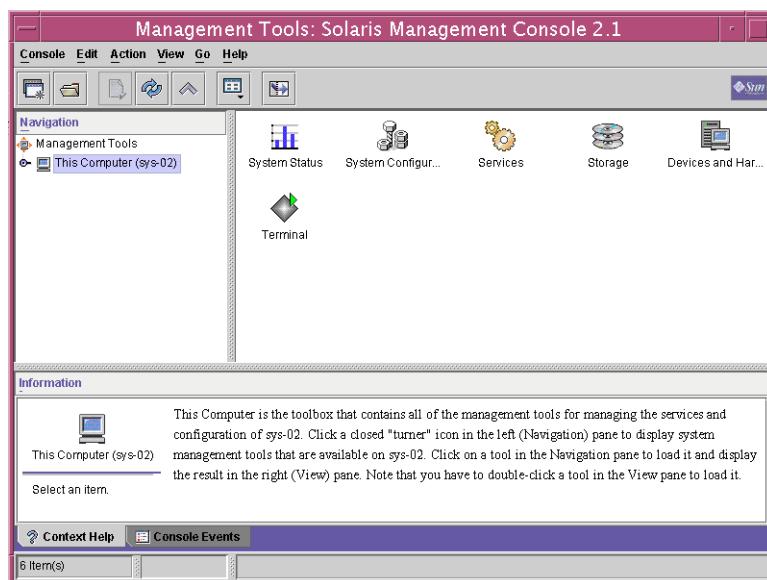


Figure 10-2 Solaris Management Console Window

Adding a User Account

The default method of adding a user account through Solaris Management Console is to add the user account with the user’s home directory automounted. The following steps demonstrate how to build a user template that adds the user account with the user’s directory under the `/export/home` directory.

To add a user account, perform the following steps:

1. Click This Computer in the Navigation pane to display the system management tools.

2. Click System Configuration to display the tool for setting up a new user account.
 3. Click Users and enter the user name and password to be used for authentication if prompted to do so by Solaris Management Console.
 4. Double-click User Templates to access the tool to create and manage user templates.
 5. From the Menu Bar, select Add User Template from the Action list.
- Figure 10-3 shows the Add User Template window.

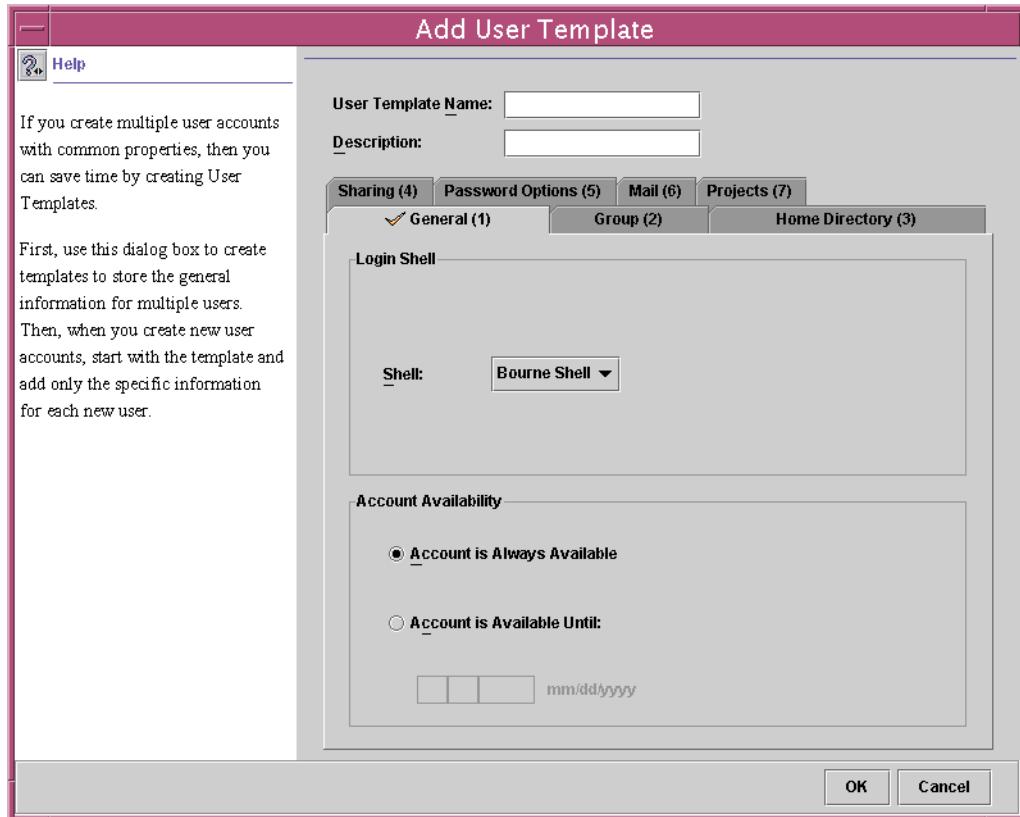


Figure 10-3 Add User Template Window

6. Type SA200user in the User Template Name field. You can provide an optional description if you would like.

7. Click the Home Directory tab. Type your system name in the Home Directory Server field. Uncheck the check box labeled Automatically Mount Home Directory.

Figure 10-4 shows the Add User Template window with the Home Directory Information completed.

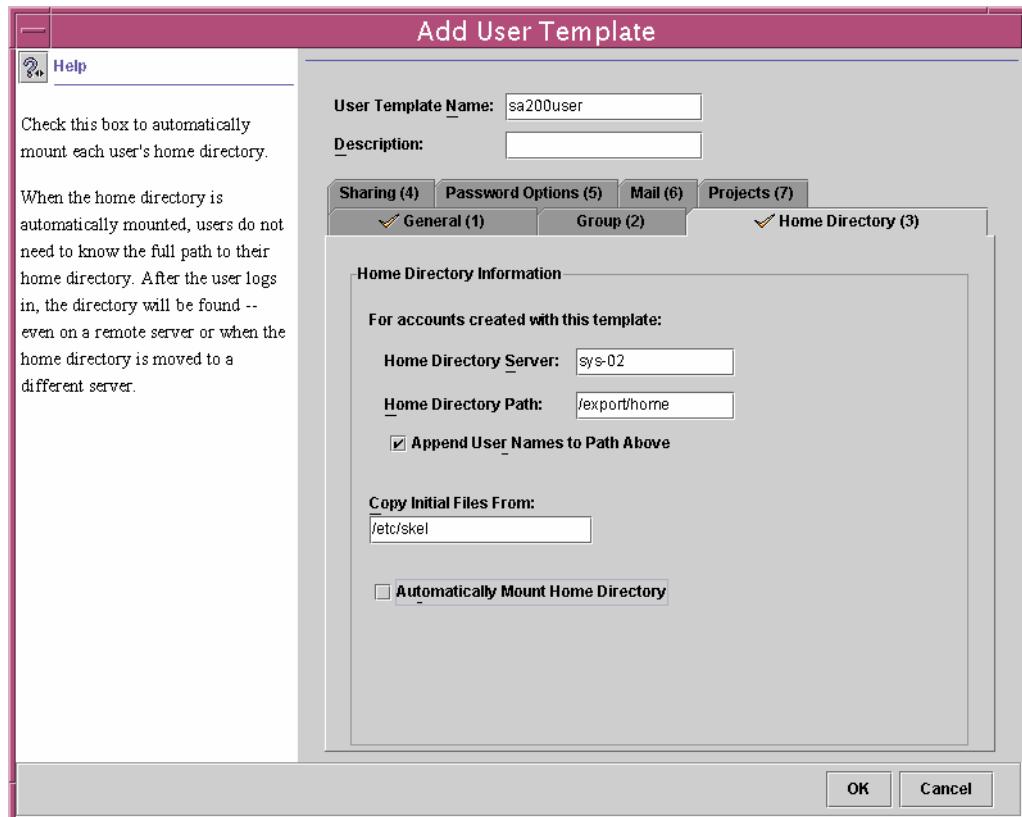


Figure 10-4 Add User Template Window (Home Directory Tab)

8. Click OK, and the Solaris Management Console (User Templates) window, as shown in Figure 10-5, reappears with the SA200user template in the View pane.

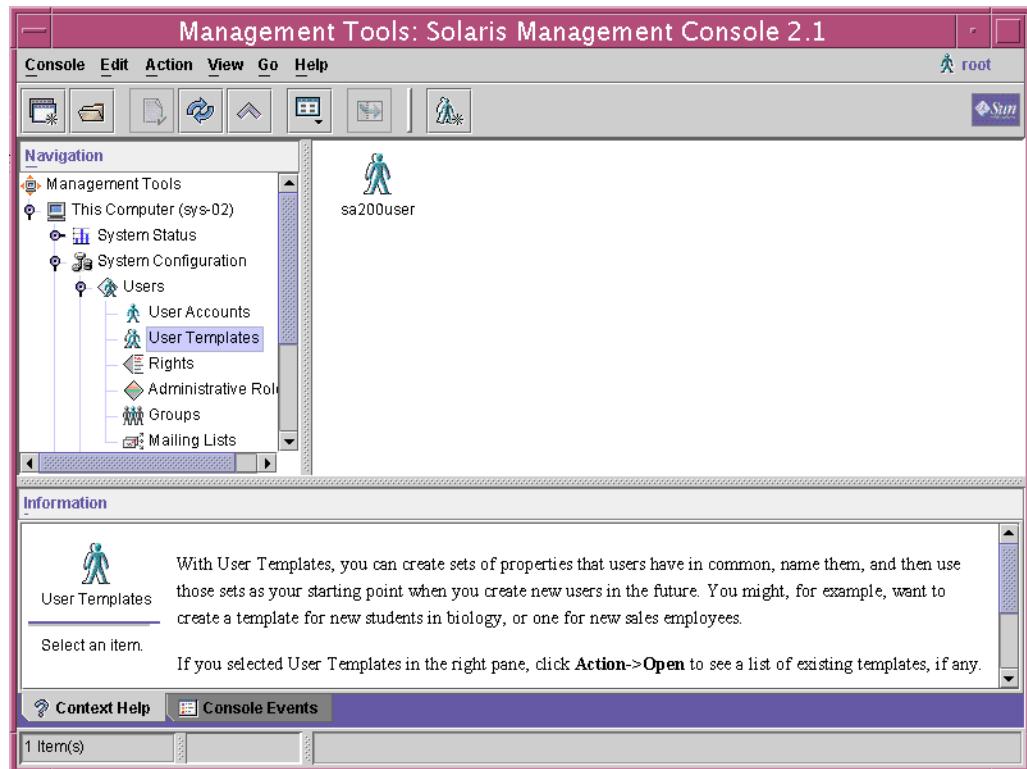


Figure 10-5 Management Tools: Solaris Management Console Window – User Templates

9. Click User Accounts from the Navigation pane, and a list of user accounts on the system appears in the View pane. See Figure 10-6.

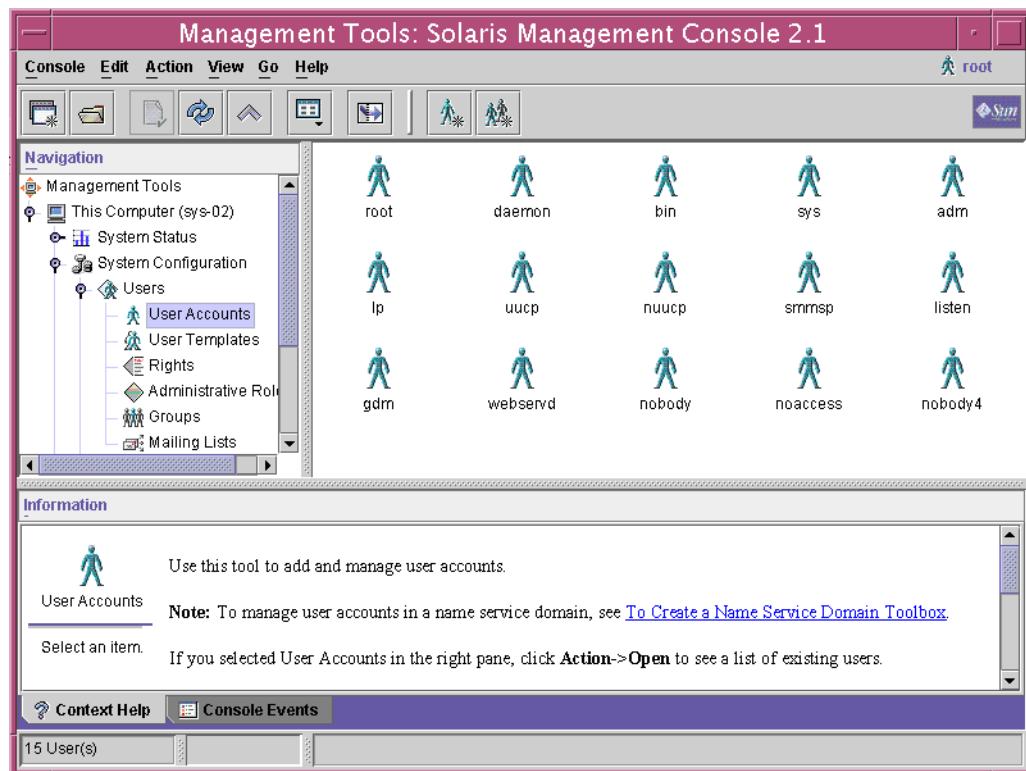


Figure 10-6 Management Tools: Solaris Management Console Window – User Accounts

10. From the Menu Bar, select Action. Then select Add User, and then select From Template. The Add User From Template window appears. See Figure 10-7.

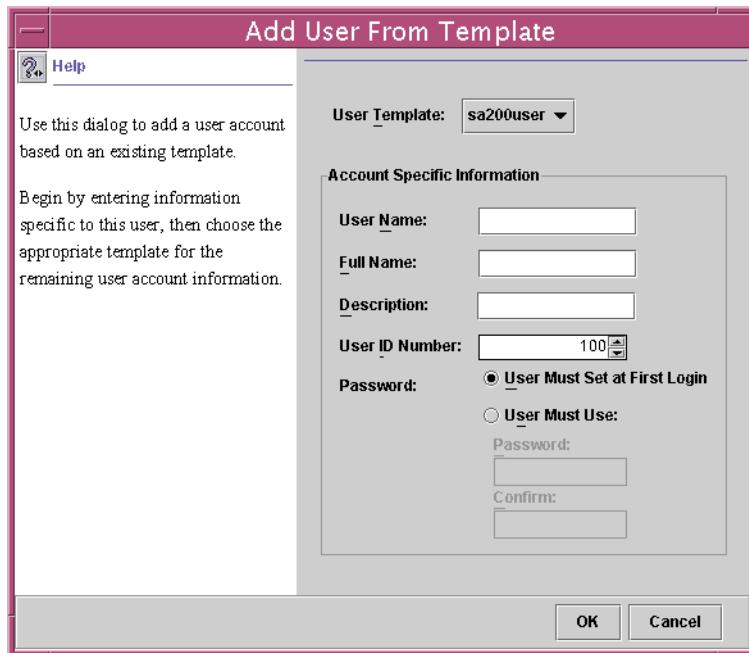


Figure 10-7 Add User From Template Window

Because you only have one template created, it is the default template available from the User Template pull-down list.

11. In the field beside User Name, enter the login ID of the user you want to create. A full name and description are optional.
12. Click the button User Must Use and fill in the password and confirmation fields with the password 123pass.
13. Click OK and the Solaris Management Console (User Accounts) window reappears with the user account you just created in the View pane.

14. Double-click the user account you just created. The User Properties window appears, as shown in Figure 10-8. You can view and modify the properties of that user account.

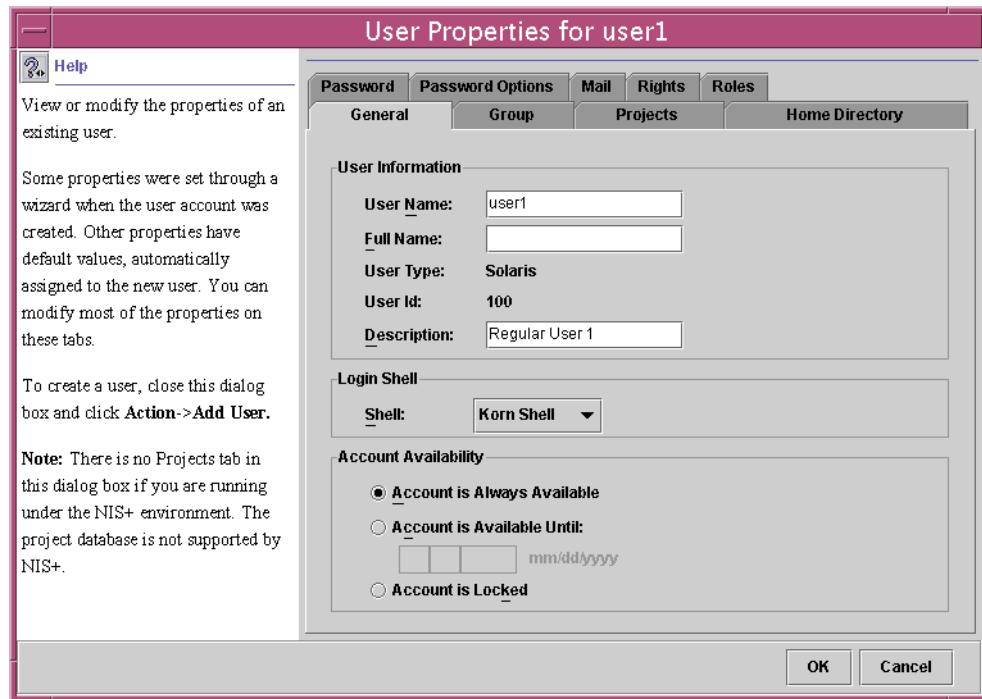


Figure 10-8 User Properties Window

15. Click the Group tab.

The screen changes to reveal a list of groups. Figure 10-9 shows the information under the Group tab, including the primary group to which the user belongs and a list of available groups.

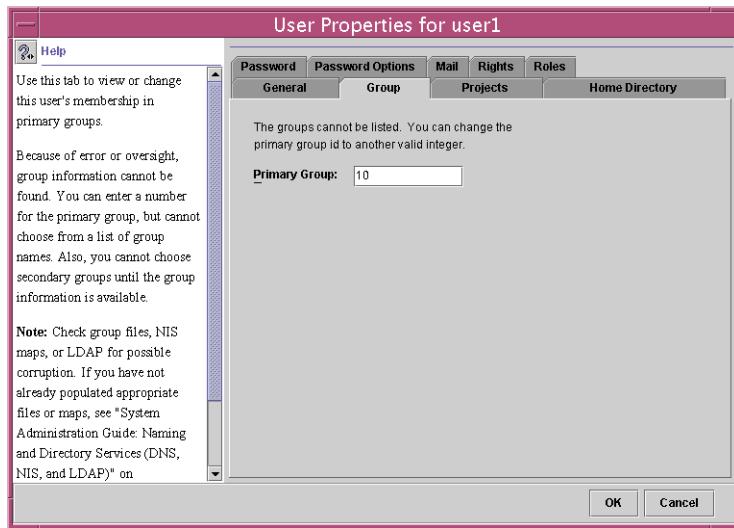


Figure 10-9 User Properties Window – Adding Groups

16. You can click a group listed under Available Groups, then click Add, and the group moves into the Member Of column.
17. Add the groups to which you want the user to belong, and then click OK.

Deleting a User Account

Figure 10-10 shows the initial steps you take to remove a user account from the system.

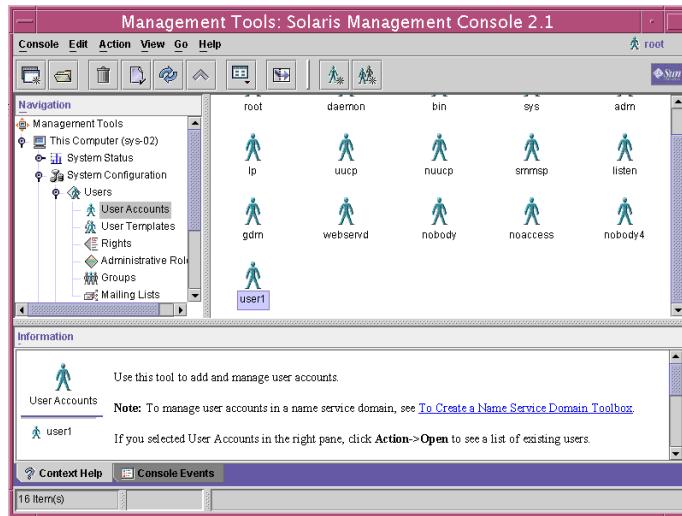


Figure 10-10 Management Tools: Solaris Management Console Window – Deleting a User Account Window

1. Highlight the user account in the User Accounts window.
2. From the Menu Bar, click Edit. Select Delete from the Edit menu.

Figure 10-11 shows the warning window that appears asking you to verify that you want to delete the user account.



Figure 10-11 Warning: Delete User

This window also contains options to remove the user's home directory and to remove the user's mailbox.

3. Check the appropriate boxes, and then click Delete. The user account is deleted.

Troubleshooting Login Issues

Some of the most common problems you might encounter as a system administrator are user login problems. There are two categories of login problems: login problems when the user logs in at the command line and login problems when the user logs in from the Common Desktop Environment (CDE).

The CDE uses more configuration files, so there are more potential problems associated with logging in from the CDE. When you troubleshoot a login problem, first determine whether you can log in from the command line. Attempt to log in from another system by using either the telnet command or the rlogin command, or click Options from the CDE login panel and select Command Line Login. If you can log in successfully at the command line, then the problem is with the CDE configuration files. If you cannot log in at the command line, then the problem is more serious and involves key configuration files.

Login Problems at the Command Line

Table 10-13 presents an overview of common login problems that occur when the user logs in at the command line.

Table 10-13 Login Problems at the Command Line

Login Problem	Description
Login incorrect	This message occurs when there are problems with the login information. The most common cause of an incorrect login message is a mistyped password. Make sure the that correct password is being used, and then attempt to enter it again. Remember that passwords are case-sensitive, so you cannot interchange uppercase letters and lowercase letters. In the same way, the letter "o" is not interchangeable with the numeral "0" nor is the letter "l" interchangeable with the numeral "1."
Permission denied	This message occurs when there are login, password, or NIS+ security problems. Most often, an administrator has locked the user's password or the user's account has been terminated.

Table 10-13 Login Problems at the Command Line (Continued)

Login Problem	Description
Password will not work at lockscreen	A common error is to have the Caps Lock key on, which causes all letters to be uppercase. This does not work if the password contains lowercase letters.
No shell	This message occurs when the user's shell does not exist, is typed incorrectly, or is wrong in the /etc/passwd file.
No directory! Logging in with home=/	This message occurs when the user cannot access the home directory for one of the following reasons: An entry in the /etc/passwd file is incorrect, or the home directory has been removed or is missing, or the home directory exists on a mount point that is currently unavailable.
Choose a new password (followed by the New password: prompt)	This message occurs the first time a user logs in and chooses an initial password to access the account.
Couldn't fork a process!	This message occurs then the server could not fork a child process during login. The most common cause of this message is that the system has reached its maximum number of processes. You can either kill some unneeded processes (if you are already logged into that system as root) or increase the number of processes your system can handle.

Login Problems in the CDE

Problems associated with logging into the CDE range from a user being unable to login (and returning to the CDE login screen), to the custom environment not loading properly. In general, the system does not return error messages to the user from the CDE. The following is a list of files and directories that provide troubleshooting information about the CDE:

- `/usr/dt/bin/Xsession`

This file is the configuration script for the login manager. This file should not be edited. The first user-specific file that the `Xsession` script calls is the `$HOME/.dtprofile` file.
- `$HOME/.dtprofile`

By default, the file does not contain much content, except for examples. It contains a few echo statements for session logging purposes, and the `DTSOURCEPROFILE` variable is set. But it also contains information about how it might be edited. The user can edit this file to add user-specific environment variables.
- `DTSOURCEPROFILE=true`

This line allows the user's `$HOME/.login` file (for csh users) or the `$HOME/.profile` (for other shell users) to be sourced as part of the startup process.

Sometimes a `.login` or `.profile` file contains problem commands that cause the shell to crash. If the `.dtprofile` file is set to source a `.login` or `.profile` file that has problem commands, desktop startup might fail.

Consequently, no desktop appears. Instead, the system redisplays the Solaris OS CDE login screen. Startup errors from the `.login` or `.profile` file are usually noted in the `$HOME/.dt/startlog` file. Use a Failsafe login Session or a command-line login to debug problem commands in the `.login` or `.profile` files.

- `$HOME/.dt/sessions`

This directory structure contains files and directories that configure the display of the user's custom desktop and determine the applications that start when the user logs in. Look for recent changes to files and for changes to the directory structure. For example, examine the home directory and the home.old directory or a current directory and the current.old directory. Compare the changes. The changes could provide information on a new application or on changes in the saved desktop that cause the user's login to fail.

- `$HOME/.dt`

Upon removing the entire .dt directory structure, log out, and log back in again for the system to rebuild a default .dt file structure. This action allows the user to get back into the system if the problem with the CDE files cannot be resolved.

Table 10-14 shows the locations of and information found in error logs for the CDE.

Table 10-14 CDE Error Log Locations

Location	Error Log
/var/dt/Xerrors	The Solaris OS CDE login window system errors that occur prior to user login
<code>\$HOME/.dt/startlog</code>	The Solaris OS CDE errors that occur during the startup of the Xsession script, while processing the .dtprofile, .login, or .profile file
<code>\$HOME/.dt/errorlog.old</code> <code>\$HOME/.dt/errorlog.older</code>	The Solaris OS CDE errors that occur after the Xsession script start up
<code>\$HOME/.dt/sessionlogs</code>	Directory of session logs for Session Manager and Window Manager errors

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Adding User Accounts and Group Entries (Level 1)

In this exercise, you use the Solaris Management Console, as well as the `smuser`, `smgroup`, `usermod`, `userdel`, `groupadd`, and `groupdel` commands, to create, modify, and delete multiple user accounts and group entries.

Preparation

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Refer to the lecture notes as necessary to perform the tasks listed. Refer to Table 10-15 and Table 10-16 as needed.

Table 10-15 Group Specifications

Group Name	GID Number
class1	101
class2	102

Table 10-16 User Specifications

User Name	Password	Shell	UID	Primary Group	Secondary Group
user3	123pass	Korn	1003	10	class1
user4	123pass	C	1004	10	class1
user5	123pass	Bourne	1005	10	

Exercise: Adding User Accounts and Group Entries (Level 1)

Table 10-16 User Specifications (Continued)

locked1	Select Account is Locked	Bourne	2001	10	
cleared1	Select User must set password at next login	Bourne	2002	10	

Tasks

Complete the following tasks:

- Disable the Solaris OS registration window.
(Steps 1–5 of Task 1 in the Level 2 lab)
- Working from Table 10-15 and Table 10-16 on page 10-41, create two new groups and two new users by using the groupadd, smgroup, useradd, and smuser commands.
(Steps 1–2 of Tasks 2 and 3 in the Level 2 lab)
- Launch the Solaris Management Console, and create a user template to add users that do not use automounted home directories.
(Step 3 of Task 3 in the Level 2 lab)
- Using the Solaris Management Console, add the new users user5, locked1, and cleared1 with characteristics from Table 10-16 on page 10-41.
(Steps 4–5 of Task 3 in the Level 2 lab)
- Verify that the shells you specify are set in the /etc/passwd file. Determine if the password strings for users with the same password are also the same in the /etc/shadow file. Check the password strings for the users locked1 and cleared1. Verify that the users user3 and user4 are secondary members of the class1 group.
(Steps 1–4 of Task 4 in the Level 2 lab)
- Determine what happens when you try to log in as the user locked1. Verify that you can log in as the user cleared1. Record the password requirements indicated.
(Steps 5–6 of Task 4 in the Level 2 lab)
- Establish password aging for the user user5. Determine what happens when you attempt to log in as that user. Log in as user5 and attempt to change the password from the command line. Log in as the root user when you are finished.
(Steps 1–4 of Task 5 in the Level 2 lab)

Exercise: Adding User Accounts and Group Entries (Level 1)

- Use the `groupadd` command to add a group called `class3`. Use the `usermod` command to change the UID number, home directory, and user name for the user `locked1`. Verify that the changes exist in the `/etc/passwd` file.
(Steps 1–2 of Task 5 in the Level 2 lab)
- Use the `smuser` command to change the login shell of `user5` to `ksh`. Use the `userdel` command to delete the user `user3`. Verify that the home directory has been deleted. Use the `smgroup` command to rename the group `class1` to `group1`. Use the `groupdel` command to remove the group `class2`. Verify the changes to the `/etc/group` file.
(Steps 3–7 of Task 5 in the Level 2 lab)

Exercise: Adding User Accounts and Group Entries (Level 2)

In this exercise, you use the Solaris Management Console, as well as the `smuser`, `smgroup`, `usermod`, `userdel`, `groupadd`, and `groupdel` commands, to create, modify, and delete multiple user accounts and group entries.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed. Refer to Table 10-15 and Table 10-16 on page 10-41 as needed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fnl.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Disable the Solaris OS registration window.
- Working from Table 10-15 and Table 10-16 on page 10-41, create two new groups and two new users by using the commands `groupadd`, `smgroup`, `useradd`, and `smuser`.
- Launch the Solaris Management Console, and create a user template to add users that do not use automounted home directories.
- Using the Solaris Management Console, add the new users `user5`, `locked1` and `cleared1` with characteristics from Table 10-16 on page 10-41.

Exercise: Adding User Accounts and Group Entries (Level 2)

- Verify that the shells you specify are set in the /etc/passwd file. Determine if the password strings for users with the same password are also the same in the /etc/shadow file. Check the password strings for the users `locked1` and `cleared1`. Verify that the users `user3` and `user4` are secondary members of the `class1` group.
- Determine what happens when you try to log in as the user `locked1`. Verify that you can log in as the user `cleared1`. Record the password requirements indicated.
- Establish password aging for `user5`. Determine what happens when you attempt to log in as that user. Log in as `user5` and attempt to change the password from the command line. Log in as the `root` user when you are finished.
- Use the `groupadd` command to add a group called `class3`. Use the `usermod` command to change the UID number, home directory, and user name for the user `locked1`. Verify that the changes exist in the /etc/passwd file.
- Use the `smuser` command to change the login shell of `user5` to `KS`. Use the `userdel` command to delete the user `user3`. Verify that the user's home directory has been deleted. Use the `smgroup` command to rename the group `class1` to `group1`. Use the `groupdel` command to remove the group `class2`. Verify the changes to the /etc/group file.

Tasks

Complete the following tasks.

Task 1 – Disabling the Solaris OS Registration Window

Complete the following steps:

1. Disable the Solaris OS Registration window so that it does not appear whenever a new user logs in from the CDE.
2. Log in as the `root` user (or use the `su` command to change to the `root` user).
3. Change to the /etc/default directory.
4. In the default directory, create the `solregis` file.

```
#vi solregis
```

5. In the `solregis` file, type the keyword `DISABLE=1` (note that the character “1” is the number one).
6. Save this file, and exit the editor.

Task 2 – Adding Group Entries

Complete the following steps:

Note – Refer to Table 10-15 on page 10-41 for details while adding groups.



1. As the root user, open a terminal window.
2. Add the two groups `class1` and `class2` with the `groupadd` and `sgroup` commands, respectively.

Task 3 – Adding User Accounts

Complete the following steps:

Note – Refer to Table 10-16 on page 10-41 for details while adding users with the various tools.



1. Add a user named `user3` by using the `useradd` command.
2. Add a user named `user4` by using the `smuser` command.
3. Launch the Solaris Management Console by typing `smc&` on the command line. After the Solaris Management Console appears, create a user template to add user accounts that do not use automounted home directories by performing the following:
 - a. Select This Computer, and then select System Configuration. Then select Users, and then select User Templates to open the User Templates tool.
 - b. From the Menu Bar, select Action. Then select Add User Template.
 - c. The Add User Template window appears, containing blank fields for a template name and description. Enter the name `200user` in the User Template Name field, and `SA200` for the Description field.
 - d. Click the Home Directory Tab and uncheck the Automatically Mount Home Directory check box. Enter the name of your system in the Home Directory Server field.
 - e. Click OK to create your template.

4. Click User Accounts, and add the user5 account by selecting Action, then selecting Add User, and then selecting From Template on the menu bar.

The Add User From Template window appears. Enter user5 in the User Name field, and select 1005 as the User ID Number. For the password, click User Must Use, and enter 123pass in both password fields. Click OK.

5. From the Solaris Management Console, add the additional users locked1 and cleared1 by using the 200user template. While adding the cleared1 user, select the password option User Must Set Password At Next Login. After adding both users, double-click the locked1 user and select the tab General. Under the Account Availability section, select the button Account is Locked. Also select the shell as listed in Table 10-16 on page 10-41.

Task 4 – Examining Configuration Files

Complete the following steps:

1. Examine the contents of the /etc/passwd file. What are the full path names of the shells used by user3, user4, and user5?
2. Examine the contents of the /etc/shadow file. What text is found in the password field for the users locked1 and cleared1?
3. You used the same password for user3 through user5. Are the password strings the same in the /etc/shadow file?
4. Examine the contents of the /etc/group file. Verify that user3 and user4 are both listed as secondary members of the class1 group. Are they?
5. Log out of the CDE, and attempt to log in as locked1. Are you able to log in?
6. Attempt to log in as cleared1. What happens? Attempt to use the password abcdefg. What are the system requirements for the password?

Use the password abc123. Log in as cleared1 after you establish a password to verify that the login works. Log out, and log in as the root user.

Task 5 – Establishing Password Aging

Complete the following steps:

1. Start the Solaris Management Console, and go back into the User Accounts Tool. Select user5 from the list of users. Change the password options information for user5 so that it matches the following information. Click OK when you are finished, and exit the Solaris Management Console.

User Must Keep For: 1 (one day)

Before Change Alert User: 1 (one day)

User Must Change Within: 2 (two days)

Expires If Not Used For: 1 (one day)

2. Log out of your root login session. Attempt to log in as user5. What happens? Supply a new password if necessary.
3. Complete the login as user5. Open a terminal window, and attempt to change the password you just set. What happens?
4. Log out, and log in again as the root user.

Task 6 – Modifying User Accounts and Group Entries

Complete the following steps:

1. Use the groupadd command to create a new group entry called class3 that uses GID number 103.
2. Use the usermod command to change the login name of locked1 to user6, the UID to 3001, and the home directory of locked1 to user6. Verify that the changes you request are recorded in the /etc/passwd file and the directory that was moved.
3. Use the smuser modify command to change the login shell of user5 to /bin/ksh. Verify that the changes you request are recorded in the /etc/passwd file.
4. Use the userdel command to delete the user account cleared1 and the related home directory. Verify that the /export/home/cleared1 directory no longer exists.

Exercise: Adding User Accounts and Group Entries (Level 2)

5. Use the smgroup command to change the group name of class1 to group1.
6. Use the groupdel command to remove the group entry class2.
7. Verify that the commands used to modify group entries have correctly modified the /etc/group file.

Exercise: Adding User Accounts and Group Entries (Level 3)

In this exercise, you use the Solaris Management Console, as well as the `smuser`, `smgroup`, `usermod`, `userdel`, `groupadd`, `groupmod`, and `groupdel` commands, to create, modify, and delete multiple user accounts and group entries.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed. Refer to Table 10-15 and Table 10-16 on page Module 10-41 as needed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fnl.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.



Note – Some of the commands displayed in this section are quite long and will wrap to the next line. You should consider all of the bold typeface commands that follow a command line prompt to be all one line.

Task Summary

In this exercise, you accomplish the following:

- Disable the Solaris OS registration window.
- Working from Table 10-15 and Table 10-16 on page 10-41, create two new groups and two new user accounts using the commands `groupadd`, `smgroup`, `useradd`, and `smuser`.
- Launch the Solaris Management Console and create a user template to add users that do not use automounted home directories.

Exercise: Adding User Accounts and Group Entries (Level 3)

- Using the Solaris Management Console, add the new user accounts user5, locked1, and cleared1 with characteristics from Table 10-16 on page 10-41.
- Verify that the shells you specify are set in the /etc/passwd file. Determine if the password strings for users with the same password are also the same in the /etc/shadow file. Check the password strings for the users locked1 and cleared1. Verify that the users user3 and user4 are secondary members of the class1 group.
- Determine what happens when you try to log in as the user locked1. Verify that you can log in as the user cleared1. Record the password requirements indicated.
- Establish password aging for the user user5. Determine what happens when you attempt to log in as that user. Log in as user5 and attempt to change the password from the command line. Log in as root when you are finished.
- Use the groupadd command to add a group called class3. Use the usermod command to change the UID number, home directory, and user name for the user locked1. Verify that the changes exist in the /etc/passwd file.
- Use the smuser command to change the login shell of user5 to ksh. Use the userdel command to delete the user3 account. Verify that the user's home directory has been deleted. Use the smgroup command to rename the group class1 to group1. Use the groupdel command to remove the group class2. Verify the changes to the /etc/group file.

Tasks and Solutions

Complete the following tasks.

Task 1 – Disabling the Solaris OS Registration Window

Complete the following steps:

1. Disable the Solaris OS Registration window so that it does not appear whenever a new user logs in from the CDE.
2. Log in as the root user (or use the su command to change the root user).
3. Change to the /etc/default directory.

4. In the default directory, create the file solregis.

```
#vi solregis
```

5. In the solregis file, type the keyword DISABLE=1 (note that the character “1” is the number one).
6. Save this file, and exit the editor.

Task 2 – Adding Group Entries

Complete the following steps:

Note – Refer to Table 10-15 on page 10-41 for details while adding groups.



1. As the root user, open a terminal window.
2. Add the two groups class1 and class2, with groupadd and smgroup commands, respectively.

```
# groupadd -g 101 class1  
# /usr/sadm/bin/smgroup add -- -n class2 -g 102
```

Task 3 – Adding User Accounts

Complete the following steps:

Note – Refer to Table 10-16 on page 10-41 for details while adding users with the various tools.



1. Add a user named user3 by using the useradd command.

```
# useradd -u 1003 -g 10 -G class1 -d /export/home/user3 -m -s /bin/ksh  
user3  
# passwd user3  
New Password: 123pass  
Re-enter new Password: 123pass  
passwd: password successfully changed for user3
```

2. Add a user named user4 by using the smuser command.

```
# /usr/sadm/bin/smuser add -- -n user4 -u 1004 -g 10 -G class1 -d  
/export/home/user4 -s /bin/csh -x autohome=N  
# passwd user4  
New Password: 123pass  
Re-enter new Password: 123pass  
passwd: password successfully changed for user4
```

Exercise: Adding User Accounts and Group Entries (Level 3)

3. Launch the Solaris Management Console by typing **smc&** on the command line. After the Solaris Management Console appears, create a user template to add user accounts that do not use automounted home directories by performing the following:
 - a. Select This Computer, and then select System Configuration. Then select Users, and then select User Templates to open the User Templates tool.
 - b. From the Menu Bar, select Action, and then select Add User Template.
 - c. The Add User Template window appears, containing blank fields for a template name and description. Enter the name **200user** in the User Template Name field, and **SA200** for the Description field.
 - d. Click the Home Directory Tab and uncheck the Automatically Mount Home Directory check box. Enter the name of your system in the Home Directory Server field.
 - e. Click OK to create your template.
4. Click User Accounts, and add the **user5** account by selecting Action, then selecting Add User, and then selecting From Template on the menu bar.

The Add User From Template window appears. Enter **user5** in the User Name field and select **1005** as the UID Number. For password, click the button called User Must Use, and enter **123pass** in both password fields. Click OK.
5. From the Solaris Management Console, add the users **locked1** and **cleared1** by using the **200user** template. While adding the **cleared1** user, select the password option **User Must Set Password At Next Login**. After adding both users, double-click the **locked1** user and select the tab General. Under the Account Availability section, select **Account is Locked**. Also select the shell as listed in Table 10-16 on page 10-41.

Task 4 – Examining Configuration Files

Complete the following steps:

1. Examine the contents of the /etc/passwd file. What are the full path names of the shells used by user3, user4, and user5?

user3	/bin/ksh
user4	/bin/csh
user5	/bin/sh

2. Examine the contents of the /etc/shadow file. What text is found in the password field for the users locked1 and cleared1?

locked1	*LK*
cleared1	none

3. You used the same password for user3 through user5. Are the password strings the same in the /etc/shadow file?

No.

4. Examine the contents of the /etc/group file. Verify that user3 and user4 are both listed as secondary members of the class1 group. Are they?

The names user3 and user4 should be listed in the last field for the class1 group.

5. Log out of the CDE, and attempt to log in as locked1. Are you able to log in?

No, you get a message that says login incorrect, no matter what you use as a password.

6. Attempt to log in as cleared1. What happens? Attempt to use the password abcdefg. What are the system requirements for the password? You must not press Return when you are asked for an initial password.

You must choose an initial password for this user and then log in again. The first six characters must contain at least two alphabetic characters and at least one numeric or special character.

Use the password abc123. Log in as cleared1 after you establish a password to verify that the login works. Log out, and log in as the root user.

Task 5 – Establishing Password Aging

Complete the following steps:

1. Start the Solaris Management Console, and go back into the User Accounts tool. Select user5 from the list of users. Change the password options information for user5 so that it matches the following information. Click OK when you are finished, and exit the Solaris Management Console.

User Must Keep For: 1 (one day)

Before Change Alert User: 1 (one day)

User Must Change Within: 2 (two days)

Expires If Not Used For: 1 (one day)

2. Log out of your root login session. Attempt to log in as user5. What happens? Supply a new password if necessary.

You must supply a new password before you can log in.

3. Complete the login as user5. Open a terminal window, and attempt to change the password you just set. What happens?

When you log in, a warning indicates that your password expires in two days.

When you try to change your password, the following error message appears:

```
passwd: Sorry: less than 1 days since the last change.  
Permission denied
```

4. Log out, and log in again as the root user.

Task 6 – Modifying User Accounts and Group Entries

Complete the following steps:

1. Use the groupadd command to create a new group entry called class3 that uses GID number 103.

```
# groupadd -g 103 class3
```

2. Use the usermod command to change the login name of locked1 to user6, the UID to 3001, and the home directory of locked1 to user6. Verify that the changes you request are recorded in the /etc/passwd file and that the directory was moved.

```
# usermod -u 3001 -d /export/home/user6 -m -l user6 locked1
```

The /etc/passwd file should reflect the new UID number and user name. The directory under /export/home should be renamed.

3. Use the smuser modify command to change the login shell of user5 to /bin/ksh. Verify that the changes you request are recorded in the /etc/passwd file.

```
# /usr/sadm/bin/smuser modify -- -n user5 -s /bin/ksh
```

The /etc/passwd file should show that the shell is /bin/ksh.

4. Use the userdel command to delete the user account cleared1 and the related home directory. Verify that the /export/home/cleared1 directory no longer exists.

```
# userdel -r cleared1
```

The /export/home/cleared1 directory should no longer exist.

5. Use the smgroup command to change the group name of class1 to group1.

```
# /usr/sadm/bin/smgroup modify -- -n class1 -N group1
```

6. Use the groupdel command to remove the group entry class2.

```
# groupdel class2
```

7. Verify that the commands used to modify group entries have correctly modified the /etc/group file.

The group group1 should exist. The groups class1 and class2 should not exist.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Managing Initialization Files

The environment maintained by the shell includes variables that are defined by the `login` program, the system initialization files, and the user initialization files.

When users log in to the system, their login shells look for and execute two different types of initialization files. The first type controls the system-wide environment. The second type controls the user's environment. The six shells available in the Solaris 10 OS provide basic features and a set of variables which the `root` user or a regular user can set in the initialization files to customize the shell environment.

The shells support two types of variables:

- Environment variables – Variables that provide information about the user's environment to every shell program that is started.
- Local variables – Variables that affect only the current shell. Any subshell started would not have knowledge of these variables.

Introducing System-Wide Initialization Files

As the system administrator, you maintain the system-wide initialization files. These files provide an environment for the entire community of users who log in to the system. The Solaris OS provides the system initialization files. They reside in the `/etc` directory.

The `/etc/profile` file and the `/etc/.login` file are the two main system initialization files.

The Bourne, Korn, and BASH login shells look for and execute the system initialization file `/etc/profile` during login.

The C login shell looks for and executes the system initialization file `/etc/.login` during the login process.

Note – The default files `/etc/profile` and `/etc/.login` check disk usage quotas, print the message of the day from the `/etc/motd` file, and check for mail. None of the messages are printed to the screen if the `.hushlogin` file exists in the user's home directory.



Introducing User Initialization Files

As the system administrator, you set up the user initialization files that are placed in each user account's home directory when the user is created.

The primary purpose of the user initialization files is to define the characteristics of a user's work environment, such as the command-line prompt, the environment variables, and the windowing environment.

Only the owners of the files or the `root` user can change or customize the content of these files.

Table 10-17 shows the initialization files necessary for each primary shell available in the Solaris 10 OS.

Table 10-17 Initialization Files for the Primary Shells

Shells	System-Wide Initialization Files	Primary User Initialization Files Read at Login	User Initialization Files Read When a New Shell Is Started	Shell Path Name
Bourne	<code>/etc/profile</code>	<code>\$HOME/.profile</code>		<code>/bin/sh</code>
Korn	<code>/etc/profile</code>	<code>\$HOME/.profile</code> <code>\$HOME/.kshrc</code>	<code>\$HOME/.kshrc</code>	<code>/bin/ksh</code>
C	<code>/etc/.login</code>	<code>\$HOME/.cshrc</code> <code>\$HOME/.login</code>	<code>\$HOME/.cshrc</code>	<code>/bin/csh</code>

For additional information about the Z, BASH, and TC shells available in the Solaris 10 OS, refer to the online manual pages.

Note – By default, the `root` user's login shell is the Bourne shell, and the shell entry in the `/etc/passwd` file appears as `/sbin/sh`.



When a user logs in to the system, the system invokes the user's login shell program. The shell program looks for its initialization files in a specific order, executes the commands contained in each file, and displays the shell prompt on the user's screen.

Customizing the User's Work Environment

The Solaris OS provides a set of initialization file templates. The /etc/skel directory contains the initialization file templates. Table 10-18 shows the default initialization file templates and the user initialization files for the Bourne, Korn, and C shells.

Table 10-18 Default User Initialization Files

Shell	Initialization File Templates	User Initialization Files
Bourne	/etc/skel/local.profile	\$HOME/.profile
Korn	/etc/skel/local.profile	\$HOME/.profile
C	/etc/skel/local.cshrc /etc/skel/local.login	\$HOME/.cshrc \$HOME/.login



Note – The useradd command copies files from the /etc/skel directory to the \$HOME directory. The smuser command copies files from the /etc/skel directory to the \$HOME directory and renames them to the appropriate file names.

The root user can customize these templates to create a standard set of user initialization files. A standard set of user initialization files provides a common work environment for each user. When the root user creates new user accounts, some or all of these initialization files are automatically copied to each new user's home directory.

Users can then edit their initialization files to further customize their environments for each shell.

Table 10-19 shows some of the variables available for customizing a user's shell environment.

Table 10-19 Login Variables

Variable Name	Set By	Description
LOGNAME	Login	Defines the user's login name.

Table 10-19 Login Variables (Continued)

Variable Name	Set By	Description
HOME	Login	Sets the path to the user's home directory. It is the default argument for the cd command.
SHELL	Login	Sets the path to the default shell.
PATH	Login	Sets the default path that the shell searches to find commands.
MAIL	Login	Sets the path to the user's mailbox.
TERM	Login	Defines the terminal.
LPDEST	Not set by default	Sets the user's default printer.
PWD	Shell	Defines the current working directory.
PS1	Shell	Defines the shell prompt for the Bourne or Korn shell.
prompt	Shell	Defines the shell prompt for the C shell.

 **Note** – For complete information on all variables used by the default shells, see the following man pages: sh(1), ksh(1), csh(1), zsh(1), bash(1), and tcsh(1).

A user can change the values of the predefined variables and specify additional variables.

Table 10-20 shows how to set environment variables in the user initialization files of the Bourne, Korn, and C shells.

Table 10-20 Setting Environment Variables

Shell	User's Initialization File
Bourne or Korn	<i>VARIABLE=value ; export VARIABLE</i> For example: <code>PS1="\$HOSTNAME " ; export PS1</code>
C	<code>setenv variable value</code> For example: <code>setenv LPDEST laserprinter</code>

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Modifying Initialization Files (Level 1)

In this exercise, complete the following tasks:

- Modify initialization file templates in the /etc/skel directory
- Create user accounts that use the initialization files

Preparation

This exercise requires the skills practiced in the previous exercise. The user accounts that you create in this exercise are required in later sections of the course. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Modify the template for Bourne shell users. Set the EDITOR to vi, LPDEST to printer1, EXINIT to set showmode autoindent and number, and ENV to source the .kshrc file.
(Steps 1–3 in the Level 2 lab)
- Use the Solaris Management Console to create a new user account called user9 that uses the Korn shell. Log in as the new user, and verify that all the variables you set in local.profile are set correctly in the user's environment.
(Steps 4–6 in the Level 2 lab)

Exercise: Modifying Initialization Files (Level 1)

- Create a `.kshrc` file for the new user account that includes two aliases and sets the primary prompt to echo the current working directory. Log out, and log in again as the same user to verify that the `.kshrc` file works. Log out, and log in again as the `root` user.
(Steps 7–9 in the Level 2 lab)
- Use the `useradd` command to create a new user account called `user10` that uses the Korn shell. Log in as this user, and record the list of initialization files in the home directory. Copy the appropriate file to the `.profile` file. Test the login to verify that the list of variables is set the same as those of the first user you created. Log out, and log in as the `root` user when you are finished.
(Steps 9–13 in the Level 2 lab)

Exercise: Modifying Initialization Files (Level 2)

In this exercise, complete the following tasks:

- Modify initialization file templates in the /etc/skel directory
- Create user accounts that use the initialization files

Preparation

This exercise requires the skills practiced in the previous exercise. The user accounts that you create in this exercise are required in later sections of the course. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Modify the template for Bourne shell users. Set the EDITOR to vi, LPDEST to printer1, EXINIT to set showmode autoindent and number, and ENV to source the .kshrc file.
- Use the Solaris Management Console to create a new user account called user9 that uses the Korn shell. Log in as the new user, and verify that all the variables you set in local.profile are set correctly in the user's environment.

- Create a .kshrc file for the new user account that includes two aliases and sets the primary prompt to echo the current working directory. Log out, and log in again as the same user to verify that the .kshrc file works. Log out, and log in again as the root user.
- Use the useradd command to create a new user account called user10 that uses the Korn shell. Log in as this user, and record the list of initialization files in the home directory. Copy the appropriate file to the .profile file. Test the login to verify that the list of variables is set the same as those of the first user you created. Log out, and log in as the root user when you are finished.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window.
2. Change to the /etc/skel directory.
3. Use the vi editor to edit the local.profile file, and make the following changes:
 - a. Edit the line that declares the PATH variable so that it reads as follows. Enter this text as one line (no spaces).
PATH=/usr/sbin:/sbin:/usr/sadm/bin:/usr/dt/bin:/usr/openwin/bin:/usr/bin:/usr/ucb:.
 - b. Add the following lines below the PATH variable you just edited:
**EDITOR=vi
LPDEST=printer1
EXINIT='set showmode autoindent number'
ENV=\$HOME/.kshrc**

- c. Change the line that reads:

export PATH

so that it reads:

export PATH EDITOR LPDEST EXINIT ENV

4. Use the Solaris Management Console to create a new user account with the following characteristics. Exit the Solaris Management Console when you are finished.

User Name:	user9
User ID:	1009
Primary Group:	staff
Login Shell:	Korn
Password:	123pass

5. Log out, and log in again as user9. Open a terminal window.
6. Verify that the PATH, LPDEST, EDITOR, EXINIT, and ENV variables are set according to the changes you made in the /etc/skel/local.profile file.

Do they match?

7. Create a file called .kshrc in user9's home directory.

Insert the following lines. A space follows the \$PWD\$ in the last line.

```
set -o noclobber
set -o ignoreeof
alias h=history
alias c=clear
PS1='$PWD$ '
```

8. Log out, and then log in again as user9. Open a terminal window, and verify that your new variables work.

Do they work?

9. Log out, and log in again as the root user. Use the useradd command to create a new user account called user10 with the following characteristics:

User Name:	user10
User ID:	1010
Primary Group:	10
Login Shell:	Korn
Home Directory:	/export/home/user10
Comment:	SA-200 Student
Password:	cangetin

Exercise: Modifying Initialization Files (Level 2)

10. Log out, and log in again as user10. Open a terminal window. What shell initialization files exist in your home directory?
Which of these are the same as /etc/skel/local.profile?
11. Copy the local.profile file to the .profile file.
12. Log out, and log in again as user10. Verify that the variables set for the user9 login are also set for this login.
Do they match?
13. Log out, and log in again as the root user.

Exercise: Modifying Initialization Files (Level 3)

In this exercise, complete the following tasks:

- Modify initialization file templates in the /etc/skel directory
- Create user accounts that use the initialization files

Preparation

This exercise requires the skills practiced in the previous exercise. The user accounts that you create in this exercise are required in later sections of the course. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Edit the /etc/skel/local.profile file so that it sets the PATH variable to a specific list of directories. Set the EDITOR, LPDEST, EXINIT, and ENV variables to appropriate values.
- Use the Solaris Management Console to create a new user account called user9 that uses the Korn shell. Log in as the new user, and verify that all the variables you set in local.profile are set correctly in the user's environment.

- Create a .kshrc file for the new user account that includes two aliases and sets the primary prompt to echo the current working directory. Log out, and log in again as the same user to verify that the .kshrc file works. Log out, and log in again as the root user.
- Use the useradd command to create a new user account called user10 that uses the Korn shell. Log in as this user, and record the list of initialization files in the home directory. Copy the appropriate file to the .profile file. Test the login to verify that the list of variables is set the same as those of the first user you created. Log out, and log in as the root user when you are finished.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window.
2. Change to the /etc/skel directory.

```
# cd /etc/skel
```

3. Use the vi editor to edit the local.profile file, and make the following changes:

```
# vi local.profile
```

- a. Edit the line that declares the PATH variable so that it reads as follows. Enter this text as one line (no spaces).

```
PATH=/usr/sbin:/sbin:/usr/sadm/bin:/usr/dt/bin:/usr/openwin/bin:/usr/bin:  
/usr/ucb:.
```

- b. Add the following lines below the PATH variable you just edited:

```
EDITOR=vi  
LPDEST=printer1  
EXINIT='set showmode autoindent number'  
ENV=$HOME/.kshrc
```

- c. Change the line that reads:

```
export PATH
```

so that it reads:

```
export PATH EDITOR LPDEST EXINIT ENV
```

4. Use the Solaris Management Console to create a new user with the following characteristics. Exit the Solaris Management Console when you are finished.

User Name:	user9
User ID:	1009
Primary Group:	staff
Login Shell:	Korn
Password:	123pass

5. Log out, and log in again as user9. Open a terminal window.
6. Verify that the PATH, LPDEST, EDITOR, EXINIT, and ENV variables are set according to the changes you made in the /etc/skel/local.profile file.

```
$ echo $PATH  
$ echo $LPDEST  
$ echo $EDITOR  
$ echo $EXINIT  
$ echo $ENV
```

Do they match?

These variables should match the settings made in the local.profile file.

7. Create a file called .kshrc in user9's home directory.

```
$ cd  
$ vi .kshrc
```

Insert the following lines. A space follows the \$PWD\$ in the last line.

```
set -o noclobber  
set -o ignoreeof  
alias h=history  
alias c=clear  
PS1='$PWD$ '
```

Exercise: Modifying Initialization Files (Level 3)

8. Log out, and then log in again as user9. Open a terminal window, and verify that your new variables work.

```
/export/home/user9 $ cd /tmp  
/tmp $ cd  
/export/home/user9 $ c  
export/home/user9 $ h  
1      cd /tmp  
2      cd  
3      c  
4      h
```

Do they work?

These variables should function according to the values set in .kshrc. The prompt should reflect your current directory, and the aliases should clear the screen and present a history list.

9. Log out, and log in again as the root user. Use the useradd command to create a new user account called user10 with the following characteristics:

User Name:	user10
User ID:	1010
Primary Group:	10
Login Shell:	Korn
Home Directory:	/export/home/user10
Comment:	SA-200 Student
Password:	cangetin

```
# useradd -u 1010 -g 10 -d /export/home/user10 -m -s /bin/ksh -c "SA-200  
Student" user10  
64 blocks  
# passwd user10  
New password: cangetin  
Re-enter new password: cangetin
```

10. Log out, and log in again as user10. Open a terminal window. What shell initialization files exist in your home directory?

```
$ ls -la
```

.profile, local.profile, local.login, local.cshrc

Which of these are the same as the /etc/skel/local.profile file?

The local.profile file.

11. Copy the local.profile file to the .profile file.

```
$ cp local.profile .profile
```

12. Log out, and log in again as user10. Verify that the variables set for the user9 login are also set for this login.

```
$ echo $PATH  
$ echo $LPDEST  
$ echo $EDITOR  
$ echo $EXINIT  
$ echo $ENV
```

Do they match?

These variables should match the settings made in the local.profile file.

13. Log out, and log in again as the root user.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 11

Performing System Security

Objectives

Upon completion of this module, you should be able to:

- Monitor system access
- Switch users on a system
- Control system access
- Restrict access to data in files

The course map in Figure 11-1 shows how this module fits into the current instructional goal.

Performing User and Security Administration

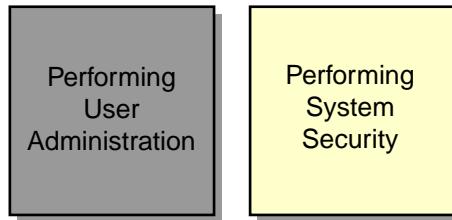


Figure 11-1 Course Map

Monitoring System Access

All systems should be monitored routinely for unauthorized user access. You can determine who is or who has been logged into the system by executing commands and examining log files.

Displaying Users on the Local System

The `who` command displays a list of users currently logged in to the local system. It displays each user's login name, the login device (TTY port), the login date and time. The command reads the binary file `/var/adm/utmpx` to obtain this information and information about where the users logged in from.

If a user is logged in remotely, the `who` command displays the remote host name, or Internet Protocol (IP) address in the last column of the output.

```
# who
root      console      Oct 17 08:21(:0)
root      pts/4        Oct 17 08:21(:0.0)
root      pts/5        Oct 17 08:21(:0.0)
user5     pts/6        Oct 17 09:20(sys-03)
root      pts/7        Oct 17 09:20(:0.0)
user3     pts/8        Oct 17 09:21(localhost)
#
```

The second field displayed by the `who` command defines the user's login device, which is one of the following:

- `console` – The device used to display system boot and error messages
- `pts` – The pseudo device that represents a login or window session without a physical device
- `term` – The device physically connected to a serial port, such as a terminal or a modem

Note – The `who` command has many options, one of which is the `-m` option. The `who -m` command outputs information about only the current terminal window.



Displaying Users on Remote Systems

The **rusers** command produces output similar to that of the **who** command, but it displays a list of the users logged in on local and remote hosts. The list displays the user's name and the host's name in the order in which the responses are received from the hosts.

A remote host responds only to the **rusers** command if its **rpc.rusersd** daemon is enabled. The **rpc.rusersd** daemon is the network server daemon that returns the list of users on the remote hosts.

Note – The **rusers** facility is managed using the Service Management Facility (SMF).



To see whether the **rusers** facility is online, issue the command:

```
# svcs -a | grep rusers
online      17:00:48 svc:/network/rpc/rusers:default
```

The following is the command format for the **rusers** command:

```
rusers -options hostname
```

The **rusers -1** command displays a long list of the login names of users who are logged in on local and remote systems. The output displays the name of the system into which a user is logged, the login device (TTY port), the login date and time, the idle time, and the login host name. If the user is not idle, no time is displayed in the idle time field. The term idle means that the user is not actively doing anything at the time on the terminal, which would denote the user is probably at screen lock or away from the terminal.

The following is an example of the **rusers** command:

```
# rusers -1
Sending broadcast for rusersd protocol version 3...
root      sys-02:console          Oct 17 08:21      (:0)
user5     sys-02:pts/6            Oct 17 09:20      1 (sys-03)
user3     sys-02:pts/8            Oct 17 09:21      1 (localhost)
root      fe80::203:baff:f:pts/2 Oct 17 09:18      1 (sys-02)
root      sys-03:pts/2           Oct 17 09:18      1 (sys-02)
Sending broadcast for rusersd protocol version 2...
```

Displaying User Information

To display detailed information about user activity that is either local or remote, use the `finger` command.

The `finger` command displays:

- The user's login name
- The home directory path
- The login time
- The login device name
- The data contained in the comment field of the `/etc/passwd` file (usually the user's full name)
- The login shell
- The name of the host, if the user is logged in remotely, and any idle time

The following is the command format for the `finger` command:

```
finger [-bfhilmpqsw] [username...]
finger [-l] [ username@hostname1 [ @hostname ] ]
```

The `-m` option matches arguments only on `username` (not the first or last name that might appear in the comment field of `/etc/passwd`).

To display information for `user5`, perform the command:

```
# finger -m user5
Login name: user5
Directory: /export/home/user5          Shell: /bin/ksh
On since Oct 17 09:20:43 on pts/6 from sys-03
1 minute 50 seconds Idle Time
No unread mail
No Plan.
```

If users create the standard ASCII files `.plan` or `.project` in their home directories, the content of those files is shown as part of the output of the `finger` command.

These files are traditionally used to outline a user's current plans or projects and must be created with file access permissions set to 644 (`rw-r--r--`).



Note – You get a response from the finger command only if the network/finger service is enabled.

```
# inetadm | grep finger
enabled    online      svc:/network/finger:default
```

Displaying a Record of Login Activity

Use the last command to display a record of all logins and logouts with the most recent activity at the top of the output. The last command reads the binary file /var/adm/wtmpx, which records all logins, logouts, and reboots.

Each entry includes the user name, the login device, the host that the user is logged in from, the date and time that the user logged in, the time of logout, and the total login time in hours and minutes, including entries for system reboot times.

The output of the last command can be extremely long. Therefore, you might want to use it with the *-n number* option to specify the number of lines to display.

The following is an example of the last command:

```
# last
user3  pts/8      localhost      Sun Oct 17 09:21  still logged in
root   console    :0            Sun Oct 17 08:21  still logged in
reboot system boot          Sun Oct 17 08:00
wtmp begins Fri Oct 15 11:36
```

(output truncated)

You can use the last command also to display information about an individual user if you supply the user's login name as an argument.

```
# last user5
user5  pts/6      sys-03        Sun Oct 17 09:20  still logged in
user5  pts/7      localhost     Sun Oct 17 09:13 - 09:15  (00:02)
(output truncated)
```

To view the last five system reboot times only, perform the command:

```
# last -5 reboot
reboot      system boot          Sun Oct 17 08:00
reboot      system down         Sun Oct 17 03:27
reboot      system boot          Sun Oct 17 03:16
reboot      system down         Sun Oct 17 03:27
reboot      system boot          Sun Oct 17 03:16
```

Recording Failed Login Attempts

When a user logs in to a system either locally or remotely, the login program consults the /etc/passwd and the /etc/shadow files to authenticate the user. It verifies the user name and password entered.

If the user provides a login name that is in the /etc/passwd file and the correct password for that login name, the login program grants access to the system.

If the login name is not in the /etc/passwd file or the password is not correct for the login name, the login program denies access to the system.

You can log failed login attempts in the /var/adm/loginlog file. This is a useful tool if you want to determine if attempts are being made to break into a system.

By default, the loginlog file does not exist. To enable logging, you should create this file with read and write permissions for the root user only, and it should belong to the sys group.

```
# touch /var/adm/loginlog
# chown root:sys /var/adm/loginlog
# chmod 600 /var/adm/loginlog
```

All failed command-line login activity is written to this file automatically after five consecutive failed attempts.

The loginlog file contains one entry for each of the failed attempts. Each entry contains the user's login name, login device (TTY port), and time of the failed attempt.

If there are fewer than five consecutive failed attempts, no activity is logged to this file. This value is configured by setting the appropriate `syslog_failed_login` parameter in the `/etc/default/login` file:

```
# tail -15 /etc/default/login

# RETRIES determines the number of failed logins that will be
# allowed before login exits. Default is 5 and maximum is 15.
# If account locking is configured (user_attr(4)/policy.conf(4))
# for a local user's account (passwd(4)/shadow(4)), that account
# will be locked if failed logins equals or exceeds RETRIES.
#
#RETRIES=5
#
# The SYSLOG_FAILED_LOGINS variable is used to determine how many failed
# login attempts will be allowed by the system before a failed login
# message is logged, using the syslog(3) LOG_NOTICE facility. For
example,
# if the variable is set to 0, login will log -all- failed login
attempts.
#
#SYSLOG_FAILED_LOGINS=5
```

Switching Users on a System

You should avoid logging in directly as the root user. This precaution helps protect the system from unauthorized access, because it reduces the likelihood that the system will be left unattended with the root user logged in. Also, critical mistakes are less likely to occur if you perform routine work as a regular system user.



Caution – As the system administrator, you should log in to a system as a regular user, and then switch to the root account only to perform administrative tasks.

Introducing the su Command

Use the `su` command to switch to the superuser or another user without logging out and back in as that user.

The following is the command format for the `su` command:

```
su - username
```

If no user name is given, then the `su` command attempts to switch to the root user.

To use the `su` command, supply the appropriate password unless you are already the root user. The root user can run the `su` command without passwords.

If the password is correct, the `su` command creates a new shell process, as specified in the shell field of that user account's `/etc/passwd` file entry.

The `su -` (dash) option specifies a complete login by reading all of the user's shell initialization files. The `-` (dash) option changes your work environment to what would be expected if you had logged in directly as that specified user. It also changes the user's home directory.

When you run the `su` command, the Effective User ID (EUID) and the Effective Group ID (EGID) are changed to the new user to whom you have switched.

Access to files and directories is determined by the value of the EUID and EGID for the effective user, rather than by the UID and GID numbers of the original user who logged in to the system.

Using the whoami Command

The whoami command displays the name of the account, whose authorization you have switched to.

Note – The whoami command resides in the /usr/ucb directory.



For example, user1 is logged into the system under that login name. This user then runs the su command to become the root user and enters the root password. The whoami command displays the user's actual authorization for accessing directories and files, for example:

```
$ whoami
user1
$ pwd
/export/home/user1
$ su
password: EnterPassword
# whoami
root
# pwd
/export/home/user1
```

Using the who am i Command

To determine the login name of the original user, use the who command with the am i option.

To use the who am i command, at the shell prompt, type the su command and the login name of the user account to which you want to switch, and press Return. Type the password for the user account, and press Return.

For example, while logged in as user3, use the su command to switch to user5:

```
$ su user5
password: EnterPassword
$ who am i
user3 pts/10 Oct 17 09:25 (sys-02)
```

An alternative to the who am i command is the who -m command.

Switching to Another Regular User

To switch to another user and have that user's environment, use the **su** command as follows:

1. At the shell prompt, display your login name and path.

```
$ who am i  
user3      pts/10          Oct 17 09:25      (sys-02)  
$ pwd  
/export/home/user3
```

2. Enter the **su** command with the dash (-) option and the login name of the user to which you want to switch. Then, enter the password for the user.

```
$ su - user5  
Password: EnterPassword
```

3. To determine the login name of the actual user, perform the **whoami** command, and press Return.

```
$ whoami  
user5
```

4. To determine the current working directory, perform the **pwd** command. The location is the effective user's home directory.

```
$ pwd  
/export/home/user5
```

5. To display the login name of the original user, perform the **who am i** command.

```
$ who am i  
user3      pts/10          Oct 17 09:25      (sys-02)
```

6. To return to the original user status and home directory, perform the **exit** command.

```
$ exit  
$ pwd  
/export/home/user3
```

Becoming the root User

In the default system configuration, direct root logins are restricted to the console. This means that you cannot remotely log in to a system as root. To remotely log in to a host as the root user, you must log in as a regular user and then run the su command to become the root user.

To become the root user, use the su command as follows:

1. Log in from the login window as a regular user, such as user1.
2. At the shell prompt in a terminal window, perform the su command. Enter the root password.

```
$ su -
Password: EnterPassword
```

3. To display the original login, perform the who am i command.

```
# who am i
user3      pts/10          Oct 17 09:25      (sys-02)
```

4. To determine the login name of the user to which you have switched, perform the whoami command.

```
# whoami
root
```

5. To determine the current working directory, perform the pwd command.

```
# pwd
/
```

6. To exit the root session and return to the original user, perform the exit command.

```
# exit
$ pwd
/export/home/user3
$
```

Monitoring su Attempts

For security reasons, you must monitor who has been using the `su` command, especially those users who are trying to gain root access on the system. You can initiate the monitoring by setting two variables in the `/etc/default/su` file.

Note – There are many variables in the `/etc/default/su` file. This course presents only a small subset of the variables.



Contents of the `/etc/default/su` File

To display the contents of the `/etc/default/su` file, perform the command:

```
# cat /etc/default/su
#ident "@(#)su.dfl1.693/08/14 SMI" /* SVr4.0 1.2 */

# SULOG determines the location of the file used to log all su attempts
#
SULOG=/var/adm/sulog

# CONSOLE determines whether attempts to su to root should be logged
# to the named device
#
#CONSOLE=/dev/console
(output edited for brevity)
SYSLOG=YES
```

In the preceding example, unsuccessful attempts to use the `su` command to access the root account are logged to the `/var/adm/messages` file. The following is an example entry from that file:

```
Oct 16 12:35:47 sys-02 su: [ID 810491 auth.crit] 'su root' failed for
user3 on /dev/pts/2
```

The CONSOLE Variable in the /etc/default/su File

By default, the system ignores the CONSOLE variable in the /etc/default/su file because of the preceding comment (#) symbol. All attempts to use the su command are logged to the console, regardless of success or failure. Here is an example of output to the console:

```
Feb 2 09:50:09 host1 su: 'su root' failed for user1 on /dev/pts/4
Feb 2 09:50:33 host1 su: 'su user3' succeeded for user1 on /dev/pts/4
```

When the comment symbol is removed, the value of the CONSOLE variable is defined for the /dev/console file. Subsequently, an additional line of output for each successful attempt to use the su command to access the root account is logged to the console. Here is an example of logged su command activity:

```
Feb 2 11:20:07 host1 su: 'su root' succeeded for user1 on /dev/pts/4
SU 02/02 11:20 + pts/4 user1-root
```

The SULOG Variable in the /etc/default/su File

The SULOG variable in the /etc/default/su file specifies the name of the file in which all attempts to use the su command to switch to another user are logged. If the variable is undefined, the su command logging is turned off.

The /var/adm/sulog file is a record of all attempts by users on the system to execute the su command. Each time the su command is executed, an entry is added to the sulog file.

The entries in this file include the date and time the command was issued, whether it was successful (shown by the plus (+) symbol for success or the hyphen (-) symbol for failure), the device from which the command was issued, and, finally, the login and the effective identity.

The following is an example of entries from the /var/adm/sulog file:

```
# more /var/adm/sulog
SU 10/17 02:51 + ??? root-uucp
SU 10/17 09:26 + pts/10 user3-root
SU 10/17 09:27 + pts/10 user3-user5
SU 10/17 09:28 + pts/10 user3-user5
SU 10/17 09:28 + pts/10 user3-root
SU 10/17 09:29 - pts/10 user3-user4
```

Controlling System Access

The more access that is available over the network, the more beneficial it is for remote system users. However, unrestricted access and sharing of data and resources can create security problems.

A local host's remote security measures are generally based on an ability to validate, limit, or block operations from remote system users.

The /etc/default/login File



Note – There are many variables in the /etc/default/login file. This course, presents only a small subset of the variables.

The /etc/default/login file establishes default parameters for users when they log in to the system. The /etc/default/login file gives you the ability to protect the root account on a system. You can restrict root access to a specific device or to a console, or disallow root access altogether.

To display the contents of the /etc/default/login file, perform the command:

```
# cat /etc/default/login
(output edited for brevity)
# If CONSOLE is set, root can only login on that device.
# Comment this line out to allow remote login by root.
#
CONSOLE=/dev/console

# PASSREQ determines if login requires a password.
#
PASSREQ=YES
```

The CONSOLE Variable in the /etc/default/login File

You can set the CONSOLE variable in the /etc/default/login file to specify one of three possible conditions that restrict access to the root account:

- If the variable is defined as CONSOLE=/dev/console, the root user can log in only at the system console. Any attempt to log in as root from any other device generates the error message:

```
# rlogin host1
Not on system console
Connection closed.
```

- If the variable is not defined, such as #CONSOLE=/dev/console, the root user can log in to the system from any device across the network, through a modem, or using an attached terminal.



Caution – If the variable does not have a value assigned to it (for example CONSOLE=) then the root user cannot log in from anywhere, not even the console. The only way to become the root user on the system is to log in as a regular user and then become root by using the su command.



Note – You can confine root logins to a particular port with the CONSOLE variable. For example, CONSOLE=/dev/term/a permits the root user to log in to the system only from a terminal that is connected to Serial Port A.

The PASSREQ Variable in the /etc/default/login File

When the PASSREQ variable in the /etc/default/login file is set to the default value of YES, then all users who had not been assigned passwords when their accounts were created are required to enter a new password as they log in for the first time. If this variable is set to NO, then null passwords are permitted. This variable does not apply to the root user.

File Transfer Protocol (FTP) Access

The Solaris OS provides an American Standard Code for Information Interchange (ASCII) file named `/etc/ftpd/ftpusers`. The `/etc/ftpd/ftpusers` file lists the names of users who are *prohibited* from connecting to the system through the FTP protocol.

Each line entry in this file contains a login name for a restricted user, for example:

username

The FTP server daemon `in.ftpd` reads the `/etc/ftpd/ftpusers` file when an FTP session is invoked. If the login name of the user matches one of the listed entries, it rejects the login session and sends the `Login failed` error message.

By default, the `/etc/ftpd/ftpusers` file lists the following system account entries:

- root
- daemon
- bin
- sys
- adm
- lp
- uucp
- nuucp
- smmsp
- listen
- gdm
- webservd
- nobody
- noaccess
- nobody4

As with any login name that you can add, these entries must match the user account names located in the /etc/passwd file.

The root entry is included in the ftpusers file as a security measure. The default security policy is to disallow remote logins for the root user. The policy is also followed for the default value set as the CONSOLE entry in the /etc/default/login file.

The /etc/hosts.equiv and \$HOME/.rhosts Files

Typically, when a remote user requests rlogin, rcp, or rsh access to a local host, the first file read by the local host is its /etc/passwd file. An entry for that particular user in this file enables that user to log in to the local host from a remote system. If a password is associated with that account, then the remote user is required to supply this password at log in to gain system access.

If there is no entry in the local host's /etc/passwd file for the remote user, access is denied.

The /etc/hosts.equiv and \$HOME/.rhosts files bypass this standard password-based authentication to determine if a remote user is allowed to access the local host, with the identity of a local user.

These files provide a remote authentication procedure to make that determination.

This procedure first checks the /etc/hosts.equiv file and then checks the \$HOME/.rhosts file in the home directory of the local user who is requesting access. The information contained in these two files (if they exist) determines if remote access is granted or denied.

The information in the /etc/hosts.equiv file applies to the entire system, while individual users can maintain their own \$HOME/.rhosts files in their home directories.

Figure 11-2 shows the flow of remote access authentication.

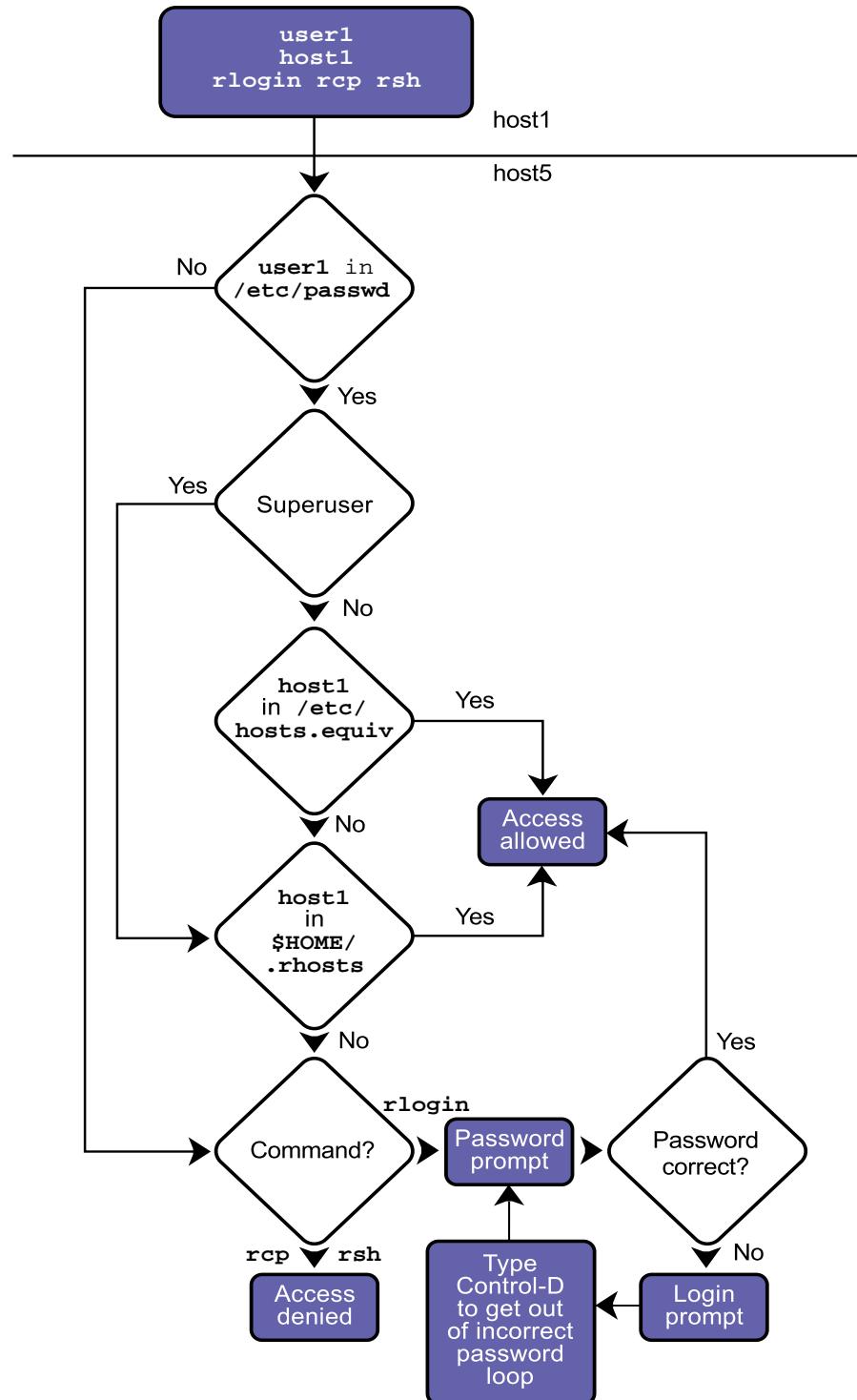


Figure 11-2 Remote Access Authentication

Entries in the /etc/hosts.equiv and \$HOME/.rhosts Files

While the /etc/hosts.equiv and \$HOME/.rhosts files have the same format, the same entries in each file have different effects.

Both files are formatted as a list of one-line entries, which can contain the following types of entries:

```
hostname  
hostname username  
+
```

If hostname is used, then users with the same UID from that named host are granted access without a password.

If hostname username is used, then the named user from that named host is granted access without a password.

Caution – If the + sign is used, this is granting any user from any host access without a password. This is particularly dangerous.



The host names in the /etc/hosts.equiv and \$HOME/.rhosts files must be the official name of the host, not one of its alias names.



Note – When logging in to a number of different systems, you can run the uname -n command to determine on which system you are currently logged in.

The /etc/hosts.equiv File Rules

For regular users, the /etc/hosts.equiv file identifies remote hosts and remote users who are considered to be *trusted*.



Note – The /etc/hosts.equiv file is not checked at all if the remote user requesting local access is the root user.

If the local host's `/etc/hosts.equiv` file contains the host name of a remote host, then all regular users of that remote host are trusted and do not need to supply a password to log in to the local host. This is provided so that each remote user is known to the local host by having an entry in the local `/etc/passwd` file; otherwise, access is denied.

This functionality is particularly useful for sites where regular users commonly have accounts on many different systems, eliminating the security risk of sending ASCII passwords over the network.

The `/etc/hosts.equiv` file does not exist by default. It must be created if trusted remote user access is required on the local host.

The `$HOME/.rhosts` File Rules

While the `/etc/hosts.equiv` file applies system-wide access for non-root users, the `.rhosts` file applies to a specific user.

All users, including the root user, can create and maintain their own `.rhosts` files in their home directories.

For example, if you run an `rlogin` process from a remote host to gain root access to a local host, the `/.rhosts` file is checked in the root home directory on the local host.

If the remote host name is listed in this file, it is a trusted host, and, in this case, root access is granted on the local host. The `CONSOLE` variable in the `/etc/default/login` file must be commented out for remote root logins.

The `$HOME/.rhosts` file does not exist by default. You must create it in the user's home directory.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: User Access (Level 1)

In this exercise, you complete the following tasks:

- Log failed login attempts
- Use the commands `finger`, `last`, `rusers`, `su`, and `whoami`
- Examine the `sulog` file
- Change the `/etc/default/login` file to allow `root` logins from any terminal
- Change the `/etc/ftpd/ftpusers` file to allow FTP access as the `root` user
- Create a `/.rhosts` file to allow `root` access from another system

Preparation

This lab requires two systems. Each system lists the other in its `/etc/inet/hosts` file. The lab also requires two specific users, `user9` and `user3`, on both systems. Both users should use the password `123pass`. Refer to the lecture notes as necessary to perform the steps listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Create a log file to record failed login attempts. Use the command-line login to make five failed login attempts. List the contents of the log file. Use commands to display information for user9 on both your system and your partner's system.
(Steps 1–7 in the Level 2 lab)
- Identify when the first root login session on your system occurred and how long the session lasted. Identify when your system last booted. List the users logged in on all systems on your network and on just your partner's system.
(Steps 8–11 in the Level 2 lab)
- Change your user identity from the root user to user9, both with and without the – (dash) option. Record the differences. List effective and real user identity during your su sessions. Locate the su log and identify which user initiated your su attempts.
(Steps 12–18 in the Level 2 lab)
- As the root user, attempt to log into your partner's system. Record error messages. Change the CONSOLE variable on your partner's system to allow root logins from any terminal. Attempt to access your partner's system again.
(Steps 19–21 in the Level 2 lab)
- As the root user, attempt to use the ftp command to access your partner's system. Change the ftp permissions file to allow root access to your partner's system.
(Step 22 in the Level 2 lab)
- As the root user, attempt to use the rlogin command to access your partner's system. Ask your partner to create a / .rhosts file that lists your system name. Attempt to use the rlogin command to access your partner's system again.
(Step 23 in the Level 2 lab)

Exercise: User Access (Level 2)

In this exercise, you complete the following tasks:

- Log failed login attempts
- Use the commands `finger`, `last`, `rusers`, `su`, and `whoami`
- Examine the `sulog` file
- Change the `/etc/default/login` file to allow root logins from any terminal
- Change the `/etc/ftpd/ftpusers` file to allow FTP access as the root user
- Create a `/.rhosts` file to allow root access from another system

Preparation

This lab requires two systems. Each system lists the other in its `/etc/inet/hosts` files. It also requires two specific users, `user9` and `user3`, on both systems. Both users should use the password `123pass`. Refer to the lecture notes as necessary to perform the steps listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Create the file `/var/adm/loginlog`. Use the command-line login to make five failed login attempts. List the contents of the `/var/adm/loginlog` file. Use the `finger` command to display information for user9 on both your system and your partner's system.
- Use the `last` command to identify when the first root login session on your system occurred and how long the session lasted. Use the `last` command to learn when your system last booted. Use the `rusers` command to list the users logged in on all systems on your network and on just your partner's system.
- Use the `su` command to change your user identity from the root user to user9, both with and without the - (dash) option. Record the differences. Use the `whoami` and `who am i` commands to list your effective and real user identity during your `su` sessions. Locate the `su` log declared in the `/etc/default/su` file, and identify which user initiated your `su` attempts.
- As the root user, attempt a session to your partner's system by using the `telnet` command. Record error messages. Change the `CONSOLE` variable on your partner's system to allow root logins from any terminal. Attempt the `telnet` session again.
- As the root user, attempt to use the `ftp` command to access your partner's system. Change the `/etc/ftpd/ftpusers` file to allow root access to your partner's system.
- As the root user, attempt to use the `rlogin` command to access your partner's system. Ask your partner to create a `/.rhosts` file that lists your system name. Attempt to use the `rlogin` command to access your partner's system again.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change the directory to `/var/adm`.
2. Use the `touch` command to create a file called `loginlog`. (Ensure permissions are set to read and write for the root user only.) If necessary, set the group ownership to `sys`.

Exercise: User Access (Level 2)

3. Log out. From the CDE Options menu, select the Command Line Login option. When the CDE login screen clears, press Return to obtain the command-line login prompt.
4. Enter `root` after the login prompt, but supply an incorrect password. Do this five times. After the fifth attempt, the CDE login screen appears again. Log in as `root`, and open a terminal window.
5. Examine the `/var/adm/loginlog` file. What does it contain?
6. Use the `finger` command to display information for the user called `user9`. What is the difference in the output between the `finger -m` command and the `finger` command with no option?
7. Use the `finger` command to display information for the same user on your partner's system. (You will need to reference your partner's system on the command line.) Try this with and without the `-m` option. Does the `-m` option change the output that the `finger` command displays?
8. Use the `last` command to display login and system reboot activity. When did the first `root` login occur, and how long did that session last?
9. Use the `last` command to display only system boot activity. When did the system last reboot?
10. Use the `rusers` command to list information about the users on all systems on your network segment.
11. Use the `rusers` command to list information for users on your partner's system. When, and on what terminal, did the first user listed log in?
12. Switch your user identity to that of `user9`. Do not use the `-` (dash) option.
13. Display some of the variables that define your environment.
14. Exit the `su` session and try to switch your user identity again, this time using the `-` (dash) option.

Are the values reported now correct for the user `root` or for `user9`?
15. Use the `whoami` and `who am i` commands to list your effective and real user identity.

What do these commands report?
16. Use the `su` command to change your user identity from `user9` to `user3`, and use the `whoami` and `who am i` commands again.

What do these commands report?

Exit both `su` sessions when you are finished.

17. Change the directory to /etc/default. Examine the /etc/default/su file, and record the value of the SULOG variable.
18. Display the file named by the SULOG variable, and identify the entry that relates to your last su command. Is user9 or the root user identified as the user who became user3?
19. As the user root, attempt to log in to your partner's system by using the telnet command. Was your attempt successful? What message appears?
20. On your partner's system, edit the /etc/default/login file, and change the line that reads:

CONSOLE=/dev/console

so that it reads:

#CONSOLE=/dev/console

21. As the root user, again attempt to log in to your partner's system by using the telnet command. If your login attempt is successful, exit the telnet session. If not, check the change you made in Step 20, and try again.
22. As the root user, attempt to use the ftp command to access your partner's system. Were you successful? Ask your partner to edit the /etc/ftpd/ftpusers file and comment out the root entry. Attempt to use the ftp command to access your partner's system again. List some files in the /tmp directory from the ftp> prompt.
23. As the root user, attempt to use the rlogin command to access your partner's system. Were you successful? Ask your partner to create a /.rhosts file and enter the name of your system on a line by itself. Attempt to use the rlogin command to access your partner's system again.

Exercise: User Access (Level 3)

In this exercise, you complete the following tasks:

- Log failed login attempts
- Use the commands `finger`, `last`, `rusers`, `su`, and `whoami`
- Examine the `sulog` file
- Change the `/etc/default/login` file to allow root logins from any terminal
- Change the `/etc/ftpd/ftpusers` file to allow FTP access as the root user
- Create a `/.rhosts` file to allow root access from another system

Preparation

This lab requires two systems that list each other in their `/etc/inet/hosts` files. It also requires two specific users, `user9` and `user3`, on both systems. Both users should use the password `123pass`. Refer to the lecture notes as necessary to perform the steps listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Create the file `/var/adm/loginlog`. Use the command-line login to make five failed login attempts. List the contents of the `/var/adm/loginlog` file. Use the `finger` command to display information for user9 on both your system and your partner's system.
- Use the `last` command to identify when the first `root` login session on your system occurred and how long the session lasted. Use the `last` command to learn when your system last booted. Use the `rusers` command to list the users logged in on all systems on your network and on just your partner's system.
- Use the `su` command to change your user identity from the `root` user to `user9`, both with and without the – (dash) option. Record the differences. Use the `whoami` and `who am i` commands to list your effective and real user identity during your `su` sessions. Locate the `su` log declared in the `/etc/default/su` file, and identify which user initiated your `su` attempts.
- As the `root` user, attempt a session to your partner's system by using the `telnet` command. Record error messages. Change the `CONSOLE` variable on your partner's system to allow `root` logins from any terminal. Attempt the `telnet` session again.
- As the `root` user, attempt to use the `ftp` command to access your partner's system. Change the `/etc/ftpd/ftpusers` file to allow `root` access to your partner's system.
- As the `root` user, attempt to use the `rlogin` command to access your partner's system. Ask your partner to create a `/.rhosts` file that lists your system name. Attempt to use the `rlogin` command to access your partner's system again.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change the directory to /var/adm.

```
# cd /var/adm
```

2. Use the touch command to create a file called loginlog. (Ensure permissions are set to read and write for the root user only.) If necessary, set the group ownership to sys.

```
# touch loginlog  
# chmod 600 loginlog  
# chgrp sys loginlog
```

3. Log out. From the CDE Options menu, select the Command Line Login option. When the CDE login screen clears, press Return to obtain the command-line login prompt.
4. Enter root after the login prompt, but supply an incorrect password. Do this five times. After the fifth attempt, the CDE login screen appears again. Log in as root, and open a terminal window.
5. Examine the /var/adm/loginlog file. What does it contain?

This file should contain a list of failed login attempts which appear similar to the following:

```
login:/dev/pts/2:Tue Dec 7 13:29:22 2004
```

6. Use the finger command to display information for the user called user9. What is the difference in output between the finger -m command and the finger command with no option?

```
# finger user9  
# finger -m user9
```

The finger command with no option lists all user accounts that have the string user in their names and comment fields. The finger -m command lists only the entry for the user named user9.

7. Use the finger command to display information for the same user on your partner's system. (You will need to reference your partner's system on the command line.) Try this with and without the -m option. Does the -m option change the output that the finger command displays?

```
# finger user9@hostname  
# finger -m user9@hostname
```

No.

8. Use the `last` command to display login and system reboot activity. When did the first root login occur, and how long did that session last?

```
# last
```

This information depends on the activity on your particular system.

9. Use the `last` command to display only system boot activity. When did the system last reboot?

```
# last reboot
```

This information depends on the activity on your particular system.

10. Use the `rusers` command to list information about the users on all systems on your network segment.

```
# rusers -l
```

11. Use the `rusers` command to list information about the users on your partner's system. When, and on what terminal, did the first user listed log in?

```
# rusers -l hostname
```

This information depends on the activity on your particular system.

12. Switch your user identity to that of user9. Do not use the - (dash) option.

```
# su user9
```

```
$
```

13. Display some of the variables that define your environment.

```
$ echo $LOGNAME  
$ echo $HOME
```

Are the values reported correct for the user root or for user9?

root

14. Exit the `su` session and try to switch your user identity again, this time using the - (dash) option.

```
$ exit  
# su - user9  
$ echo $LOGNAME  
$ echo $HOME
```

Are the values reported now correct for the user root or for user9?

user9

Exercise: User Access (Level 3)

15. Use the whoami and who am i commands to list your effective and real user identity.

```
$ /usr/ucb/whoami  
$ who am i
```

What do these commands report?

The /usr/ucb/whoami command displays the login name matching your effective UID, user9. The who am i command displays the login name matching your real UID, root.

16. Use the su command to change your user identity from user9 to user3, and use the whoami and who am i commands again.

```
$ su user3  
Password: 123pass  
$
```

What do these commands report?

```
$ /usr/ucb/whoami  
user3  
$ who am i  
root
```

Exit both su sessions when you are finished.

```
$ exit  
$ exit  
#
```

17. Change the directory to /etc/default. Examine the /etc/default/su file, and record the value of the SULOG variable.

```
# cd /etc/default  
# more su  
/var/adm/sulog
```

18. Display the file named by the SULOG variable, and identify the entry that relates to your last su command. Is user9 or the root user identified as the user who became user3?

```
# cat /var/adm/sulog  
root
```

19. As the root user, attempt to log in to your partner's system by using the telnet command. Was your attempt successful? What message appears?

```
# telnet hostname  
(telnet connection messages)
```

SunOS 5.10

```
login: root  
Password: cangetin
```

The login attempt should not succeed. It fails and the system sends the messages:

Not on system console
Connection closed by foreign host.

20. On your partner's system, edit the /etc/default/login file, and change the line that reads:

CONSOLE=/dev/console

so that it reads:

```
#CONSOLE=/dev/console
```

21. As the root user, again attempt to log in to your partner's system by using the telnet command. If your login attempt is successful, exit the telnet session. If not, check the change you made in Step 20, and try again.

```
# telnet host  
(telnet connection messages)
```

SunOS 5.10

```
login: root  
Password: cangetin  
Last login: Sun Oct 17 09:21:17 from localhost  
Sun Microsystems Inc. SunOS 5.10 s10_68 Sep. 20, 2004
```

```
# exit  
Connection closed by foreign host.  
#
```

Exercise: User Access (Level 3)

22. As the root user, attempt to use the `ftp` command to access your partner's system. Were you successful?

No, you should receive the message: Login incorrect. Login failed.

Ask your partner to edit the `/etc/ftpd/ftpusers` file and comment out the `root` entry. Attempt to use the `ftp` command to access your partner's system again. List some files in the `/tmp` directory from the `ftp>` prompt.

You should see files such as:

```
dtdbcache_:0  
sdtvolcheck402  
speckeysd.lock
```

23. As the root user, attempt to use the `rlogin` command to access your partner's system. Were you successful?

You should not be able to use the rlogin command to directly access your partner's system. You should be prompted for a password.

Ask your partner to create a `/ .rhosts` file and enter the name of your system on a line by itself. Attempt to use the `rlogin` command to access your partner's system again.

You should be able to use the rlogin command to log directly in to your partner's system now.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Restricting Access to Data in Files

After you have established login restrictions, the next task is to control access to the data on the systems. Of course, some users need to be allowed to read various files; other users need permission to change and delete files, and there are some files that no regular user should be able to access.

Users who need to share files should be in the same group in the /etc/group file.

Note – In general, you use file access permissions to determine which users or groups have permission to read, modify, or delete files.



Determining a User's Group Membership

The groups command displays group memberships for the user.

The command format for the groups command is:

```
groups [username]
```

For example, to see which groups you are a member of, perform the command:

```
# groups  
other root bin sys adm uucp mail tty lp nuucp daemon
```

To list the groups to which a specific user is a member, use the groups command with the user's name, such as user5, as an argument.

```
# groups user5  
staff class sysadmin
```

Identifying a User Account

You use the `id` command to further identify users by listing their UID number, user name, GID number, and group name. This information is useful when you are troubleshooting file access problems for users.

The `id` command also returns the EUID number and name, and the EGID number and login name. For example, if you logged in as `user1` and then used the `su` command to become `user4`, the `id` command reports the information for the `user4` account.

The command format for the `id` command is:

```
id options username
```

To view your effective user account, perform the command:

```
$ id
uid=101(user1) gid=300(class)
```

To view account information for a specific user, use a user login name with the `id` command:

```
$ id user1
uid=101(user1) gid=300(class)
```

To view information about the secondary groups of a user, use the `-a` option and a user login name, such as `user1`:

```
$ id -a user1
uid=101(user1) gid=300(class) groups=14(sysadmin)
```

Changing File and Directory Ownership

You might need to use the `chown` command to change the original owner of a file or directory to another user account on the system. By default, only the root user can change the ownership of a file or directory.



Note – Regular users can be given permission to use the `chown` command to change the ownership of files and directories owned by them. Edit the `/etc/system` file, and add the parameter: `set rstchown=0` (zero). You need to reboot the system for the changes to take effect.

The command format for the chown command is:

```
chown option(s) user_name filename(s)
```

or

```
chown option(s) UID filename(s)
```

Note – The user must exist in the /etc/passwd file.



In this example, a user named user1 created a file called file7.

```
# cd /export/home/user1
# touch file7
# ls -l file7
-rw-r--r--  1 user1      staff          672 Jun 1 15:11  file7
#
```

You can use the chown command to give ownership of this file to a new user named user9. You use the ls command to verify the new ownership.

```
# chown user9 file7
# ls -l file7
-rw-r--r--  1 user9      staff          672 Jun 1 15:12  file7
#
```

After this sequence of commands, the file is owned by user9. This file is still in the home directory of user1. The two users need to determine if the file should be moved to a new directory location.

The ownerships of subdirectories can be changed in the same manner as files, as shown in the following examples:

In this example, user1 owns a directory called dir4.

```
$ ls -lR dir4
dir4:
total 0
-rw-r--r--  1 user1      staff          0 Mar 19 16:06 file1
-rw-r--r--  1 user1      staff          0 Mar 19 16:06 file2
-rw-r--r--  1 user1      staff          0 Mar 19 16:06 file3
$
```

You would use the chown command with the -R option to give ownership of this directory and all of its contents (files and subdirectories) to user2.

```
$ chown -R user2 dir4
$ ls -lR dir4
dir4:
total 0
-rw-r--r--  1 user2    staff          0 Mar 19 16:06 file1
-rw-r--r--  1 user2    staff          0 Mar 19 16:06 file2
-rw-r--r--  1 user2    staff          0 Mar 19 16:06 file3
$
```

The -R option makes the chown command recursive. It descends through the directory and any subdirectories, setting the ownership UID number as it moves through the directory hierarchy.

The chown command can also change both the individual and group ownership of a file or subdirectory simultaneously.

```
$ chown user3:class file2
```

Additionally, you can use the -R option to descend a directory hierarchy recursively, changing individual and group ownership of the directory and its contents simultaneously. The following example demonstrates this kind of change to the dir1 directory.

```
$ mkdir dir4
$ chown -R user3:class dir1
$ ls -lR dir1
dir1:
total 0
-rw-r--r--  1 user3    class         0 Mar 19 16:18 file1
-rw-r--r--  1 user3    class         0 Mar 19 16:18 file2
```

Changing File and Directory Group Membership

The `chgrp` command can be used by the root user or the file's owner to change the group ownership of files and directories to another group on the system. However, the file owner must also belong to the new group.



Note – Regular users can be given permission to use the `chgrp` command to change a file's or directory's group ownership to groups of which the user is not a member. Edit the `/etc/group` file, and add a parameter: `set rstchown=0` (zero). You must reboot the system for the changes to take effect.

The command format for the `chgrp` command is:

```
chgrp groupname filename(s)
```

or

```
chgrp GID filename(s)
```



Note – The *groupname* must exist in the `/etc/group` file.

For example, the `file4` file currently is a member of a group named `staff`.

```
# ls -l file4
-rw-rw-r-- 1 user1 staff          874 Jun 1 15:08 file4
#
```

You would use the `chgrp` command to give this file to a new group named `class` and use the `ls` command to verify the new group ownership.

```
# chgrp class file4
# ls -l file4
-rw-rw-r-- 1 user1 class          874 Jun 1 15:09 file4
#
```

When you are finished, all users who are members of the group called `class` have read and write access to this file.

Using File Permissions

Three types of special permissions are available for executable files and directories. These are:

- The setuid permission
- The setgid permission
- The Sticky Bit permission

The setuid Permission on Executable Files

When the set-user identification (setuid) permission is set on an executable file, a user or process that runs this executable file is granted access based on the owner of the file (usually the root user), instead of on who started the executable.

This setting allows a user to access files and directories that are typically accessible only by the owner of the executable. Note that many executable programs must be run by the root user, or by sys or bin to work properly.

Use the ls command to check the setuid permission.

```
# ls -l /usr/bin/su
-r-sr-xr-x  1 root      sys          22292 Jan 15 17:49 /usr/bin/su
```

The setuid permission displays as an “s” in the owner’s execute field.



Note – If a capital “S” appears in the owner’s execute field, it indicates that the setuid bit is on, and the execute bit “x” for the owner of the file is off or denied.

The root user and the owner can set the setuid permissions on an executable file by using the chmod command and the octal value 4###.

For example:

```
# chmod 4555 executable_file
```



Caution – Except for those setuid executable files that exist by default in the Solaris OS, you should disallow the use of setuid programs or at least restrict their use.

To search for files with setuid permissions and to display their full path names, perform the command:

```
# find / -perm -4000
```

The setgid Permission on Executable Files

The set-group identification (setgid) permission is similar to the setuid permission, except that when the process runs, it runs as if it were a member of the same group in which the file is a member. Also, access is granted based on the permissions assigned to that group.

For example, the `write` program has a `setgid` permission that allows users to send messages to other users' terminals.

Use the `ls` command to check the `setgid` permission.

```
# ls -l /usr/bin/write
-r-xr-sr-x 1 root      tty          11484 Jan 15 17:55 /usr/bin/write
```

The `setgid` permission displays as an "s" in the group's execute field.



Note – If a lowercase letter "l" appears in the group's execute field, it indicates that the `setgid` bit is on, and the execute bit for the group is off or denied. This indicates that mandatory file and record locking occurs during file access for those programs that are written to request locking.

The root user and the owner can set `setgid` permissions on an executable file by using the `chmod` command and the octal value 2###. Here is the command-line format:

```
# chmod 2555 executable_file
```

The setgid Permission on Directories

The `setgid` permission is a useful feature for creating shared directories.

When a `setgid` permission is applied to a directory, files created in the directory belong to the group of which the directory is a member.

For example, if a user has write permission in the directory and creates a file there, that file is a member of the same group as the directory and not the user's group.

To create a shared directory, you must set the setgid bit using symbolic mode. Here is the format for that mode:

```
# chmod g+s shared_directory
```

To search for files with setgid permissions and display their full path names, perform the command:

```
# find / -perm -2000
```

Sticky Bit Permission on Public Directories

The Sticky Bit is a special permission that protects the files within a publicly writable directory.

If the directory permissions have the Sticky Bit set, a file can be deleted only by the owner of the file, the owner of the directory, or by the root user. This prevents a user from deleting other users' files from publicly writable directories.

Use the ls command to determine if a directory has the Sticky Bit permission set.

```
# ls -ld /tmp
drwxrwxrwt  6  root    sys        719 May 31 03:30      /tmp
```

The Sticky Bit displays as the letter "t" in the execute field for other.



Note – If a capital "T" appears in the execute field for other, it indicates that the Sticky Bit is on; however, the execute bit is off or denied.

The root user and the owner can set the Sticky Bit permission on directories by using the chmod command and the octal value 1# ##. Here is the command-line format:

```
# chmod 1777 public_directory
```

To search for directories that have Sticky Bit permissions and display their full path names, execute the following command:

```
# find / -type d -perm -1000
```



Note – For more detailed information on the Sticky Bit, execute the man sticky command.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Restricting Access to Data on Systems (Level 1)

In this exercise, you complete the following tasks:

- Practice using commands related to user identity and file ownership
- Assign a user to the sysadmin group
- Assign special file permissions to files

Preparation

Refer to lecture notes as necessary to perform the steps listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Using the commands presented in the lecture, identify the groups of which root is a member. Compare the output from these commands. Add a user account called user11 with the useradd command. Verify the list of groups of which user11 is a member. Use the Solaris Management Console to create a new user account called user12. Add user11 to the sysadmin group.
(Steps 1–7 in the Level 2 lab)
- Log in as user11 and create a new file called file1. Attempt to change its user ownership. Record error messages. Change the group ownership of file1 to sysadmin. Switch the user identity to the root user, and change ownership of file1 to user12.
(Steps 8–11 in the Level 2 lab)

Exercise: Restricting Access to Data on Systems (Level 1)

- As user11, create a new file called `file2`. Set `setuid` and `setgid` permissions on `file2`. Remove all execute permissions from `file2`. Record the permissions listed as you change them.
(Steps 12–15 in the Level 2 lab)
- Record the permissions associated with the `/tmp` directory. As user11, create a new file called `test1` in the `/tmp` directory. As user12, attempt to remove this file. Record the result. As user11, create a new directory called `dir1` in `/export/home/user11`. Set permissions for the `dir1` directory to 777. Create a file called `test2` in the `dir1` directory. As user12 attempt to remove this file. Record the result. Log in again as the root user.
(Steps 16–21 in the Level 2 lab)

Exercise: Restricting Access to Data on Systems (Level 2)

In this exercise, you complete the following tasks:

- Practice using commands related to user identity and file ownership
- Assign a user to the sysadmin group
- Assign special file permissions to files

Preparation

Refer to lecture notes as necessary to perform the steps listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Using the commands groups, id, and id -a, identify the groups of which the root user is a member. Compare the output from these commands. Add a user account called user11 with the useradd command. Verify the list of groups of which user11 is a member. Use the Solaris Management Console to create a new user account called user12. Add user11 to the sysadmin group.
- Log in as user11 and create a new file called file1. Attempt to change its user ownership. Record error messages. Change the group ownership of file1 to sysadmin. Switch your user identity to the root user, and change ownership of file1 to user12.

- As user11, create a new file called `file2`. Use the `chmod` command to set `setuid` and `setgid` permissions on `file2`. Use the `chmod` command to remove all execute permissions from `file2`. Record the permissions listed as you change them.
- Record the permissions associated with the `/tmp` directory. As user11, create a new file called `test1` in the `/tmp` directory. As user12, attempt to remove this file. Record the result. As user11, create a new directory called `dir1` in `/export/home/user11`. Set permissions for the `dir1` directory to 777. Create a file called `test2` in the `dir1` directory. As user12 attempt to remove this file. Record the result. Log in again as the root user.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. Use the `groups` command to display the groups of which root is a member. Record the list that the `groups` command displays.
2. Use the `id` command both without and then with the `-a` option.
Does the `id` command report the primary or a secondary group for the root user?
Compare the `id -a` command output with that from the `groups` command in Step 1. What additional information does the `id -a` command provide?
3. Use the `useradd` command to create a new user account called `user11` with the following characteristics:

User Name:	user11
User ID:	1011
Primary Group:	10
Login Shell:	Korn
Home Directory:	<code>/export/home/user11</code>
Comment:	SA200 User
Password:	123pass

4. List the groups of which `user11` is a member.

5. Open a terminal window, and launch the Solaris Management Console.
6. Open the User Accounts tool. Select Add User from the Action menu. Then select From Template. Create a user account from the following information. Exit the Solaris Management Console when you are finished.

User Name: user12

User ID: 1012

Password: 123pass

7. From a terminal window, use the `usermod` command to add `user11` to group 14. Verify that the change took place. Log out.
8. Log in as `user11`. Open a terminal window, and use the `touch` command to create a file called `file1`. Verify that `user11` and the group `staff` own `file1`.
9. Attempt to change the owner of `file1` from `user11` to `user12`. What error message displays?
10. Attempt to change the group ownership of `file1` from `staff` to `sysadmin`. Verify the change. Did it work?
11. Switch your user identity to the root user, and change the directory to `/export/home/user11`. Change the owner of `file1` from `user11` to `user12`. Verify the change. Did it work? Exit your `su` session when you are finished.
12. In the home directory for `user11`, use the `touch` command to create a file called `file2`. Display and record the permissions associated with `file2`.
13. Use the `chmod` command to add `setuid` and execute permissions to `file2`. Display and record the permissions associated with `file2`. What changed?
14. Use the `chmod` command to add `setuid` and `setgid` permissions to `file2`. Display and record the permissions associated with `file2`. What changed?
15. Use the `chmod` command with octal arguments to remove all execute permissions from `file2`. Display and record the permissions associated with `file2`. What changed?

Exercise: Restricting Access to Data on Systems (Level 2)

16. Change the directory to / (root), and list the permissions associated with the /tmp directory. Is the Sticky Bit set on /tmp? Do all users have write permission in the /tmp directory?
17. Change the directory to /tmp. Create a file called test1 in the /tmp directory. Verify that user11 and the group staff own test1 and that 644 (rw-r--r--) permissions apply. Do they?
18. Switch your user identity to user12. In the /tmp directory, attempt to remove the test1 file. What messages appear? Exit your su session when you are finished.
19. In the home directory for user11, create a directory called dir1. Change permissions for the dir1 directory to 777. Create a file called test2 below the dir1 directory.
20. Switch your user identity to user12. Attempt to remove the file test2 from the dir1 directory. Verify that the test2 file no longer exists. Exit your su session when you are finished.
21. Log out, and log in again as the root user.

Exercise: Restricting Access to Data on Systems (Level 3)

In this exercise, you complete the following tasks:

- Practice using commands related to user identity and file ownership
- Assign a user to the sysadmin group
- Assign special file permissions to files

Preparation

Refer to lecture notes as necessary to perform the steps listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Using the commands groups, id, and id -a, identify the groups of which the root user is a member. Compare the output from these commands. Add a user account called user11 by using the useradd command. Verify the list of groups of which user11 is a member. Use the Solaris Management Console to create a new user account called user12. Add user11 to the sysadmin group.
- Log in as user11 and create a new file called file1. Attempt to change its user ownership. Record error messages. Change the group ownership of file1 to sysadmin. Switch your user identity to the root user, and change ownership of file1 to user12.

Exercise: Restricting Access to Data on Systems (Level 3)

- As user11, create a new file called `file2`. Use the `chmod` command to set `setuid` and `setgid` permissions on `file2`. Use the `chmod` command to remove all execute permissions from `file2`. Record the permissions listed as you change them.
- Record the permissions associated with the `/tmp` directory. As user11, create a new file called `test1` in the `/tmp` directory. As user12, attempt to remove this file. Record the result. As user11, create a new directory called `dir1` in `/export/home/user11`. Set permissions for the `dir1` directory to 777. Create a file called `test2` in the `dir1` directory. As user12 attempt to remove this file. Record the result. Log in again as the root user.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Use the `groups` command to display the groups of which root is a member. Record the list that the `groups` command displays.

```
# groups
```

```
other root bin sys adm uucp mail tty lp nuucp  
daemon
```

2. Use the `id` command both without and then with the `-a` option.

```
# id
```

Does the `id` command report the primary or a secondary group for the root user?

The id command reports the primary group.

```
# id -a
```

Compare the `id -a` command output with that from the `groups` command in Step 1. What additional information does the `id -a` command provide?

The id -a command reports group ID numbers in addition to group names for all groups.

3. Use the useradd command to create a new user called user11 with the following characteristics:

User Name:	user11
User ID:	1011
Primary Group:	10
Login Shell:	Korn
Home Directory:	/export/home/user11
Comment:	SA200 User
Password:	123pass

```
# useradd -u 1011 -g 10 -d /export/home/user11 -m -s /bin/ksh -c "SA200
User" user11
64 blocks
# passwd user11
New password: 123pass
Re-enter new password: 123pass
passwd (SYSTEM): passwd successfully changed for user11
#
```

4. List the groups of which user11 is a member.

```
# id -a user11
staff
```

5. Open a terminal window, and run the Solaris Management Console.

```
# smc &
```

6. Open the User Accounts tool. Select Add User from the Action menu. Then select With Template. Create a user account from the following information. Exit the Solaris Management Console when you are finished.

User Name:	user12
User ID:	1012
Password:	123pass

7. From a terminal window, use the usermod command to add user11 to group 14. Verify that the change took place. Log out.

```
# usermod -G 14 user11
# id -a user11
```

Exercise: Restricting Access to Data on Systems (Level 3)

8. Log in as user11. Open a terminal window, and use the touch command to create a file called file1. Verify that user11 and the group staff own file1.

```
$ touch file1  
$ ls -l file1
```

9. Attempt to change the owner of file1 from user11 to user12. What error message appears?

```
$ chown user12 file1  
chown: file1: Not owner
```

10. Attempt to change the group ownership of file1 from staff to sysadmin. Verify the change. Did it work?

```
$ chgrp sysadmin file1  
$ ls -l file1
```

Yes.

11. Switch your user identity to the root user, and change the directory to /export/home/user11. Change the owner of file1 from user11 to user12. Verify the change. Did it work? Exit your su session when you are finished.

```
$ su -  
Password: cangetin  
# pwd  
/  
# cd /export/home/user11  
# chown user12 file1  
# ls -l  
-rw-r--r--    1 user12 sysadmin 0 Apr 17  2002 file1  
# exit  
$
```

Yes.

12. In the home directory for user11, use the touch command to create a file called file2. Display and record the permissions associated with file2.

```
$ touch file2  
$ ls -l file2
```

The permissions for file2 should read -rw-r--r-.

13. Use the chmod command to add setuid and execute permissions to file2. Display and record the permissions associated with file2. What changed?

```
$ chmod 4555 file2  
$ ls -l file2
```

The permissions for file2 should read -r-sr-xr-x.

14. Use the `chmod` command to add `setuid` and `setgid` permissions to `file2`. Display and record the permissions associated with `file2`. What changed?

```
$ chmod 6555 file2  
$ ls -l file2
```

The permissions for file2 should read -r-sr-sr-x.

15. Use the `chmod` command with octal arguments to remove all execute permissions from `file2`. Display and record the permissions associated with `file2`. What changed?

```
$ chmod 6444 file2  
$ ls -l file2
```

The permissions for file2 should read -r-Sr-lr--.

16. Change the directory to `/` (root), and list the permissions associated with the `/tmp` directory. Is the Sticky Bit set on the `/tmp` directory? Do all users have write permission in `/tmp`?

```
$ cd /  
$ ls -ld tmp
```

Yes to both.

17. Change the directory to `/tmp`. Create a file called `test1` in the `/tmp` directory. Verify that `user11` and the group `staff` own `test1` and that `644` (`rw-r--r--`) permissions apply. Do they?

```
$ cd tmp  
$ touch test1  
$ ls -l test1
```

Yes.

18. Switch your user identity to `user12`. In the `/tmp` directory, attempt to remove the `test1` file. What messages appear? Exit your `su` session when you are finished.

```
$ su user12  
Password: 123pass  
$ rm test1  
rm: test1: override protection 644 (yes/no)? y  
rm: test1 not removed: Permission denied  
$ exit  
$
```

Exercise: Restricting Access to Data on Systems (Level 3)

19. In the home directory for user11, create a directory called dir1. Change permissions for the dir1 directory to 777. Create a file called test2 below the dir1 directory.

```
$ cd  
$ mkdir dir1  
$ chmod 777 dir1  
$ touch dir1/test2
```

20. Switch your user identity to user12. Attempt to remove the file test2 from the dir1 directory. Verify that the test2 file no longer exists. Exit your su session when you are finished.

```
$ su user12  
Password: 123pass  
$ rm dir1/test2  
$ ls -l dir1  
$ exit  
$
```

21. Log out, and log in again as the root user.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 12

Configuring and Using Printer Services

Objectives

Upon completion of this module, you should be able to:

- Identify network printing fundamentals
- Configure and administer printer services
- Start and stop the line printer (LP) print service
- Specify a destination printer
- Use the LP print service
- Use common print commands

The course map in Figure 12-1 shows how this module fits into the current instructional goal.

Managing Network Printers and System Processes

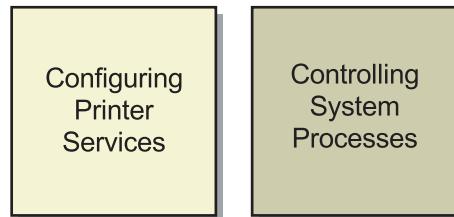


Figure 12-1 Course Map

Introducing Network Printing Fundamentals

The Solaris OS LP print service provides a complete printing environment that allows the sharing of printers across systems and a set of software utilities that enable users to print files while they continue to work on other tasks.

In this Solaris release, modifications have been made to incorporate support for a wide array of printers. This functionality differs greatly from previous Solaris software releases.

In previous releases, it was only possible to print to printers that understood PostScript™ natively, or plain ASCII text. The list of supported printer types, and information about whether these printer types accepted PostScript or ASCII text, was limited. Now, through the use of additional transformation software, raster image processor (RIP), and PostScript Printer Description (PPD) files, you can print to a wider range of printers. The database of printer description files is called the *foomatic* database.

Raster Image Processor (RIP)

The RIP enables you to print to printers that do not have resident PostScript processing capabilities. The Solaris printing software now provides the print server RIP and supporting technologies. The RIP occurs behind the scenes. However, to use the appropriate driver, you need to configure each printer, by using either Solaris Print Manager or a new option to the `lpadmin` command.

PostScript Printer Description (PPD)

PostScript is a language developed by Adobe® to describe a print document. This language removed the need for application developers to write support for many different makes and models of printers into their applications. An application which created PostScript output could print to any PostScript-capable printer.

When a printer vendor creates a printer which has features not referenced by PostScript, a PostScript Printer Description (PPD) file describes the device dependent features. It was also created by Adobe to allow printer manufacturers to implement their own special features into PostScript.

Print Management Tools

The LP print service software contains the following components for the set up and administration of printers in the Solaris OS:

- Solaris OS Print Manager – A graphical user interface (GUI) that provides the ability to configure and manage printers.
- LP print service commands – A command-line interface that configures and manages printers. These commands also provide functionality not available in the other print management tools.

Client-Server Model

The Solaris OS print service is implemented in a client-server model.

Print Server

A print server is any system that is configured to manage a printer directly connected to it or that is attached to the network. The print server makes the printers available to other systems on the network and provides spooling for the client's print requests.

Print Client

A print client is a system that sends print requests to a print server.

Types of Printer Configurations

As a system administrator, printers are configured so that users have access to one or more printers.

Printers should be distributed over several print servers. If one print server becomes unavailable, print requests can be quickly and easily routed to other print servers on the network.

The Solaris OS supports local, network, and remote printer configurations.

Local Printer

A local printer is physically connected to a system and is accessed from that system.

Network Printer

A network printer is physically attached to the network and has its own host name and Internet Protocol (IP) address. A network printer provides print services to clients but is not directly connected to a print server.

Remote Printer

A remote printer is one that users access over the network, that is, a printer that is either physically connected to a remote system or physically attached to the network.

Refer to Figure 12-2 when reviewing the concept of local, network, and remote printers.

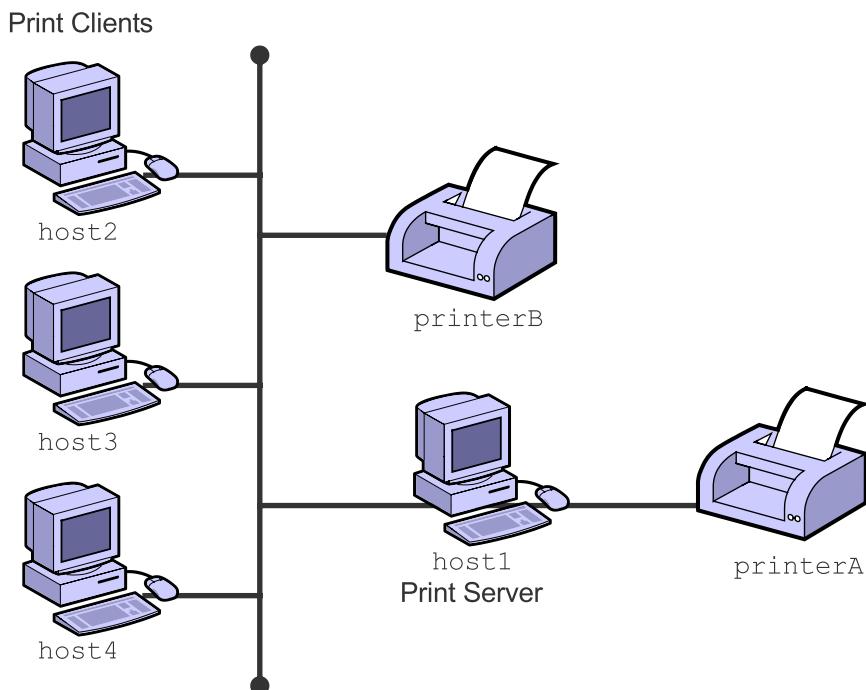


Figure 12-2 Local, Network, and Remote Printers

The printer named **printerA**, connected to the system named **host1**, is a local printer for any user logged on to that system.

The printer named `printerB` is a network printer that is controlled by the print server, `host1`. This is a network printer for any users logged in on `host1`, `host2`, `host3`, or `host4`.

For users who are logged in to `host2`, `host3`, or `host4`, both `printerA` and `printerB` can be accessed as remote printers.

Basic Functions of the Solaris OS LP Print Service

Basic functions of the Solaris OS LP print service include initialization, queuing, tracking, fault notification, and filtering.

Initialization

The Solaris OS LP print service initializes a printer prior to sending it a print request. The initialization function ensures that the printer is in a known state.

Queuing

The Solaris OS LP print service queues the print requests. The queuing function schedules the print requests that are waiting to be sent to the printer.

Tracking

The Solaris OS LP print service tracks the status of every print request. The tracking function enables the root user to manage all of the requests and enables other users to view or cancel their own requests. This function also logs any errors that have occurred during the printing process.

Fault Notification

The Solaris OS LP print service provides fault notification if a problem occurs in the print service. The fault notification function prints an error message on the console or sends an email to the root user, depending on how the service has been configured.

Filtering

The Solaris OS LP print service provides filtering capabilities that convert print jobs to the appropriate type of file for the destination printer.

LP Print Service Directory Structure

The Solaris OS LP print service includes a directory structure, files, and logs. The following section describes some of the more important components of this structure.

The /usr/bin Directory

This directory contains the LP print service user commands, such as the `lp`, `lpstat`, and `cancel` commands.

The /usr/sbin Directory

This directory contains the LP print service administrative commands, such as the `lpadmin`, `lpusers`, and `lpshut` commands.

The /usr/share/lib/terminfo Directory

This directory contains the `terminfo` database directories, which describe the capabilities of printers and terminals.

The /usr/lib/lp Directory

This directory contains the `lpsched` daemon, binary files that the LP print service uses, PostScript™ filters, and standard printer interface programs. Two important subdirectories in the `/usr/lib/lp` directory are the `model` and `postscript` directories.

The /usr/lib/lp/model Directory

This directory contains four default printer interface programs or shell scripts, called the `standard`, `standard_foomatic`, `netstandard`, and the `netstandard_foomatic` scripts.

The `standard` script supports local printers using PostScript or ASCII. For example, when a print request is queued for printing, the print service runs the printer's `standard` script to:

- Initialize the printer port, if necessary
- Initialize the actual printer, using the `terminfo` database to find the appropriate control sequences

- Print a banner page, if necessary
- Print the correct number of copies, as specified by the user's print request

The netstandard script specifically supports network printers. It collects the spooler and print database information needed to perform network printing and passes the information to a print output module. The netpr module opens the network connection to the printer and sends the data to the printer.

Note – The netpr module is located in the /usr/lib/lp/bin directory.



The standard_foomatic and netstandard_foomatic scripts are for printers which takes advantage of the new Raster Image Processor (RIP) feature and PostScript Printer Definition (PPD) files from /usr/lib/lp/model/ppd.

When a printer is configured, the appropriate model script is copied from /usr/lib/lp/model to /etc/lp/interfaces/*printer_name*.

The root user can then modify any printer's interface script. For example, to turn off the printing of a banner page, edit the /etc/lp/interfaces/*printer_name* file on the print server. Change the nobanner line from:

nobanner="no"

to

nobanner="yes"

The /usr/lib/lp/postscript Directory

This directory contains all PostScript filter programs provided by the Solaris OS LP print service.



Note – Print filters are programs that the print server uses to convert the content type of a queued print request from other formats to PostScript, handle special printing modes, like two-sided printing, and detect faults.

These filters have companion descriptor files in the /etc/lp/fd directory that tell the LP print service the characteristics and location of the filters.

The /etc/lp Directory

This directory contains a hierarchy of LP server configuration directories and files.

You can view the contents of these configuration files. However, you should not edit these files directly. To make configuration changes, use the lpadmin command or printmgr GUI.

There are three subdirectories in the /etc/lp directory that are important to printer configuration. These are the fd, interfaces, and printers directories.

- The /etc/lp/fd directory contains a set of print filter descriptor files. These files describe the characteristics of the filter and point to the actual filter program.

Note – The /etc/lp/filter.table file contains a filter lookup table.

- 
- The /etc/lp/interfaces directory contains each printer's interface script file. When a printer is configured, the print service places a copy of the appropriate default interface script from the /usr/lib/lp/model directory into the /etc/lp/interfaces/*printername* file.
 - The /etc/lp/printers directory contains a subdirectory for each printer served by the system. Each subdirectory contains configuration information and alert files for an individual printer.

For example, the configuration file for a printer named printerB can contain the following information:

```
# cat /etc/lp/printers/printerB/configuration
Banner: optional
Content types: postscript
Device: /dev/null
Interface: /usr/lib/lp/model/netstandard
Printer type: PS
Modules:
Options: dest=printerB,protocol=bsd
```

The `/var/spool/lp` Directory

This directory contains a list of current requests that are in the print queue.

The `lpsched` daemon for each system keeps track of print requests in the following directories:

- `/var/spool/lp/tmp/system-name`
- `/var/spool/lp/requests/system-name`

With a local print request, the `/var/spool/lp/tmp/system-name` directory contains one file, and the `/var/spool/lp/requests/system-name` directory contains another file.

With a remote print request, the `/var/spool/lp/tmp/system-name` directory contains two files, and the `/var/spool/lp/requests/system-name` directory contains one file.

Only the `root` user or `lp` users can access the information in the `/var/spool/lp/requests/system-name` directory.

Only the user who submitted the print request, the `root` user, or the `lp` user can access the information in the `/var/spool/lp/tmp/system-name` directory.

These files remain in their directories only as long as the print request is in the queue. After completing the print request, the print service combines the information in the files and appends it to the `/var/lp/logs/requests` file.

Note – The `/var/spool/print` directory contains the client-side request staging area for the LP print service.



The /var/lp/logs Directory

This directory contains an ongoing history of print requests. The log file /var/lp/logs/requests contains information about completed print requests that are no longer in the print queue.

Print Requests From the Network

The /usr/sbin/inetd Internet Service Daemon

The Internet services daemon, inetd, is a Service Management Facility (SMF) restarter process for many network services. It is usually started up by SMF at system boot time. The inetd service listens for requests for network services which are currently enabled. The service which handles incoming print requests from the network is
`svc:/application/print/server:default`.

To check the status of the print service, use the `svcs -a` command:

```
# svcs -a |grep 'print'  
disabled      16:59:17 svc:/application/print/server:default  
online        16:59:49 svc:/application/print/cleanup:default  
offline       16:59:35 svc:/application/print/ipp-listener:default  
offline       17:00:43 svc:/application/print/rfc1179:default
```

Use the `svcadm` command to enable or disable the service. Changes made to the state of the service persist across reboots:

```
# svcadm enable svc:/application/print/server:default  
# svcs -a | grep 'print/server'  
online        19:01:09 svc:/application/print/server:default
```

When a request arrives, the `inetd` daemon executes the server program that is associated with the service. Print servers listen for print requests with the `inetd` daemon, and upon hearing a request, start up the `in.lpd` daemon.

The /usr/lib/print/in.lpd Program

The inetd daemon starts the in.lpd program, sometimes referred to as the protocol adapter. The in.lpd program implements the network listening service for the print protocol. The print protocol provides a remote interface that enables systems to interact with a local spooling system. This protocol defines standard requests from the print client to the print server, such as requests to start queue processing, to transfer print jobs, to retrieve print status, and to cancel print jobs.

Upon the receipt of a connect request, the in.lpd program starts and services the connection. The in.lpd program closes the connection and exits after servicing the request.

Internet Printing Protocol (IPP) Listener

The IPP listener for the Solaris OS listens for Hypertext Transfer Protocol (HTTP) requests on port 631. The listener receives print client requests and communicates those requests to the printing system.

After the print server has been configured, the IPP listening service automatically starts:

```
# svcs ipp-listener  
online      19:01:11 svc:/application/print/ipp-listener:default
```

A print client needs to know the print server name and the name of a printer to print to. For example, on a Microsoft Windows system, a network printer can be configured with the network path:
http://server-name:631/printers/printer-name.

The /usr/lib/lp/lpsched Daemon

The LP print service has a scheduler daemon called `lpsched`. The scheduler daemon updates the LP system files with information about printer setup and configuration. It also manages requests issued to the system by the `lp` and `lpr` commands.

The `lpsched` daemon schedules all of the local print requests on a print server. It also tracks the status of printers and filters on the print server. When a printer finishes a request, the `lpsched` daemon schedules the next request, if there is one in the queue on the print server.

Each print server, has by default, only one `lpsched` daemon running. It is started by the `svc:/application/print/server:default` service when the system is booted. The parent `lpsched` daemon spawns a child `lpsched` processes to service print jobs.

Solaris OS Printing Process

Users submit print requests from print clients by using the `lp` or `lpr` commands.

 **Note** – The Solaris OS print service accepts both the System V Interface Definition (SVID) `/usr/bin/lp` command and the Berkeley Software Distribution (BSD) `/usr/ucb/lpr` command to submit print requests.

Users should use these commands to print text files. These commands do not print documents created in applications such as the StarOffice™ software. Most third-party applications require you to print from a selection menu within the application.

The function of the `lp` and `lpr` commands is to queue print requests for printing on a destination printer.

Locating the Destination Printer

The Solaris OS LP print service checks several resources to locate the destination printer for a print request.

Figure 12-3 shows the resources checked as it identifies the appropriate printer for a print request.

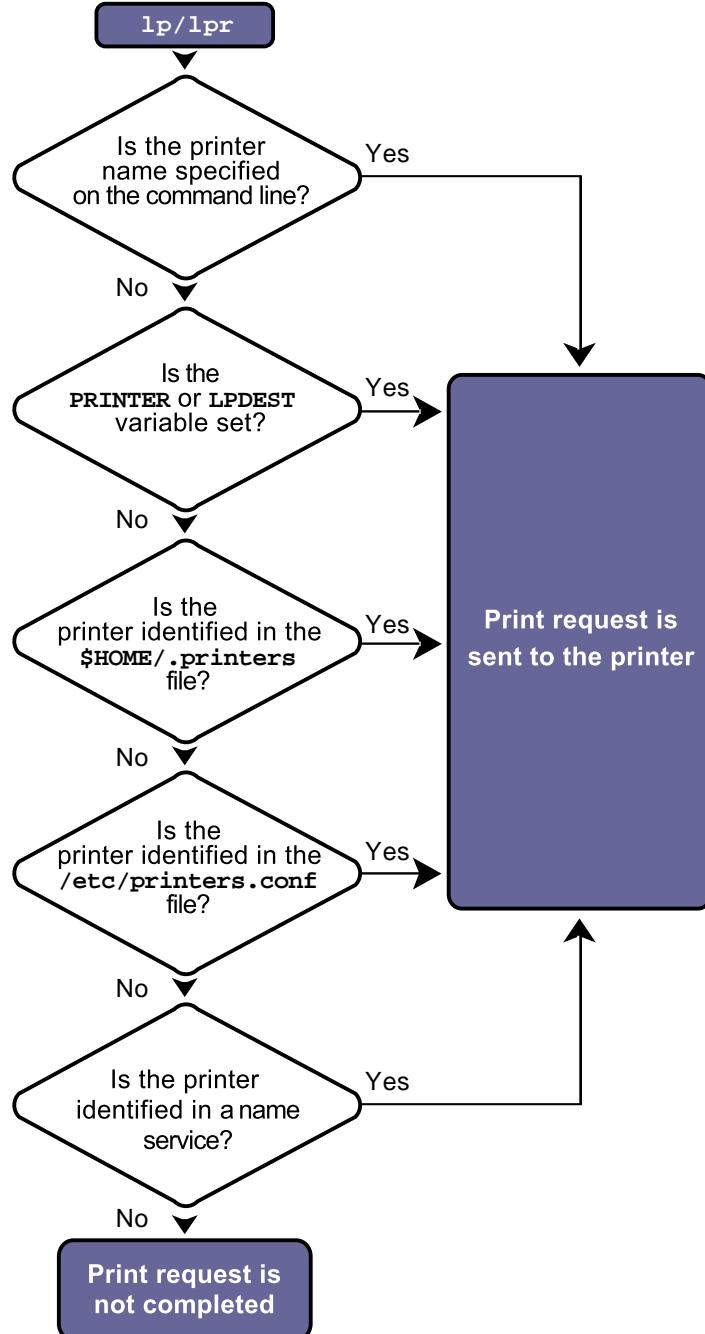


Figure 12-3 Locating the Destination Printer

If the command line does not specify a named printer destination, the user's shell environment is checked.

You can set the LPDEST or PRINTER environment variables to a default printer name. The lp command checks LPDEST and then PRINTER. The lpr command reverses the order when searching for a printer.

If neither variable specifies a named printer destination, then the Solaris OS LP print service checks for the _default variable in the following files:

- The \$HOME/.printers file

Users can create their own .printers file in their home directory to set the default printer name. They should add the following line to the file:

```
_default printername
```

If the \$HOME/.printers file does not exist or does not specify a printer name destination, then the Solaris OS LP print service checks the /etc/printers.conf file.

- The /etc/printers.conf file

Each entry in the /etc/printers.conf file describes a printer destination. For example, if host1 is the print server's name and printerA is the printer's name, the entry in this file appears as follows:

```
_default:\n      :use=printerA:\nprinterA:\n      :bsdaddr=host1,printerA,Solaris\n      :description=printerA
```

If the _default variable is not set, then the _default variable in the name service database (for example, Network Information Service (NIS)) is checked.

- The printers.confbyname file

The printers.confbyname file is the NIS version of the /etc/printers.conf file. In this case, the _default variable entry in the name service map called printers.confbyname defines the print server and printer name destination:

```
_default:bsdaddr=servername,printername:
```

If the destination printer name cannot be located in any of these configuration resources, the print request cannot be completed.



Note – The last three files described in the following paragraphs rely on the printers: entry in the NIS version of the /etc/nsswitch.conf file.

An example of the /etc/nsswitch.conf file syntax is:

```
printers: user files nis
```

where:

```
user = Checks $HOME/.printers file  
files = Checks /etc/printers.conf file  
nis = Checks printers.confbyname file
```

Introducing the Local Print Process

When a user submits a print request to a local printer, the lp or lpr command sends the request to the lpsched daemon. The lpsched daemon is also called the print scheduler.

Figure 12-4 shows the role of the `lpsched` daemon in the printing process.

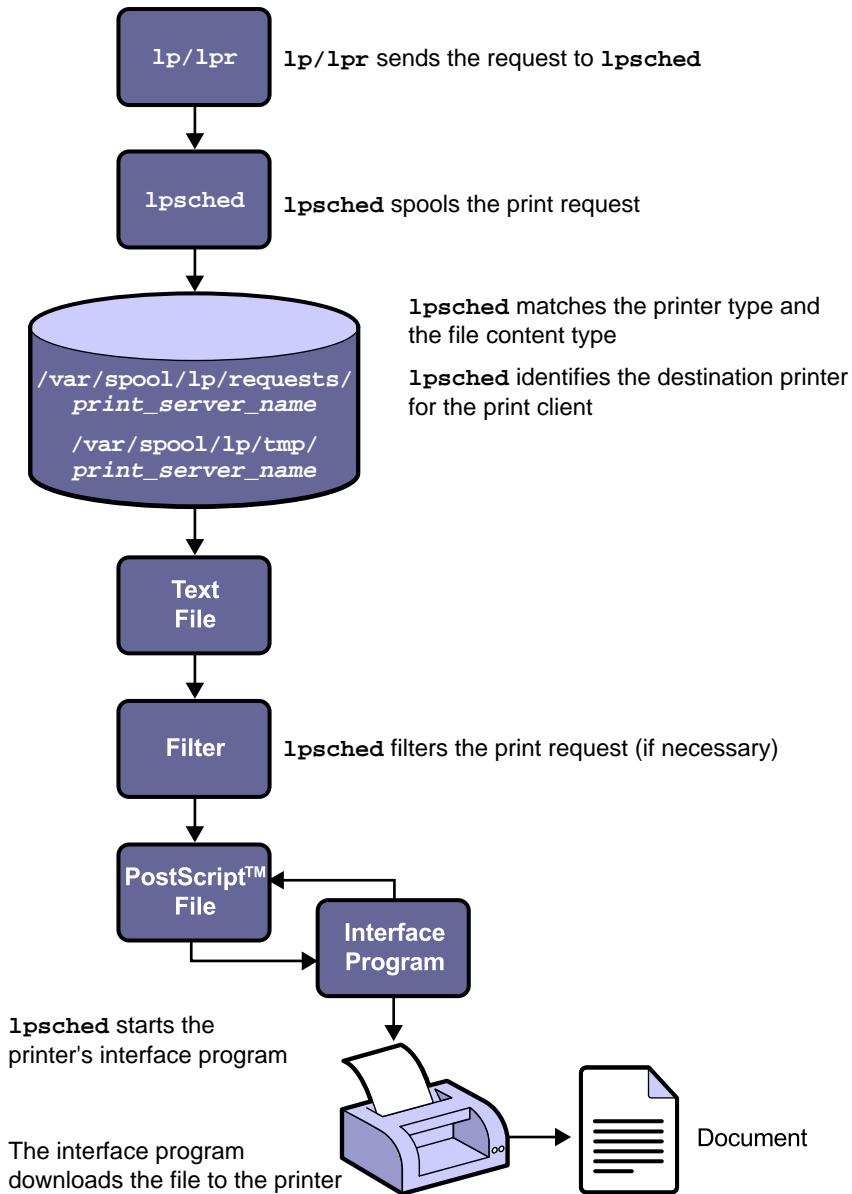


Figure 12-4 Local Printing Process

The `lpsched` daemon matches the printer type and identifies the default printer for the system. It then filters the print job.

The `lpsched` daemon keeps track of print requests in the following directories:

- `/var/spool/lp/requests/system_name`
- `/var/spool/lp/tmp/system_name`

If the printer is free, the `lpsched` daemon starts the printer's interface program. The interface program performs the following functions:

- Initializes the printer port
- Initializes the printer
- Prints the banner page
- Prints the correct number of file copies
- Sends any fault notifications

Remote Print Process

When a user submits a print request to a remote printer, the `lp` or `lpr` command sends the print request directly to the print server.

The print server processes the print request and sends the print request to the destination printer to be printed.

Figure 12-5 shows a remote print request submitted from a print client to a print server in the Solaris OS.

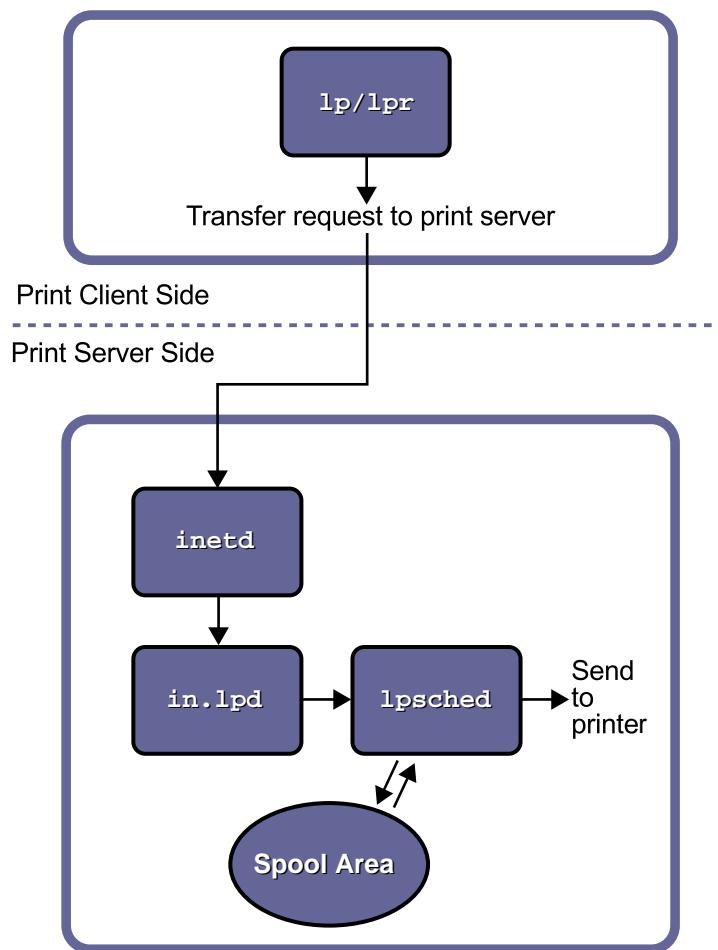


Figure 12-5 Solaris OS Remote Printing

The client's print command communicates directly with the print service on the server to transfer a print request to the printer.

The print server listens for print requests with the Internet services daemon `inetd`. When the `inetd` daemon hears a request for a print service on the network, it starts the `in.1pd` program. The `in.1pd` program is also called the print protocol adapter. The `in.1pd` program starts on demand and exits when the network request finishes.

The print protocol adapter translates the print request, communicates it to the print spooler, and returns the results to the print requester.

The print protocol adapter contacts the `lpsched` daemon to start the printer's interface program and to transfer the print request to the destination printer.

Configuring Printer Services

Configuring printer services in the Solaris OS involves a number of key tasks. Table 12-1 shows these tasks.

Table 12-1 Main Tasks for Configuring Printer Services

Tasks	Description
Setting up the printer	Physically connecting the printer to a system or the network
Setting up the print server	Configuring the system that is to manage and provide access to the printer
Setting up the print client	Configuring the system to access a remote printer
Verifying printer access	Checking that the print server recognizes all print clients and that each print client recognizes the print server



Note – When a network of systems is not running a name service, such as NIS, enter each print server's host name and IP address in the /etc/inet/hosts file on the print client when you are setting up the printer services.

Using the Solaris OS Print Manager

The Solaris OS Print Manager enables you to set up and manage printers.

The Solaris OS Print Manager is the preferred method for managing printers. When used with a name service such as NIS, it centralizes printer information and simplifies printer administration.



Note – The Solaris OS Print Manager recognizes existing printer information on print servers, print clients, and in the name service databases.

The following steps demonstrate how to configure a network printer with the Solaris Print Manager. As the `root` user, start the Solaris OS Print Manager with the following command:

```
# /usr/sbin/printmgr &
```

You can also start the Solaris OS Print Manager by selecting the Printer Administrator from the Tools option on the Common Desktop Environment (CDE) Workspace menu and entering the host name of the workstation to continue.

Either method displays the Solaris OS Print Manager main window, with Figure 12-6 overlaid on top of it.



Figure 12-6 Solaris Print Manager: Select Naming Service Window

1. Click OK to select the default, `files`.

Figure 12-7 remains on the screen.

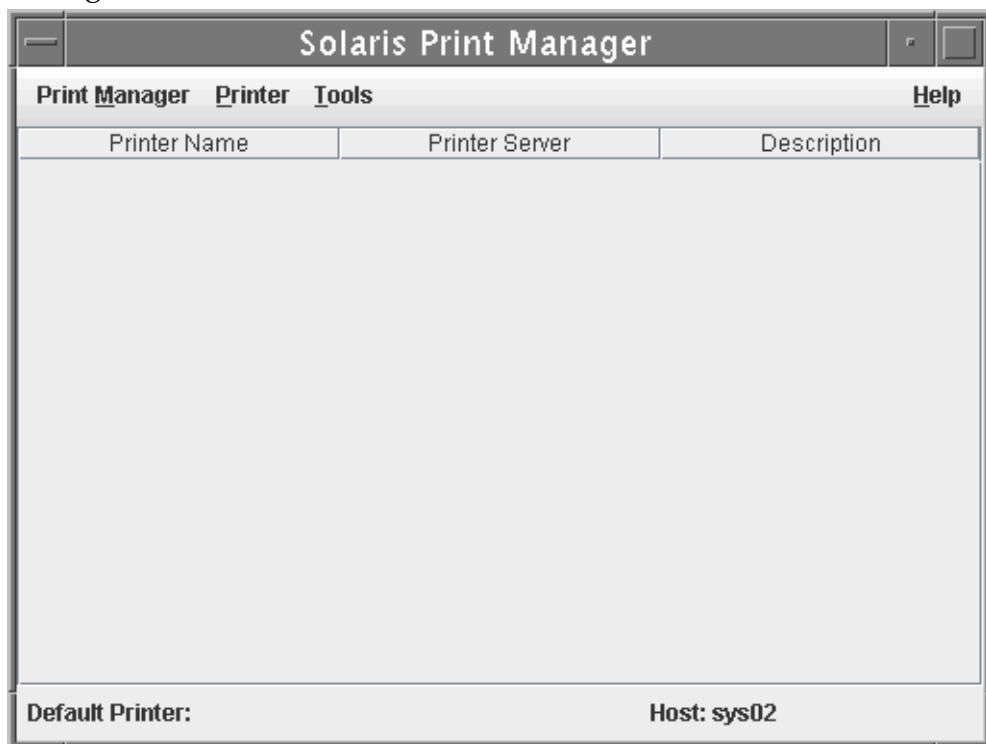


Figure 12-7 Solaris Print Manager Window

2. Click the Printer menu in this window. Figure 12-8 shows possible menu selections on the Printer menu.

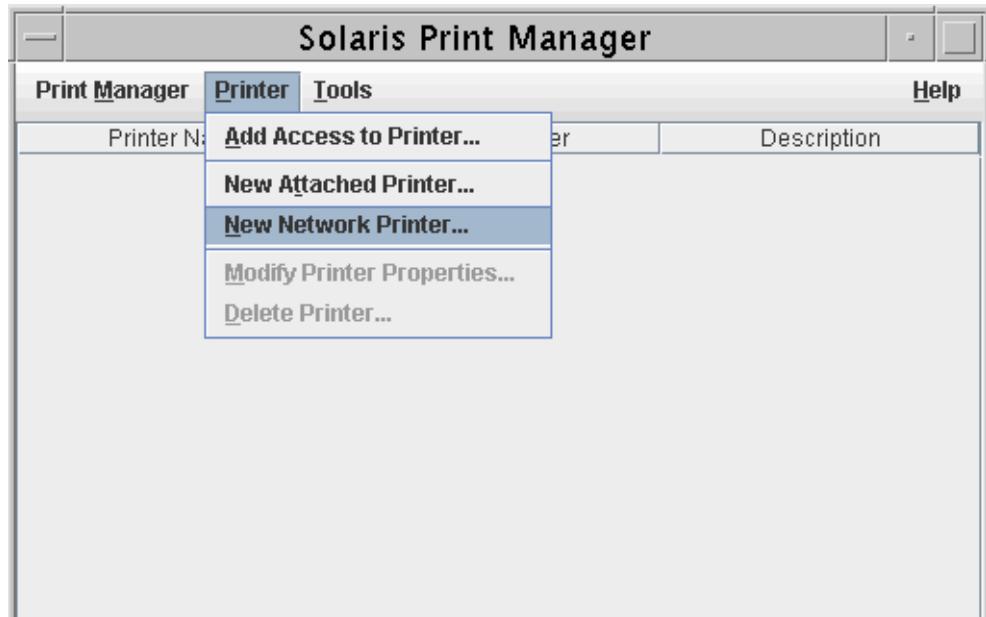


Figure 12-8 Solaris Print Manager Printer Menu



Note – By clicking Print Manager and selecting Show Command Line Console, you can see the command-line equivalents to each of the actions taken to configure printers. You can then save these steps as commands to perform similar actions in the future or build your own scripts for configuring printers.

Menu selections include:

- Add Access to Printer – Selected from a print client to set up access to printers that are controlled by a print server. The host name and IP address of the print server must be in the print client's /etc/inet/hosts file or in a name service database (for example, NIS).
- New Attached Printer – Selected from a print server to configure a printer that is physically connected to it. The print server provides the queuing capabilities, filtering, and printing administration.
- New Network Printer – Selected from a print server to configure a printer that is directly attached to the network. The print server provides the queuing capabilities, filtering, and printing administration. The network printer's name and its IP address must be entered either in the print server's /etc/inet/hosts file or in a name service database.

Configuring a New Network Printer

Table 12-2 shows the information you would use to configure a new local or network printer.

Table 12-2 Information Fields for Configuring a New Printer

Required Field	Available in releases prior to Solaris 10	Available in releases Solaris 10 and later
Printer Name	A unique name for the printer. The name can contain a maximum of 14 alphanumeric characters, including dashes and underscores. This is the name entered on the command line with a print command.	

Table 12-2 Information Fields for Configuring a New Printer
(Continued)

Required Field	Available in releases prior to Solaris 10	Available in releases Solaris 10 and later
Printer Server	Defaults to the name of the system on which you are currently running the Solaris OS Print Manager. This system is the print server for this network printer.	
Description	This field is optional. A printer's description commonly contains information to help users identify the printer, for example, physical location or printer type.	
Printer Port	Only required for attached printers.	
Printer Type	Yes Not, by default, for the Solaris 9 OS /04 release	PPD is enabled by default in the Print Manager. This allows you to choose a printer from the range of supported printers in <code>/usr/lib/lp/model/pd/system/foomatic</code> .
File Content Type	Yes Not, by default, for the Solaris 9 OS /04 release	Yes, by deselecting the Use PPD files options in the Print Manager drop-down menu.
Printer Make	No Yes, available in the Solaris 9 OS /04 release only	Yes
Printer Model	No	Yes. A list of supported printer models for the selected printer make. The corresponding PPD files are in: <code>/usr/lib/lp/model/pd/system/foomatic/make</code>

Table 12-2 Information Fields for Configuring a New Printer
(Continued)

Required Field	Available in releases prior to Solaris 10	Available in releases Solaris 10 and later
Printer Driver	No Yes, available in the Solaris 9 OS /04 release	Defaults to the foomatic PostScript printer driver.
Fault Notification	The list of choices for how the superuser is notified of printer errors. These include: Write to Superuser, Mail to Superuser, or None.	
Destination	The network printer's unique access name. The Destination access name can be either the name of the printer or its IP address as defined in the /etc/inet/hosts file or in a name service database. The Destination access name is used only by the print subsystem when it is making the network connection to the physical printer or the printer-host device. It becomes part of the printer configuration database and is associated with the network printer's IP address.	
Protocol	For a network printer: The Internet protocol that is used to communicate with the printer for file transfer. The choices are Berkeley BSD Printer Protocol and raw Transmission Control Protocol (TCP). In general, the TCP protocol is more generic across printers. The printer vendor documentation supplies the information about the protocol to select.	
Options	Identifies two options, the Default Printer option and the Always Print Banner option, which, by default, are disabled. To enable an option, click in the appropriate box (a check mark appears).	
User Access List	Specifies print clients that can print to this printer. By default, the word <i>all</i> allows every print client access to this printer.	
Default Printer	Allows this printer to become the system default that is used by all users who have not set their own, preferred, default printer.	
Always Print Banner	Sets whether or not a banner page is printed for each print job request.	

From the print server, use the following procedure to set up the configuration information to provide access to a new network printer.

1. From the Printer menu, select the New Network Printer option.

Figure 12-9 shows the window that appears.

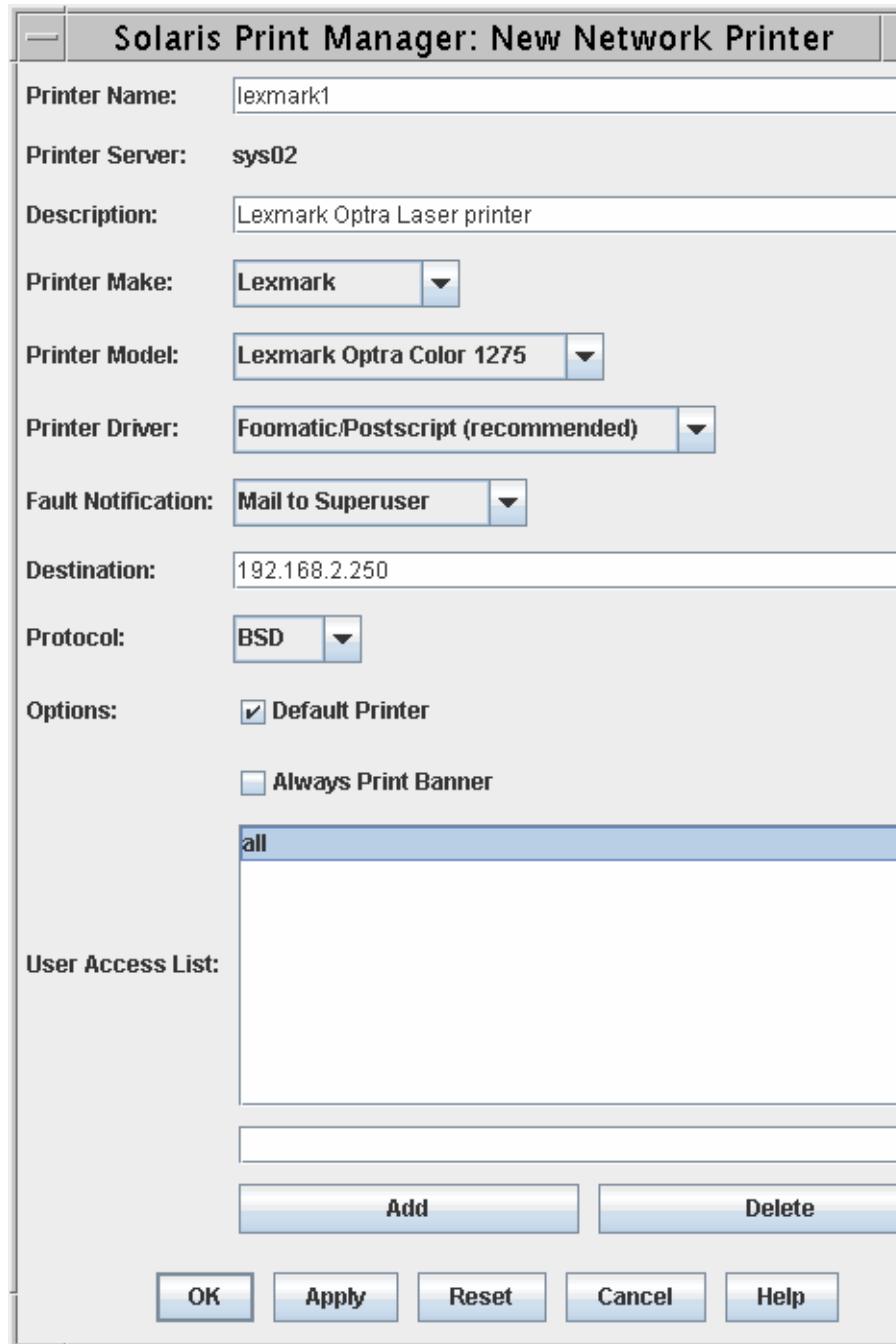


Figure 12-9 Solaris Print Manager: New Network Printer Window

2. In the Printer Name field, enter the new printer name, for example, `printerA`.
3. Click the Description field, and enter a printer description of your choice.
4. For the purposes of this demonstration, select Lexmark from the list of Printer Makes.

The LP print service uses information in the foomatic database to initialize the printer, as well as to communicate the correct sequence of codes to the printer.

To view the contents of the `foomatic` directory, type the following command:

```
# ls /usr/lib/lp/model/system/foomatic
Alps          Brother      DEC          Generic      Imagen      Minolta
Olivetti      Raven        Sonyprint
Anitech       CItoh        Dell         HP           Infotec     Mitsubishi
PCPI          Ricoh        Star
Apollo        Canon        Dymo         Heidelberg Kodak      NEC
Panasonic    Samsung      Tally
Apple         Citizen      Epson        Hitachi     Kyocera    Oce
Pentax        Seiko        Tektronix
Avery         Compaq      Fujitsu      IBM         Lexmark    Okidata
QMS           Sharp        Xerox
```

The `foomatic` directory contains many subdirectories that are named with a manufacturer.

5. Click the drop-down menu to select a Printer Model. All supported models for the chosen make are displayed. For this example, select Lexmark Optra E310.

The models for a Lexmark printer are located in the subdirectory:
`/usr/lib/lp/model/ppd/system/foomatic/Lexmark`

```
# ls /usr/lib/lp/model/ppd/system/foomatic/Lexmark
Lexmark-1000-lm1100.ppd.gzLexmark-Optra_Eplus-hpijs.ppd.gz
Lexmark-1020-lm1100.ppd.gzLexmark-Optra_Eplus-ljet4.ppd.gz
Lexmark-1020_Business-pcl3.ppd.gzLexmark-Optra_K_1220-Postscript.ppd.gz
Lexmark-1100-lm1100.ppd.gzLexmark-Optra_M410-Postscript.ppd.gz
Lexmark-2030-pbm212030.ppd.gzLexmark-Optra_M412-Postscript.ppd.gz
Lexmark-2050-c2050.ppd.gzLexmark-Optra_Rplus-Postscript.ppd.gz
<output omitted>
```

6. The default Printer Driver is the Foomatic/Postscript driver which will use Raster Image Processing (RIP) and PostScript Printer Description (PPD) files to match print documents to the document type understood by the printer.
7. Click Fault Notification, and select the Mail to Superuser option.
8. Click the Destination field, and type a Destination access name.

If the network printer is not recognized by its name from the hosts table or IP address, you might need to use the vendor-supplied access name for the network printer, which is *sometimes* qualified by a designated port number. These are both explicitly defined in the printer vendor's documentation.

Table 12-3 shows the format for a Destination entry.

Table 12-3 Destination Entry Format

Destination	Protocol
<i>printer_name</i>	BSD
<i>system_name:printer</i>	BSD
<i>IP_ADDR</i>	BSD
<i>IP_ADDR:port_number</i> ¹	TCP
<i>printer_node_name:port_number</i>	TCP

1. The port number is print server dependent. For example, LexMark uses Port 9100.
9. Leave the Internet protocol set to BSD.
10. Click in the Default Printer box to enable the Default Printer option.

Note – If enabled, the Default Printer option designates this printer as the default printer for print jobs from this system.



11. You can (optionally) click in the Always Print Banner box to enable the Always Print Banner option.
12. Accept the default, `all`, for the User Access List. This allows all users on all systems to use the printer.

To restrict user access to this printer, you can enter the values shown in Table 12-4 in the text field below the User Access List window.

Table 12-4 User Access Values

Value	Definition
<code>user-name</code>	The specified user, for example <code>user1</code> , can access the printer from any system.
<code>system-name!user-name</code>	The specified user from the named system can access the printer, for example, <code>host2!user4</code> .
<code>system-name!all</code>	All users from the named system only can access the printer, for example, <code>host5!all</code> .
<code>all!user-name</code>	The specified user from all systems can access the printer, for example, <code>all!user1</code> .

Note – To delete an entry from the User Access List, select the entry, and click Delete.



13. To accept the new network printer's configuration information, click OK.

Figure 12-10 shows the Solaris Print Manager window, which displays the newly configured printer.

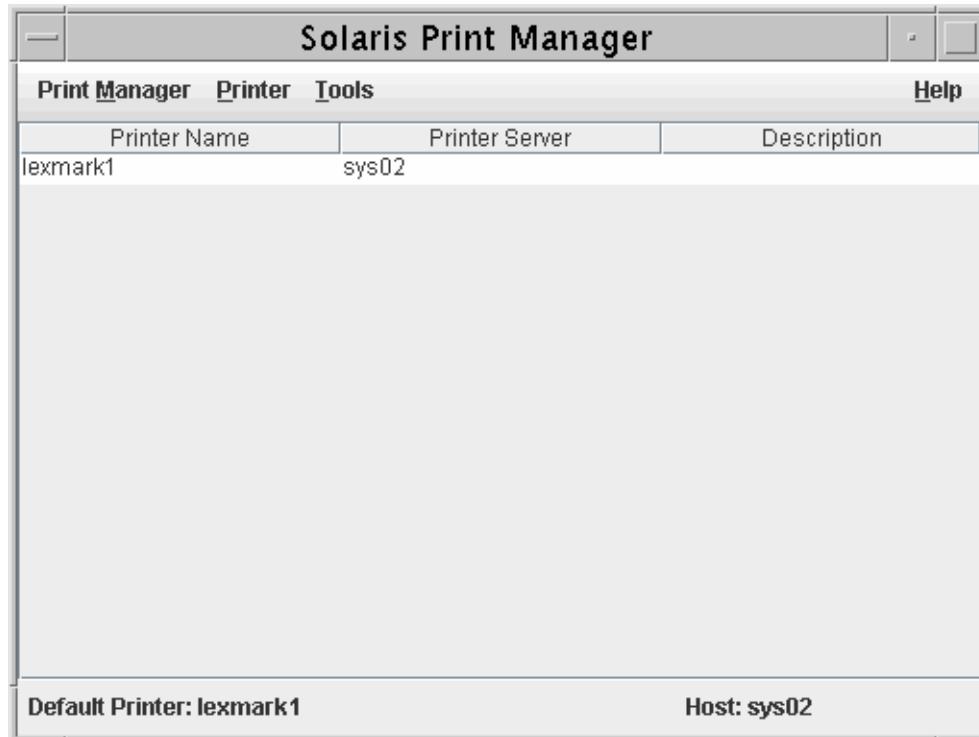


Figure 12-10 Solaris Print Manager Window: Configured Printer

Note – Information entered in this window populates the /etc/printers.conf and /etc/lp/printers/*printername* files.

14. To close the Solaris OS Print Manager window, select the Exit option from the Print Manager menu.

Administering Printer Services

You use the `lpadmin` command to configure the LP print services from the command line.

You could use this command to perform the following tasks:

- Defining printer devices and printer names
- Specifying interface programs (custom or standard) and printer options
- Specifying PostScript Printer Description (PPD) files for support of features not defined in normal PostScript
- Defining printer types and file content types
- Defining allow and deny user lists
- Specifying fault recovery
- Removing printers and printer classes

The `lpadmin` command is most commonly used by the `root` user for the purpose of:

- Configuring printers
- Setting or changing a system's default printer destination
- Removing a printer's configuration from the LP print service

Setting the System's Default Printer

The root user can run the `lpadmin` command to set an individual printer or a printer class to be the system's default destination for all print requests.

```
# lpadmin -d printername
# lpadmin -d printer-classname
```

For example, to set a system's default destination printer, perform the command:

```
# lpadmin -d printerE
```

To verify that the system's default destination printer has been set, perform the command:

```
# lpstat -d
system default destination: printerE
```

To verify an individual user's default destination printer, perform the command:

```
$ lpstat -d
system default destination: users_printer
```

The print request issued is sent by default to `printerE`.

```
# lp myfile
request id is printerE-514 (1 file)
```

Removing a Client's Printer Configuration

To remove a printer's configuration manually on the client side, perform the following:

1. Log in as the `root` user on the print client that has access to the printer to be removed from the LP print service.
2. Delete information about the printer from the print client by performing an `lpadmin` command.

```
# lpadmin -x printername
```

where `-x` deletes the specified printer.

For example, the following command deletes `printerD` from the system.

```
# lpadmin -x printerD
```

Information for the specified printer is deleted from the print client's `/etc/printers.conf` file.

Repeat Steps 1 and 2 for each print client that has access to the printer.

Removing a Server's Printer Configuration

 **Note** – The `reject` and `disable` commands are explained later in this module.

To remove a printer's configuration manually on the server side, perform the following:

1. Log in as the `root` user on the print server on which the printer is configured.
2. Stop queuing print requests on the printer.

```
# reject printerD
```

3. Stop the printer.

```
# disable printerD
```

4. Delete the printer from the print server.

```
# lpadmin -x printerD
```

This action deletes configuration information for the printer from the print server's `/etc/lp/printers` directory and `/etc/printers.conf` file.

Starting and Stopping the LP Print Service

The LP print service is started by the `lpsched` daemon and is shut down by the `lpshut` command.

Starting the LP Print Service

The `lpsched` daemon starts or restarts the LP print service. Printers that are restarted with a `lpsched` command from the command line, reprint, in their entirety, the print requests stopped by the `lpshut` command.

The following is an example of starting the `lpsched` daemon from the command line:

```
# svcadm enable application/print/server
```

Stopping the LP Print Service

When the command to disable the print service is invoked, any printers that are currently printing, stop printing.

The `lp` print service can be disabled. This method should be used to stop the print server service.

```
# svcadm disable print/server
# svcs -a | grep print
disabled          19:12:16 svc:/application/print/server:default
online           16:59:49 svc:/application/print/cleanup:default
online           19:01:10 svc:/application/print/rfc1179:default
offline          19:12:16 svc:/application/print/ipp-listener:default
```

Specifying a Destination Printer

In the Solaris OS, users submit print requests by using the `lp` command or the `lpr` command.

Note – The Solaris OS LP print service accepts both the SVID `/usr/bin/lp` command and the BSD `/usr/ucb/lpr` command to submit print requests.



Using the `lp` Command

The `lp` command is located in the `/usr/bin` directory. The `lp` command submits a print job to the default printer or to another printer by specifying the printer name. Perform one of the following commands:

```
$ /usr/bin/lp filename  
$ /usr/bin/lp -d printername filename
```

Using the `lpr` Command

The `lpr` command is located in the `/usr/ucb` directory. The `lpr` command functions in the same manner as the `lp` command. It submits a print job to the default printer or to another printer.

```
$ /usr/ucb/lpr filename  
$ /usr/ucb/lpr -P printername filename
```

The preceding examples of the print commands demonstrate the atomic style. You can also use the Portable Open Systems Interface (POSIX) style to specify a destination printer.

To submit a print request that uses the POSIX style, include the print command and an option, followed by the printer server name, a colon, and the printer name as configured on the print server.

The full command-line entry is as follows:

```
$ /usr/bin/lp -d hostname:printername filename  
$ /usr/ucb/lpr -P hostname:printername filename
```

Using the LP Print Service

The LP print service is a set of software commands, utilities, and filters that allow users to print files and the root user to set up and manage the print operations.

Table 12-5 lists some of the more commonly used print service administration commands.

Note – You must be the root user to use these commands.



Table 12-5 LP Print Service Administration Commands

Command Name	Description
accept	Permits print requests to be queued for the specific printers
reject	Prevents print requests from being queued for the specific printers
enable	Activates the specified printers
disable	Deactivates the specified printers
lpmove	Moves print requests from one printer destination to another

Accepting Print Jobs

As the root user, you use the accept command on the print server to permit print requests to be queued on the specified printers.

Using the accept Command

You use the accept command to allow queuing of print requests for the named destinations. A destination specifies the name of a printer or printer class.

The format for the command is:

```
# /usr/sbin/accept destination(s)
```

In the following example, the root user has enabled the queuing of print requests on printerD.

```
# accept printerD
destination "printerD" now accepting requests
```

Rejecting Print Jobs

As the root user, you use the `reject` command on the print server to prevent print requests from queuing on the specified printers.

Using the `reject` Command

You use the `reject` command to prevent print requests from queuing and stop users from submitting requests to the printer queues.

The format for the command is:

```
# /usr/sbin/reject -r "reason" destination(s)
```

The following example shows how to use the option `-r "reason"` to enter an explanation for the rejection of print requests for a printer. A user can see that text by issuing the `lpstat -a` or `lpstat -t` command.

```
# reject -r "Replacing Toner Cartridge" printerD
destination "printerD" will no longer accept requests
```

Enabling Printers

On the print server, as the root user, you can use the `enable` command to activate the specified printers.

Using the `enable` Command

The `enable` command activates the printers, which enables the printing of requests submitted to the print queues.

The format for the command is:

```
# /usr/bin/enable destination(s)
```

The following example shows how to enable printerD.

```
# enable printerD
printer "printerD" now enabled
```

Disabling Printers

On the print server, as the root user, you can use the disable command to deactivate the specified printers.

Using the disable Command

The disable command deactivates printers, which disables them from printing print requests waiting in the print queues.

By default, any requests currently printing on the printer when the disable command is issued are reprinted in their entirety when the printer is enabled again.

The format for the command is:

```
# /usr/bin/disable -c | -W -r "reason" destination
```

Table 12-6 shows the options for the disable command:

Table 12-6 Options for the disable Command

Option	Definition
-c	Cancels the current job and disables the printer. The current job is not printed later.
-W	Waits until the current job is finished before disabling the printer.
-r	Assigns a reason for the disabling of the printer.

The following example shows how to use the disable command with options.

```
# disable -W -r "Printer down for maintenance" printerD
printer "printerD" now disabled
```

Moving Print Jobs

You use the `lpmove` command to move one or all print requests from one printer destination to another printer destination.

Using the `lpmove` Command

The format for the `lpmove` command is:

```
# /usr/sbin/lpmove source_destination target_destination
```

To move one or all print requests by using the `lpmove` command, complete the following steps.

1. Become the `root` user on the print server.
2. Use the `reject` command to prevent any further print requests from being sent to the print queue. This step notifies users that the printer is not accepting requests.

```
# reject -r "PrinterC is down for repairs" printerC
destination "printerC" will no longer accept requests
```

3. Use the `lpstat` command to display the print queue to see how many print requests are to be moved. This step is needed to identify print request identification numbers (IDs) only if selected print requests are going to be moved to another printer.

```
# lpstat -o
printerC-29    sys41!user1    61426    Jan 07 12:30
printerC-30    sys41!user1    9560     Jan 07 12:30
printerC-31    sys42!user2    845      Jan 07 12:30
printerC-32    sys42!user2    845      Jan 07 12:30
printerC-33    sys42!user2    845      Jan 07 12:30
```

4. Use the `lpstat` command to verify that the destination printer is accepting print requests.

```
# lpstat -a printerA
printerA accepting requests since Tue Jan 1
```

5. Move the print requests.

- a. For example, to move all print requests from printerC over to printerA, perform the following command:

```
# lpmove printerC printerA  
move in progress ...  
total of 5 requests moved from printerC to printerA
```

- b. For example, to move one or more individual print requests from printerC to printerA, perform the following command:

```
# lpmove printerC-32 printerC-33 printerA  
total of 2 requests moved to printerA
```

6. If all print requests were moved from printerC, in step 5a, printerC has a `reject` automatically applied to it. When printerC is available again, use the `accept` command to allow print jobs to queue to printerC.

```
# accept printerC  
destination "printerC" now accepting requests
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using the LP Print Service (Level 1)

In this exercise, you use the Solaris OS print manager to set up a print spooler that sends output to a local terminal window, adds access to a remote printer, and uses print management commands.

Preparation

The host name and IP address of the system that controls the printer you want to access must exist in the /etc/inet/hosts file. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Open two terminal windows. Record the pseudo-terminal device used by one of them. In the other window, run the Solaris OS print manager, and define a local Lexmark printer that uses the first window's terminal as its output device. Test the new printer.
(Steps 1–7 in the Level 2 lab)
- Use the Solaris OS print manager to gain access to a printer defined on another system. Test the remote printer.
(Steps 9–13 in the Level 2 lab)
- Manipulate your Lexmark printer to:
 - Disable printer output
 - Queue four files for printing

Exercise: Using the LP Print Service (Level 1)

- List all print jobs
- Cancel two jobs by listing their request IDs
- Cancel the remaining jobs by using their associated user names
- Enable printing again
- Reject print requests and supply a reason
- View the reason
- Accept print requests on the default printer
(Steps 14–24 in the Level 2 lab)
- Remove both printers

Exercise: Using the LP Print Service (Level 2)

In this exercise, you use the Solaris OS print manager to set up a print spooler that sends output to a local terminal window, adds access to a remote printer, and uses print management commands.

Preparation

The host name and IP address of the system that controls the printer you want to access must exist in the /etc/inet/hosts file. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Open two terminal windows. Record the pseudo-terminal device used by one of them. In the other window, run the Solaris OS print manager, and define a local Lexmark printer that uses the first window's terminal as its output device. Test the new printer.
- Use the Solaris OS print manager to gain access to a printer defined on another system. Test the remote printer.
- Use the following commands to manipulate your Lexmark printer:
 - enable
 - disable
 - lp
 - lpstat

- accept
- reject
- cancel
- Manipulate your Lexmark printer to:
 - Disable printer output
 - Queue four files for printing
- List all print jobs
- Cancel two jobs by listing their request IDs
- Cancel the remaining jobs by using their associated user names
- Enable printing again
- Reject print requests and supply a reason
- View the reason
- Accept print requests on the default printer
- Remove both printers

Tasks

Complete the following steps:

1. Log in as the `root` user, and open two terminal windows. In one of the windows, use the `tty` command to identify the pseudo terminal device that it uses. Use this device name as the port for the new printer. For example, the device name in the following output is `/dev/pts/5`:

```
# tty  
/dev/pts/5
```

Device name:

2. In the other terminal window, run the Solaris OS print manager.
3. In the Select Naming Service panel, verify that `files` is selected, and click OK. From the print manager menu, select the Show Command Line Console option. Position the Command Line Console in a convenient location.
4. From the Printer menu, select the New Attached Printer option.

5. Fill in the fields according to Table 12-7. To name your printer, use a name different from that of your system.

Table 12-7 Configuration Fields

Field	Selection or Entry
Printer name	Your choice.
Description	Your choice.
Printer Port	Select the Other option. Enter the device name of the terminal window found in Step 1.
Printer Make	Lexmark.
Printer Model	Lexmark Optra E310.
Printer Driver	Foomatic/Postscript (recommended).
Fault Notification	Write to superuser .
Default Printer	Select the box.
Always Print Banner	Do not select the box.
User Access List	No change.

6. Click OK when you are finished. Notice the command-line entries that appear in the console window.

7. Test your printer configuration by printing the /etc/inet/hosts file to the default printer. Observe the output on the other terminal window.

You should see the contents of the /etc/inet/hosts file converted to the format a Lexmark Optra E310 would expect, scrolling through the other window.

8. From the Printer menu, select the Add Access to Printer option.

9. Fill in the fields according to Table 12-8.

Table 12-8 Configuration Fields

Field	Selection or Entry
Printer name	Enter the name of a printer on another system.
Printer server	Enter the name of the system on which the preceding printer is defined. Ensure this system name and IP address are in your /etc/inet/hosts file.
Description	Your choice.
Default printer	Do not select the box.

10. Click OK when you are finished.

Notice the command-line entries that appear in the console window.

11. Test your new configuration by printing the /etc/inet/hosts file to the remote printer. Observe the output on the other system.

You should see the contents of the /etc/inet/hosts file converted to the format a Lexmark Optra E310 would expect, scroll through the window on the other system

12. In an available terminal window, use the lpstat command to display the current status information of the printers on your system.

13. Disable print output for your default printer.

14. Send the following four files to your default printer:
/etc/inet/hosts, /etc/inittab, /etc/dfs/dfstab, and
/etc/skel/local.profile.

15. Check the print queue to find the request ID for each job.

The four print jobs should be listed with sequential numbers.

16. Use the request IDs to cancel two of the requests. Verify the result.
Use the following syntax to cancel the requests:

```
# cancel printernam1-# printernam1-#
```

Two of the print jobs should be gone.

17. Cancel the other two jobs by indicating the user who sent them.
Verify the result.

18. Enable printing for your default printer. Use the following syntax:

```
# enable printernam1
```

19. Set your default printer to reject requests and display a reason for doing so. For example:

```
# reject -r "Printer is down for maintenance" printename1
```

20. Attempt to send a job to the default printer. Observe the messages displayed.

```
# lp /etc/inet/hosts
```

Your message should say *printename1*: Requests are not being accepted.

21. Use the lpstat command to display the reason that the printer is not accepting requests. Use the following syntax:

```
# lpstat -a printename1
```

Your message should say *printename1*: *your reason from step 20*.

22. Set your default printer to accept requests again.

```
# accept printename1
```

23. Test your printer configuration by printing the /etc/inet/hosts file to the default printer. Observe the output on the other terminal window.

```
# lp /etc/inet/hosts
```

You should see the contents of the /etc/inet/hosts file converted to the format a Lexmark Optra E310 would expect, scrolling through the other window.

24. Before removing the printers, prevent any further print requests from being queued:

```
# reject -r "removing printer" printename1
# reject -r "removing printer" printename2
```

25. Remove both printers.

```
# lpadm -x printename1
# lpadm -x printename2
```

Exercise: Using the LP Print Service (Level 3)

In this exercise, you use the Solaris OS print manager to set up a print spooler that sends output to a local terminal window, adds access to a remote printer, and uses print management commands.

Preparation

The host name and IP address of the system that controls the printer you want to access must exist in the /etc/inet/hosts file. Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Open two terminal windows. Record the pseudo-terminal device used by one of them. In the other window, run the Solaris OS print manager, and define a local Lexmark printer that uses the first window's terminal as its output device. Test the new printer.
- Use the Solaris OS print manager to gain access to a printer defined on another system. Test the remote printer.
- Use the following commands to manipulate your Lexmark printer:
 - enable
 - disable
 - lp
 - lpstat

- accept
- reject
- cancel
- Manipulate your Lexmark printer to:
 - Disable printer output
 - Queue four files for printing
- List all print jobs
- Cancel two jobs by listing their request IDs
- Cancel the remaining jobs by using their associated user names
- Enable printing again
- Reject print requests and supply a reason
- View the reason
- Accept print requests
- Remove the printers

Tasks and Solutions

Complete the following steps:

1. Log in as the root user and open two terminal windows. In one of the windows, use the `tty` command to identify the pseudo-terminal device it uses. Use this device name as the port for the new printer. For example, the device name in the following output is `/dev/pts/5`:

```
# tty  
/dev/pts/5
```

Device name: *Your device name varies.*

2. In the other terminal window, run the Solaris OS print manager.

```
# /usr/sbin/printmgr &
```

3. In the Select Naming Service panel, verify that `files` is selected, and click OK. From the print manager menu, select the Show Command Line Console option. Position the Command Line Console in a convenient location.
4. From the Printer menu, select the New Attached Printer option.
5. Fill in the fields according to Table 12-7 on page 12-45. To name your printer, use a name different from that of your system.

Exercise: Using the LP Print Service (Level 3)

6. Click OK when you are finished. Notice the command-line entries that appear on the console window.
7. Test your printer configuration by printing the /etc/inet/hosts file to the default printer. Observe the output on the other terminal window.

lp /etc/inet/hosts

You should see the contents of the /etc/inet/hosts file converted to the format a Lexmark Optra E310 would expect, scroll through the other window.

8. From the Printer menu, select the Add Access to Printer option.
9. Fill in the fields according to Table 12-8 on page 12-46.
10. Click OK when you are finished.

Notice the command-line entries that appear in the console window.

11. Test your new configuration by printing the /etc/inet/hosts file to the remote printer. Observe the output on the other system.

lp -d printername2 /etc/inet/hosts

You should see the contents of the /etc/inet/hosts file converted to the format a Lexmark Optra E310 would expect, scrolling through the other window.

12. In an available terminal window, use the lpstat command to display the current status information of the printers on your system.

lpstat -t

13. Disable print output for your default printer.

disable printername1

14. Send the following four files to your default printer:
/etc/inet/hosts, /etc/inittab, /etc/dfs/dfstab, and
/etc/skel/local.profile.

lp /etc/inet/hosts

lp /etc/inittab

lp /etc/dfs/dfstab

lp /etc/skel/local.profile

15. Check the print queue to find the request ID for each job.

lpstat -o

The four print jobs should be listed with sequential numbers.

16. Use the request IDs to cancel two of the requests. Verify the result.
Use the following syntax to cancel the requests:

```
# cancel printernam1-# printernam1-#
# lpstat -o
```

Two of the print jobs should be gone.

17. Cancel the other two jobs by indicating the user who sent them.
Verify the result. For example:

```
# cancel -u root
# lpstat -o
```

18. Enable printing for your default printer.

```
# enable printernam1
```

19. Set your default printer to reject requests, and display a reason for doing so. For example:

```
# reject -r "Printer is down for maintenance" printernam1
```

20. Attempt to send a job to the default printer. Observe the messages displayed.

```
# lp /etc/inet/hosts
```

Your message should say *printernam1*: Requests are not being accepted.

21. Use the lpstat command to display the reason that the printer is not accepting requests. Use the following syntax:

```
# lpstat -a printernam1
```

Your message should say *printernam1*: *your reason from step 20*.

22. Set your default printer to again accept requests.

```
# accept printernam1
```

23. Test your printer configuration by printing the /etc/inet/hosts file to the default printer. Observe the output on the other terminal window.

```
# lp /etc/inet/hosts
```

You should see the contents of the /etc/inet/hosts file converted to the format a Lexmark Optra E310 would expect, scrolling through the other window.

Exercise: Using the LP Print Service (Level 3)

24. Before removing the printers, prevent any further print requests from being queued:

```
# reject -r "removing printer" printernamel
destination "printernamel" will no longer accept requests
# reject -r "removing printer" printernametwo
destination "printernametwo" will no longer accept requests
```

25. Remove both printers.

```
# lpadmin -x printernamel
# lpadmin -x printernametwo
```

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 13

Controlling System Processes

Objectives

Upon completion of this module, you should be able to:

- View system processes
- Kill frozen processes
- Schedule an automatic one-time execution of a command
- Schedule an automatic recurring execution of a command

The course map in Figure 13-1 shows how this module fits into the current instructional goal.

Managing Network Printers and System Processes

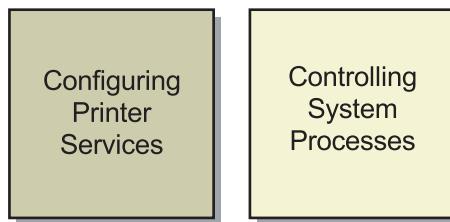


Figure 13-1 Course Map

Viewing System Processes

A process is any program that is running on the system. All processes are assigned a unique process identification (PID) number, which is used by the kernel to track and manage the process. The PID numbers are used by the root and regular users to identify and control their processes.

Using the CDE Process Manager

The Solaris OS Common Desktop Environment (CDE) provides a Process Manager to monitor and control processes that are running on the local system.

To start the Process Manager, click the Find Process control on the Tools subpanel of the Front Panel. Figure 13-2 shows the Tools menu.

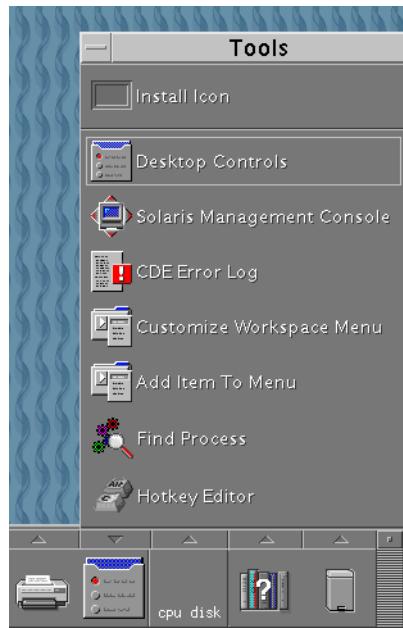
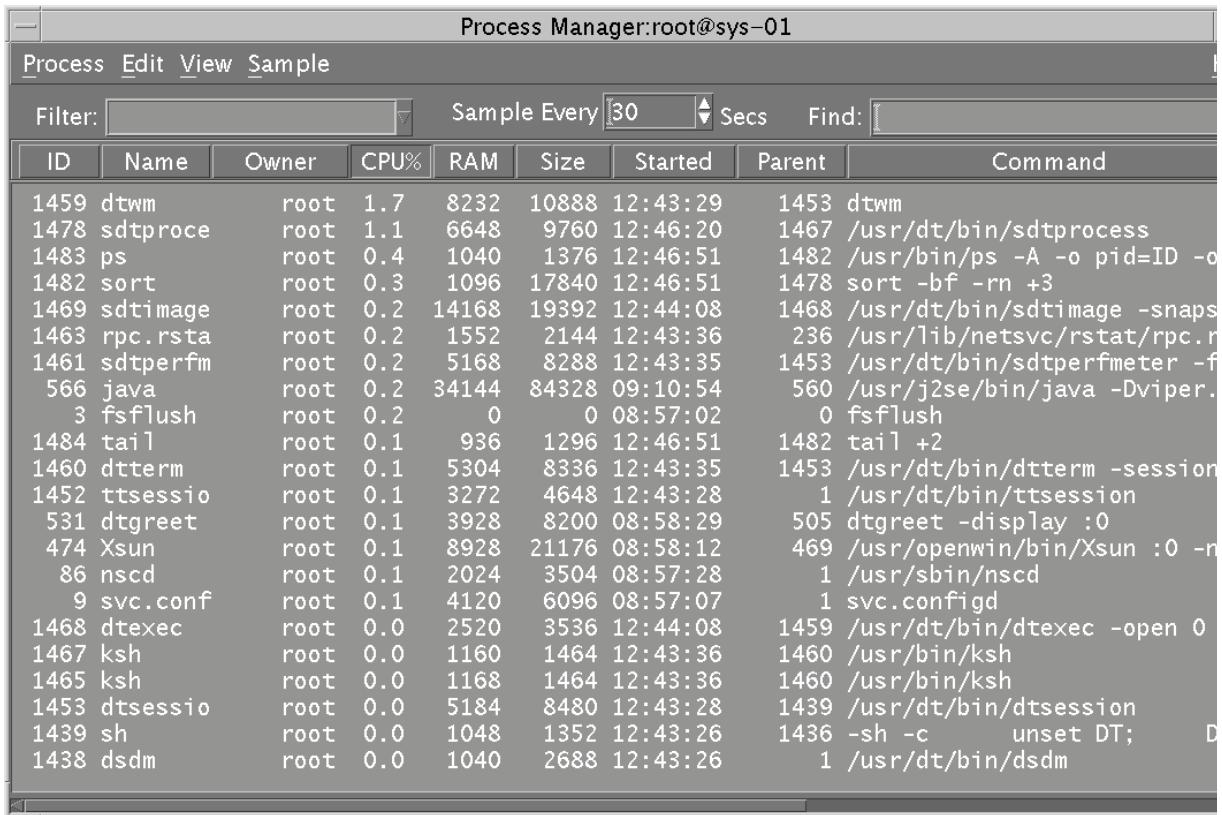


Figure 13-2 Tools Menu

You can also start the CDE Process Manager from the command line by typing the following:

```
# /usr/dt/bin/sdtprocess &
```

Figure 13-3 shows the window that appears.



The screenshot shows a terminal window titled "Process Manager:root@sys-01". The window contains a table with columns: ID, Name, Owner, CPU%, RAM, Size, Started, Parent, and Command. The table lists various processes running on the system, such as dtwm, sdtproce, ps, sort, sdtimage, rpc.rsta, sdtperfm, java, fsflush, tail, dtterm, ttsessio, dtgreet, Xsun, nscd, svc.conf, dtexec, ksh, dtsessio, sh, and dsdm. The "Command" column provides the full path to each process's executable file.

ID	Name	Owner	CPU%	RAM	Size	Started	Parent	Command
1459	dtwm	root	1.7	8232	10888	12:43:29	1453	dtwm
1478	sdtproce	root	1.1	6648	9760	12:46:20	1467	/usr/dt/bin/sdtprocess
1483	ps	root	0.4	1040	1376	12:46:51	1482	/usr/bin/ps -A -o pid=ID -o
1482	sort	root	0.3	1096	17840	12:46:51	1478	sort -bf -rn +3
1469	sdtimage	root	0.2	14168	19392	12:44:08	1468	/usr/dt/bin/sdtimage -snaps
1463	rpc.rsta	root	0.2	1552	2144	12:43:36	236	/usr/lib/netsvc/rstat/rpc.r
1461	sdtperfm	root	0.2	5168	8288	12:43:35	1453	/usr/dt/bin/sdtperfmon -f
566	java	root	0.2	34144	84328	09:10:54	560	/usr/j2se/bin/java -Dviper.
3	fsflush	root	0.2	0	0	08:57:02	0	fsflush
1484	tail	root	0.1	936	1296	12:46:51	1482	tail +2
1460	dtterm	root	0.1	5304	8336	12:43:35	1453	/usr/dt/bin/dtterm -session
1452	ttsessio	root	0.1	3272	4648	12:43:28	1	/usr/dt/bin/ttsession
531	dtgreet	root	0.1	3928	8200	08:58:29	505	dtgreet -display :0
474	Xsun	root	0.1	8928	21176	08:58:12	469	/usr/openwin/bin/Xsun :0 -n
86	nscd	root	0.1	2024	3504	08:57:28	1	/usr/sbin/nscd
9	svc.conf	root	0.1	4120	6096	08:57:07	1	svc.configd
1468	dtexec	root	0.0	2520	3536	12:44:08	1459	/usr/dt/bin/dtexec -open 0
1467	ksh	root	0.0	1160	1464	12:43:36	1460	/usr/bin/ksh
1465	ksh	root	0.0	1168	1464	12:43:36	1460	/usr/bin/ksh
1453	dtsessio	root	0.0	5184	8480	12:43:28	1439	/usr/dt/bin/dtsession
1439	sh	root	0.0	1048	1352	12:43:26	1436	-sh -c unset DT;
1438	dsdm	root	0.0	1040	2688	12:43:26	D	1 /usr/dt/bin/dsdm

Figure 13-3 CDE Process Manager Window

The Process Manager can sort processes alphabetically (Name) or numerically (ID), depending on the column that is selected.

You can initiate a search by typing text into the Find field.

To terminate a process, highlight it and press Control-C, select the Kill option from the Process menu, or select the kill option from the options that are available when you press the right mouse button.

Using the prstat Command

The prstat command examines and displays information about active processes on the system.

This command enables you to view information by specific processes, user identification (UID) numbers, central processing unit (CPU) IDs, or processor sets. By default, the prstat command displays information about all processes sorted by CPU usage. To use the prstat command, perform the command:

```
# prstat
  PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
 1641 root     4864K 4520K cpu0    59    0  0:00:00 0.5% prstat/1
 1635 root     1504K 1168K sleep   59    0  0:00:00 0.3% ksh/1
      9 root     6096K 4072K sleep   59    0  0:00:29 0.1% svc.configd/11
    566 root     82M    30M sleep   29   10  0:00:36 0.1% java/14
 1633 root     2232K 1520K sleep   59    0  0:00:00 0.1% in.rlogind/1
    531 root     8200K 2928K sleep   59    0  0:00:12 0.1% dtgreet/1
    474 root     21M    7168K sleep   59    0  0:00:11 0.1% Xsun/1
    236 root     4768K 2184K sleep   59    0  0:00:03 0.0% inetd/4
     86 root     3504K 1848K sleep   59    0  0:00:01 0.0% nscd/24
      7 root     5544K 1744K sleep   59    0  0:00:06 0.0% svc.startd/12
    154 root     2280K  824K sleep   59    0  0:00:01 0.0% in.routed/1
    509 root     6888K 2592K sleep   59    0  0:00:02 0.0% httpd/1
    240 root     5888K 1256K sleep   59    0  0:00:01 0.0% sendmail/1
    145 root     2944K  816K sleep   59    0  0:00:01 0.0% httpd/1
    347 daemon   2608K  776K sleep   59    0  0:00:00 0.0% nfsmapid/3
    206 root     1288K  600K sleep   59    0  0:00:00 0.0% utmpd/1
    344 daemon   2272K 1248K sleep  60   -20  0:00:00 0.0% sendmail/1
    107 root     2584K  784K sleep   59    0  0:00:00 0.0% sysseventd/14
    123 root     3064K  880K sleep   59    0  0:00:00 0.0% picld/4
    146 lp       2976K  448K sleep   59    0  0:00:00 0.0% httpd/1
Total: 53 processes, 171 lwps, load averages: 0.02, 0.04, 0.07
#
```

To quit the prstat command, type **q**.

Table 13-1 shows the column headings and their meanings in a `prstat` report.

Table 13-1 Column Headings for the `prstat` Report

Default Column Heading	Description
PID	The PID number of the process.
USERNAME	The login name or UID of the owner of the process.
SIZE	The total virtual memory size of the process.
RSS	The resident set size of the process in kilobytes, megabytes, or gigabytes.
STATE	The state of the process: <ul style="list-style-type: none"> • <code>cpu</code> – The process is running on the CPU. • <code>sleep</code> – The process is waiting for an event to complete. • <code>run</code> – The process is in the run queue. • <code>zombie</code> – The process terminated, and the parent is not waiting. • <code>stop</code> – The process is stopped.
PRI	The priority of the process.
NICE	The value used in priority computation.
TIME	The cumulative execution time for the process.
CPU	The percentage of recent CPU time used by the process.
PROCESS/NLWP	The name of the process/the number of lightweight processes (LWPs) in the process.



Note – The kernel and many applications are now multithreaded. A thread is a logical sequence of program instructions written to accomplish a particular task. Each application thread is independently scheduled to run on an LWP, which functions as a virtual CPU. LWPs in turn, are attached to kernel threads, which are scheduled to run on actual CPUs.



Note – Use the priocntl(1) command to assign processes to a priority class and to manage process priorities. The nice(1) command is only supported for backward compatibility to previous Solaris OS releases. The priocntl command provides more flexibility in managing processes.

Table 13-2 shows the options for the prstat command.

Table 13-2 Options for the prstat Command

Option	Description
-a	Displays separate reports about processes and users at the same time.
-c	Continuously prints new reports below previous reports.
-n <i>nproc</i>	Restricts the number of output lines.
-p <i>pidlist</i>	Reports only on processes that have a PID in the given list.
-s <i>key</i>	Sorts output lines by <i>key</i> in descending order. The five possible keys include: cpu, time, size, rss, and pri. You can use only one key at a time.
-S <i>key</i>	Sorts output lines by <i>key</i> in ascending order.
-t	Reports total usage summary for each user.
-u <i>euidlist</i>	Reports only processes that have an effective user ID (EUID) in the given list.
-U <i>uidlist</i>	Reports only processes that have a real UID in the given list.

Using the Solaris Management Console Process Tool

The Solaris Management Console provides a tool for monitoring and managing system processes. You open the Process Tool by clicking This Computer, and then clicking System Status. Then click Process.

Figure 13-4 shows the Solaris Management Console Process Tool.

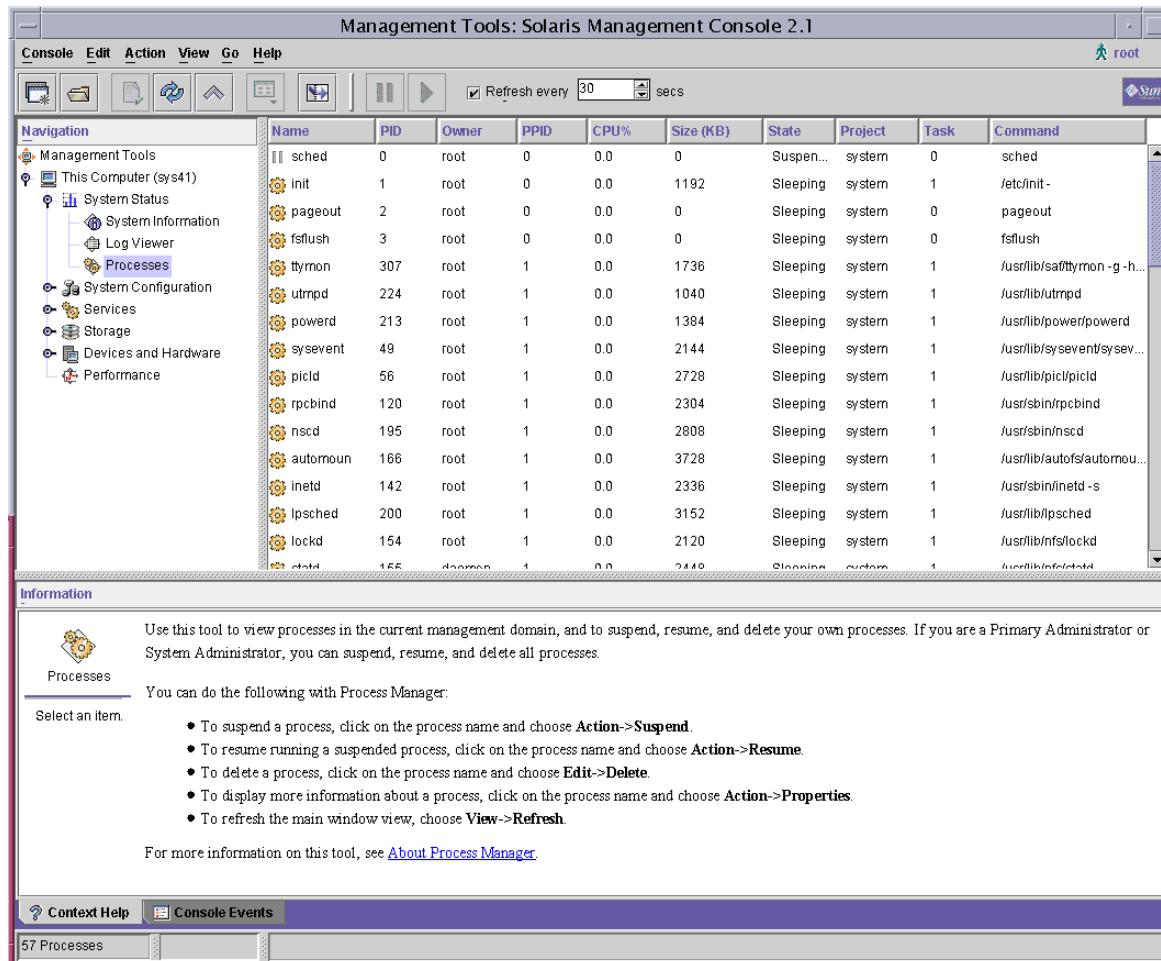


Figure 13-4 Solaris Management Console – Process Tool Window

From the Process Tool, you can do the following:

- Suspend a process. To do this, click the process name, and choose Suspend from the Action menu.
- Resume running a suspended process. To do this, click the process name, and choose Resume from the Action menu.
- Kill (delete) a process. To do this, click the process name, and choose Delete from the Edit menu.
- Display more information about a process. To do this, click the process name, and choose Properties from the Action menu.
- Refresh the main window view. To do this, choose Refresh from the View menu.

Killing Frozen Processes

You use the `kill` command or the `pkill` command to send a signal to one or more running processes. You would typically use these commands to kill an unwanted process.

Using the `kill` and `pkill` Commands

You use the `kill` or `pkill` commands to terminate one or more processes.

The format for the `kill` command is:

```
kill -signal PID
```

To show all of the available signals used with the `kill` command:

```
kill -l
```

The format for the `pkill` command is:

```
pkill -signal Process
```

Before you can terminate a process, you must know its name or PID. Use either the `ps` or `pgrep` command to locate the PID for the process.

The following examples uses the `pgrep` command to locate the PID for the `mail` processes.

```
# pgrep -l mail
241 sendmail
240 sendmail
#
# pkill sendmail
```

The following examples use the `ps` and `pkill` commands to locate and terminate the `sendmail` process.

```
# ps -e | grep sendmail
 241 ?          0:00 sendmail
 240 ?          0:02 sendmail
# kill 241
```

Killing Frozen Processes

To terminate more than one process at the same time, use the following syntax:

```
# kill -signal PID PID PID PID  
# pkill signal process process
```

You use the `kill` command without a signal on the command line to send the default Signal 15 to the process. This signal usually causes the process to terminate.

Table 13-3 shows some signals and names.

Table 13-3 Process Signal Numbers and Names

Signal Number	Signal Name	Event	Default Action
1	SIGHUP	Hangup	Exit
2	SIGINT	Interrupt	Exit
9	SIGKILL	Kill	Exit
15	SIGTERM	Terminate	Exit

- 1, SIGHUP – A hangup signal to cause a telephone line or terminal connection to be dropped. For certain daemons, such as `inetd` and `in.named`, a hangup signal will cause the daemon to reread its configuration file.
- 2, SIGINT – An interrupt signal from your keyboard—usually from a Control-C key combination.
- 9, SIGKILL – A signal to kill a process. A process cannot ignore this signal.
- 15, SIGTERM – A signal to terminate a process in an orderly manner. Some processes ignore this signal.

A complete list of signals that the kill command can send can be found by executing the command `kill -l`, or by referring to the man page for signal:

```
# man -s3head signal
```

Some processes can be written to ignore Signal 15. Processes that do not respond to a Signal 15 can be terminated by force by using Signal 9 with the `kill` or `pkill` commands. You use the following syntax:

```
# kill -9 PID
# pkill -9 process
```



Caution – Use the `kill -9` or `pkill -9` command as a last resort to terminate a process. Using the `-9` signal on a process that controls a database application or a program that updates files can be disastrous. The process is terminated instantly with no opportunity to perform an orderly shutdown.

Performing a Remote Login

When a workstation is not responding to your keyboard or mouse input, the CDE might be frozen. In such cases, you may be able to remotely access your workstation by using the `rlogin` command or by using the `telnet` command from another system.

Killing the Process for a Frozen Login

After you are connected remotely to your system, you can invoke the `pkill` command to terminate the corrupted session on your workstation.

In the following examples, the `rlogin` command is used to log in to `sys42`, from which you can issue a `pkill` or a `kill` command.

```
# rlogin sys-02
Password:
Last login: Sun Oct 24 13:44:51 from sys-01
Sun Microsystems Inc.      SunOS 5.10          s10_68  Sep. 20, 2004
# pkill -9 Xsun
or
# ps -e | grep Xsun
 442 ?          0:01 Xsun
# kill -9 442
```

Suspending and Terminating Processes with SMC

The SMC GUI tool allows an authorized user to suspend or terminate processes.

To suspend a process, you must first select the process from the list displayed. Once selected, you must select the Suspend option from the Action menu, as shown in Figure 13-5:

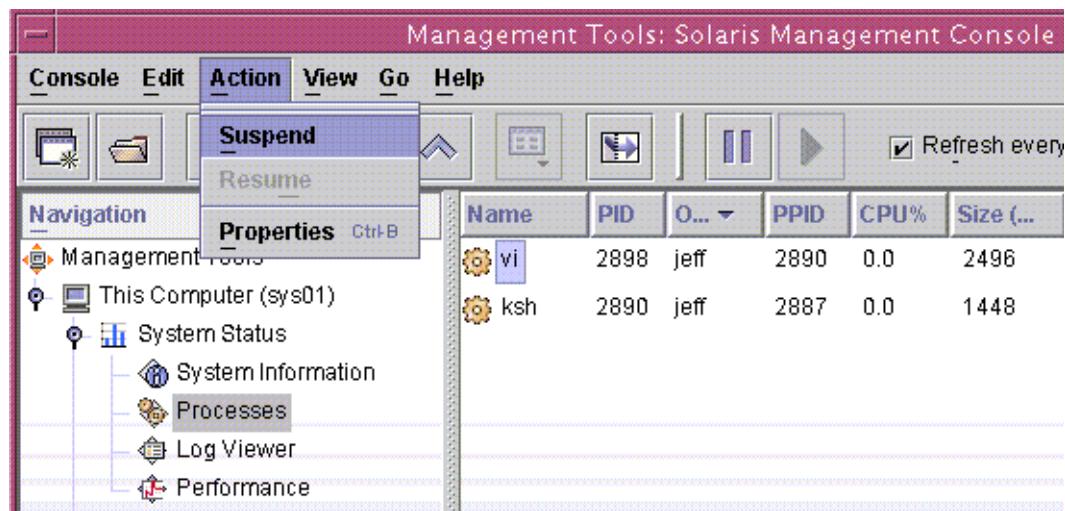


Figure 13-5 Action Menu Options

Having suspended a process, that process remains in its suspended state until it is resumed using the Resume option from the Action menu.

The SMC Process window has a filter mechanism which allows you to display only those processes whose details match the filter being applied. This is especially useful when you need to terminate processes that belong to a specific user.

In the examples shown, the filter has been applied so that only the processes belonging to the user called `jeff` are displayed.

Processes can be terminated using the Trash Can icon. The process must be selected from the list of processes before it can be terminated, as shown in Figure 13-6:

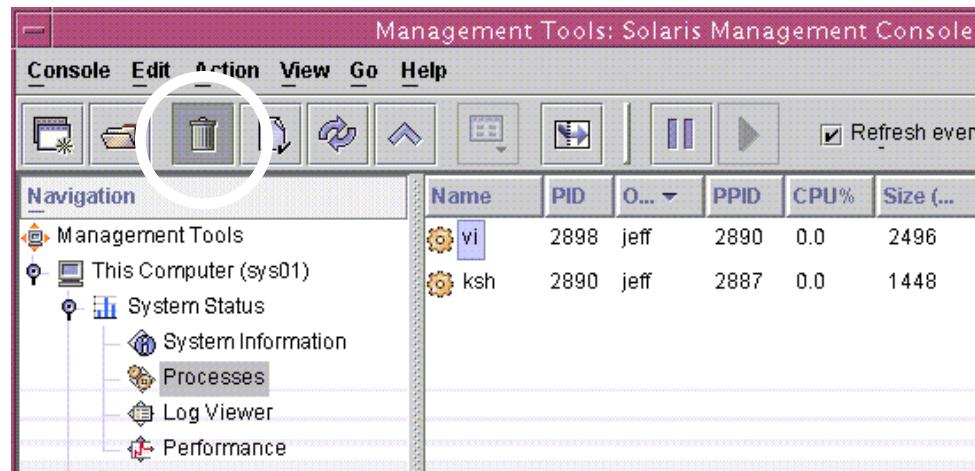


Figure 13-6 List of Processes

When the Trash Can icon is used, you are asked to confirm the deletion, as shown in Figure 13-7:



Figure 13-7 Warning: Delete Process? Dialog Box

Select Delete to confirm the deletion of the process.

Scheduling an Automatic One-Time Execution of a Command

Use the `at` command to automatically execute a job only once at a specified time.

Using the `at` Command

The format for the `at` command is:

```
at -m -q queuename time date
at -r job
at -l
```

Table 13-4 shows the options you can use to instruct the cron process on how to execute an `at` job.

Table 13-4 Options for the `at` Command

Option	Description
<code>-m</code>	Sends mail to the user after the job has finished
<code>-r job</code>	Removes a scheduled <code>at</code> job from the queue
<code>-q queuename</code>	Specifies a specific queue
<code>time</code>	Specifies a time for the command to execute
<code>-l</code>	Reports all jobs scheduled for the invoking user
<code>date</code>	Specifies an optional date for the command to execute, which is either a month name followed by a day number or a day of the week

For example, to create an at job to run at 9:00 p.m. to locate and verify the file type of core files from the /export/home directory, perform the command:

```
# at 9:00 pm
at> find /export/home -name core -exec file {} \; >> /var/tmp/corelog
at> <EOT>
commands will be executed using /sbin/sh
job 1098648000.a at Mon Oct 25 21:00:00 2004
```

To display information about the execution times of jobs, perform the command:

```
# at -l 1098648000.a
1098648000.a      Mon Oct 25 21:00:00 2004
```

To display the jobs queued to run at specified times by chronological order of execution, perform the command:

```
# atq
Rank      Execution Date      Owner      Job      Queue      Job Name
1st      Oct 25, 2004 21:00    root      1098648000.a    a      stdin
```

To view all the at jobs currently scheduled in the queue, perform the command:

```
# ls -l /var/spool/cron/atjobs
total 4
-r-Sr--r--  1 root      root      1044 Oct 25 13:48 1098648000.a
```

You can also use the at command to remove a job from the at queue.

For example, to remove job 1098648000.a from the at queue, perform the command:

```
# at -r 1098648000.a
# atq
Rank      Execution Date      Owner      Job      Queue      Job Name
```

Controlling Access to the at Command

As the root user, you control who has access to the at command with the at.deny and at.allow files.

The /etc/cron.d/at.deny File

By default, the Solaris OS includes the /etc/cron.d/at.deny file. This file identifies users who are prohibited from using the at command. The file format is one user name per line. The file initially contains:

```
daemon  
bin  
nuucp  
listen  
nobody  
noaccess
```

A user who is denied access to the at command receives the following message when attempting to use the command:

```
at: you are not authorized to use at. Sorry.
```

If only the /etc/cron.d/at.deny file exists but is empty, then all logged-in users can access the at command.

The /etc/cron.d/at.allow File

The /etc/cron.d/at.allow file does not exist by default, so all users (except those listed in the /etc/cron.d/at.deny file) can create at jobs. By creating the /etc/cron.d/at.allow file, you create a list of only those users who are allowed to execute at commands.

The /etc/cron.d/at.allow file consists of user names, one per line.

The interaction between the at.allow and the at.deny files follows these rules:

- If the at.allow file exists, only the users listed in this file can execute at commands.
- If the at.allow file does not exist, all users, except for users listed in the at.deny file, can execute at commands.
- If neither file exists, only the root user can use the at command.
- If a user is listed in both files, the user is denied.

Scheduling an Automatic Recurring Execution of a Command

You can use the cron facility to schedule regularly recurring commands. Users can submit a command to the cron facility by modifying their crontab file.

All crontab files are maintained in the /var/spool/cron/crontabs directory and are stored as the login name of the user that created the cron job.

The cron daemon is responsible for scheduling and running these jobs.

Note – The clock daemon, cron, starts at system boot and runs continuously in the background.



Introducing the crontab File Format

A crontab file consists of lines of six fields each. The fields are separated by spaces or tabs. The first five fields provide the date and time the command is to be scheduled. The last field is the full path to the command.

Note – If the command field contains a percent (%) character, then all subsequent characters are passed to the command as standard input.



Scheduling an Automatic Recurring Execution of a Command

These first five fields are separated by spaces and indicate when the command will be executed. See Figure 13-8.

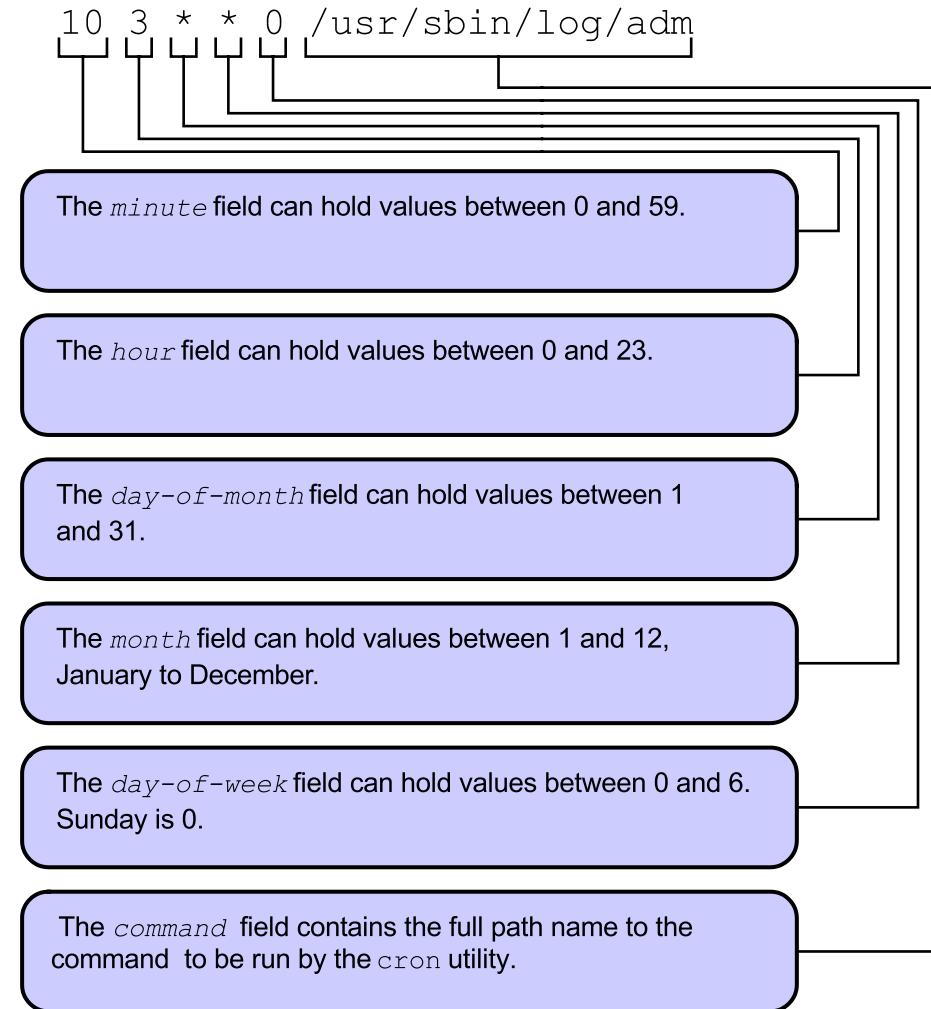


Figure 13-8 First Five Fields in a crontab File

The first five fields follow the format rules shown in Table 13-5.

Table 13-5 Rules for the crontab Fields

Value	Rule	Example
<i>n</i>	Matches if field value is <i>n</i>	As shown in the preceding figure for hour or minute, a 3 or 10
<i>n,p,q</i>	Matches if field value is <i>n</i> , <i>p</i> , or <i>q</i>	Every 10 minutes would be represented by 0,10,20,30,40,50
<i>n-p</i>	Matches if field has values between <i>n</i> and <i>p</i> inclusive	The hours between 1:00 a.m. and 4:00 a.m. would be represented by 1-4
*	Matches all legal values	As in the preceding example for the month, representing every month.

Using the crontab Command

The crontab command enables the user to view, edit, or remove a crontab file.

Viewing a crontab File

To view the contents of the root crontab file, run the crontab -l command as the root user.

```
# crontab -l
#ident  "@(#)root      1.21      04/03/23 SMI"
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscared_clean ] &&
/usr/lib/gss/gsscared_clean
#10 3 * * * /usr/lib/krb5/kprop_script __slave_kdcs__
```

This is the same command that users run to view the contents of their own crontab file.

As the root user, you can view the contents of any regular user's crontab file by performing the command:

```
# crontab -l username
```

Editing a crontab File



Caution – If you accidentally enter the crontab command on the command line without an option (-l, -e, -r), press the interrupt keys Control-C to exit. Do not press Control-D, this action overwrites the existing crontab file with an empty file.

To create or edit a crontab file, follow these steps:

1. Check that the EDITOR variable is set to the editor you want to use. This instructs the cron utility which editor to use to open the file.

```
# EDITOR=vi  
# export EDITOR
```

2. Run the following crontab command to open your crontab file, and add the appropriate entry.

```
# crontab -e  
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console  
:wq
```



Note – If the users do not redirect the standard output and standard error of their commands in the crontab file, any generated output or errors are mailed electronically to the user.

Removing a crontab File

The correct way to remove a crontab file is to invoke the command:

```
# crontab -r username
```

Typical users can remove only their own crontab file. The root user can delete any user's crontab file.

Controlling Access to the crontab Command

You can control access to the crontab command with two files in the /etc/cron.d directory—the cron.deny file and the cron.allow file.

These files permit only specified users to perform crontab tasks, such as creating, editing, displaying, or removing their own crontab files.

The /etc/cron.d/cron.deny File

The Solaris OS provides a default cron.deny file. The file consists of a list of user names, one per line, of the users who are not allowed to use cron. The following is an example of the contents of a cron.deny file:

```
daemon  
bin  
nuucp  
listen  
nobody  
noaccess
```

The /etc/cron.d/cron.allow File

The /etc/cron.d/cron.allow file does not exist by default, so all users (except those listed in the cron.deny file) can access their crontab file. By creating a cron.allow file, you can list only those users who can access crontab commands.

The file consists of a list of user names, one per line.

The interaction between the cron.allow and the cron.deny files follows these rules:

- If the cron.allow file exists, only the users listed in this file can create, edit, display, or remove crontab files.
- If the cron.allow file does not exist, all users, except for users listed in the cron.deny file, can create, edit, display, or remove crontab files.
- If neither file exists, only the root user can run the crontab command.
- If a user is listed in both files, the user is denied.

Using the Solaris™ Management Console Job Scheduler Tool

The Solaris™ Management Console contains a Scheduled Jobs tool to create and schedule jobs on your system. Users can manage jobs if the following conditions exist:

- Their user name appears in the /etc/cron.d/cron.allow file.
- Their user name does not appear in the /etc/cron.d/cron.deny file.
- The /etc/cron.d/cron.allow and /etc/cron.d/cron.deny files do not exist, and you are the root user.

To open the Job Scheduler from the Solaris Management Console, click This Computer and then click Service, and finally, click Scheduled Jobs.

See Figure 13-9 for an example of the Solaris Management Console Job Scheduler window.

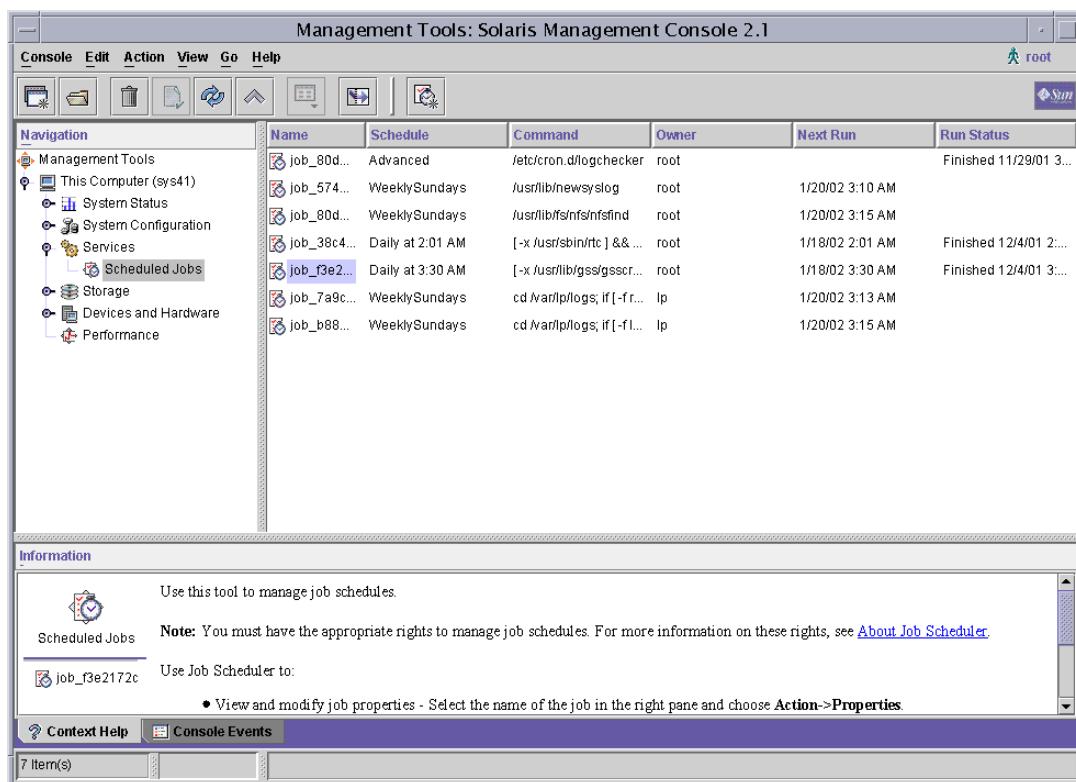


Figure 13-9 Solaris Management Console – Scheduled Jobs Window

You can use the Job Scheduler to:

- View and modify job properties
Select the name of the job in the view pane, and choose Properties from the Action menu.
- Delete a job
Select the job name and choose Delete from the Edit menu. The root user can delete all jobs. Users can only view and delete their own jobs.
- Add a scheduled job
Choose Add Scheduled Job on the Action menu.
- Enable and disable job logging, and set search paths
Choose Scheduled Job Policies on the Action menu.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using Process Control (Level 1)

In this exercise, you use the Process Tool and the `prstat` command to monitor and kill processes. You create an `at` job and create an entry in a `crontab` file.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following tasks:

- Start the Process Tool. Run the `prstat` command in a window. In a separate window, run the `find` / command. Make note of the CPU percentages for the `find` command, as displayed by the `prstat` command and the Process Tool. Open a third window, and identify the PID of the shell running in it. Use the Process Tool to show the ancestry of the shell process. Use the Process Tool to kill the shell process. Use the Process Tool to send the `TERM` signal to the `prstat` process. Exit the Process Tool when you are finished.

(Steps 1–6 in the Level 2 lab)

- Identify the device associated with your current terminal, and display the current time of day. Submit an `at` job that echoes `Test Complete` to your current window. Have the job run five minutes from the current time, and submit it to the queue called `x`. Display the `at` job in the queue.

(Steps 7–10 in the Level 2 lab)

- Set the `EDITOR` variable to `vi`. Use the `crontab` command to determine when the `logadm` process is scheduled to run. Use the `crontab` command to edit the `crontab` file for the `root` user. Add an entry that sends the message `It works!` to your current window five minutes from the current time.

(Steps 11–14 in the Level 2 lab)

Exercise: Using Process Control (Level 2)

In this exercise, you use the Process Tool and the `prstat` command to monitor and kill processes. You create an `at` job and create an entry in a `crontab` file.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Start the Process Tool. Run the `prstat` command in a window. In a separate window, run the `find` / command. Make note of the CPU percentages for the `find` command, as displayed by the `prstat` command and the Process Tool. Open a third window, and identify the PID of the shell running in it. Use the Process Tool to show the ancestry of the shell process. Use the Process Tool to kill the shell process. Use the Process Tool to send the `TERM` signal to the `prstat` process. Exit the Process Tool when you are finished.
- Identify the device associated with your current terminal, and display the current time of day. Submit an `at` job that echoes `Test Complete` to your current window. Have the job run five minutes from the current time, and submit it to the queue called `x`. Display the `at` job in the queue.
- Set the `EDITOR` variable to `vi`. Use the `crontab` command to determine when the `logadm` process is scheduled to run. Use the `crontab` command to edit the `crontab` file for the `root` user. Add an entry that sends the message `It works!` to your current window five minutes from the current time.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Start the Process Tool either by selecting the Find Process option from the Front Panel Tools menu in CDE or by invoking the appropriate command from the command line.
In the Process Tool display, sort the listing according to `CPU%`, and change the sample time to five seconds.
2. Open a second terminal window, and run the `prstat` command.
3. Position the Process Tool and the window in which the `prstat` command is running so that you can observe both simultaneously. In an available window, run the `find` command to list all files on your system. Observe how the Process Tool and the `prstat` command display statistics for the `find` command.

What is the maximum percentage of recent CPU time used by the `find` command as it executes?

4. Open a third terminal window, and run the `ps` command to determine the PID of the shell. Record the PID you find.
5. In the Process Tool, locate and select the shell process you identified in the previous step. Select the Show Ancestry option from the Process menu in the Process Tool. What is the name and PID of the first process listed?
6. Close the Show Ancestry window. Again, select the shell process you identified in Step 4. From the Process menu in the Process Tool, select the Kill option. What happens?
7. In the Process Tool, use the Find function to locate the `prstat` process. Select the Signal option from the Process menu. In the Signal fill-in field, enter the TERM signal, and click OK. What happens to the `prstat` process? Close the Process Tool when you are finished.
8. Identify the device associated with your current terminal by using the `tty` command, and display the current time of day.
9. Submit an at job that echoes Test Complete to your current window. Have the job run five minutes from the current time, and submit it to the queue called `x`.
10. Display the at job in the queue.
11. Open a new window and set and export the `EDITOR` environment variable to use the `vi` editor to edit `crontab` files.

If you are using the Bourne or Korn shell, perform the command:

```
# EDITOR=vi  
# export EDITOR
```

If you are using the C shell, perform the command:

```
# setenv EDITOR vi
```

12. Use the `crontab` command to view the current `crontab` file for the root user.
13. When is the `logadm` process scheduled to run?
14. Use the `crontab` command to edit the `crontab` file for the root user. Add an entry that sends the message It works! to your current window five minutes from now. For example, if the current time is 10:25, make an entry in your `crontab` file for the 30th minute of the same hour.

Save the file, and quit the `vi` edit session. In about five minutes, you should see the result in your window.

Exercise: Using Process Control (Level 3)

In this exercise, you use the Process Tool and the prstat command to monitor and kill processes. You create an at job and create an entry in a crontab file.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Task Summary

In this exercise, you accomplish the following:

- Start the Process Tool. Run the `prstat` command in a window. In a separate window, run the `find` / command. Make note of the CPU percentages for the `find` command, as displayed by the `prstat` command and the Process Tool. Open a third window, and identify the PID of the shell running in it. Use the Process Tool to show the ancestry of the shell process. Use the Process Tool to kill the shell process. Use the Process Tool to send the `TERM` signal to the `prstat` process. Exit the Process Tool when you are finished.
- Identify the device associated with your current terminal, and display the current time of day. Submit an `at` job that echoes `Test Complete` to your current window. Have the job run five minutes from the current time, and submit it to the queue called `x`. Display the `at` job in the queue.
- Set the `EDITOR` variable to `vi`. Use the `crontab` command to determine when the `logadm` process is scheduled to run. Use the `crontab` command to edit the `crontab` file for the `root` user. Add an entry that sends the message `It works!` to your current window five minutes from the current time.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Start the Process Tool either by selecting the Find Process option from the Front Panel Tools menu in CDE or by invoking the appropriate command from the command line.

```
# /usr/dt/bin/sdtprocess &
```

In the Process Tool display, sort the listing according to CPU%, and change the sample time to five seconds.

2. Open a second terminal window, and run the prstat command.

```
# prstat
```

3. Position the Process Tool and the window in which the prstat command is running so that you can observe both simultaneously. In an available window, run the find command to list all files on your system. Observe how the Process Tool and the prstat command display statistics for the find command.

```
# find /
```

What is the maximum percentage of recent CPU time used by the find command as it executes?

This varies according to your system configuration. Some systems might display values in the 20-percent range.

4. Open a third terminal window, and run the ps command to determine the PID of the shell. Record the PID you find.

```
# ps
```

Your value appears here.

5. In the Process Tool, locate and select the shell process you identified in the previous step. Select the Show Ancestry option from the Process menu in the Process Tool. What is the name and PID of the first process listed?

The PID varies. On systems running the CDE, the first process listed should be /usr/dt/bin/dtlogin.

6. Close the Show Ancestry window. Again, select the shell process you identified in Step 4. From the Process menu in the Process Tool, select the Kill option. What happens?

The process stops, and the window no longer appears.

7. In the Process Tool, use the Find function to locate the prstat process. Select the Signal option from the Process menu. In the Signal fill-in field, enter the TERM signal, and click OK. What happens to the prstat process? Close the Process Tool when you are finished.

The prstat process terminates, and the prompt appears in the window in which it ran.

8. Identify the device associated with your current terminal by using the tty command, and display the current time of day.

```
# tty
(something like /dev/pts/4 should appear
# date
(current date/time appears)
```

9. Submit an at job that echoes Test Complete to your current window. Have the job run five minutes from the current time, and submit it to the queue called x.

```
# at -q x 13:30
at> echo "Test Complete" > /dev/pts/# (# is from the tty command)
at> <Control-D>
commands will be executed using /sbin/sh
job 958163400.x at Fri Oct 29 13:30:00 2004
#
```

10. Display the at job in the queue.

```
# atq
```

11. Open a new window and set and export the EDITOR environment variable to use the vi editor to edit crontab files.

If you are using the Bourne or Korn shell, perform the command:

```
# EDITOR=vi
# export EDITOR
```

If you are using the C shell, perform the command:

```
# setenv EDITOR vi
```

12. Use the crontab command to view the current crontab file for the root user.

```
# crontab -l
```

Exercise: Using Process Control (Level 3)

13. When is the logadm process scheduled to run?

Ten minutes after 3:00 a.m. on all days

14. Use the crontab command to edit the crontab file for the root user. Add an entry that sends the message It works! to your current window five minutes from now. For example, if the current time is 10:25, make an entry in your crontab file for the 30th minute of the same hour.

```
# tty  
/dev/pts/#  
# date  
Thu Nov  4 10:25:14 PDT 2004  
# crontab -e
```

Add the following line, but substitute the correct time and terminal device:

```
30 10 * * * /usr/bin/echo "It works!" > /dev/pts/#
```

Save the file, and quit the vi edit session. In about five minutes, you should see the result in your window.

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications
-

Module 14

Performing File System Backups

Objectives

Upon completion of this module, you should be able to:

- Identify the fundamentals of backups
- Back up an unmounted file system

The course map in Figure 14-1 shows how this module fits into the current instructional goal.

Performing System Backups and Restores

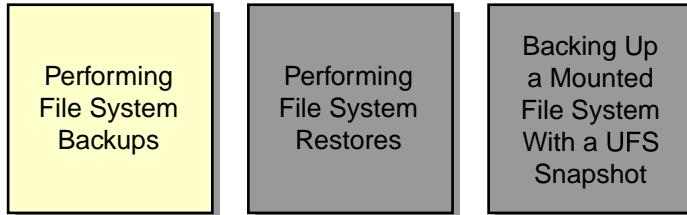


Figure 14-1 Course Map

Introducing the Fundamentals of Backups

A crucial function of system administration is to backup file systems. Backups safeguard against data loss, damage, or corruption. Backup tapes are often referred to as dump tapes.

Importance of Routine File System Backups

To back up file systems, you copy file systems to removable media, such as a tape. You perform backups on a regular basis to prevent loss of data due to:

- Accidental deletion of files
- Hardware failures
- Problems with re-installations or system upgrades
- System crashes
- System break-ins by unauthorized users, compromising data integrity
- Natural disasters

Tape Media Types

Table 14-1 shows typical tape media that you can use to store file systems during the backup process. Select media based on the availability of equipment and your preference.

Table 14-1 Tape Media Types

Media Type	Capacity
1/4-inch cartridge (QIC) ¹ cartridge tape	8 Gbytes
8-mm cartridge tape	40 Gbytes
4-mm digital audio tape (DAT) ² cartridge tape	24 Gbytes
DLT ³ 1/2-inch cartridge tape	50 Gbytes Up to 80 Gbytes with compression
SDLT ⁴ cartridge tape	160 Gbytes Up to 320 Gbytes with compression
LTO ⁵ cartridge tape	100 Gbytes (Generation One) 200 Gbytes (Generation Two)

1. QIC stands for quarter-inch tape.
2. DAT stands for digital audio tape.
3. DLT stands for digital linear tape.
4. SDLT stands for super digital linear tape.
5. LTO stands for linear tape open.

The capacities in the table are approximate. Tape capacity increases with new technology. Check the documentation that comes with the tape device to determine the capacity.

Tape Drive Naming

All tape drives have logical device names that you use to reference the device on the command line. Figure 14-2 shows the format that all logical device names use.

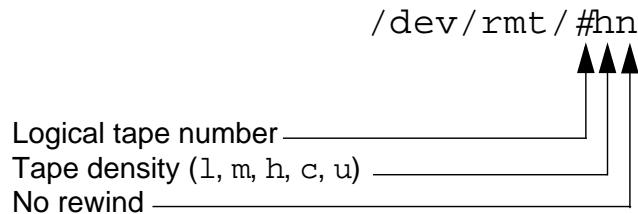


Figure 14-2 Logical Device Name Format

The logical tape numbers in the tape drive names always start with 0. For example:

- The first instance of a tape drive:
`/dev/rmt/0`
- The second instance of a tape drive:
`/dev/rmt/1`
- The third instance of a tape drive:
`/dev/rmt/2`

Two optional parameters further define the logical device name:

- Tape density – Five values can be given in the tape device name: l (low), m (medium), h (high), c (compressed), or u (ultra compressed).
- No rewind – The letter n at the end of a tape device name indicates that the tape should not be rewound when the current operation completes.

Tape densities depend on the tape drive. Check the manufacturer's documentation to determine the correct densities for the tape media.

Tape drives that support data compression contain internal hardware that performs the compression. If you back up a software-compressed file to a tape drive with hardware compression, the resulting file may be larger in size.

Tape Drive Control

You use the `mt` command (magnetic tape control) to send instructions to the tape drive. Not all tape drives support all `mt` commands.

The format for the `mt` command is:

```
mt -f tape-device-name command count
```

You use the `-f` option to specify the tape device name, typically a no-rewind device name. If no `-f` option is used, the default tape device file `/dev/rmt/0` is used.

Using the `mt` Command

Table 14-2 lists some of the `mt` commands that you can use to control a magnetic tape drive.

Table 14-2 Definitions of `mt` Commands

Command	Definition
<code>mt status</code>	Displays status information about the tape drive
<code>mt rewind</code>	Rewinds the tape
<code>mt offline</code>	Rewinds the tape and, if appropriate, takes the drive unit offline and if the hardware supports it, unloads
<code>mt fsf count</code>	Moves the tape forward <i>count</i> records

Assuming the tape was rewound to the start of tape, the following command positions the tape at the beginning of the third tape record.

```
# mt -f /dev/rmt/0n fsf 2
```

Strategies for Scheduled Backups

The most common method to schedule backups is to perform cumulative incremental backups daily. This schedule is recommended for most situations.

To set up a backup schedule, determine:

- The file systems to back up
- A backup device (for example, tape drive)
- The number of tapes to use for the backup
- The type of backup (for example, full or incremental)
- The procedures for marking and storing tapes
- The time it takes to perform a backup

Determining File System Names to Back Up

Display the contents of the `/etc/vfstab` file. Then view the `mount point` column to find the name of the file system that you want to back up.

Determining the Number of Tapes

You determine the number of tapes for a backup according to the size of the file system you are backing up.

To determine the size of the file system, use the `ufsdump` command with the `S` option. The following are the command formats:

```
# ufsdump OS filesystem_name
<number reported>
```

or

```
# ufsdump 3S filesystem_name
<number reported>
```

The numeric option determines the appropriate dump level. The output is the estimated number of bytes that the system requires for a complete backup.

Divide the reported bytes by the capacity of the tape to determine how many tapes you need to backup the file system.

Determining Back Up Frequency and Levels

You determine how often and at what level to backup each file system. The level of a backup refers to the amount of information that is backed up.

Identifying Incremental and Full Backups

You can perform a full backup or an incremental backup of a file system. A full backup is a complete file system backup. An incremental backup copies only files in the file system that have been added or modified since a previous lower-level backup.

You use dump level 0 to perform a full backup. You use dump levels 1 through 9 to schedule incremental backups. The level numbers have no meaning other than their relationship to each other as a higher or lower number.

Figure 14-3 shows *an example* of a file system backup performed in incremental levels.

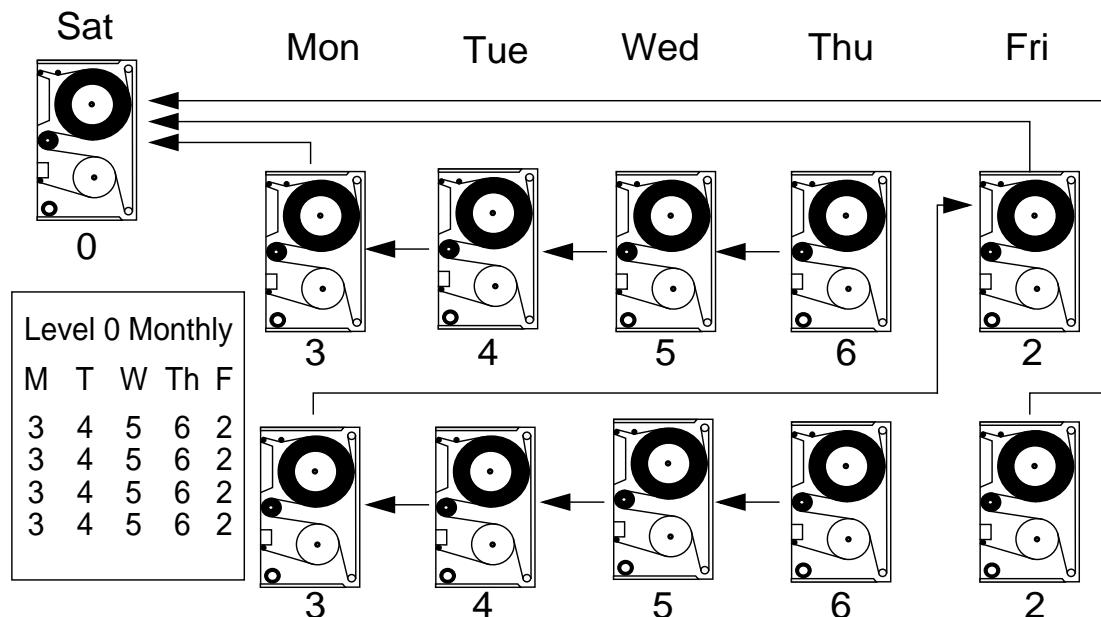


Figure 14-3 Sample of Incremental Backup Strategy

Table 14-3 defines the elements of the sample incremental backup strategy shown in Figure 14-3 on page 14-7.

Table 14-3 Incremental Back Up Level Definitions

Level	Example
0 (Full)	Performed once each month.
3	Performed every Monday. The backup copies new or modified files since the last lower-level backup (for example, 0).
4	Performed every Tuesday. The backup copies new or modified files since the last lower-level backup (for example, 3).
5	Performed every Wednesday. The backup copies new or modified files since the last lower-level backup (for example, 4).
6	Performed every Thursday. The backup copies new or modified files since the last lower-level backup (for example, 5).
2	Performed every Friday. The backup copies new or modified files since the last lower-level backup, which is the Level 0 backup at the beginning of the month.

Note – Many system administrators use the crontab utility to start a script that runs the ufsdump command.



The /etc/dumpdates File

The /etc/dumpdates file records backups if the -u option is used with the `ufsdump` command. Each line in the /etc/dumpdates file shows the file system that was backed up and the level of the last backup. It also shows the day, the date, and the time of the backup.

The following is an example /etc/dumpdates file:

```
# cat /etc/dumpdates
/dev/rdsk/c0t2d0s6  0 Fri Nov  5  19:12:27  2004
/dev/rdsk/c0t2d0s0  0 Fri Nov  5  20:44:02  2004
/dev/rdsk/c0t0d0s7  0 Tue Nov  9  09:58:26  2004
/dev/rdsk/c0t0d0s7  1 Tue Nov  9  16:25:28  2004
```

When an incremental backup is performed, the `ufsdump` command consults the /etc/dumpdates file. It looks for the date of the next lower-level backup. Then, the `ufsdump` command copies to the backup media all of the files that were modified or added since the date of that lower-level backup.

When the backup is complete, the /etc/dumpdates file records a new entry that describes this backup. The new entry replaces the entry for the previous backup at that level.

You can view the /etc/dumpdates file to determine if the system is completing backups. If a backup does not complete because of equipment failure, the /etc/dumpdates file does not record the backup.



Note – When you are restoring an entire file system, check the /etc/dumpdates file for a list of the most recent dates and levels of backups. Use this list to determine which tapes are needed to restore the entire file system. The tapes should be physically marked with the dump level and date of the backup.

Backing Up an Unmounted File System

Check that the file system is inactive, or unmounted, before you back the system up. If the file system is active, the output of the backup can be inconsistent, and you could find it impossible to restore some of the files correctly.

The `ufsdump` Command

The standard Solaris OS command for `ufs` file system backups is `/usr/sbin/ufsdump`.

The format for the `ufsdump` command is:

`ufsdump option(s) argument(s) filesystem_name`

You can use this command to back up a complete or a partial file system. Backups are often referred to as dumps.

Options for the `ufsdump` Command

Table 14-4 defines several common options for the `ufsdump` command.

Table 14-4 Options for the `ufsdump` Command

Option	Description
0–9	Back up level. Level 0 is a full backup of the file system. Levels 1 through 9 are incremental backups of files that have changed since the last lower-level backup. When no backup level is given, the default is level 9.
v	Verify. After each tape is written, the system verifies the contents of the media against the source file system. If any discrepancies occur, the system prompts the operator to insert new media and repeat the process. Use this option only on an unmounted file system. Any activity in the file system causes the system to report discrepancies.
s	Size estimate. This option allows you to estimate the amount of space that will be needed on the tape to perform the level of backup you want.
l	Autoload. You use this option with an autoloading (stackloader) tape drive.
o	Offline. When the backup is complete, the system takes the drive offline, rewinds the tape (if you use a tape), and, if possible, ejects the media.
u	Update. The system creates an entry in the <code>/etc/dumpdates</code> file with the device name for the file system disk slice, the backup level (0–9), and the date. If an entry already exists for a backup at the same level, the system replaces the entry.
n	Notify. The system sends messages to the terminals of all logged-in users who are members of the <code>sys</code> group to indicate that the <code>ufsdump</code> command requires attention.
f <i>device</i>	Specify. The system specifies the device name of the file system backup. When you use the default tape device, <code>/dev/rmt/0</code> , you do not need the -f option. The system assumes the default.

Tape Back Ups

You use the `ufsdump` command to create file system backups to tape. The dump level (0–9) specified in the `ufsdump` command determines which files to back up.

Using the `ufsdump` Command

Perform the following steps to use the `ufsdump` command to start a tape backup:

1. Become the root user to change the system to single-user mode, and unmount the file systems.

```
# /usr/sbin/shutdown -y -g300 "System is being shutdown for backup"
```

```
Shutdown started.      Mon Oct 11 12:22:33 BST 2004
```

```
Broadcast Message from root (pts/1) on host1 Mon Oct 11 12:22:33...
The system host1 will be shut down in 5 minutes
System is being shutdown for backup
(further output omitted)
```

2. Verify that the `/export/home` file system was unmounted with the `shutdown` command. If not, unmount it manually.
3. Check the integrity of the file system data with the `fsck` command.

```
# fsck /export/home
```

4. Perform a full (Level 0) backup of the `/export/home` file system.

```
# ufsdump 0uf /dev/rmt/0 /export/home
ufsdump 0uf /dev/rmt/0 /export/home
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Mon Oct 11 12:30:44 2004
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsck/c0t0d0s7 (host1:/export/home) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 1126 blocks (563KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 1086 blocks (543KB) on 1 volume at 1803 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Mon Oct 11 12:42:12 2004
#
```

Remote Backups to a Tape

You can use the `ufsdump` command to perform a backup on a remote tape device.

The format for the `ufsdump` command is:

```
ufsdump options remotehost:tapedevice filesystem
```

To perform remote backups across the network, the system with the tape drive must have an entry in its `/.rhosts` file for every system that uses the tape drive.

Using the `ufsdump` Command

The following example shows how to perform a full (Level 0) backup of the `/export/home` file system on the `host1` system, to the remote tape device on the `host2` system.

```
# ufsdump 0uf host2:/dev/rmt/0 /export/home
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Mon Oct 11 13:30:44 2004
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsck/c0t0d0s7 (host1:/export/home) to
host2:/dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 320 blocks (160KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 318 blocks (159KB) on 1 volume at 691 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Mon Oct 11 13:44:12 2004
#
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Backing Up a File System (Level 1)

In this exercise, you back up an available file system on your system.

Preparation

This exercise requires a system that is configured with a tape drive and a file system that is available to unmount. This exercise assumes that the /export/home file system exists on a separate partition from the / (root) file system and can be unmounted. Identify the slice on which the /export/home file system resides. Get a tape that is appropriate for your system from the instructor.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, a tape should already be present in your tape drive. Do not eject the tape with an mt command, or you cannot continue with the lab.

Tasks

Complete the following tasks:

- Use the mt command to rewind the tape to the beginning.
- Use the ufsdump command to create a tape backup of the /export/home file system. Make sure that the /etc/dumpdates file is updated.
(Steps 1–4 in the Level 2 lab)
- Add files and directories to the /export/home file system.
(Steps 5–6 in the Level 2 lab)

Exercise: Backing Up a File System (Level 1)

- Use the `ufsdump` command to do an incremental backup of the `/export/home` file system.
(Steps 7–9 in the Level 2 lab)
- Use the `mt` command to remove the tape from the tape drive.
- Review the `/etc/dumpdates` file.
(Steps 10–12 in the Level 2 lab)

Exercise: Backing Up a File System (Level 2)

In this exercise, you back up an available file system on your system.

Preparation

This exercise requires a system that is configured with a tape drive and a file system that is available to unmount. This exercise assumes that the /export/home file system exists on a separate partition from the / (root) file system and can be unmounted. Identify the slice on which the /export/home file system resides. Get a tape that is appropriate for your system from the instructor.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, a tape should already be present in your tape drive. Do not eject the tape with an mt command, or you cannot continue with the lab.

Task Summary

In this exercise, you accomplish the following:

- Use the mt command to rewind the tape to the beginning.
- Use the ufsdump command to create a tape backup of the /export/home file system.
- Add files and directories to the /export/home file system.
- Use the ufsdump command to do an incremental backup of the /export/home file system.
- Use the mt command to remove the tape from the tape drive.
- Review the /etc/dumpdates file.

Tasks

Complete the following steps:

1. Unmount the /export/home file system. If your system reports that the /export/home file system is busy, use the `umount -f` command.
2. Insert a tape into your tape drive.
3. Use the `mt` command to rewind the tape to the beginning.
4. Use the `uudepm` command to create a backup for the /export/home file system. Make sure that the /etc/dumpdates file is updated.
5. Mount the /export/home file system.
6. Copy the contents of the /etc/uucp directory to the /export/home directory.
7. Unmount the /export/home file system.
8. Move the tape to the next tape record.
9. Use the `uudepm` command to create an incremental backup for the /export/home file system, using a non-rewinding device.
10. Rewind and eject the tape from the tape drive.
11. Set the tape aside for use with subsequent labs.
12. Review the contents of the /etc/dumpdates file.
13. Mount the /export/home file system.

Exercise: Backing Up a File System (Level 3)

In this exercise, you back up an available file system on your system.

Preparation

This exercise requires a system that is configured with a tape drive and file system that is available to unmount. This exercise assumes that the /export/home file system exists on a separate partition from the / (root) file system and can be unmounted. Identify the slice on which the /export/home file system resides. Get a tape that is appropriate for your system from the instructor.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, a tape should already be present in your tape drive. Do not eject the tape with an mt command, or you cannot continue with the lab.

Task Summary

In this exercise, you accomplish the following:

- Use the mt command to rewind the tape to the beginning.
- Use the ufsdump command to create a tape backup of the /export/home file system.
- Add files and directories to the /export/home file system.
- Use the ufsdump command to do an incremental backup of the /export/home file system.
- Use the mt command to remove the tape from the tape drive.
- Review the /etc/dumpdates file.

Tasks and Solutions

Complete the following steps:

1. Unmount the /export/home file system. If your system reports that the /export/home file system is busy, use the `umount -f` command.

```
# umount /export/home
```

2. Insert a tape into your tape drive.
3. Use the `mt` command to rewind the tape to the beginning.

```
# mt rewind
```

4. Use the `ufsdump` command to create a backup tape for the /export/home file system, where `c#t#d#s#` represents. If you cannot remember which device the /export/home file system was mounted on, view the contents of the /etc/vfstab file with the `more` command.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsck/c#t#d#s#
```

You should see output similar to:

```
ufsdump 0uf /dev/rmt/0 /dev/rdsck/c0d0t0s7
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Mon Oct 11 12:30:44 2004
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsck/c0t0d0s7 (sys43:/export/home) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 1126 blocks (563KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 1086 blocks (543KB) on 1 volume at 1803 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Mon Oct 11 12:44:12 2004
```

5. Mount the /export/home file system.

```
# mount /export/home
```

6. Copy the contents of the /etc/uucp directory to the /export/home directory.

```
# cp -r /etc/uucp /export/home
```

7. Unmount the /export/home file system.

```
# umount /export/home
```

8. Move the tape to the next tape record.

```
# mt -f /dev/rmt/0n fsf 1
```

9. Use the `ufsdump` command to create an incremental backup for the `/export/home` file system, using a non-rewinding device.

```
# ufsdump 1uf /dev/rmt/0n /dev/rdsk/c#t#d#s#
```

You should see output similar to:

```
ufsdump 1uf /dev/rmt/0n /dev/rdsk/c0d0t0s7
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Mon Oct 11 13:13:03 2004
DUMP: Date of last level 0 dump: Mon Oct 11 12:30:44 2004
DUMP: Dumping /dev/rdsk/c0t0d0s7 (sys43:/export/home) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 320 blocks (160KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 318 blocks (159KB) on 1 volume at 691 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 1 dump on Mon Oct 11 13:22:36 2004
```

10. Rewind and eject the tape from the tape drive.

```
# mt -f /dev/rmt/0 offline
```

11. Set the tape aside for use with subsequent labs.
12. Review the contents of the `/etc/dumpdates` file.

```
# more /etc/dumpdates
```

You should see one line showing information for the Level 0 dump and another line for the Level 1 dump, for example:

<code>/dev/rdsk/c0t0d0s7</code>	0	Mon Oct 11 12:30:44 2004
<code>/dev/rdsk/c0t0d0s7</code>	1	Mon Oct 11 13:13:03 2004

13. Mount the `/export/home` file system.

```
# mount /export/home
```

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 15

Performing File System Restores

Objectives

Upon completion of this module, you should be able to restore ufs file systems.

The following course map in Figure 15-1 shows how this module fits into the current instructional goal.

Performing System Backups and Restores

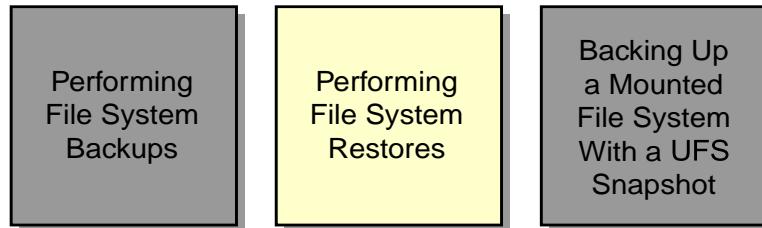


Figure 15-1 Course Map

Restoring a ufs File System

You restore a file system to rebuild a damaged file system, to reinstall or upgrade the Solaris OS software, or to reorganize file systems on existing or new disks.

Restoring a Regular File System

When you are restoring data to a system, consider the following questions:

- Can the system boot on its own (regular file system restore)?
- Do you need to boot the system from CD-ROM, DVD, or network (critical file system restore)?
- Do you need to boot the system from CD-ROM, DVD, or network and repair the boot drive (special case recovery)?

To restore files or file systems, determine the following:

- The file system backup tapes that are needed
- The device name to which you will restore the file system
- The name of the temporary directory to which you will restore individual files
- The type of backup device to be used (local or remote)
- The backup device name (local or remote)

To restore a regular file system, such as the /export/home or /opt file system, back up to the disk, you use the `ufsrestore` command. The `ufsrestore` command copies files to the disk, relative to the current working directory, from backup tapes that were created by the `ufsdump` command.

You can use the `ufsrestore` command to reload an entire file system hierarchy from a Level 0 backup and related incremental backups. You can also restore one or more single files from any backup tape.

The format for the `ufsrestore` command is:

`ufsrestore option(s) argument(s) filesystem`

Table 15-1 describes some options that you can use with the `ufsrestore` command.

Table 15-1 Options for the `ufsrestore` Command

Option	Description
<code>t</code>	Lists the table of contents of the backup media.
<code>r</code>	Restores the entire file system from the backup media.
<code>x file1 file2</code>	Restores only the files named on the command line.
<code>i</code>	Invokes an interactive restore.
<code>v</code>	Specifies verbose mode. This mode displays the path names to the terminal screen as each file is restored.
<code>f device</code>	Specifies the tape device name. When not specified, the <code>/dev/rmt/0</code> device file is used.

When you restore an entire file system from a backup tape, the system creates a `restoresymtable` file. The `ufsrestore` command uses the `restoresymtable` file for check-pointing or passing information between incremental restores. You can remove the `restoresymtable` file when the restore is complete.

Using the `ufsrestore` Command to Restore a Regular File System

The following procedure demonstrates how to use the `ufsrestore` command to restore the `/opt` file system on the `c0t0d0s5` slice.

1. Create the new file system structure.

```
# newfs /dev/rdsk/c0t0d0s5
# mount /dev/dsk/c0t0d0s5 /opt
# cd /opt
```

2. Mount the file system to the `/opt` directory, and change to that directory.

Restoring a `ufs` File System

3. Restore the entire `/opt` file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```



Note – Always restore a file system by starting with the Level 0 backup tape, continuing with the next-lower-level tape, and continuing through the highest-level tape.

4. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

5. Unmount the new file system.

```
# cd /
```

```
# umount /opt
```

6. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdsk/c0t0d0s5
```

7. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s5
```



Note – The system administrator should always back up the newly created file system because the `ufsrestore` command repositions the files and changes the inode allocation.

8. `init 6`

Restoring the `/usr` File System

To restore the `/usr` file system, boot from the Solaris 10 Software 1 CD-ROM or DVD, and then use the `ufsrestore` command to restore files back to the `/usr` partition.



Note – If the `/` (root), `/usr`, or `/var` file systems are unusable because of some type of corruption or damage, the system will not boot.

Using the `ufsrestore` Command to Restore a Critical File System

The following procedure demonstrates how to restore the `/usr` file system on Slice 6 of the boot disk.

1. Insert the Solaris 10 Software 1 CD-ROM or DVD, and boot from it with the single-user mode option.

```
ok boot cdrom -s
```

2. Create the new file system structure.

```
# newfs /dev/rdsck/c0t0d0s6
```

3. Mount the file system to the mount point `/a`, and change to that directory.

```
# mount /dev/dsk/c0t0d0s6 /a
```

```
# cd /a
```

4. Restore the entire `/usr` file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```



Note – Remember to restore a file system by starting with the Level 0 backup tape, continuing with the next-lower-level tape, and continuing through the highest-level tape.

5. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

6. Unmount the new file system.

```
# cd /
```

```
# umount /a
```

7. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdsck/c0t0d0s6
```

8. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsck/c0t0d0s6
```

9. Reboot the system.

```
# init 6
```

Performing a Special Case Recovery of the / (root) File System

You perform a special case recovery to recover the / (root) file system if there is damage to the boot block.

To restore the / (root) file system, boot from the Solaris 10 Software 1 CD-ROM or DVD, and use the **ufsrestore** command.

The following procedure demonstrates how to restore the / (root) file system on Slice 0 of the boot disk.

1. Insert the Solaris 10 Software 1 CD-ROM or DVD, and boot from it with the single-user mode option.

```
ok boot cdrom -s
```

2. Create the new file system structure.

```
# newfs /dev/rdsk/c0t0d0s0
```

3. Mount the file system to the mount point /a and change to that directory.

```
# mount /dev/dsk/c0t0d0s0 /a
```

```
# cd /a
```

4. Restore the / (root) file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```

 **Note** – Always restore a file system by starting with the Level 0 backup tape, and continuing with the next-lower-level tape, and continuing through the highest-level tape.

5. Remove the **restoresymtable** file.

```
# rm restoresymtable
```

6. Install the **bootblk** in Sectors 1 through 15 of the boot disk. To do this, change to the directory that contains the **bootblk**, and enter the **installboot** command.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs
```

```
# installboot bootblk /dev/rdsk/c0t0d0s0
```

7. Unmount the new file system.

```
# cd /
```

```
# umount /a
```

8. Use the fsck command to check the restored file system.

```
# fsck /dev/rdsck/c0t0d0s0
```

9. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsck/c0t0d0s0
```

10. Reboot the system.

```
# init 6
```

Invoking an Interactive Restore

The ufsrestore i command invokes an interactive interface. Through the interface, you can browse the directory hierarchy of the backup tape and select individual files to extract. The term volume is used by ufsrestore and should be considered a single tape.

Using the ufsrestore i Command

The following procedure demonstrates how to use the ufsrestore i command to extract individual files from a backup tape.

1. Become the root user, and change to the temporary directory that you want to receive the extracted files.

```
# cd /export/home/tmp
```

2. Perform the ufsrestore i command.

```
# ufsrestore ivf /dev/rmt/0
```

Verify volume and initialize maps

Media block size is 64

Dump date: Mon Oct 11 12:30:44 2004

Dumped from: the epoch

Level 0 dump of /export/home on sys43:/dev/dsk/c0t0d0s7

Label: none

Extract directories from tape

Initialize symbol table.

3. Display the contents of the directory structure on the backup tape.

```
ufsrestore > ls
```

```
.:
```

2 *./	13 directory1	15 directory3	11 file2
2 *.../	14 directory2	10 file1	12 file3

4. Change to the target directory on the backup tape.

```
ufsrestore > cd directory1  
ufsrestore > ls  
.directory1:  
3904 ./ 2 *../ 3905 file1 3906 file2 3907 file3
```

5. Add the files you want to restore to the extraction list.

```
ufsrestore > add file1 file2  
Make node ./directory1
```

Files you want to restore are marked with an asterisk (*) for extraction. If you extract a directory, all of the directory contents are marked for extraction.

In this example, two files are marked for extraction. The **ls** command displays an asterisk in front of the selected file names, **file1** and **file2**.

```
ufsrestore > ls  
.directory1:  
3904 *./ 2 *../ 3905 *file1 3906 *file2 3907 file3
```

6. To delete a file from the extraction list, use the **delete** command.

```
ufsrestore > delete file1
```

The **ls** command displays the **file1** file without an asterisk.

```
ufsrestore > ls  
.directory1:  
3904 *./ 2 *../ 3905 file1 3906 *file2 3907 file3
```

7. To view the files and directories marked for extraction, use the **marked** command.

```
ufsrestore > marked  
.directory1:  
3904 *./ 2 *../ 3906 *file2
```

8. To restore the selected files from the backup tape, perform the **extract** command:

```
ufsrestore > extract  
Extract requested files  
You have not read any volumes yet.  
Unless you know which volume your file(s) are on you should start  
with the last volume and work towards the first.  
Specify next volume #: 1
```



Note – The `ufsrestore` command has to find the selected files. If you used more than one tape for the backup, first insert the tape with the highest volume number and type the appropriate number at this point. Then repeat, working towards Volume #1 until all files have been restored.

```
extract file ./directory1/file2
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
```



Note – Answering `y` sets ownership and permissions of the temporary directory to those of the mount point on the tape.

9. To exit the interactive restore after the files are extracted, perform the command:

```
ufsrestore> quit
```

10. Move the restored files to their original or permanent directory location, and delete the files from the temporary directory.

```
# mv /export/home/tmp/directory1/file2 /export/home
# rm -r /export/home/tmp/directory1
```



Note – You can use the `help` command in an interactive restore to display a list of available commands.

Performing an Incremental Restore

When performing incremental restores, start with the last volume and work towards the first. The system uses information in the `restoresymtable` file to restore incremental backups on top of the latest full backup.

The following procedure demonstrates how to restore the `/export/home` file system from incremental tapes.



Note – This procedure makes use of the interactive restore to assist in showing the concept of incremental restores. You would typically use a command, such as `ufsrestore rf`, for restoring entire file systems.

1. View the contents of the `/etc/dumpdates` file for information about the `/export/home` file system.

```
# more /etc/dumpdates |grep c0t0d0s7
/dev/rdsk/c0t0d0s7      0 Wed Apr  7 09:55:34 2004
/dev/rdsk/c0t0d0s7      1 Web Apr  7 09:57:30 2004
```

2. Create the new file system structure for the `/export/home` file system.

```
# newfs /dev/rdsk/c0t0d0s7
```

3. Mount the file system and change to that directory.

```
# mount /dev/dsk/c0t0d0s7 /export/home
```

```
# cd /export/home
```

4. Insert the Level 0 backup tape.

5. Restore the `/export/home` file system from the backup tapes.

```
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump date: Wed Apr  7 09:55:34 2004
Dumped from: the epoch
Level 0 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Make node ./directory1
Make node ./directory2
Make node ./directory3
Extract new leaves.
Check pointing the restore
extract file ./file1
extract file ./file2
extract file ./file3
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
#
```

6. Load the next lower-level tape into the tape drive.

```
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump date: Wed Apr 07 09:57:30 2004
Dumped from: Wed Apr 07 09:55:34 2004
Level 1 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Begin incremental restore
Initialize symbol table.
Extract directories from tape
Mark entries to be removed.
Calculate node updates.
Make node ./directory4
Make node ./directory5
Make node ./directory6
Find unreferenced names.
Remove old nodes (directories).
Extract new leaves.
Check pointing the restore
extract file ./file4
extract file ./file5
extract file ./file6
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
#
```

Alternative Steps

The following steps are an alternative to the previous Steps 5 and 6.

5. Restore the /export/home file system from the backup tapes. (This example uses an interactive, verbose restore to provide more detailed information.)

```
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump date: Mon Oct 11 13:10:12 2004
Dumped from: the epoch
Level 0 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
2 *./          8 directory2      5 file2
2 *../          9 directory3      6 file3
7 directory1    4 file1          3 lost+found/
```

The system lists files from the last Level 0 backup.

```
ufsrestore > add *
Warning: ./lost+found: File exists
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./file1
extract file ./file2
extract file ./file3
extract file ./directory1
extract file ./directory2
extract file ./directory3
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
Directories already exist, set modes anyway? [yn] n
ufsrestore > q
#
```

6. The information in the /etc/dumpdates file shows an incremental backup that was taken after the Level 0 backup. Load the next tape and perform the incremental restore.

```
# ufsrestore iv
Verify volume and initialize maps
Media block size is 64
Dump date: Wed Apr 07 09:57:30 2004
Dumped from: Wed Apr 07 09:55:34 2004
Level 1 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
 2 *./          13  directory4      15  directory6      11  file5
 2 *.../         14  directory5      10  file4        12  file6
ufsrestore > add *
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./file4
extract file ./file5
extract file ./file6
extract file ./directory4
extract file ./directory5
extract file ./directory6
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
ufsrestore > q
#
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Recovering Backup Files and File Systems (Level 1)

In this exercise, you read the backup tape from the previous exercise. You back up the / (root) file system, restore a single file from tape, and destroy and restore the / (root) file system.

Preparation

This exercise requires a system that is configured with a tape drive and a / (root) file system that is separate from the /usr and /var file systems. Identify the slice that holds the / (root) file system. From your instructor, get a tape appropriate for your system.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, a tape should already be present in your tape drive. Do not eject the tape with an mt command, or you cannot continue with the lab.

Tasks

Complete the following:

- Read the contents of both ufsdump files on the backup tape written in the previous exercise.
(Steps 1–3 in Task 1 of the Level 2 lab)
- Reboot the system to run level S. Use the ufsdump command to create a backup tape of the / (root) file system on your system. Verify that the tape contains valid data for this file system. Allow the system to continue to boot to run level 3.
(Steps 1–5 in Task 2 of the Level 2 lab)

Exercise: Recovering Backup Files and File Systems (Level 1)

- Use the `ufsrestore i` command to restore the `/etc/inet/hosts` file from tape, and place it below the `/var/tmp` directory.
(Steps 1–6 in Task 3 of the Level 2 lab)
- Remove the `/kernel`, `/platform`, and `/devices` directories recursively. Abort the operating system, and attempt to boot the system from disk. Record what happens. Boot the system from the Solaris 10 Software 1 of 4 CD-ROM or DVD to run level S. Create a new file system on the `/` (root) slice. Use the `ufsrestore` command to reload the `/` (root) file system. Install a new boot block. Reboot the system, and eject the CD-ROM/DVD.
(Steps 1–11 in Task 4 of the Level 2 lab)

Exercise: Recovering Backup Files and File Systems (Level 2)

In this exercise, you read the backup tape from the previous exercise. You back up the / (root) file system, restore a single file from tape, and destroy and restore the / (root) file system.

Preparation

This exercise requires a system that is configured with a tape drive and a / (root) file system that is separate from the /usr and /var file systems. Identify the slice that holds the / (root) file system. From the instructor, get a tape appropriate for your system.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, a tape should already be present in your tape drive. Do not eject the tape with an mt command, or you cannot continue with the lab.

Task Summary

In this exercise, you accomplish the following:

- Read the contents of both ufsdump files on the backup tape written in the previous exercise.
- Reboot the system to run level S. Use the ufsdump command to create a backup tape of the / (root) file system on your system. Verify that the tape contains valid data for this file system. Allow the system to continue to boot to run level 3.

Exercise: Recovering Backup Files and File Systems (Level 2)

- Use the `ufsrestore i` command to restore the `/etc/inet/hosts` file from tape, and place it below the `/var/tmp` directory.
- Remove the `/kernel`, `/platform`, and `/devices` directories recursively. Abort the operating system, and attempt to boot the system from disk. Record what happens. Boot the system from the Solaris 10 OS Software 1 of 4 CD-ROM or DVD to run level S. Create a new file system on the `/` (root) slice. Use the `ufsrestore` command to reload the `/` (root) file system. Install a new boot block. Reboot the system, and eject the CD-ROM/DVD.

Tasks

Complete the following tasks.

Task 1 – Read Your Previous Backup Tape

Complete the following steps:

1. Locate the backup tape written in the previous exercise, and load it into your tape drive.
2. Use the interactive restore command to view the contents of the first Level 0 backup tape. Verify that the files are from the /export/home directory that you backed up. Enter **q** to quit the interactive restore.
3. Using a non-rewind device, move the tape to the next record, and view the contents of the second, incremental backup. Verify that the files you see are from the incremental backup. (The uucp directory is the one you added after the Level 0 backup.)

Task 2 – Create a Backup of the / (root) File System

Complete the following steps:

1. Log in as the root user, and open a terminal window. Shut down the system to run level 0. Then, boot the system to run level S. Supply the root password as required to enter run level S.
2. Verify that a tape is in your tape drive.
3. Use the **ufsdump** command to create a backup tape for the / (root) file system.
4. Verify that the / (root) file system is on the tape.
5. Allow the system to continue to boot to run level 3.

Task 3 – Restore the /etc/inet/hosts File From a Tape

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change to the /var/tmp directory.
2. Enter the **ufsrestore i** command to retrieve the /etc/inet/hosts file from the tape.
3. Change to the /etc/inet directory on the tape, and list the files in the directory.

4. Add the hosts file to the list of files to extract, and display the list.
5. Extract the hosts file from tape. Specify volume number 1. Do not set the owner and mode for ., and then quit the ufsrestore command.
6. Verify that the etc/inet/hosts file exists below the /var/tmp directory.

Task 4 – Destroy and Restore the / (root) File System

Complete the following steps:

1. Change to the / (root) directory, and remove the following critical system directories and their contents: /kernel, /platform, and /devices.
2. Press the Stop-A key sequence to abort the operating system. Attempt to boot the system from the boot disk.
What happens?
3. Insert the Solaris 10 Software 1 of 4 CD-ROM or DVD. Boot the system from the CD-ROM/DVD to run level S.
4. Use the newfs command to create a new file system on the / (root) slice. (The slice should match the one you used earlier in the exercise when you created a backup of the / (root) file system.) Run the fsck command on the file system that you create.
5. Verify that your root backup tape is in the tape drive. Mount the new file system as the /a file system. Change to the /a directory.
6. Use the ufsrestore command to load the / (root) data into the new file system.
7. Remove the restoresymtable file.
8. Install a new boot block in Sectors 1 through 15 of the / (root) slice, by changing to the directory containing the boot block and entering the installboot command.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs  
# installboot bootblk /dev/rdsck/c0t0d0s0
```

9. Change to the / (root) directory, and unmount the new file system.
10. Reboot the system.
11. Log in as the root user, and open a terminal window. Eject the Solaris 10 Software 1 of 4 CD-ROM or DVD.

Exercise: Recovering Backup Files and File Systems (Level 3)

In this exercise, you read the backup tape from the previous exercise. You back up the / (root) file system, restore a single file from tape, and destroy and restore the / (root) file system.

Preparation

This exercise requires a system that is configured with a tape drive and a / (root) file system that is separate from the /usr and /var file systems. Identify the slice that holds the / (root) file system. From the instructor get a tape appropriate for your system.

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

In the RLDC environment, a tape should already be present in your tape drive. Do not eject the tape with an mt command, or you cannot continue with the lab.

Task Summary

In this exercise, you accomplish the following:

- Read the contents of both ufsdump files on the backup tape written in the previous exercise.
- Reboot the system to run level S. Use the ufsdump command to create a backup tape of the / (root) file system on your system. Verify that the tape contains valid data for this file system. Allow the system to continue to boot to run level 3.

- Use the `ufsrestore i` command to restore the `/etc/inet/hosts` file from tape, and place it below the `/var/tmp` directory.
- Remove the `/kernel`, `/platform`, and `/devices` directories recursively. Abort the operating system, and attempt to boot the system from disk. Record what happens. Boot the system from the Solaris 10 Software 1 of 4 CD-ROM or DVD to run level S. Create a new file system on the `/` (root) slice. Use the `ufsrestore` command to reload the `/` (root) file system. Install a new boot block. Reboot the system, and eject the CD-ROM or DVD.

Tasks and Solutions

Complete the following tasks.

Task 1 – Read Your Previous Backup Tape

Complete the following steps:

1. Locate the backup tape written in the previous exercise, and load it into your tape drive.
2. Use the interactive restore command to view the contents of the first Level 0 backup tape. Verify that the files are from the `/export/home` directory that you backed up. Enter `q` to quit the interactive restore.

```
# ufsrestore iv
Verify volume and initialize maps
Media block size is 64
Dump date: Mon Oct 11 12:30:44
Dumped from: the epoch
Level 0 dump of /export/home on sys43:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
2 *./          3712 default/          6  file3
2 *../          4   file1           3  lost+found/
7  core          5   file2
```

You should see the files from your /export/home directory.

```
ufsrestore > quit
```

3. Using a non-rewind device, move the tape to the next record, and view the contents of the second, incremental backup. Verify that the files you see are from the incremental backup. (The uucp directory is the one you added after the Level 0 backup.) Quit the interactive restore.

```
# mt -f /dev/rmt/0n fsf 1
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump date: Wed Apr 07 09:57:30 2004
Dumped from: Wed Apr 07 09:55:34 2004
Level 1 dump of /export/home on sys43:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
2 *./          2 *../      7424  uucp/
ufsrestore > q
#
```

Task 2 – Create a Backup of the / (root) File System

Complete the following steps:

1. Log in as the root user, and open a terminal window. Shut down the system to run level 0. Then boot the system to run level S. Supply the root password as required to enter run level S.

```
# init 0
(shutdown messages)
ok boot -s
```

2. Verify that a tape is in your tape drive.
3. Use the ufsdump command to create a backup tape for the / (root) file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsck/c0t0d0s0
```

4. Verify that the / (root) file system is on the tape.

```
# ufsrestore tvf /dev/rmt/0
```

The screen should scroll directory structures under / (root) first, followed by files.

5. Allow the system to continue to boot to run level 3.

```
# <Control-D>
```

Task 3 – Restore the /etc/inet/hosts File From a Tape

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change to the /var/tmp directory.

```
# cd /var/tmp
```

2. Enter the ufsrestore i command to retrieve the /etc/inet/hosts file from the tape.

```
# ufsrestore if /dev/rmt/0
```

```
ufsrestore > ls
```

You should see files and directories for the / (root) file system.

3. Change to the /etc/inet directory on the tape, and list the files in the directory.

```
ufsrestore > cd /etc/inet
```

```
ufsrestore > ls
```

You should see files and directories for the /etc/inet file system.

4. Add the hosts file to the list of files to extract, and display the list.

```
ufsrestore > add hosts
```

```
ufsrestore > marked
```

You should see the hosts file listed.

5. Extract the hosts file from tape. Specify volume number 1. Do not set the owner and mode for ., and then quit the ufsrestore command.

```
ufsrestore > extract
```

```
Extract requested files
```

```
You have not read any volumes yet.
```

```
Unless you know which volume your file(s) are on you should start with the last volume and work towards the first.
```

```
Specify next volume #: 1
```

```
set owner/mode for '.'? [yn] n
```

```
ufsrestore > q
```

6. Verify that the etc/inet/hosts file exists below the /var/tmp directory.

```
# ls etc/inet/hosts
```

```
etc/inet/hosts
```

Task 4 – Destroy and Restore the / (root) File System

Complete the following steps:

1. Change to the / (root) directory, and remove the following critical system directories and their contents: /kernel, /platform, and /devices.

```
# cd /
# rm -r /kernel /platform /devices
```

2. Press the Stop-A key sequence to abort the operating system. Attempt to boot the system from the boot disk.

ok boot

What happens?

The system fails to boot and displays the message:

Boot load failed.

The file just loaded does not appear to be executable

3. Insert the Solaris 10 Software 1 CD-ROM or DVD. Boot the system from the CD-ROM or DVD to run level S.

ok boot cdrom -s

4. Run the newfs command to create a new file system on the / (root) slice. (The slice should match the one you used earlier in the exercise when you created a backup of the / (root) file system.) Enter the fsck command on the file system that you create.

```
# newfs /dev/rdsk/c0t0d0s0
# fsck /dev/rdsk/c0t0d0s0
```

5. Verify that your root backup tape is in the tape drive. Mount the new file system as the /a file system. Change to the /a directory.

```
# mount /dev/dsk/c0t0d0s0 /a
# cd /a
```

6. Use the ufsrestore command to load the / (root) data into the new file system.

```
# ufsrestore rf /dev/rmt/0
```

7. Remove the restoresymtable file.

```
# rm restoresymtable
```

8. Install a new boot block in Sectors 1 through 15 of the / (root) slice, by changing to the directory containing the boot block and entering the installboot command.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs
# installboot bootblk /dev/rdsk/c0t0d0s0
```

Exercise: Recovering Backup Files and File Systems (Level 3)

9. Change to the / (root) directory, and unmount the new file system.

```
# cd /
# umount /a
```

10. Reboot the system.

```
# init 6
```

11. Log in as the root user, and open a terminal window. Eject the Solaris 10 Software 1 CD-ROM or DVD.

```
# eject cdrom
```

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 16

Backing Up a Mounted File System With a UFS Snapshot

Objectives

Upon completion of this module, you should be able to:

- Create a UFS snapshot
- Back up the snapshot file

The following course map in Figure 16-1 shows how this module fits into the current instructional goal.

Performing System Backups and Restores

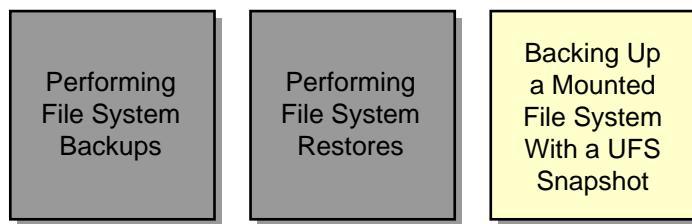


Figure 16-1 Course Map

Creating a UFS Snapshot

The UFS Copy on Write Snapshots feature provides administrators an online backup solution for ufs file systems. This utility enables you to use a point-in-time copy of a ufs file system, called a snapshot, to create an online backup. You can create the backup while the file system is mounted and the system is in multiuser mode.

 **Note** – The UFS snapshots are similar to the Sun StorEdge™ Instant Image product. Instant Image allocates space equal to the size of the entire file system that is being captured. However, the file system data saved by UFS snapshots occupies only as much disk space as needed.

Using the fssnap Command

You use the fssnap command to create, query, or delete temporary read-only snapshots of ufs file systems.

The format for the fssnap command is:

```
/usr/sbin/fssnap -F FSType -V -o special_option(s) mount-point | special
```

Table 16-1 shows some of the options for the fssnap command.

Table 16-1 Options for the fssnap Command

Option	Description
-d	Deletes the snapshot associated with the given file system. If the -o <i>unlink</i> option was used when you built the snapshot, the backing-store file is deleted together with the snapshot. Otherwise, the backing-store file (which contains file system data) occupies disk space until you delete it manually.
-F <i>FSType</i>	Specifies the file system type to be used.
-i	Displays the state of an <i>FSType</i> snapshot.
-V	Echoes the complete command line but does not execute the command.
-o	Enables you to use <i>special_options</i> , such as the location and size of the backing-store (bs) file.

To create a UFS snapshot, specify a backing-store path and the actual file system to be captured. The following is the command format:

```
# fssnap -F ufs -o bs=backing_store_path /file-system
```



Note – The *backing_store_path* can be a raw device, the name of an existing directory, or the name of a file that does not already exist.

The following example uses the fssnap command to create a snapshot of the /export/home file system.

```
# fssnap -F ufs -o bs=/var/tmp /export/home
/dev/fssnap/0
```

The snapshot subsystem saves file system data in a file called a backing-store file before the data is overwritten. Some important aspects of a backing-store file are:

- A backing-store file is a bit-mapped file that takes up disk space until you delete the UFS snapshot.
- The size of the backing-store file varies with the amount of activity on the file system being captured.
- The destination path that you specify on the fssnap command line must have enough free space to hold the backing-store file.
- The location of the backing-store file must be different from that of the file system you want to capture in a UFS snapshot.
- A backing-store file can reside on different types of file systems, including another ufs file system or a mounted nfs file system.

The fssnap command creates the backing-store file and two read-only virtual devices. The block virtual device, /dev/fssnap/0, can be mounted as a read-only file system. The raw virtual device, /dev/rfssnap/0, can be used for raw read-only access to a file system.

These virtual devices can be backed up with any of the existing Solaris OS backup commands. The backup created from a virtual device is a backup of the original file system when the UFS snapshot was taken.



Note – When a UFS snapshot is first created, the file system locks temporarily. Users might notice a slight pause when writing to this file system. The length of the pause increases with the size of the file system. There is no performance impact when users are reading from the file system.

Limiting the Size of the Backing-Store File

Before creating a UFS snapshot, use the `df -k` command to check that the backing-store file has enough disk space to grow. The size of the backing-store file depends on how much data has changed since the previous snapshot was taken.

You can limit the size of the backing-store file by using the `-o maxsize=n` option of the `fssnap` command, where `n` (`k`, `m`, or `g`) is the maximum size of the backing-store file specified in Kbytes, Mbytes, or Gbytes.



Caution – If the backing-store file runs out of disk space, the system automatically deletes the UFS snapshot, which causes the backup to fail. The active `ufs` file system is not affected. Check the `/var/adm/messages` file for possible UFS snapshot errors.



Note – You can force an unmount of an active `ufs` file system, for which a snapshot exists (for example, with the `umount -f` command). This action deletes the appropriate snapshot automatically.

The following example creates a snapshot of the `/export/home` file system, and limits the backing-store file to 500 Mbytes.

```
# fssnap -F ufs -o bs=/var/tmp,maxsize=500m /export/home  
/dev/fssnap/0
```

Displaying Information for a ufs File System Snapshot

You can use either `fssnap` command to display UFS snapshot information.

The following example displays a list of all the current UFS snapshots on the system. The list also displays the corresponding virtual device for each snapshot.

```
# fssnap -i  
0   /export/home  
1   /usr  
2   /database
```

You use the `-i` option to the `/usr/lib/fs/ufs/fssnap` command to display detailed information for a specific UFS snapshot that was created by the `fssnap` command.

The following example shows the details for the `/export/home` snapshot.

```
# /usr/lib/fs/ufs/fssnap -i /export/home  
Snapshot number          : 0  
Block Device             : /dev/fssnap/0  
Raw Device               : /dev/rfssnap/0  
Mount point              : /export/home  
Device state             : idle  
Backing store path       : /var/tmp/snapshot0  
Backing store size       : 0 KB  
Maximum backing store size : 512000 KB  
Snapshot create time     : Mon Oct 11 08:58:33 2004  
Copy-on-write granularity : 32 KB
```

Backing Up the UFS Snapshot File

The virtual devices that contain the UFS snapshot act as standard read-only devices, which enable you to back up the virtual device in the same manner as you would back up a file system.

Performing a Backup of a UFS Snapshot

You can use the `tar` command or the `ufsdump` command to back up a UFS snapshot.

Using the `tar` Command to Back Up a Snapshot File

If you use the `tar` command to back up the UFS snapshot, mount the snapshot before backing it up. The following procedure demonstrates how to do this type of mount.

1. Create the mount point for the block virtual device.

```
# mkdir -p /backups/home.bkup
```

2. Mount the block virtual device to the mount point.

```
# mount -F ufs -o ro /dev/fssnap/0 /backups/home.bkup
```

3. Change directory to the mount point.

```
# cd /backups/home.bkup
```

4. Use the `tar` command to write the data to tape.

```
# tar cvf /dev/rmt/0 .
```

Using the `ufsdump` Command

If you use the `ufsdump` command to back up a UFS snapshot, you can specify the raw virtual device during the backup.

```
# ufsdump 0uf /dev/rmt/0 /dev/rfssnap/0
```

Verify that the UFS snapshot is backed up.

```
# ufsrestore tf /dev/rmt/0
```

Performing an Incremental Backup Using a UFS Snapshot

Incremental backups of snapshots contain files that have been modified since the last UFS snapshot. You use the `ufsdump` command with the `N` option to create an incremental UFS snapshot, which writes the name of the device being backed up, rather than the name of the snapshot device to the `/etc/dumpdates` file.

The following example shows how to use the `ufsdump` command to create an incremental backup of a file system.



Note – It is important to note the use of the `N` argument when backing up a snapshot. This argument ensures proper updates to the `/etc/dumpdates` file.

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdsck/c1t0d0s0 /dev/rfssnap/0
```

Next you would verify that the UFS snapshot is backed up to tape.

```
# ufsrestore tf /dev/rmt/0
```

To understand incremental backups of snapshots, consider the following demonstration:

1. Create a snapshot of the `/extra` file system that is going to be backed up while the file system is mounted.

```
# fssnap -o bs=/var/tmp /extra
/dev/fssnap/0
#
```

2. Verify that the snapshot was successful, and view detailed information about the snapshot.

```
# fssnap -i
0      /extra
# /usr/lib/fs/ufs/fssnap -i /extra
Snapshot number          : 0
Block Device              : /dev/fssnap/0
Raw Device                : /dev/rfssnap/0
Mount point               : /extra
Device state              : idle
Backing store path        : /var/tmp/snapshot0
Backing store size        : 0 KB
Maximum backing store size : Unlimited
Snapshot create time       : Mon Oct 11 10:34:21 2004
Copy-on-write granularity : 32 KB
```

Backing Up the UFS Snapshot File

3. Make a directory that will be used to mount and view the snapshot data.

```
# mkdir /extrasnap  
#
```

4. Mount the snapshot to the new mount point, and compare the size of the file system and the snapshot device.

```
# mount -o ro /dev/fssnap/0 /extrasnap  
# df -k |grep extra  
/dev/dsk/c1t0d0s0 1294023 9 1242254 1% /extra  
/dev/fssnap/0 1294023 9 1242254 1% /extrasnap
```

5. Edit a file under the /extra directory and make it larger, and then compare the size of the file system and the snapshot device.

```
# vi file1  
(yank and put text, or read text in from another file)
```

```
# df -k |grep extra  
/dev/dsk/c1t0d0s0 1294023 20 1242243 1% /extra  
/dev/fssnap/0 1294023 9 1242254 1% /extrasnap
```

Observe that the file system grew in size while the snapshot file did not.

6. Perform a full backup with the N option of the ufsdump command.

```
# ufsdump 0ufN /dev/rmt/0 /dev/rdsck/c1t0d0s0 /dev/rfssnap/0  
DUMP: Writing 32 Kilobyte records  
DUMP: Date of this level 0 dump: Mon Oct 11 10:49:38 2004  
DUMP: Date of last level 0 dump: the epoch  
DUMP: Dumping /dev/rfssnap/0 (sys41:/extrasnap) to /dev/rmt/0.  
DUMP: Mapping (Pass I) [regular files]  
DUMP: Mapping (Pass II) [directories]  
DUMP: Estimated 262 blocks (131KB).  
DUMP: Dumping (Pass III) [directories]  
DUMP: Dumping (Pass IV) [regular files]  
DUMP: Tape rewinding  
DUMP: 254 blocks (127KB) on 1 volume at 1814 KB/sec  
DUMP: DUMP IS DONE  
DUMP: Level 0 dump on Mon Oct 11 11:03:46 2004
```

7. Verify the backup.

```
# ufsrestore tf /dev/rmt/0  
2 .  
3 ./file1  
4 ./file2  
5 ./file3  
6 ./file4  
#
```

8. Unmount the back up device and remove the snapshot.

```
# umount /extrasnap
# fssnap -d /extra
# rm /var/tmp/snapshot0
#
```

9. Make some changes to the /extra file system, such as copying some files, and then re-create the snapshot.

```
# cp file1 file5
# cp file1 file6
# fssnap -o bs=/var/tmp /extra
/dev/fssnap/0
#
```

10. Re-mount the snapshot device, and compare the size of the file system and the snapshot device.

```
# mount -o ro /dev/fssnap/0 /extrasnap
# df -k |grep extra
/dev/dsk/c1t0d0s0    1294023      46 1242217      1%      /extra
/dev/fssnap/0          1294023      46 1242217      1%      /extrasnap
#
```

11. Perform an incremental backup with the N option of the ufsdump command.

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdsck/c1t0d0s0 /dev/rfssnap/0
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Mon Oct 11 13:13:03 2004
DUMP: Date of last level 0 dump: Mon Oct 11 12:30:44 2004
DUMP: Dumping /dev/rfssnap/0 (sys41:/extrasnap) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 294 blocks (147KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 254 blocks (127KB) on 1 volume at 1693 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 1 dump on Mon Oct 11 13:22:36 2004
#
```

12. Verify the backup.

```
# ufsrestore tf /dev/rmt/0
2      .
7      ./file5
8      ./file6
#
```

Notice that the backup of the snapshot contains only the files that were added since the previous Level 0 backup.

Restoring Data From a UFS Snapshot Backup

The backup created from a virtual device is a backup of the original file system when the UFS snapshot was taken.

You restore a UFS snapshot from a backup tape in the same manner as you would the backup of an original file system. Data written to a tape by `ufsdump` is simply data, whether it is a snapshot or a file system.

To restore the `demo` directory from the snapshot backup of the `/usr` file system, complete the following steps:

1. Load the tape that contains the snapshot backup of the `/usr` file system into the tape drive.
2. Change to the `/usr` file system.

```
# cd /usr  
3. Perform the a ufsrestore command.
```

```
# ufsrestore if /dev/rmt/0  
ufsrestore > add demo  
ufsrestore > extract  
Specify next volume #: 1  
set owner/mode for '.'? [yn] n  
ufsrestore > quit
```

4. Verify that the `demo` directory exists, and eject the tape.

Deleting a UFS Snapshot

Deleting a UFS snapshot from the system is a multistep process and order-dependant. First, unmount the snapshot device, and then delete the snapshot. Finally, remove the backing-store file.

```
# umount /dev/fssnap/0  
# fssnap -d /export/home  
# rm /backing_store_file
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Working With UFS Snapshots (Level 1)

In this exercise, you create a UFS snapshot of the /opt file system, display detailed information for the UFS snapshot, and then remove the snapshot and backing-store file.

Tasks

Complete the following tasks:

- Create a snapshot of the /opt file system
- View the contents of the backing-store directory
- Display detailed information about the snapshot
- Remove the snapshot from the system

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Exercise: Working With UFS Snapshots (Level 2)

In this exercise, you create a UFS snapshot of the /opt file system, display detailed information for the UFS snapshot, and then remove the snapshot and backing-store file.

Task Summary

In this exercise, you accomplish the following:

- Create a snapshot of the /opt file system
- View the contents of the backing-store directory
- Display detailed information about the snapshot
- Remove the snapshot from the system

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks

Complete the following steps:

1. Create a snapshot of the /opt file system without specifying a file name for the backing-store file.
2. View the contents of the /var/tmp file system.
What is the default name assigned to a backing-store file?
3. Display the detailed information about the snapshot.
What is the maximum backing-store file size for the snapshot?

Exercise: Working With UFS Snapshots (Level 2)

4. Delete the snapshot from the system.
5. View the contents of the /var/tmp file system.
Has the backing-store file been removed?
6. Remove the backing-store file that you created in Step 1.

Exercise: Working With UFS Snapshots (Level 3)

In this exercise, you create a UFS snapshot of the /opt file system, display detailed information for the UFS snapshot, and then remove the snapshot and backing-store file.

Task Summary

In this exercise, you accomplish the following:

- Create a snapshot of the /opt file system
- View the contents of the backing-store directory
- Display detailed information about the snapshot
- Remove the snapshot from the system

Remote Lab Data Center (RLDC)

In addition to being able to use local classroom equipment, this lab has also been designed to use equipment located in a remote lab data center. Directions for accessing and using this resource can be found at:

<http://fn1.brom.suned.com/>

Ask your instructor for the particular SSH (Secure Shell) configuration file you should use to access the appropriate remote equipment for this exercise.

Tasks and Solutions

Complete the following steps:

1. Create a snapshot of the /opt file system without specifying a file name for the backing-store file.

```
# fssnap -F ufs -o bs=/var/tmp /opt
```

2. View the contents of the /var/tmp file system.

What is the default name assigned to a backing-store file?

snapshot0

Exercise: Working With UFS Snapshots (Level 3)

3. Display the detailed information about the snapshot.

```
# /usr/lib/fs/ufs/fssnap -i /opt
```

What is the maximum backing-store file size for the snapshot?

Unlimited

4. Delete the snapshot from the system.

```
# fssnap -d /opt
```

5. View the contents of the /var/tmp file system. Has the backing-store file been removed?

No

6. Remove the backing-store file you created in Step 1.

```
# rm /var/tmp/snapshot0
```

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Index

A

abort key sequence 8-8
accept command 12-35
accessing removable media 5-27
adding software packages
 command 6-9
adjusting a link counter 4-21
adm account 10-5
administering Volume
 Management 5-26
at command
 allowing access 13-17
 controlling access 13-16
 denying access 13-16
 executing 13-15
 overview 13-14
autoconfiguration 9-17
automatic execution of
 commands 13-17

B

backing-store file
 definition 16-3
 limiting size 16-4
backup
 before installation 1-13
 definition of 14-2
 frequency and levels 14-7
 full 14-7
 incremental 14-7
 information 14-9

level definitions 14-8
number of tapes 14-6
remote 14-13
restore file system 15-3
restoresymtable 15-2
restoring snapshot 16-10
scheduling 14-7
 strategies 14-6
backup superblock 4-7
banner command 8-12
bin account 10-5
blocks, cylinder group 4-8
boot
 block 4-7
 device 9-15
 disk device path name 8-28
 process 9-14, 9-22
 secondary program 9-16
boot -a command 9-21
boot command 3-17, 8-12
boot PROM
 boot sequence 9-15
 commands 8-11
 definition of 8-3
 overview 8-2
bootblk command 4-7
bootblk program 9-15
boot-device parameter 8-25
bus configuration 3-14
busy file system 5-19

C

CDE process manager 13-2

CD-ROM

 drive 5-24

CD-ROM drive

 location 5-24

change group command 11-40

change owner command 11-38

Changes 9-20

changing NVRAM parameters 8-16, 8-26

checking

 file systems 5-14

 software packages command 6-12

chgrp command 11-40

chown command 11-38

cluster configuration 1-6

command-line tools 10-14

commands

 .project 11-4

 /etc/dumpdates 14-9

 /etc/init.d/lp stop 12-33

 /usr/dt/bin/ sdtprocess 13-2

 /usr/sadm/admin/bin/

 printmgr 12-20

 accept 12-35

 at 13-14

 banner 8-12

 boot 3-17, 8-12

 boot -a 9-21

 boot PROM 8-11

 bootblk 4-7

 chgrp 11-40

 chown 11-38

 devalias 8-23

 devfsadm 3-18

 df 4-25

 disable 12-35

 du 4-28

 eprom 8-26

 enable 12-35

 finger 11-4

 fmthard 3-47

 format 3-16, 3-31

 fsck 4-17, 4-18, 4-19, 5-14

 fssnap 16-2

fssnap -i 16-5

fstyp 5-17

fuser 5-19

groupadd 10-23

groupdel 10-26

groups 11-36

help 8-14

id 11-37

init 9-45

kill 13-9

last 11-5

ln 2-15

lp 12-12

lpadmin 12-31

lpmove 12-35

lpr 12-12

ls 2-11

mkdir 2-14

mount 5-4, 5-11, 5-16, 5-27

mt 14-5

newfs 4-14, 5-15

newgrp 10-9

nvalias 8-24

nvunalias 8-25

patchadd 7-4, 7-9

patchrm 7-10

pkgadd 6-9

pkgchk 6-12

pkginfo 6-7

pkgrm 6-14

printenv 8-15

probe 8-17

probe-ide 8-19

probe-scsi 8-18

probe-scsi-all 8-19

prstat 13-4

prtconf 3-15

prtvtoc 3-46

quot 4-30

reject 12-35

rmmount 5-25

rusers 11-3

set-defaults 8-17

setenv 8-16

show-devs 8-22

show-disks 8-24

showrev 7-4
shutdown 9-45
smgroup add 10-23
smgroup delete 10-26
smgroup modify 10-25
smuser 10-14
smuser add 10-18
smuser delete 10-22
smuser modify 10-21
su 11-8, 11-11
tar 7-8, 16-6
touch 2-12
tunefs 4-16
ufsdump 14-10, 14-13, 16-6
ufsrestore i 15-7
umount 5-18
unmount -f 5-20
umountall 5-19
unzip 7-7
useradd 10-15
usermod 10-20
usfrestore 15-3
verify 3-45
volcheck 5-23
who 11-2
whoami 11-9
zcat 7-8

configuration
 /etc/system file 9-16
 kernel 9-18
 new network printer 12-22 to 12-29
 printer 12-30
 printer services 12-19
 removing printer 12-31
 Volume Management files 5-25

configuration files 5-25

consistency
 cylinder group block 4-18
 data block 4-18
 inode 4-18

CONSOLE variable 11-13, 11-15

controller number 3-8

corrupted file system 4-17

course goals Preface-xix

creating
 custom device aliases 8-26

new run control scripts 9-36
 ufs file systems 4-14

creating mount points 5-15

creating new file systems 5-15

cron daemon 13-17

crontab file
 accessing 13-21
 definition of 13-17
 editing 13-20
 format 13-17
 removing 13-20
 viewing contents 13-19

customize disk window 1-23

cylinder 3-4

cylinder group block consistency 4-18

cylinder group blocks 4-8

cylinder groups 4-7

D

daemon account 10-5
daemon, network listening service 12-11

daemons
 /usr/sbin/vold 5-23
 cron 13-17
 in.lpd 12-11
 inetd 12-10
 Internet services 12-10
 lpsched 12-15
 network server 11-3
 rpc.rusersd 11-3
 scheduler 12-12

data block
 definition of 4-11
 fragmentation 4-12

data block consistency 4-18

data blocks 2-10

data compression 14-5

data organization on disk platters 3-3

date mounted 5-8

destination printer default 12-31

destination printer, locating 12-13

dealias command 8-23

devfsadm command 3-18

device
 class 3-18

- loading drivers 3-18
- name 5-8
- naming conventions 3-10
- path names 8-22
- device aliases
 - creating 8-26
 - removing 8-25
- device configuration tree 3-11
- device driver 3-12
- device drivers directory 9-17
- device tree 8-20
- df command 4-25
- DHCP 1-18
- different types of file systems 5-16
- directories
 - / 2-3
 - / (root) 2-2
 - /bin 2-5
 - /dev 2-3, 2-4, 3-10
 - /dev/dsk 3-10
 - /dev/rdsk 3-10
 - /devices 2-5
 - /etc/lp/fd 12-7
 - /etc/init.d 9-29
 - /etc/lp 12-8
 - /etc/lp/fd 12-8
 - /etc/lp/interfaces 12-8
 - /etc/lp/printers 12-8
 - /etc/rc#.d 9-27
 - /export 2-3, 2-5
 - /home 2-3, 2-5
 - /kernel 2-3, 9-17
 - /kernel/drv 9-17
 - /mnt 2-3
 - /opt 2-3
 - /platform 2-3
 - /platform/`uname -i`/kernel 9-17
 - /platform/`uname -m`/kernel 9-17
 - /sbin 2-3, 5-4
 - /sur/share/lib/terminfo 12-6
 - /tmp 2-5, 5-8
 - /usr 2-3, 2-5
 - /usr/bin 12-6
 - /usr/kernel 9-17
 - /usr/kernel/drv 9-17
 - /usr/lib/lp 12-6
 - /usr/lib/lp/postscript 12-7
 - /usr/lib/lp/model 12-6
 - /usr/sbin 12-6
 - /usr/share/lib/terminfo 12-6
 - /var 2-4, 2-5
 - /var/lp/logs 12-10
 - /var/sadm 7-4
 - /var/sadm/patch 7-3
 - /var/spool/lp 12-9
 - /var/spool/pkg 6-15
 - description 2-13
 - home 10-2
 - lost+found directory 4-18
 - make command 2-14
 - structure 2-2
 - symbolic links 2-14
- directory hierarchy 2-2
- directory location, removable media 5-24
- disable command 12-35
- disk
 - backup 3-35
 - formatting with Solaris Management Console Storage Manager 3-71
 - label 4-6
 - labels 3-35
 - number 3-8
 - overlapping slices 3-34
 - partition 3-36
 - repartitioning 3-35
 - slide 3-36
 - undesirable conditions 3-32
 - wasted space 3-33
- disk blocks 3-4
- disk label fields 3-46
- disk names
 - logical 3-10
 - physical 3-11
- disk number
 - IDE 3-8
 - SCSI 3-8
- disk partitioning
 - \$ value 3-43
 - blocks 3-41
 - cylinders 3-41
 - flag 3-41
 - part 3-41

procedure 3-38
size 3-41
SMC 3-71
tag 3-41
disk platter 3-4
disk slice
 controller number 3-8
 defining 3-32
 disk number 3-8
 file system 2-2
 naming convention 3-8
 offset 3-32
 slice number 3-8
 target number 3-8
disk space
 by user command 4-30
 usage command 4-25
disk structure 3-2
disk-based file systems 4-2
diskette drive 5-24
diskette drive, location 5-24
displaying
 details on all packages 6-8
 software package information 6-7
distributed file system 4-2
downloading patches 7-6
du command 4-28
Dynamic Host Configuration Protocol
 (DHCP) 1-18
dynamic kernel 9-18

E

editing /etc/system 9-21
EDITOR variable 13-20
EEPROM command 8-26
enable command 12-35
enabling login checking 11-6
end user system support software
 group 1-9
environment variables
 LPDEST 12-14
 PRINTER 12-14

F
failed login file 11-6
file system
 /var/run 5-8
 backup 14-2
 backup information 14-9
 busy 5-19
 checking 5-14
 corruption 4-17
 create new 5-15
 creating a ufs 4-14
 determining the type of 5-16
 disk-based 4-2
 display capacity 4-26
 distributed 4-2
 forced unmount 5-19
 HSFS 5-17
 manually unmounting 5-18
 minfree 4-15
 monitoring 4-25
 mount point 5-2
 mounting different types 5-16
 mounting manually 5-11
 mounting new 5-15
 name 14-6
 PCFS 5-17
 pseudo 4-3, 5-8
 restoring 15-3
 root type 9-20
 state flag 4-17
 structure 2-2
 UFS 4-2, 4-4
 unmount all 5-19
 unmounting 5-18
 unmounting busy 5-19
 virtual 5-5
file system inconsistencies
 resolving 4-20
files
 \$HOME/.printers 12-14
 \$HOME/.rhosts 11-17
 .plan 11-4
 /etc/vfstab 5-11
 /etc/cron.d/at.allow 13-16
 /etc/cron.d/at.deny 13-16

/etc/default/fs 5-16
/etc/default/login 11-14, 11-15
/etc/default/passwd 10-10
/etc/default/su 11-12, 11-13
/etc/dfs/fstypes 5-16
/etc/format.dat 3-35
/etc/ftpd/ftpusers 11-16
/etc/group 10-2, 10-3, 10-8
/etc/hosts.equiv 11-17
/etc/inetd.conf 12-10
/etc/mnttab 5-8
/etc/passwd 10-3
/etc/printers.conf 12-14
/etc/rmmount.conf
 configuration 5-25
/etc/shadow 10-3, 10-6
/etc/system 9-18, 9-20, 9-21
/etc/system configuration 9-16
/etc/vfstab 5-5, 5-16
/etc/vold.conf configuration 5-25
/reconfigure 3-17
/var/adm/loginlog 11-6
/var/adm/utmpx 11-2
/var/adm/wtmpx 11-5
/var/lp/logs/requests 12-9
/var/sadm/install/contents 6-2
backing-store 16-3
configuration 5-25
creating regular files 2-12
crontab 13-17
data blocks 2-9
description of 2-11
failed login 11-6
file names 2-9
inodes 2-9
list command 2-11
regular 2-12
repairing 5-21
switch user log 11-13
types 2-11
 unreferenced 4-20
finger command 11-4
firmware 8-2
fmthard command 3-47
forced unmount 5-19
forceunload parameter 9-20
format command 3-16, 3-31
format hard disk 3-47
F PROM 8-3
fragmentation 4-12
free list 4-21
fsck command
 definition of 5-14
 interactive mode 4-19
 non-interactive mode 4-18
fsck program
 at bootup 4-17
 definition of 4-17
 lost+found directory 4-18
fssnap command 16-2
fssnap -i command 16-5
fstyp command 5-17
FTP, restricting access 11-16
full backup 14-7
fuser command 5-19

G

genunix static core 9-16
geographic location 1-12
GID 10-2, 10-5
group file syntax 10-9
groupadd command 10-23
groupdel command 10-26
groups command 11-36

H

hard disk
 cylinder 3-4
 format 3-47
 head actuator arm 3-3
 read/write heads 3-3
 Slice 2 3-4
 structure 3-2
 track 3-4
hard sector 3-4
hardware requirements 1-5
head actuator arm 3-3
help command 8-14
help screen 3-65

host IP address 1-11
host name 1-11
HSFS file system 5-17

I

id command 11-37
IDE configuration 2-18, 3-9
IDE controller devices 8-19
identifying devices 8-17
in.lpd daemon 12-11
incremental backup 14-7
 snapshot 16-7
incremental restore 15-9
indirect pointers 4-11
inetd daemon 12-10
information pane 3-69
init command 9-45
init phase 9-22
inode
 allocated and unreferenced 4-20
 definition of 2-9, 4-9
 direct pointers 4-11
 indirect pointers 4-11
inode consistency 4-18
install_cluster 7-12
installation
 backup 1-13
 custom JumpStart 1-3
 hardware requirements 1-5
 interactive 1-14
 pre-installation 1-11
 pre-installation information 1-11
 software arrangement 1-6
 web version 1-2
 WebStart 3.0 1-2
 WebStart Flash 1-3
installation options 1-2
installing patches 7-9
instance names 3-12
integrated device electronics (IDE) 2-18
interactive installation 1-14
interactive mode 8-13
Internet services daemon 12-10

J
Jaz drive 5-24

K

kernel
 configuring 9-18
 genunix 9-16
 initialization phase 9-16
 modules 9-17
 search path 9-20
kill command 13-9

L

language 1-12, 1-16
last command 11-5
line printer command 12-12
link command 2-15
link counter 4-21
list command 2-11
listen account 10-6
listing
 device path names 8-22
 NVRAM parameters 8-15
 system configuration 3-15
ln command 2-15
load device drivers 3-18
local print process 12-15
location bar 3-69
logical device names 3-10
login
 device types 11-2
 displaying activity 11-5
 enabling checking 11-6
 failed 11-6
 problems in CDE 10-38
 shell 10-2
 troubleshooting 10-36
login device types
 pts 11-2
 term 11-2
login ID 10-4
login incorrect 10-36

logs, printer requests 12-10
lp account 10-5
lp command 12-12
LP Print Service 12-35
lpadmin command 12-31
LPDEST environment variable 12-14
lpmove command 12-35
lpr command 12-12
lpsched daemon 12-15
ls command 2-11

M

magnetic tape control command 14-5
make directory command 2-14
minfree space 4-15
mkdir command 2-14
moddir 9-20
modes
 interactive 8-13
 reconfiguration 8-13
 single-user 8-13
 verbose 8-13
monitoring
 switch user attempts 11-12
 system access 11-2

mount
 checking file system 5-14
 manually 5-14
 options 5-8, 5-12
 procedure 5-15
 removable media 5-25
mount 5-11, 5-27
mount command 5-4, 5-16
mount point
 /etc/mnttab file 5-8
 column 14-6
 creating 5-15
 definition of 5-2
mounting process 5-2
mt command 14-5

N

name service type 1-12
navigation pane 3-68
netstandard script 12-7
network listening service daemon 12-11
network server daemon 11-3
newfs command 4-14, 5-15
newgrp command 10-9
noaccess account 10-6
nobody account 10-6
nobody4 account 10-6
nuucp account 10-5
nvalias command 8-24
NVRAM
 changing parameters 8-16, 8-26
NVRAM listing parameters 8-15
nvunalias command 8-25

O

OpenBoot architecture 8-2
overlapping disk slices 3-34

P

PASSREQ variable 11-15
password
 aging 10-2, 10-3
 encryption 10-3
 file syntax 10-4
 user account 10-2
patch
 checking current 7-4
 downloading 7-6
 formats 7-2
 ftp utility 7-5
 installing 7-9
 removing 7-10
patch clusters, installing 7-11
patchadd 7-4
patchadd command 7-9
patchrm command 7-10
path names, boot disk 8-28
PCFS file system 5-17

PCMCIA card 5-24
permission denied 10-36
permissions
 setgid 11-42
 setuid 11-41
 Sticky Bit 11-43
physical device names 3-11
physical disk structure 3-2
pkgadd command 6-6, 6-9
pkgchk command 6-6, 6-12
pkginfo command 6-6, 6-7
pkgrm command 6-6, 6-14
pointers
 direct 4-11
 indirect 4-11
Portable Open Systems Interface
 (POSIX) 12-34
PostScript filter programs 12-7
power on self test (POST) 8-6, 8-7, 8-9, 9-15
poweroff command 9-45
PRI 13-5
print client 12-3
print management tools 12-3
Print Manager 12-3, 12-19
print server
 configuration hierarchy 12-8
 definition of 12-3
 fault notification 12-5
 initialization 12-5
 queuing 12-5
 tracking 12-5
print services, configuring 12-19
printenv command 8-15
printer
 add access 12-22
 attached 12-22
 configuration 12-3
 configuration files 12-8
 configuring 12-30
 configuring network 12-22 to 12-29
 interface program files 12-8
 local 12-4
 locating destination 12-13
 network 12-4, 12-22
 print filter descriptor files 12-8
 process 12-15
remote 12-4
remote process 12-17
removing a configuration 12-31
request log 12-9
restart command 12-33
specifying destination 12-34
subdirectory of local printers 12-8
system default 12-31
temporary shutdown command 12-33
PRINTER environment variable 12-14
printers
 netstandard script 12-7
 PostScript filter programs 12-7
 standard script 12-6
printers.confbyname file 12-14
printing
 accepting jobs 12-35
 clearing hung processes 13-9
 disabling queuing 12-37
 enabling queuing 12-36
 moving jobs 12-38
 overview 12-3
 rejecting jobs 12-36
 terminating a hung login 13-11
probe- commands 8-17
probe-ide command 8-19
probe-scsi command 8-18
probe-scsi-all command 8-19
process
 stopping 5-20
process manager 13-2
process manager window 13-3
PROCESS/NLWP 13-5
processes
 /sbin/init 9-16
prstat command 13-4
prtconf command 3-15
prtvtoc command 3-46
pseudo file system 4-3, 5-8

Q

quot command 4-30

R

read/write heads 3-3
reboot command 9-45
reconfiguration boot 3-17
reconfiguration mode 8-13
reconnecting allocated unreferenced files 4-20
recovery, special case 15-6
regular files 2-12
reject command 12-35
remote
 backup 14-13
 displaying users 11-3
 print process 12-17
remote system users 11-3
removable media device 5-23
removable media, accessing 5-27
remove software package command 6-14
removing
 custom device aliases 8-25
removing a patch 7-10
repartitioning a disk 3-35
restore
 /opt file system 15-3
 /usr 15-5
 /usr file system 15-4
 /var 15-5
 incremental 15-9
 interactive 15-7
 regular file system 15-2
 root file system 15-6
restoresymtable file 15-2
restoring UFS file system 15-2
restricting ftp access 11-16
restricting root access 11-14
rmmount command 5-25
root
 access 11-8
 account 10-5
 file system type 9-20
 password 1-12
 restricting access 11-14
rpc.rusersd daemon 11-3
RSS 13-5

run control (rc) scripts
 creating 9-36
 definition of 9-25
 directory 9-29
run control scripts
 starting 9-28
 stopping 9-28
run levels
 definition of 9-12
 determining current 9-13
 scripts 9-27
rusers command 11-3

S

salvaging the free list 4-21
scheduler daemon 12-12
scheduling backups 14-7
scripts
 netstandard 12-7
 standard 12-6
SCSI configuration 3-9
SCSI controller devices 8-18
search path for kernel modules 9-20
secondary boot program 9-16
sector 3-4
set-defaults command 8-17
setenv command 8-16
setgid permission 11-42
setuid permission 11-41
shadow file syntax 10-7
show-devs command 8-22
show-disks command 8-24
showrev 7-4
shutdown command 9-45
shutdown procedures 9-44
SIGHUP signal 13-10
SIGINT signal 13-10
SIGKILL signal 13-10
SIGTERM signal 13-10
single-user mode 8-13
SIZE 13-5
Slice 2 3-4
slice number 3-8
smgroup add command 10-23
smgroup delete command 10-26

smgroup modify command 10-25
smmsp account 10-6
smuser add command 10-18
smuser command 10-14
smuser delete command 10-22
smuser modify command 10-21
snapshot
 backing up 16-6
 displaying 16-5
 incremental backup 16-7
 restoring backup 16-10
software
 adding packages 6-9
 administration 6-2
 arrangement 1-6
 checking packages 6-12
 clusters 1-6, 1-7
 packages 1-6, 1-7
 packages installed 6-2
 remove packages 6-14
software groups
 End User System Support 1-9
software packages 6-2
Solaris Management Console
 definition 3-64
 disk tools 3-71
 information pane 3-69
 location bar 3-69
 navigation pane 3-68
 restarting 3-66
 scheduler tool 13-22
 starting 3-64
 status bar 3-70
 toolbox editor 3-67
 tools 3-65
 usage 4-31
 view pane 3-69
Solaris Management Console Users Tool
 adding a user 10-27
 definition 10-27
 deleting a user 10-35
Solaris OS run levels 9-12
Solaris OS upgrade 1-4
special case recovery 15-6
spool directory 6-15
standard script 12-6
STATE 13-5
state flag 4-17
static core 9-16
status bar 3-70
Sticky Bit permission 11-43
stop all processes 5-20
Stop key 8-7
su command 11-8, 11-11
subnet mask 1-12
SULOG variable 11-13
SunService program 7-5
SunSolve database 7-5
sununinstall 1-23
SUNW 6-7
superblock
 backup 4-7, 4-22
 consistency 4-17
 corrupted 4-22
 definition of 4-7
 list alternates 4-22
switch user
 complete login 11-8
 log file 11-13
 overview 11-8
symbolic link
 creating 2-15
 definition of 2-14
sys account 10-5
sysinit 9-23
system access
 monitoring 11-2
 user information 11-4
system configuration listing 3-15
system default printer 12-31
system shutdown 9-44

T

tape device 14-3
tape drive control 14-5
tar command 7-8, 16-6
target number 3-8
time mounted 5-8
time zone 1-12
touch command 2-12
track 3-4

troubleshooting, login 10-36
tunefs command 4-16

U

UFS file system 4-2, 4-4, 15-2
ufsboot program 9-16
ufsdump command 14-10, 14-13, 16-6
ufsrestore command 15-3
ufsrestore i command 15-7
UID 10-2, 10-4
Ultra workstations 8-3
umount command 5-18
unallocated block count 4-21
unix static core 9-16
unmount -f command 5-20
umountall command 5-19
unmounting file systems 5-18
unmounting process 5-2
unreferenced files 4-20
unzip command 7-7
upgrade
 live 1-5
 standard 1-4

user
 becoming root 11-11
 displaying effective 11-8
 displaying information 11-4
 displaying real 11-9
 home directory 10-2
 switching 11-8
 switching log 11-13
 switching monitoring 11-12
 switching regular 11-10

user account
 command-line tools 10-14
 home directory 10-2
 login shell 10-2
 password 10-2
 password aging 10-2

user accounts
 adding a user with Solaris
 Management Console 10-27
 creating 10-15
 creating a group 10-23
 deleting a group 10-26

deleting a user with Solaris
 Management Console 10-35
managing 10-14
modifying 10-20
user name 10-2
useradd command 10-15
usermod command 10-20
uucp account 10-5

V

variables
 CONSOLE 11-13, 11-15
 EDITOR 13-20
 PASREQ 11-15
 SULOG 11-13
verbose mode 8-13
verify command 3-45
view pane 3-69
virtual device 16-3
virtual file system table 5-5
volcheck command 5-23
Volume Management
 administering 5-26
 configuration files 5-25
 daemon 5-23
 definition of 5-23
 starting 5-26
 stopping 5-26
volume table of contents (VTOC) 3-35, 4-6

W

WebStart 3.0 1-2
WebStart Flash installation 1-3
who command 11-2
whoami command 11-9
workstations 8-3

Z

zcat command 7-8
Zip drive 5-24