

Solaris™ 8 Operating Environment System Administration I

SA-238

Student Guide



Sun Microsystems, Inc.
MS BRM01-209
500 Eldorado Boulevard
Broomfield, Colorado 80021
U.S.A.

Revision A.2, August 2000

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, 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.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the logo Sun, Solaris, SunOS, ONC, NFS, JumpStart, Solstice AdminSuite, OpenBoot, HotJava, Ultra, Solaris Web Start, HotJava, UltraSPARC, Ultra Enterprise, SunService, SunSolve, and OpenWindows 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.

The OPEN LOOK and Sun Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government approval required when exporting the product.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g) (2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015 (b)(6/95) and DFAR 227.7202-3(a).

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.



Please
Recycle



Adobe PostScript

Copyright 2000 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto, California 94303, 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.

Des parties de ce produit pourront être dérivées du système Berkeley 4.3 BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Sun, Sun Microsystems, le logo Sun, Solaris, SunOS, ONC, NFS, JumpStart, Solstice AdminSuite, OpenBoot, HotJava, Ultra, Solaris Web Start, HotJava, UltraSPARC, Ultra Enterprise, SunService, SunSolve, and OpenWindows 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 marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

L'accord du gouvernement américain est requis avant l'exportation du produit.

Le système X Window est un produit de X Consortium, Inc.

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.



Contents

About This Course	xxi
Course Goal	xxi
Course Overview	xxii
Course Map	xxiii
Module-by-Module Overview	xxv
Course Objectives	xxix
Skills Gained by Module	xxx
Guidelines for Module Pacing	xxxii
Topics Not Covered	xxxii
How Prepared Are You?	xxxiii
Introductions	xxxiv
How to Use Course Materials	xxxv
Course Icons and Typographical Conventions	xxxvi
Icons	xxxvi
Typographical Conventions	xxxvii
 Introducing the Solaris 8 Operating Environment System	
Administration	1-1
Objectives	1-1
Additional Resources	1-1
Roles of the System Administrator	1-2
Administering Standalone Systems	1-3
Administering Client/Server Systems	1-5
System Administration Terms	1-7
Check Your Progress	1-9
 Adding Users	2-1
Objectives	2-1
Additional Resources	2-2
Setting Up User Accounts	2-3
Managing User Accounts	2-4
Managing User Accounts with <code>admintool</code>	2-5
Creating a New Group in the <code>/etc/group</code> File	2-7
Adding a New User Account	2-9

Password Aging	2-13
Modifying a User Account	2-17
Storing User and Group Account Information.....	2-21
The /etc/passwd File	2-22
Default System Account Entries	2-23
The /etc/shadow File	2-25
The /etc/group File	2-27
Creating and Managing Accounts from the Command-line....	2-29
Creating User Accounts	2-30
Command Format.....	2-30
Options	2-30
Adding a User with <code>useradd</code>	2-31
Modifying User Accounts.....	2-32
Command Format.....	2-32
Options	2-32
Example	2-32
Deleting User Accounts	2-33
Command Format.....	2-33
Options	2-33
Examples	2-33
Adding Group Accounts.....	2-34
Command Format.....	2-34
Options	2-34
Example	2-34
Modifying Group Accounts	2-35
Command Format.....	2-35
Options	2-35
Example	2-35
Deleting Group Accounts	2-36
Command Format.....	2-36
Example	2-36
Exercise: Adding Users and Groups	2-37
Preparation.....	2-37
Task Summary	2-37
Tasks	2-38
Exercise Summary.....	2-42
Task Solutions.....	2-42
Understanding Initialization Files.....	2-45
System-Wide Initialization Files	2-45
User Initialization Files	2-45
Customizing the Work Environment.....	2-47
Shell Variables	2-47
Setting Environment Variables in User Initialization Files.....	2-48
Using the Initialization File Templates	2-49
Exercise: Modifying Initialization Files	2-50

Preparation.....	2-50
Task Summary.....	2-50
Tasks	2-51
Exercise Summary.....	2-53
Task Solutions.....	2-54
Check Your Progress	2-55
System Security	3-1
Objectives	3-1
Additional Resources	3-2
Managing System Security Overview.....	3-3
Managing Login and Access Control.....	3-4
The <code>pwconv</code> Command.....	3-4
Recording Failed Login Attempts	3-4
Monitoring System Access.....	3-6
Displaying Users on the System	3-6
Login Device Types	3-6
Displaying User Information	3-7
Command Format.....	3-7
Displaying User Information	3-7
Displaying a Record of Login Activity	3-8
Displaying Users on Remote Systems.....	3-9
Command Format.....	3-9
Accessing root Privileges	3-10
Using the <code>su</code> Command to Become Another User.....	3-10
Effective User ID and Effective Group ID.....	3-11
Using the <code>whoami</code> Command.....	3-11
Displaying the Effective Current Username	3-11
Using the <code>su</code> Command to Become root.....	3-12
Using the <code>su</code> Command to Become Another Regular User.....	3-13
The <code>sysadmin</code> Group.....	3-14
Managing User Access	3-15
Monitoring su Attempts	3-16
The <code>CONSOLE</code> Variable	3-16
The <code>SULOG</code> Variable	3-17
Restricting root Access.....	3-18
The <code>CONSOLE</code> Variable	3-19
Implementing System-Wide Password Aging	3-20
The <code>/etc/default/passwd</code> File Variables.....	3-20
Exercise: User Access.....	3-22
Preparation.....	3-22
Task Summary.....	3-22
Tasks	3-23
Exercise Summary.....	3-27
Task Solutions.....	3-28
Restricting Access to Data in Files.....	3-30

Determining a User's Group Membership.....	3-31
Identifying a User Account.....	3-32
Command Format.....	3-32
Changing a File's Ownership with the <code>chown</code> Command	3-33
Command Format.....	3-33
Changing File Ownership.....	3-33
Changing Directory Ownership	3-34
Changing User and Group Ownership Simultaneously ...	3-34
Changing a File's Ownership With the <code>chgrp</code> Command.....	3-35
Command Format.....	3-35
Special File Permissions	3-36
The <code>setuid</code> Permission	3-37
The <code>setgid</code> Permission	3-38
Shared Directories.....	3-38
Searching for <code>setgid</code> Files and Directories	3-39
The Sticky Bit Permission	3-40
Searching for Directories with a Sticky Bit Permission	3-40
Exercise: File Owners, Groups, and Special Permissions	3-41
Preparation.....	3-41
Task Summary.....	3-41
Tasks	3-42
Exercise Summary.....	3-47
Task Solutions.....	3-48
Access Control Lists.....	3-51
ACL Entries.....	3-52
Adding and Modifying ACL Permissions on a File	3-54
Command Format.....	3-54
Examples of Modifying ACL Entries on a File	3-54
Determining if a File Has an ACL	3-55
Deleting an ACL Entry on a File.....	3-56
Command Format.....	3-56
Replacing an Entire ACL on a File	3-57
Command Format.....	3-57
An Example of Setting an ACL on a File	3-57
Another Example of Setting an ACL on a File.....	3-58
Exercise: Using Access Control Lists.....	3-59
Preparation.....	3-59
Task Summary.....	3-59
Tasks	3-60
Exercise Summary.....	3-64
Task Solutions.....	3-65
Managing Remote Access Issues	3-67
The <code>/etc/hosts.equiv</code> and <code>\$HOME/.rhosts</code> Files	3-68
Remote Access Authentication	3-69
Entries in <code>/etc/hosts.equiv</code> and <code>\$HOME/.rhosts</code>	3-70
The <code>/etc/hosts.equiv</code> File	3-71

The \$HOME/.rhosts File	3-71
Restricting FTP Logins	3-72
The /etc/shells File	3-73
Exercise: Managing Remote Security Issues	3-74
Preparation.....	3-74
Task Summary.....	3-74
Tasks	3-75
Exercise Summary.....	3-79
Task Solutions.....	3-80
Check Your Progress	3-81
The Directory Hierarchy	4-1
Objectives	4-1
Additional Resources	4-1
The Solaris Operating Environment File Types	4-2
Identifying File Types.....	4-3
File Names, Inodes, and Data Blocks	4-4
Regular Files	4-5
Directories	4-6
Symbolic Links	4-7
Device Files	4-9
Character Device Files	4-10
Block Device Files	4-11
Hard Links	4-12
The root Subdirectories.....	4-14
Exercise: Identifying File Types	4-18
Preparation.....	4-18
Task Summary.....	4-18
Tasks	4-18
Exercise Summary.....	4-22
Task Solutions.....	4-23
Check Your Progress	4-25
Device Configuration.....	5-1
Objectives	5-1
Additional Resources	5-1
Basic Architecture of a Disk.....	5-2
Physical Disk Structure	5-2
Components of a Disk Platter	5-4
Defining Disk Slices	5-6
The Boot Disk.....	5-7
Disk Slice Naming Convention.....	5-8
Device Naming Conventions	5-11
Logical Device Names	5-11
Physical Device Names	5-12
Instance Names	5-14
Listing a System's Devices.....	5-15

The /etc/path_to_inst File.....	5-15
Sample /etc/path_to_inst File	5-16
The prtconf Command.....	5-16
The format Command.....	5-18
Reconfiguring Devices	5-19
Configuring the Solaris 8 Operating Environment Devices	5-20
devfsadm Options	5-20
Configuring a Device Before the Solaris 8	
Operating Environment	5-22
Adding a New Disk or Tape Drive	5-22
Adding a New Disk Device	5-22
Adding a New Tape Drive	5-23
Exercise: Configuring and Naming Disks	5-24
Preparation.....	5-24
Task Summary.....	5-24
Tasks	5-25
Exercise Summary.....	5-28
Task Solutions.....	5-29
Check Your Progress	5-31
Disks, Slices, and Format	6-1
Objectives	6-1
Additional Resources	6-1
Disk Slices and the format Utility.....	6-2
Disk Labels and Partition Tables	6-3
Disk Partition Table	6-4
Defining Disk Slices.....	6-6
Defining Disk Partitions.....	6-7
Undesirable Conditions	6-7
Wasted Disk Space.....	6-7
Overlapping Disk Slices.....	6-7
Locations of Disk Partition Tables.....	6-9
Disk Partitioning	6-10
Saving a Partition Table to the /etc/format.dat File....	6-16
Locating and Using the Customized Partition Table.....	6-16
Repartitioning a Disk with the modify Command.....	6-18
Using the modify Command	6-18
Using the Free Hog Slice	6-20
Viewing the Disk's VTOC.....	6-22
Reading a Disk's VTOC Using the verify Command	6-22
Reading a Disk's VTOC Using the prtvtoc Command	6-23
The fmthard Command	6-24
Exercise: Disks, Slices, and Format.....	6-25
Preparation.....	6-25
Task Summary.....	6-25
Tasks	6-26

Exercise Summary.....	6-33
Task Solutions.....	6-34
Check Your Progress	6-35
The Solaris Operating Environment ufs File System	7-1
Objectives	7-1
Additional Resources	7-1
File System Types Supported by the Solaris Operating	
Environment	7-2
Disk-Based File System	7-2
Distributed File Systems	7-3
Pseudo File System	7-3
Introducing the Solaris Operating Environment	
ufs File System.....	7-4
Basic Disk Structures	7-6
The Disk Label (VTOC).....	7-6
The Boot Block.....	7-6
The Superblock.....	7-6
Backup Superblocks.....	7-6
Cylinder Groups.....	7-8
Inodes.....	7-9
Direct Pointers	7-11
Indirect Pointers	7-11
Data Blocks.....	7-12
Data Blocks and Fragmentation.....	7-12
Shadow Inode.....	7-14
Creating ufs File Systems.....	7-15
Creating a ufs File System	7-15
Exercise: Creating UFS File Systems	7-17
Preparation.....	7-17
Task Summary	7-17
Tasks	7-18
Exercise Summary.....	7-21
Task Solutions.....	7-22
Check Your Progress	7-24
Mounting File Systems	8-1
Objectives	8-1
Additional Resources	8-2
Working With File Systems	8-3
Identifying Mounted File Systems	8-5
The mount Command	8-5
The /etc/mnttab File	8-5
Mount Table Changes in /etc/mnttab	8-7
The /var/run File System.....	8-7
Mounting File Systems.....	8-8
The /usr/sbin/mount Command	8-8

Command Format.....	8-8
Mounting a Local File System Manually	8-8
Using Options With the mount Command.....	8-9
Automatic Mounting of File Systems.....	8-12
The Virtual File System Table: /etc/vfstab.....	8-12
The /etc/vfstab File	8-12
The /usr/sbin/ mountall Command	8-14
Checking File Systems Before Mounting.....	8-14
Unmounting File Systems.....	8-15
The /usr/sbin/umount Command.....	8-15
Automatic Unmounting of File Systems	8-16
The /usr/sbin/ umountall Command	8-16
Commands to Unmount a Busy File System	8-17
Using the fuser Command.....	8-17
Using the umount -f Command.....	8-18
Procedure for Mounting a New File System.....	8-19
Removable Media Device Management.....	8-20
Accessing Mounted Diskettes and CD-ROMs.....	8-20
Administering Volume Management	8-21
Administering Volume Management	8-22
Accessing a Diskette or CD-ROM Without Volume Management	8-23
Mounting Different Types of File Systems.....	8-24
Specifying a hsf s File System Type.....	8-24
Specifying a pcfs File System Type.....	8-24
Determining a File System's Type.....	8-25
Finding a File System's Type.....	8-25
The fstyp Command	8-26
Exercise: Mounting File Systems	8-27
Preparation.....	8-27
Task Summary	8-27
Tasks	8-28
Exercise Summary.....	8-31
Task Solutions.....	8-32
Check Your Progress	8-33
Maintaining File Systems.....	9-1
Objectives	9-1
Additional Resources	9-1
The File System Check Program	9-2
Data Inconsistencies Checked by fsck.....	9-2
Phases of fsck	9-3
Non-Interactive Mode	9-4
Interactive Mode	9-5
Using the fsck Command.....	9-5
Troubleshooting with fsck	9-7

Reconnecting an Allocated Unreferenced File.....	9-7
Adjusting a Link Counter	9-8
Salvaging the Free List	9-8
Using Backup Superblocks	9-8
Monitoring File System Usages	9-11
The df Command.....	9-11
The du Command.....	9-12
The ff Command.....	9-14
The quot Command	9-14
Troubleshooting	9-16
Repairing Important Files if Boot Fails	9-16
Exercise: Maintaining File Systems	9-18
Preparation.....	9-18
Task Summary.....	9-18
Tasks	9-19
Exercise Summary.....	9-22
Task Solutions.....	9-23
Check Your Progress	9-24
Scheduled Process Control.....	10-1
Objectives	10-1
Additional Resources	10-1
Processes Running on the System	10-2
Viewing Processes and PIDs	10-2
CDE Process Manager	10-3
The prstat Command.....	10-5
Scheduling the Automatic Execution of Commands.....	10-7
The crontab Command.....	10-7
The crontab File Format	10-8
crontab for the root User	10-9
Using crontab -l to View a Crontab File	10-10
Editing a crontab File.....	10-10
Controlling crontab Access.....	10-10
Removing a crontab File	10-11
The at Command.....	10-12
Command Format.....	10-12
Options	10-12
Executing the at Command	10-13
Denying at Access.....	10-13
Allowing at Access.....	10-14
Exercise: Process Control	10-15
Preparation.....	10-15
Task Summary.....	10-15
Tasks	10-16
Exercise Summary.....	10-19
Task Solutions.....	10-20
Check Your Progress	10-21

The Solaris Operating Environment LP Print Service	11-1
Objectives	11-1
Additional Resources	11-2
Solaris Operating Environment LP Print Service	11-3
Print Management Tools.....	11-3
Client-Server Model.....	11-4
Types of Printer Configurations	11-4
LP Print Service Functions.....	11-5
Configuring Printer Services	11-7
Print Server Requirements.....	11-7
The Solaris 8 Print Manager	11-9
Starting the Solaris Print Manager	11-9
Configuring a New Network Printer	11-11
Printing the Solaris Operating Environment	11-18
Examples of Using the Print Command	11-18
Examples of Specifying a Destination Printer	11-18
Submitting a Print Request Atomic Style	11-19
Submitting a Print Request POSIX Style	11-19
Locating the Destination Printer	11-20
The LP Print Service Directory Structure	11-22
LP Print Service Directories.....	11-23
The /usr/bin Directory	11-23
The /usr/sbin Directory	11-23
The /usr/share/lib/terminfo Directory	11-23
The /usr/lib/lp Directory	11-23
The /etc/lp Directory.....	11-25
The /var/spool/lp Directory.....	11-26
The /var/lp/logs Directory	11-26
LP Print Service Daemons	11-27
The Internet Service Daemon/usr/sbin/inetd	11-27
The /usr/lib/print/in.lpd Program	11-27
The /usr/lib/lpsched Daemon.....	11-27
The /usr/lib/saf/listen Daemon	11-28
The lpNet Daemon.....	11-28
The Solaris Operating Environment Printing Process.....	11-29
The Local Print Process	11-29
The Remote Print Process	11-31
Remote Printing in a Solaris 2.6 to Solaris 8 Operating Environment	11-31
Remote Printing in a Solaris 2.0 to Solaris 2.5.1 Environment	11-33
LP Print Service Commands.....	11-34
The accept and reject Commands.....	11-35
Using the accept Command to Allow Queuing	11-35
Using the reject Command to Prevent Queuing	11-35
The enable and disable Commands	11-36

Using the enable Command to Activate a Printer	11-36
Using the disable Command to Deactivate a Printer	11-36
The lpmove Command.....	11-37
Configuring the LP Print Service Using lpadmin Command	11-38
Creating Printer Classes.....	11-39
Printer Priority Within a Class.....	11-39
Creating a Printer Class	11-40
Setting or Changing a System's Default Printer.....	11-41
Manually Removing a Printer's Configuration	11-42
Halting and Restarting the LP Print Service	11-43
Exercise: LP Print Service.....	11-44
Preparation.....	11-44
Task Summary.....	11-44
Tasks	11-45
Exercise Summary.....	11-48
Check Your Progress	11-49
The Boot PROM	12-1
Objectives	12-1
The Boot PROM Concept.....	12-2
The NVRAM Component.....	12-2
Power On Self Test (POST).....	12-4
The OpenBoot Goal	12-4
Basic BootPROM Configurations.....	12-6
Systems Containing a Single System Board.....	12-6
Systems Containing Multiple System Boards.....	12-6
Controlling the POST Phase	12-8
Halting the Solaris Operating Environment	12-8
Basic Boot PROM Commands.....	12-10
The banner Command.....	12-10
The boot Command	12-11
Command Format.....	12-11
Options	12-11
The help Command	12-12
Detailed Help.....	12-13
The printenv Command	12-13
The setenv Command.....	12-15
The reset Command.....	12-15
The set-defaults Command	12-16
Device Tree.....	12-17
Viewing Device Path Names.....	12-19
Boot Disk Device Path Example.....	12-20
Using probe- Commands to Identify Devices.....	12-21
A probe- Warning Message.....	12-21
The probe-scsi Command.....	12-22
The probe-scsi-all Command	12-22

The probe - ide Command.....	12-23
Identifying the System's Boot Device	12-24
Creating Custom Device Aliases	12-25
The nvalias and nvunalias Commands	12-25
The nvedit Command.....	12-26
Changing NVRAM Parameters with the eeeprom Command.	12-28
Examples	12-28
Interrupting an Unresponsive System.....	12-29
Exercise: OpenBoot PROM	12-30
Preparation.....	12-30
Task Summary.....	12-30
Tasks	12-31
Exercise Summary.....	12-36
Task Solutions.....	12-37
Check Your Progress	12-39
/sbin/ij System Boot Process	13-1
Objectives	13-1
Additional Resources	13-2
The Solaris Operating Environment Run Levels.....	13-3
Determining a System's Current Run Level	13-4
The Boot Process	13-5
Boot PROM Phase	13-7
Boot Programs Phase.....	13-7
The kernel Initialization Phase.....	13-8
Configuring the kernel.....	13-10
Sample /etc/system File.....	13-12
The init Phase.....	13-14
The /etc/inittab File	13-15
Default /etc/inittab File	13-17
The init Process.....	13-18
Run Control Scripts	13-20
The /sbin Directory	13-20
The /etc/rc#.d Directories.....	13-21
The /etc/init.d Directory	13-22
Summary of Run Control Scripts and Functions	13-23
Creating a New Run Control Script	13-24
System Shutdown Procedures	13-26
The /sbin/init Command.....	13-26
The /usr/sbin/shutdown Command.....	13-27
The /usr/sbin/halt Command.....	13-28
The /usr/sbin/poweroff Command	13-28
The /usr/sbin/reboot Command	13-29
Exercise: The Boot Process.....	13-30
Preparation.....	13-30
Task Summary.....	13-30

Tasks	13-31
Exercise Summary.....	13-34
Task Solutions.....	13-35
Check Your Progress	13-36
Installing the Solaris 8 Operating Environment on a Standalone System.....	14-1
Objectives	14-1
The Solaris Operating Environment Software Installation Options	14-2
Hardware Requirements of a Solaris 8 Operating Environment Installation.....	14-4
The Solaris 8 Operating Environment Installation CD-ROM...	14-5
The Solaris 8 Operating Environment SPARC Platform Edition CD-ROM.....	14-5
International Versions of the Solaris 8 Operating Environment	14-5
Intel Versions of the Solaris 8 Operating Environment.....	14-6
Choosing the Correct CD for Your Installation Requirements.....	14-6
The Solaris Operating Environment Software Arrangement...	14-7
Software Packages	14-7
Software Clusters	14-8
Cluster Configurations.....	14-8
The Solaris Operating Environment Software Groups	14-9
Planning an Installation on a Standalone System	14-11
Pre-Installation Information	14-12
Software Installation Using Solaris Web Start.....	14-14
Installing the Solaris 8 Operating Environment.....	14-26
Additional Software	14-39
Exercise: The Solaris Operating Environment.....	14-40
Preparation.....	14-40
Task Summary.....	14-40
Tasks	14-41
Exercise Summary.....	14-45
Check Your Progress	14-46
Administration of Software Packages	15-1
Objectives	15-1
Additional Resources	15-1
Software Packages	15-2
The pkginfo Command.....	15-3
Command Format.....	15-3
Displaying Detailed Information for All Packages.....	15-3
Displaying Detailed Information for a Specific Package ..	15-4
Displaying Information for Software Packages on CD-ROM.....	15-4

The <code>pkgrm</code> Command	15-6
Command Format.....	15-6
The <code>pkgadd</code> Command.....	15-8
Command Format.....	15-8
The <code>pkgchk</code> Command.....	15-9
Command Format.....	15-9
The <code>/var/sadm/install/contents</code> File	15-11
Identifying the Directory Location of a Command.....	15-12
Search the Solaris Operating Environment CD-ROM for	
Command Information	15-12
Adding and Removing Packages With <code>admintool</code>	15-13
To Display Software Package Information.....	15-13
Managing Software With <code>admintool</code>	15-18
Adding a Software Package	15-18
Using a Spool Directory	15-23
Spooling Packages	15-23
Removing Packages From the Spool Directory	15-23
Package Administration Summary	15-24
Package Command Summary.....	15-24
Package Administration File and Directory Summary ...	15-24
Exercise: Software Package Administration Commands.....	15-25
Preparation.....	15-25
Task Summary.....	15-25
Tasks	15-25
Exercise Summary.....	15-29
Task Solutions.....	15-30
Check Your Progress	15-31
Managing Software Patches.....	16-1
Objectives	16-1
Additional Resources	16-1
Patch Administration	16-2
Patch Distribution.....	16-3
World Wide Web Patch Access.....	16-4
SunSolve Site.....	16-5
An Additional URL for Patch Access.....	16-6
Anonymous <code>ftp</code> Patch Access	16-7
An Additional <code>ftp</code> Site for Patch Access.....	16-7
The <code>ftp</code> Patch Access Procedure	16-7
Downloading Patches.....	16-9
Patch Informational Documents.....	16-10
Listing Patch Documents Using <code>ftp</code>	16-10
The <code>/var/sadm/patch</code> Directory	16-12
Patch Formats	16-13
Preparing Patches for Installation	16-13
Patch Contents.....	16-14

The patchadd and patchrm Commands	16-15
Installing a Patch	16-16
Installing a Patch in the Solaris 2.6 Operating Environment and Later Versions	16-16
Installing a Patch in a Pre-Solaris 2.6 Operating Environment 16-17	
Checking Current Patch Status	16-19
Removing a Patch	16-20
Removing a Patch from the Solaris 2.6 and Later Operating Environments	16-20
Removing a Patch from the Pre-Solaris 2.6 Operating Environments	16-20
Exercise: Patches Maintenance.....	16-21
Preparation.....	16-21
Task Summary.....	16-21
Tasks	16-21
Exercise Summary.....	16-24
Task Solutions.....	16-25
Check Your Progress	16-26
Backup and Recovery	17-1
Objectives	17-1
Additional Resources	17-1
Backing Up and Restoring File Systems	17-2
Importance of Regular File System Backups	17-2
Tape Device Types.....	17-3
Tape Device Naming	17-4
Logical Tape Device Names	17-4
Data Compression.....	17-5
Types of File System Backups	17-6
The ufsdump Command.....	17-6
Command Format.....	17-6
Common Options.....	17-6
The /etc/dumpdates File.....	17-8
Scheduling Backups.....	17-9
A Sample Backup Strategy	17-10
Planning File System Backups	17-11
Finding File System Names.....	17-11
Determining the Number of Tapes	17-11
Backing Up to Tape.....	17-12
Performing Remote Backups.....	17-13
Command Format.....	17-13
Restoring File Systems	17-14
Command Format.....	17-14
Common Options.....	17-14
The restorestymtable File	17-15

Preparing to Restore File Systems	17-15
Restoring the root (/) File System.....	17-16
Restoring the /usr and /var File Systems.....	17-17
Restoring Regular File Systems.....	17-17
Invoking an Interactive Restore	17-18
Controlling the Tape Drive.....	17-20
Command Format.....	17-20
Examples of Handling Multiple Archives.....	17-20
Exercise: Backup and Recovery	17-21
Preparation.....	17-21
Task Summary.....	17-21
Tasks	17-22
Exercise Summary.....	17-25
Task Solutions.....	17-26
Check Your Progress	17-27
New Features of the Solaris 8 Operating Environment.....	A-1
fsck – Handling Error Messages.....	B-1
The Phases of the fsck Command	B-1
Initialization Phase.....	B-1
Phase 1	B-3
Phase 2	B-5
Phase 3	B-10
Phase 4	B-12
Phase 5	B-13
Cleanup Phase	B-14
Adding Network Printers.....	C-1
Adding a Network Printer.....	C-1
Using Printer Vendor Supplied Tools.....	C-1
Setting Up the LexMark Optra Model Network Printer	C-2
Setting Up a Sun System as the Network Printer Server	C-4
Installing the Software Packages	C-4
Configuring the Network Printer Software	C-6
Setting Up an HP LaserJet 4000TN Network Printer	C-11
Installing the HP JetAdmin Utility for UNIX	C-12
Testing the Installation of the HP Network Printer	C-18
Enabling Access to a Network Printer	C-19

About This Course

Course Goal

Administering the Solaris™ 8 Operating Environment involves many tasks, including standalone installation, file system management, backups, process control, user administration, and device management. Students taking this class will gain the necessary knowledge and skills to perform these essential system administration tasks in the Solaris 8 Operating Environment.

This course also prepares system administrators for the follow-on course, SA-288: *Solaris 8 System Administration II*.

Course Overview

The primary objective of this course is to teach new system administrators the basics of administering Sun workstations.

Attending this course provides hands-on experience in installing and maintaining a standalone workstation in the UNIX[®] environment.

You will perform basic administration tasks, such as installing a standalone system, adding users, backing up and restoring file systems, and adding printer support. The procedures needed to perform these system administration tasks are emphasized. The course also introduces the concepts of file systems and disk management.

Course Map

The following course map enables you to see what you have accomplished and where you are going in reference to the course goal:

Introduction

Introducing the
Solaris 8
Operating
Environment
System
Administration

Users, Initialization Files, and Security

Adding
Users

System
Security

Devices, Disks, and File Systems

The Directory
Hierarchy

Device
Configuration

Disks, Slices,
and Format

The Solaris
Operating
Environment
ufs
File System

Mounting
File Systems

Maintaining
File Systems

Processes and Printing

Scheduled
Process
Control

The Solaris
Operating
Environment
LP Print
Service

System Firmware, Boot Process, and Run Levels

The
Boot Prom

System Boot
Process

Software Installation and Administration

Installing the Solaris 8 Operating Environment on a Standalone System	Administering Software Packages	Administering Software Patches	Backup and Recovery

Module-by-Module Overview

This course contains the following modules:

- Module 1 – “Introducing the Solaris 8 Operating Environment System Administration”

This module defines the roles of a Solaris Operating Environment system administrator and describes some common system administration terms used in the Solaris Operating Environment.

- Module 2 - "Adding Users"

This module introduces the task of adding users: creating new groups and user accounts, setting up user environments, identifying fields in the `/etc/passwd`, `/etc/shadow`, and `/etc/group` files.

Lab exercise – Add, modify, and delete user accounts and groups using `admintool` and command line tools. Create a `.profile` and `.kshrc` file for a Korn shell user.

- Module 3 - "System Security"

This module focuses on accounts, commands, and files that have an affect on basic system security, including how to set access control lists on files, and identifying `setuid`, `setgid`, and sticky permissions.

Lab exercise – Modify the content of a system security file, create ACLs on files

- Module 4 - "The Directory Hierarchy"

This module describes the main file types in the Solaris Operating Environment and defines the function of the main subdirectories located in the `root` directory.

- Module 5 - "Device Configuration"

This module describes the device naming conventions used in the Solaris 8 Operating Environment, and commands to display and reconfigure device configurations.

Lab exercise – Identify the devices and device names attached to a system.

- Module 6 - "Disks, Slices, and Format"

This module covers the use of the `format` utility to view a partition table, define disk slices, label a disk, and modify pre-existing disk slices.

Lab exercise – Use the `format` utility to create and save a working partition table on an unused disk, and modify the size of a disk slice.

- Module 7 – "Solaris Operating Environment ufs File System"

The module defines three common file system types, introduces the structure of a ufs file system, and describes the procedures for creating a new ufs file system.

Lab exercise - Create a new ufs file system on an unused disk slice using the `newfs` command.

- Module 8 - "Mounting File Systems"

This module describes the concepts and procedures involved in mounting and unmounting file systems, and using the `/etc/vfstab` file to mount file systems automatically at boot time.

Lab exercise – Add entries to the `/etc/vfstab` file and mount a new file system.

- Module 9 - "Maintaining File Systems"

This module describes the `fsck` utility for checking and repairing file systems, and introduces commands for monitoring file system usage.

Lab exercise – Display file system usage information and practice using the `fsck` utility to repair a corrupted file system.

- Module 10 - "Scheduled Process Control"

This module introduces commands for viewing and controlling the processes running on the system; and describes the procedures for automating repetitive tasks.

Lab exercise – Run the process manager and the `prstat` command to view and control processes running on the system, and automate the execution of commands using the `at` command and by creating a `crontab` file.

- Module 11 - "The Solaris Operating Environment LP Print Service"

This module covers the functions of the print service, introduces the LP administration commands, and procedures for adding a printer for access by users.

Lab exercise – Configure a printer and use various LP print commands.

- Module 12 – "The Boot PROM"

This module introduces the main functions of the OpenBoot™ programmable read-only memory (PROM) and NVRAM; it describes the use of boot PROM commands, how to determine the default boot device, how to modify parameters, and procedures for creating custom device aliases.

Lab exercise – Create custom device aliases and modify parameters.

- Module 13 – "System Boot Process"

This module focuses on the phases of the boot process, and discusses the various commands used to change system run levels.

Lab exercise – Use commands to change your system's run level, and add a new run control script.

- Module 14 – "Installing Solaris Operating Environment 8 on a Standalone System"

This module describes the procedures for installing the Solaris 8 Operating Environment software.

Lab exercise – Install software on a standalone workstation.

- Module 15 – "Administering of Software Packages"

This module focuses on displaying software package information, and adding and deleting software packages.

Lab exercise – Identify installed packages, remove a package, and add a package.

- Module 16 – “Administering Software Patches”

This module covers the procedures for adding and backing out software patches.

Lab exercise – Install and back out a software patch.

- Module 17 – “Backup and Recovery”

The module focuses on how to back up and restore file systems.

Lab exercise – Restore the `root` file system.

Course Objectives

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

- Define basic system administration tasks and terms
- Add users and groups to the system
- Configure user initialization files
- Implement basic system security
- Create ACLs (access control lists) on files
- Identify disks configured on a system
- Define disk slices on a new disk
- Create and mount a file system
- Repair a corrupted file system
- View and manage processes
- Configure and administer printers
- Identify the default boot device
- Describe the boot process
- Change system run levels
- Install the Solaris 8 Operating Environment software on a standalone workstation
- Add software packages
- Add a software patch
- Perform a root file system backup and restore

✓

Skills Gained by Module

The skills for *Solaris 8™ Operating Environment System Administration I* are shown in column 1 of the following matrix. The black boxes indicate the main coverage for a topic; the gray boxes indicate the topic is briefly discussed.

Skills Gained	Module																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Define basic system administration tasks and terms																	
Add users and groups to the system																	
Configure user initialization files																	
Implement basic system security																	
Create ACLs on files																	
Identify disks configured on a system																	
Define disk slices on a new disk																	
Create and mount a file system																	
Repair a corrupted file system																	
View and manage processes																	
Configure and administer printers																	
Identify the default boot device																	
Describe the boot process																	
Change system run levels																	
Install the Solaris 8 Operating Environment software on a standalone workstation																	
Add software packages																	
Add software patch																	
Perform a root file system backup and restore																	

Guidelines for Module Pacing

The following table provides a rough estimate of pacing for this course:

Module	Day 1	Day 2	Day 3	Day 4	Day 5
"About This Course"	A.M.				
"Introducing the Solaris 8 Operating Environment System Administration"	A.M.				
"Adding Users"	A.M.				
"System Security"	P.M.	A.M.			
"The Directory Hierarchy"		A.M.			
"Device Configuration"		A.M.			
"Disks, Slices, and Format"		P.M.			
"The Solaris Operating Environment ufs File System"		P.M.			
"Mounting File Systems"			A.M.		
"Maintaining File Systems"			A.M.		
"Scheduled Process Control"			P.M.		
"The Solaris Operating Environment LP Print Service"			P.M.		
"The Boot PROM"				A.M.	
"System Boot Process"				A.M./ P.M.	
"Installing the Solaris 8 Operating Environment on a Standalone"				P.M.	
"Administering Software Packages"					A.M.
"Administering Software Patches"					A.M.
"Backup and Recovery"					P.M.

Topics Not Covered

This course does not cover the topics shown below. Topics listed here are covered in other courses offered by Sun Educational Services:

- Basic UNIX commands – Covered in SA-118: *Fundamentals of Solaris 8 for System Administrators*
- The vi editor – Covered in SA-118: *Fundamentals of Solaris 8 for System Administrators*
- Basic UNIX file security – Covered in SA-118: *Fundamentals of Solaris 8 for System Administrators*
- JumpStart™ – Covered in SA-288: *Solaris 8™ Operating Environment System Administration II*
- Solstice™ AdminSuite™ – Covered in SA-288: *Solaris 8™ Operating Environment System Administration II*
- NFS™ environment configuration – Covered in SA-288: *Solaris 8™ Operating Environment System Administration II*
- Naming services – Covered in SA-288: *Solaris 8™ Operating Environment System Administration II*
- Troubleshooting – Covered in ST-350: *Sun Systems Fault Analysis Workshop*
- System tuning – Covered in SA-400: *Concepts and Tuning*

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

How Prepared Are You?

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

- Can you use basic UNIX® commands to navigate the Solaris Operating Environment directory tree, to search for or manipulate directories and file?
- Can you use the vi text editor to create or modify files?
- Can you change access permissions on files and directories?

Introductions

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

- Name
- Company affiliation
- Title, function, and job responsibility
- System administrator experience
- Reasons for enrolling in this course
- Expectations for the course

How to Use Course Materials

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

- **Course map** – An overview of the course content appears in the "About This Course" module so you can see how each module fits into the overall course goal.
- **Objectives** - What you should be able to accomplish after completing this module is listed here.
- **Lecture** – The instructor will present information specific to the topic of the module. This information will help you learn the knowledge and skills necessary to succeed with the exercises.
- **Exercise** – Lab exercises will give you the opportunity to practice your skills and apply the concepts presented in the lecture.
- **Check your progress** – Module objectives are restated, sometimes in question format, so that before moving on to the next module you are sure that you can accomplish the objectives of the current module.

Course Icons and Typographical Conventions

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

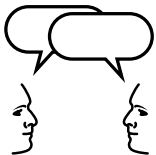
Icons



Additional resources – Indicates additional reference materials are available.



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



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



Exercise objective – Indicates the objective for the lab exercises that follow. The exercises are appropriate for the material being discussed.

Note – Additional important, reinforcing, interesting, or special information.



Caution – A potential hazard to data or machinery.



Warning – Anything that poses personal danger or irreversible damage to data or the operating system.

Typographical Conventions

Courier is used for the names of commands, files, and directories, as well as on-screen computer output. For example:

```
Use ls -al to list all files.  
system% You have mail.
```

It is also used to represent parts of the Java™ programming language such as class names, methods, and keywords. For example:

The `getServletInfo` method is used to...
The `java.awt.Dialog` class contains `Dialog` (Frame parent)

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

```
system% su  
Password:
```

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

To delete a file, type `rm filename`.

Palatino italics is used for book titles, new words or terms, or words that are emphasized. For example:

Read Chapter 6 in *User's Guide*.
These are called *class* options.
You *must* be root to do this.

Introducing the Solaris 8 Operating Environment System Administration

1 

Objectives

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

- Define the roles of a Solaris Operating Environment system administrator
- Define common system administration terms

Additional Resources



Additional resources – The following references can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10
- *Solaris 8 System Administration Guide, Volume II*, Part Number 805-7229-10
- *Solaris 8 System Administration Guide, Volume III*, Part Number 806-0916-10

Roles of the System Administrator

The system administrator is responsible for the smooth operation of day-to-day activities on each system. The scope and variety of tasks that a Solaris Operating Environment system administrator performs have been placed into the following two course categories:

- The first category encompasses all the major skills and activities required to administer a standalone system and are covered in this course: *SA-238 Solaris 8 Operating Environment System Administration I*.
- The second category includes those skills and activities required to successfully administer a basic client/server configuration and are covered in the course: *SA-288 Solaris 8 Operating Environment System Administration II*.

Administering Standalone Systems

The tasks described in this category are necessary to perform system administration duties on SunTM Microsystems systems in a standalone environment. These are also the required prerequisite skills for mastering the topics outlined in the second course category.

The following lists the essential activities for all system administrators:

- Managing users accounts
Setting up login accounts for new users and removing accounts when users no longer require system access.
- Maintaining system security
Monitoring and controlling system access, maintaining passwords, assigning special privileges to selected users, and controlling file access.
- Configuring new devices
Adding and configuring new peripheral devices on systems.
- Installing and partitioning disk drives
Partitioning disks to handle new or larger file systems to satisfy increased storage requirements on systems.
- Managing file systems
Creating, mounting, and maintaining file systems to ensure access to system, application, and user data.
- Scheduling system-related jobs
Scheduling jobs to run automatically during off-peak hours when system loads are at a minimum.
- Maintaining print services
Installing, maintaining, and removing printers and print services.

- Managing the boot PROM

Using basic boot PROM commands to select alternative boot devices, creating alternative device alias names, and customizing boot PROM environment variables.

- Configuring system initialization files

Modifying the run control scripts and files used to control system operations during boot.

- Installing the Solaris Operating Environment software

Preparing and installing the Solaris 8 Operating Environment software on standalone systems.

- Administering software package and patches

Adding or removing necessary software packages and patches.

- Performing backup and recovery operations

Backing up and restoring file systems on a regular schedule.

- Managing disaster recovery

Recovering critical file systems and rebooting successfully.

Administering Client/Server Systems

The following tasks are necessary to perform system administration duties on Sun systems within a client/server environment and are covered in the *SA-288 Solaris 8 Operating Environment System Administration Part II* course.

- Configure a network environment

Configure a system to function in a networked client/server environment.
- Set up the `syslog` utility

Set up system logging utilities, basic diagnostics, and availability enhancements.
- Configure and administer a Network file system (NFS) environment

Configure distributed file systems and administer NFS servers and NFS clients.
- Configure `cacheFS` file systems

Improve system performance by configuring a `cacheFS` file system. Monitor `cacheFS` file system statistics and maintain logs of the `cacheFS` file system.
- Use automount

Configure the system for shared resources to be mounted only if requested. Set up multiple paths to shared resources to mount the least busy path on demand.
- Set up name services

Select the proper name service to match system capabilities and requirements. Set up systems to use name services.
- Configure boot protocols

Configure a server for thin client support.

✓ *The Sun Ray 1 network appliance is an example of a thin client.*

- Install and configure Solstice AdminSuite™

Install and configure products associated with the Solaris 8 Operating Environment Administration Package.

- Install the Solaris Operating Environment using the Jumpstart program

Set up an automatic installation process for unattended installations.

System Administration Terms

The following list defines some common system administration terms.

- Host – Another word for a computer system.
- Host name – A unique name given to a computer system by the system administrator to distinguish it from other hosts on the network. The command `uname -n` displays the assigned host name.
- Internet (IP) address – A number that represents the host address and the network address, for example: 192.134.117.25. A host's IP address identifies where a host is on the Internet, which allows network traffic to be directed to that host. This software address is placed in the `/etc/inet/hosts` file.
- Ethernet address – A host's unique hardware address. A number displayed as 12 hexadecimal digits. For example, 08:00:20:1c:54:7e. This address is stored in the NVRAM (non-volatile random access memory) chip.
- Server – A host that provides one or more services to hosts on a network.
- Client – A host that uses services provided by the server.

Note – Servers and clients are two types of hosts in a distributed computing environment.

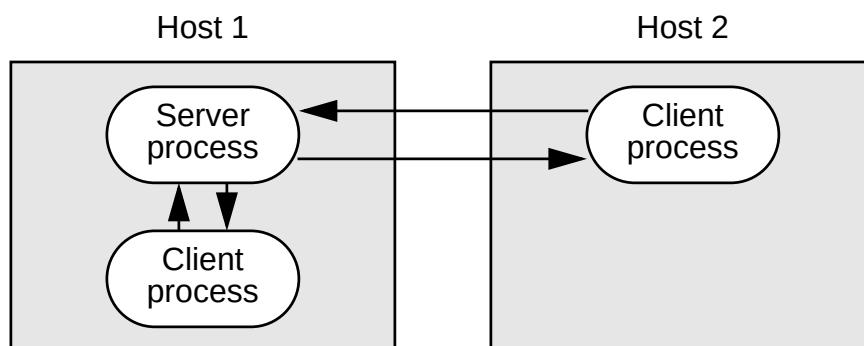


Figure 1-1 Example of Two Types of Hosts

A wide variety of server and client processes can be operating in a network environment. For example:

- A *file server* is a host that shares its disk storage and files with other hosts on the network.
- A *print server* provides network printing services to other hosts.
- An *application server* provides applications to various hosts.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Define the roles of a Solaris Operating Environment system administrator
- ☐ Define common system administration terms

Objectives

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

- Create and manage user accounts on the local system using the `admintool` utility
- Describe the format of the files `/etc/passwd` and `/etc/shadow` for securing login access
- Describe the format of the `/etc/group` file for maintaining shared and restricted access to files and directories
- Add, modify, and delete user accounts on the local system with the commands `useradd`, `usermod`, and `userdel`
- Add, modify, and delete group accounts for the local system with the commands `groupadd`, `groupmod`, and `groupdel`
- Define the two different types of shell initialization files
- Describe the shell startup activities during login for the three main Solaris Operating Environment shells
- List the shell initialization files used to set up a user's work environment at login
- Describe the purpose of the `/etc/skel` directory
- Modify the initialization files to customize a user's work environment

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10

✓ **The files `/var/adm/utmp` and `/var/adm/wtmp` are obsolete in the Solaris 8 Operating Environment. These files have been replaced with `/var/adm/utmpx` and `/var/adm/wtmpx`.**

Setting Up User Accounts

An important system administration task is setting up user accounts for each user requiring system access. Each user account consists of five main components:

- User name – A unique name a user enters to log in to a system, also called a login name.
- Password – A combination of six to eight letters, numbers, or special characters that a user must enter with the login name to gain access to a system.
- User's home directory – A directory the user is placed in after login, for creating and storing files.
- User's login shell – The user's work environment is set up by the initialization files defined by the user's login shell. There are six possible login shells in the Solaris Operating Environment, which include the Bourne shell, Korn shell, C shell, Z shell, BASH shell, and the TC shell.
- User initialization files – Shell scripts that determine how a user's work environment is to be set up when the user logs in to a system.

Managing User Accounts

You can add, modify, and delete user accounts on the system using either command-line tools or the graphical interface utility called `admintool`.

However, before you can add user accounts to the system, you must determine the following information for each new user:

- Login name – Each user's name must be unique and consist of two to eight letters (A_Z, a-z) and numbers (0-9). The first character must be a letter, and at least one character must be a lowercase letter. User names cannot contain spaces.
- User identification (UID) number – The user's unique numerical ID for the system. UID numbers for regular users range from 100 to 60000. All UID numbers must be unique.

Note – As of the Solaris 2.6 Operating Environment, the maximum value for a UID is 2147483647. However, the UIDs over 60000 do not have full functionality and are incompatible with some the Solaris Operating Environment features. So avoid using UIDs over 60000 to be compatible with earlier versions of the operating system.

- Group identification (GID) number – The unique numerical ID of the group to which the user belongs. Each GID number must be an integer between 100 to 60000.

Note – You can add a user to predefined groups of users listed in the `/etc/group` file.

- Comment – Identifies the user. Generally contains the full name of the user and optional information such as a phone number or location.
- home directory – Identifies the user's home directory pathname.
- Login Shell – Identifies the user's login shell.
- Password Aging – Optional feature to make users change their passwords on a regular basis.

Managing User Accounts with **admintool**

The administration utility, **admintool**, enables system administrators to maintain and modify local system files from the following categories:

- Users
- Groups
- Hosts
- Printers
- Serial ports
- Software

Note – You execute the **admintool** utility from the Common Desktop Environment (CDE) or OpenWindows™ environment.

To set up and manage user accounts with **admintool**, log in as **root** and run the following command from a terminal window in a CDE environment.

admintool &

The `admintool` window then displays.

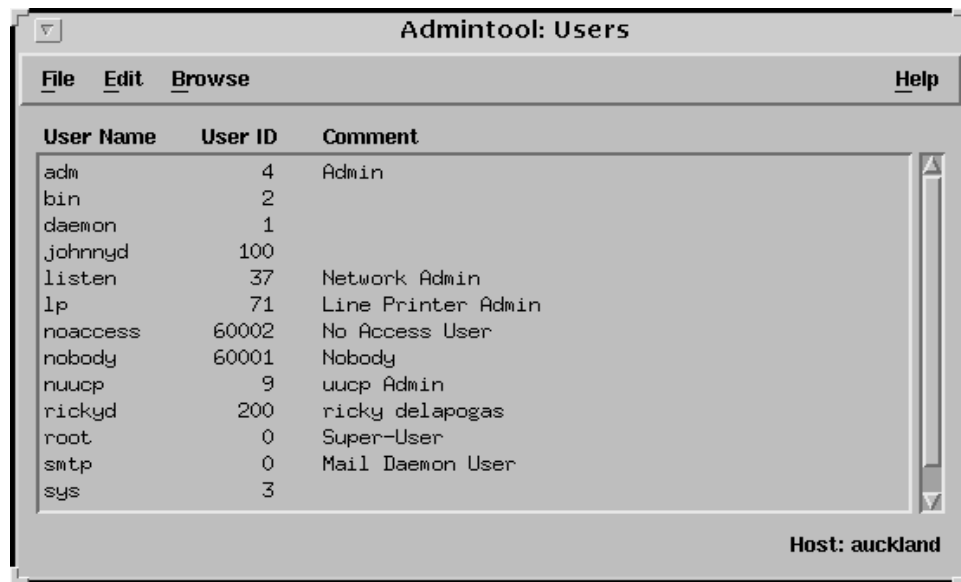


Figure 2-1 The `admintool` Users Window

The following are the general tasks required to create a new user account.

- To add a new group, select the Add Group window from the Browse menu.
- To create a user account, select the Add User window from the Browse menu and specify the new user information:
 - ▼ User name and UID
 - ▼ Primary GID
 - ▼ Secondary GID
 - ▼ Real name as a comment
 - ▼ Login shell
 - ▼ Password
 - ▼ home directory information

Creating a New Group in the /etc/group File

To add a new group to the /etc/group file:

1. From the Browse menu, select Groups.

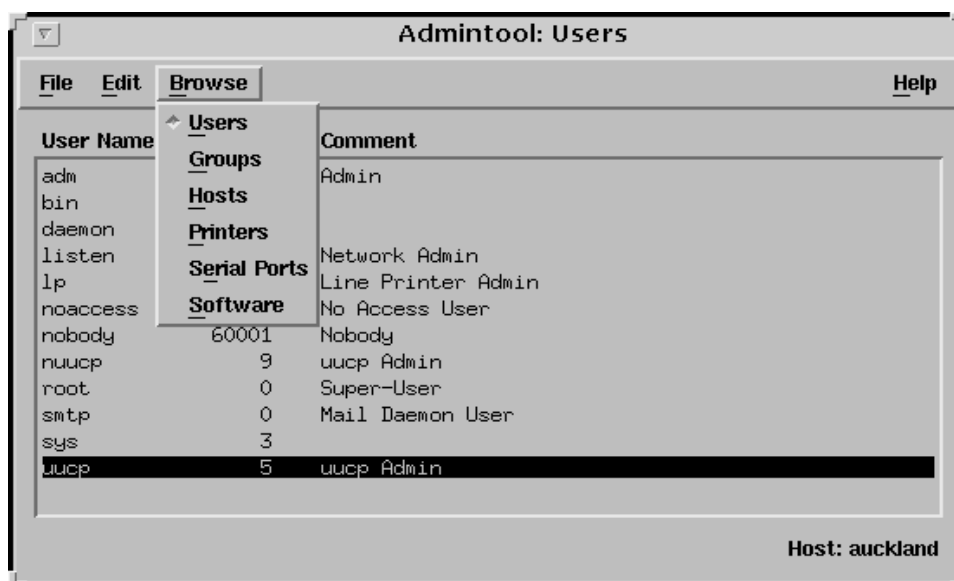


Figure 2-2 Browse Menu

The Group Database window is displayed.

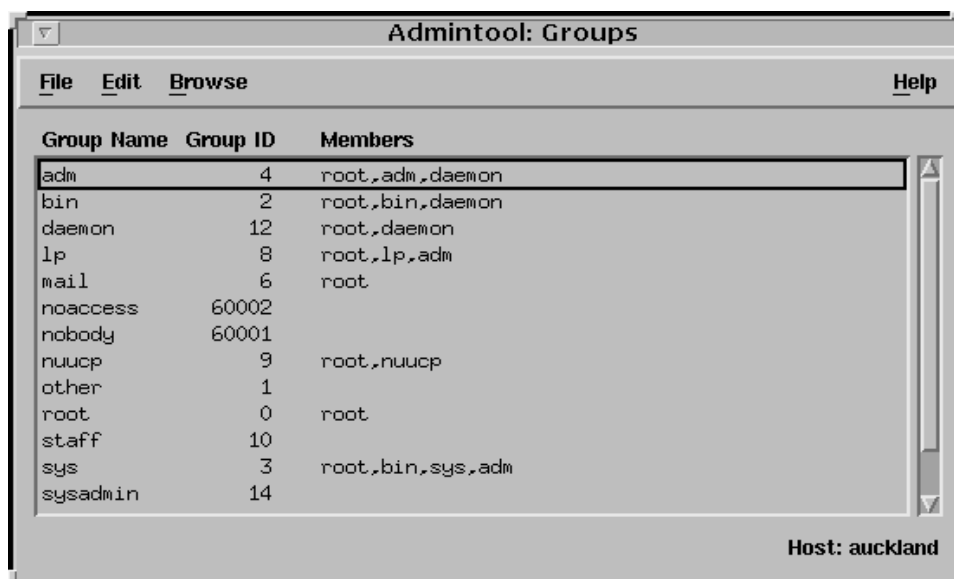


Figure 2-3 Groups Database Window

2. From the Edit menu, select Add.

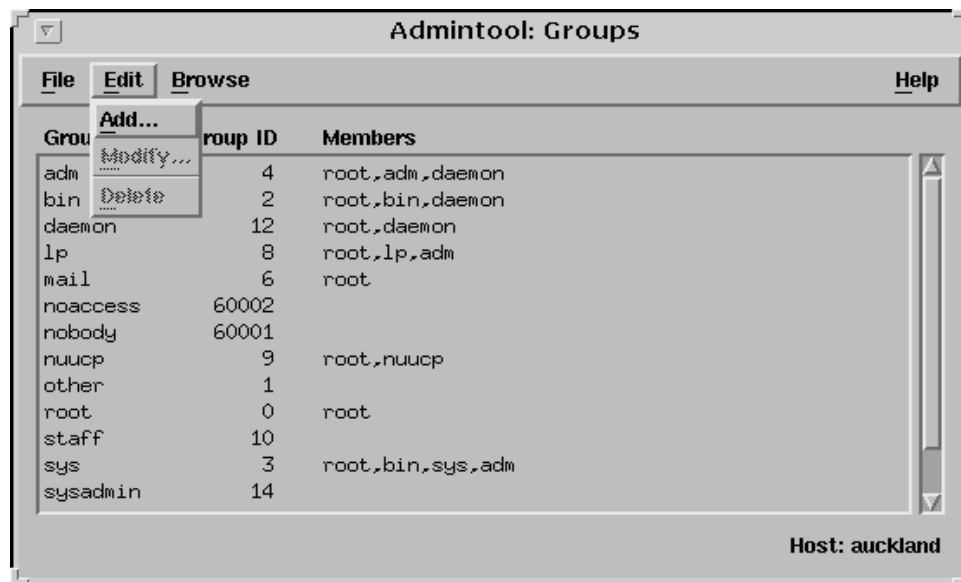


Figure 2-4 Edit Menu

The Add Group window is displayed.

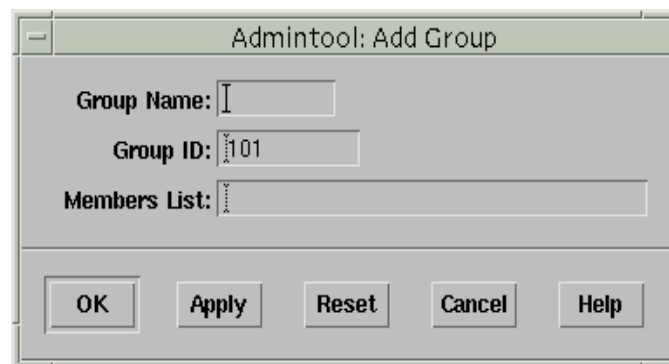


Figure 2-5 admintool Add Group Window

3. Enter the following information:

- ▼ In the Group name field, type **class**
- ▼ In the Group ID (GID) field, type **300**
- ▼ In the Members List field, add any secondary members.

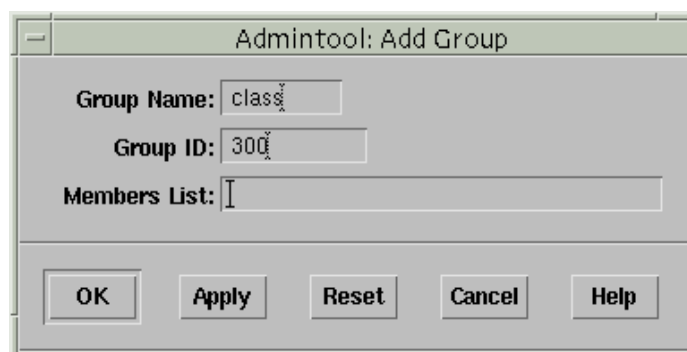


Figure 2-6 Add Group Name and ID

4. Click on OK.

Adding a New User Account

To add a new user account:

1. From the Browse menu, select Users.

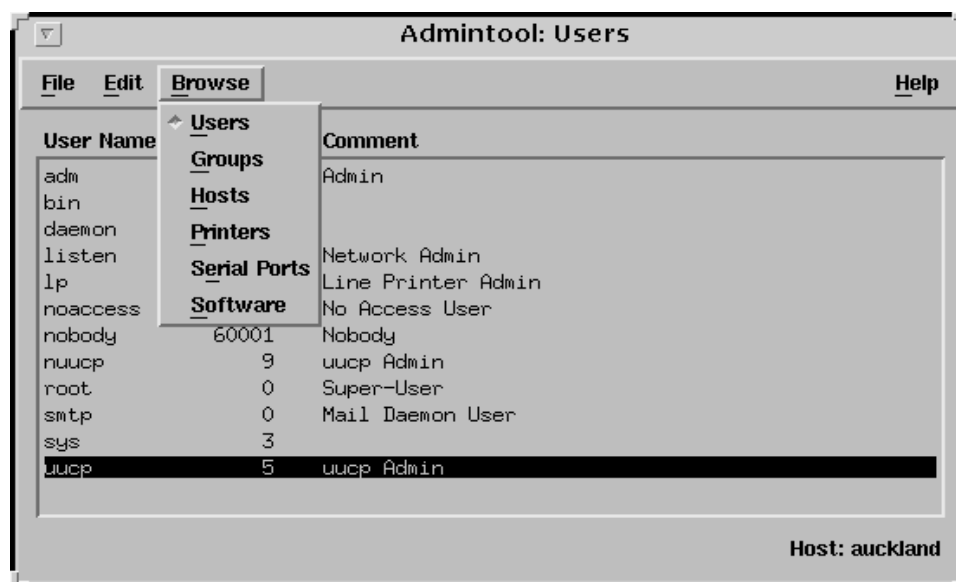


Figure 2-7 Users Window from the Browse Menu

2. From the Edit menu, select Add.

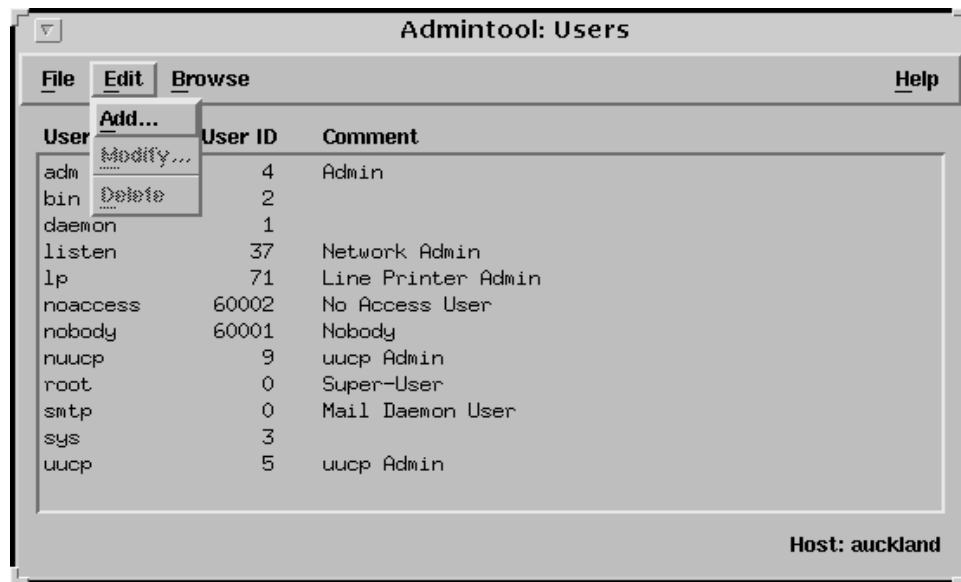


Figure 2-8 Edit Menu – Add

The Edit menu contains the following selections:

- ▼ Add – Creates a new user account.
- ▼ Modify – Allows you to view or modify an existing account.
- ▼ Delete – Deletes selected components of a user's account.

3. Specify the User Identity values for the fields listed.
 - a. In the User Name field, type your first name.
 - b. In the User ID field, type the UID provided by `admintool`.
 - c. In the Primary Group field, type **class**.
 - d. In the Secondary Groups field, specify 14 (sysadmin).

Note – The sysadmin group (GID 14) enables non-privileged users to modify system files using `admintool`.

- e. In the Comment field, type your full name.
- f. Click on the Login Shell button to specify your preferred shell.

The screenshot shows a window titled "Admintool: Add User". It is divided into three main sections: "USER IDENTITY", "ACCOUNT SECURITY", and "HOME DIRECTORY".

USER IDENTITY

- User Name:
- User ID:
- Primary Group:
- Secondary Groups:
- Comment:
- Login Shell: ☐ /bin/sh

ACCOUNT SECURITY

- Password: ☐
- Min Change: days
- Max Change: days
- Max Inactive: days
- Expiration Date: ☐ ☐ ☐
- (dd/mm/yy)
- Warning: days

HOME DIRECTORY

- Create Home Dir: ☒
- Path:

At the bottom of the window are five buttons: OK, Apply, Reset, Cancel, and Help.

Figure 2-9 User Identification Information in the Add User Window

4. To specify a user's password, select one of the available choices described in Table 2-1.

Table 2-1 Password Status Choices

Password Status	Description
Cleared until first login	Account does not have a password. The user is prompted to enter a new password at initial login (by default).
Account is locked	Account is locked. The user cannot log in until you unlock the account.
No password—setuid only	No one can log in to the account, but you can run account programs, such as <code>lp</code> or <code>uucp</code> .
Normal password	You can assign a password to the account while adding the new user.

- From the Password menu, select Normal Password. You must enter the password twice for verification. Supply a password and click on OK.

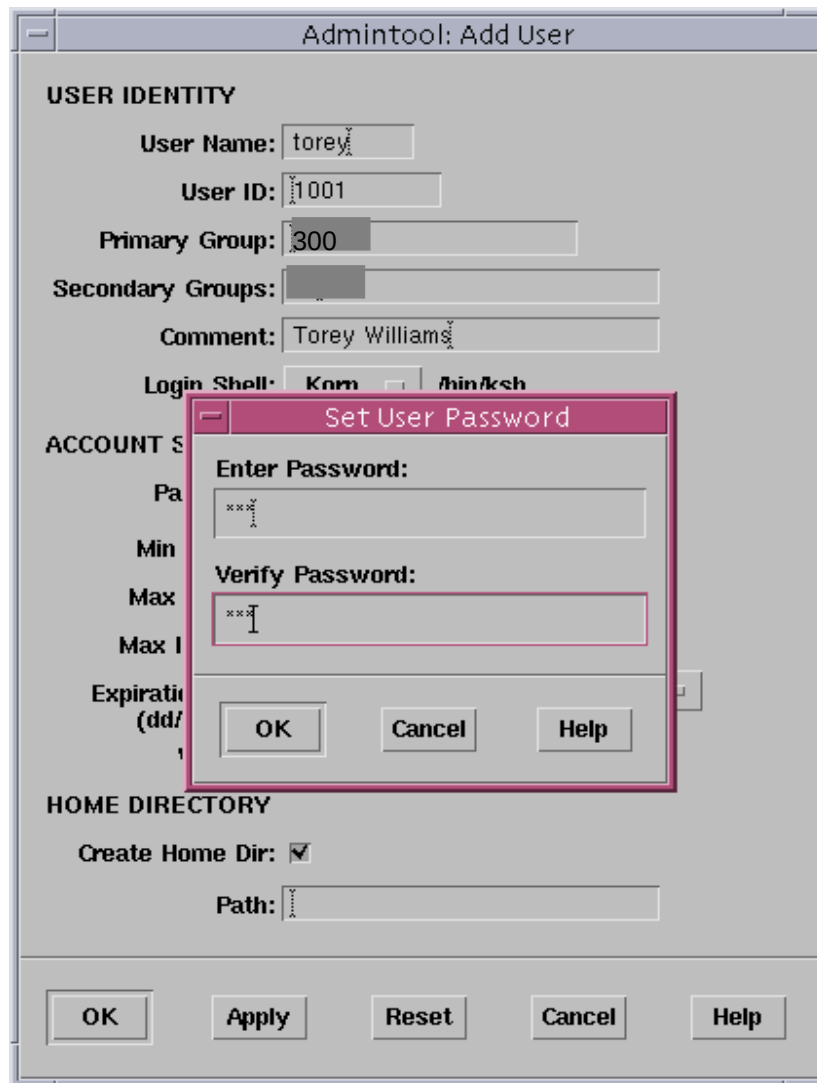


Figure 2-10 Set User's Password Window

Password Aging

Password aging features are included in the Account Security section of the Add User window.

Passwords should be changed on a regular basis to reduce unauthorized system access.

The Solaris 8 Operating Environment provides several options for managing passwords on a per-user basis. Table 2-2 describes the different password aging parameters.

Table 2-2 Password Aging Parameters

Parameter	Meaning
Min Change	The minimum number of days required between password changes
Max Change	The maximum number of days the password is valid
Max Inactive	The number of days of inactivity allowed for that user
Expiration Date	An absolute date specifying when the login can no longer be used
Warning	The number of days the user is warned before the password expires

Users receive the following message at login if they attempt to change their password before the Min Change parameter:

Sorry, less than *n* days since last change.

If users exceed the Max Change parameter, the following message is displayed:

Your password has expired. Choose a new one.

6. Specify the Account Security values for the fields listed. For example:
 - a. In the Min Change field type **7**.
 - b. In the Max Change field type **90**.
 - c. In the Max Inactive field, type **30**.
 - d. In the Expiration Date fields, select **1**, **Dec**, and **2000**.
 - e. In the Warning field, type **5**.

The window should reflect the values shown in Figure 2-11.

The screenshot shows the 'Admintool: Add User' window. It is divided into three main sections: 'USER IDENTITY', 'ACCOUNT SECURITY', and 'HOME DIRECTORY'.
 In the 'USER IDENTITY' section, the 'User Name' is 'torey', 'User ID' is '1001', 'Primary Group' is '300', 'Secondary Groups' is empty, 'Comment' is 'Torey Williams', and 'Login Shell' is 'Korn' with '/bin/ksh' selected.
 In the 'ACCOUNT SECURITY' section, 'Password' is set to 'Normal Password...', 'Min Change' is '7' days, 'Max Change' is '90' days, 'Max Inactive' is '30' days, 'Expiration Date' is '1' of 'Dec' '2000', and 'Warning' is '5' days.
 In the 'HOME DIRECTORY' section, 'Create Home Dir' is checked, and the 'Path' field is empty.
 At the bottom are buttons for 'OK', 'Apply', 'Reset', 'Cancel', and 'Help'.

Figure 2-11 Password Aging Parameters

The home Directory

7. To specify the home directory location, set the Path field to `/export/home/username`.
8. Click OK to create the new user account.

Note – `admintool` copies and renames only the `/etc/skel` initialization file(s) for the login shell selected for the new user. For example, `admintool` copies and renames only the `.profile` file for the Korn and Bourne shells and places it in the user's home directory. It copies and renames only `.cshrc` and `.login` files for C shell users.

The screenshot shows the 'Admintool: Add User' dialog box. It is divided into three main sections: USER IDENTITY, ACCOUNT SECURITY, and HOME DIRECTORY. At the bottom are buttons for OK, Apply, Reset, Cancel, and Help.

USER IDENTITY	
User Name:	torey
User ID:	1001
Primary Group:	10
Secondary Groups:	14
Comment:	Torey Williams
Login Shell:	Korn /bin/ksh

ACCOUNT SECURITY	
Password:	Normal Password...
Min Change:	7 days
Max Change:	90 days
Max Inactive:	30 days
Expiration Date:	1 Dec 2000 (dd/mm/yy)
Warning:	5 days

HOME DIRECTORY	
Create Home Dir:	☒
Path:	/export/home/torey

Figure 2-12 The home Directory Specification

Modifying a User Account

When a user no longer requires login access to the system, you should secure or delete that user's account.

To secure an account no longer in use, you can simply lock it. Once locked, no one can log in to that account; however, potentially important shared files in the home directory are still available to other users on the system.

Locking a User Account

To lock a user's account:

1. As root, launch `admintool` from a terminal window in a CDE environment, (if the utility is not already running).

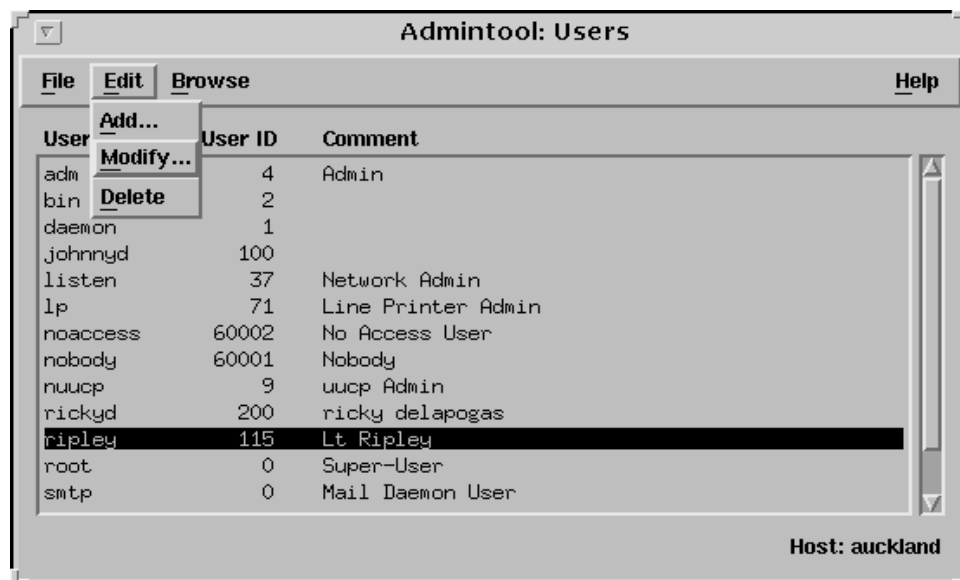


Figure 2-13 Lock a User Account Window

2. In the User Account window, select the login name of the account created earlier.
3. From the Edit menu, select Modify.

The Modify User window is displayed with the selected user's current values completed.

Figure 2-14 Modify User Window

4. From the Password menu, select Account Is Locked to lock the account.
5. Click on OK.
6. Verify that the account is locked by viewing the user account entry in the `/etc/shadow` file.

cat /etc/shadow

The locked user account should show the password field set to *LK*, which is an unmatchable password that indicates the account is locked.

Note – You can also lock a user account from the command line using the command: `passwd -l username`.

Deleting a User Account

After archiving or otherwise accounting for the user's files, you can delete the user account.

If you delete a user account, use `admintool` to delete or retain the user's home directory and its contents.

1. Select the login name of the user to delete.
2. From the Edit menu, select Delete.

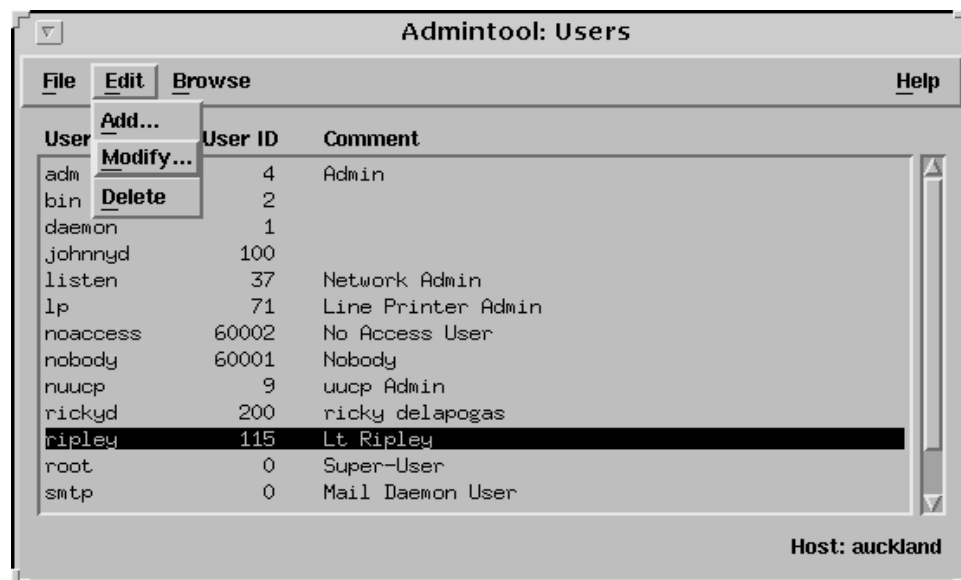


Figure 2-15 Edit Menu – Delete

The Delete dialog box is displayed.



Figure 2-16 Delete Warning Window

3. To delete the user, the user's home directory and its contents from the system, click the Delete Home Directory box and then click on Delete.

By not selecting the Delete Home Directory box, you remove only the account information for the user.

Note – Be sure to note the user's UID before removal if you intend to search the system for files owned by that user.

Files that were owned by the deleted user account are now tracked by the system by the UID number that had been assigned to that user.

You can use the `find` command to locate and remove these files, if necessary. For example:

To locate all files owned by a user, type:

```
# find / -user UID
```

To locate and remove all files owned by the user, type:

```
# find / -user UID -exec rm {} \;
```


Storing User and Group Account Information

The Solaris Operating Environment stores user account and group account information in the following system files:

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`

Authorized system users have login account entries in the `/etc/passwd` file.

All passwords are encrypted and maintained in a separate shadow file named `/etc/shadow`. To further control user passwords, you can often enforce password aging, which is maintained in the `/etc/shadow` file.

The `/etc/group` file defines the default system group accounts. You use this file to create new group accounts or modify existing group accounts on the system.

The /etc/passwd File

Due to the critical nature of the /etc/passwd file, you seldom, if ever, open this file to edit it directly. Instead, you use `admintool` or the command-line tools: `useradd`, `usermod`, and `userdel` to maintain this file.

The following is a sample /etc/passwd file, containing initial system account entries:

```
root:x:0:1: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:
smtp:x:0:0:Mail Daemon User:/:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
```

Each line entry in this file contains the following seven fields separated by colons:

loginID:x:UID:GID:comment:home_directory:login_shell

- **loginID** – Represents the user's login name. It should be unique. The field is a string of no more than eight characters consisting of alphabetic and numeric characters, period (.), underscore (_), and hyphen (-). The first character must be a letter, and it must contain at least one lowercase character.
- **x** – Represents a placeholder for the user's encrypted password, which is kept in the /etc/shadow file.
- **UID** – Contains the UID 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 60001 is reserved for the `nobody` account. UID 60002 is reserved for the `noaccess` account. Duplicate UIDs are allowed but should be avoided. If two users have the same UID, they have identical access to each users files.

- **GID** – Contains the GID 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.)
- **comment** – Contains the user's full name.
- **home_directory** – Contains the full pathname to the user's home directory.
- **login_shell** – Defines the user's login shell, which can be /bin/sh, /bin/ksh, /bin/csh, /bin/zsh, /bin/bash, or /bin/tcsh.

Default System Account Entries

Table 2-3 describes the default system account entries located in the /etc/passwd file.

Table 2-3 Default System Account Entries

User Name	User ID	Description
root	0	Superuser account. Has almost no restrictions and overrides all other logins, protections, and permissions; has access to the entire system.
daemon	1	System account that controls background processing.
bin	2	Administrative account that owns most of the commands.
sys	3	Administrative account that owns many system files.
adm	4	Administrative account that owns certain administrative files.
lp	71	Print service account that owns the object and spooled data files for the printer.
smtp	0	The smtp mailer uses the Simple Mail Transfer Protocol (SMTP) to transfer a message. SMTP is the standard mail protocol used on the Internet.

Table 2-3 Default System Account Entries (Continued)

User Name	User ID	Description
uucp	5	The uucp account that owns the object and spooled data files for the UNIX-to-UNIX copy program (UUCP).
nuucp	6	The uucp account used by remote systems to login to the host and start file transfers.
listen	37	Network listener account.
nobody	60001	Anonymous user account, assigned by an NFS server when an unauthorized <code>root</code> user makes a request. The <code>nobody</code> user account is assigned to software processes that do not need any special permissions.
noaccess	60002	Account assigned to a user or a process that needs access to a system through some application without actually logging into the system.
nobody4	65534	SunOS™ 4.0 or 4.1 version of the <code>nobody</code> account. ¹

1. The `nobody` account is used for securing NFS resources. When a user is logged in as `root` on an NFS client and attempts to access a remote file resource, the UID is changed from 0 to the UID of `nobody` (60001); `nobody` gets the same access permissions as those defined for everyone else.

The /etc/shadow File

Due to the critical nature of the /etc/shadow file, you should never edit it directly. Instead, you maintain the file's fields using `admintool` or the commands `useradd`, `usermod`, or `passwd`. The /etc/shadow file can be read only by a user with `root` permission.

The following is an example of the /etc/shadow file containing its initial system account entries:

```
root:LXeoktCoMtwZN:6445::::::
daemon:NP:6445::::::
bin:NP:6445::::::
sys:NP:6445::::::
adm:NP:6445::::::
lp:NP:6445::::::
smtp:NP:6445::::::
uucp:NP:6445::::::
nuucp:NP:6445::::::
listen:*LK*::::::
nobody:NP:6445::::::
noaccess:NP:6445::::::
nobody4:NP:6445::::::
```

Each line entry contains the following nine fields, separated by colons:

loginID:password:lastchg:min:max:warn:inactive:expire:

- `loginID` – Contains the user's login name.
- `password` – Contains a 13-character encrypted password, or the string `*LK*`, which indicates a locked account, or the string `NP`, which indicates no password.
- `lastchg` – Indicates the number of days between January 1, 1970, and the last password modification date.
- `min` – Contains the minimum number of days required between password changes.
- `max` – Contains the maximum number of days the password is valid before the user is prompted to enter a new password at login.
- `warn` – Contains the number of days the user is warned before the password expires.

- `inactive` – Contains the number of inactive days allowed for that user before the user's account is locked.
- `expire` – Contains the date when the user account expires. Once exceeded, the user can no longer log in.

The ninth field is reserved for future use, and is currently not used.

The /etc/group File

Each user must belong to a group, which is referred to as the user's primary group and specified by the GID located in the user's account entry within the /etc/passwd file.

Each user can also belong up to 15 additional groups, known as secondary groups, which are specified in /etc/group file only.

The following is a sample of the default entries in an /etc/group file.

```
# cat /etc/group
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,tty,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
sysadmin::14:lister,torey
nobody::60001:
noaccess::60002:
nogroup::65534:
#
```

Each line entry in the /etc/group file contains the following four fields, each separated by a colon character.

groupname:group-password:GID:username-list

- **groupname** – Contains the name assigned to the group. Group names can contain a maximum of eight characters.
- **group-password** – Contains an asterisk or is an empty field. This field is a relic of earlier versions of UNIX. There is no utility to set a password on a group. To place a password on a group, cut and paste an existing password from the /etc/shadow file into the /etc/group file entry.

Note – A group password is used by the `newgrp` command. This command is used to log a user into a new group. If that new group has a password, and the user is not a member of that group, the password has to be entered before `newgrp` will continue.

- **GID** – Contains the group's GID number. It must be unique on the local system and should be unique across the organization. Numbers 0 to 99, 60001, and 60002 are reserved for system group accounts. User-defined groups can range from 100 to 60000.
- **username-list** – 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.+

Creating and Managing Accounts from the Command-line

You can use the following command-line tools to add, modify, and delete user accounts and group accounts on the local system.

- `useradd` – Adds a new user account to 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 (creates) a new group account on the system
- `groupmod` – Modifies a group account on the system
- `groupdel` – Deletes a group account from the system

Creating User Accounts

You can add new user accounts on the local system using the `useradd` command. This command adds an entry for the new user into the `/etc/passwd` and `/etc/shadow` files.

The `useradd` command also automatically copies all the initialization files in the `/etc/skel` directory to the user's new `home` directory.

Command Format

```
useradd [ -u uid ][ -g gid ][ -G gid [,gid,.. ]][ -d dir ][ -m ][ -s  
shell ][ -c comment ] loginname
```

Options

You can use the following options with the `useradd` command:

- `-u uid` – Sets the unique UID for the new user.
- `-g group` – Specifies a predefined group's ID or name.
- `-G group` – Defines the new user's secondary group memberships.
- `-d dir` – Defines the full pathname for the user's `home` directory.
- `-m` – Creates the new `home` directory if it does not already exist.
- `-s shell` – Defines the full pathname for the shell program to be used as the user's login shell. If not defined, it defaults to `/bin/sh`.
- `-c comment` – Typically used to specify the user's full name and location.
- `-o` – Allows a UID to be duplicated.
- `-e expire` – Sets an expiration date on the user account. Specifies the date (mm/dd/yy) on which a user can no longer log in and access the account. The account is locked.

- `-f inactive` – Sets the number of inactive days allowed on a user account. If the account is not logged into during the specified number of days it is locked.
- `-k skel_dir` – Specifies an alternative directory location containing customized initialization files to be copied into the user's home directory. (The default is `/etc/skel`.)

Adding a User with useradd

You can use the `useradd` command to create an account for a user named `user1`, assign the UID, add the user to the group `other`, create a home directory in `/export/home`, and set the login shell for the account.

```
# useradd -u 100 -g other -d /export/home/newuser1 -m -s /bin/ksh -c  
"Regular User Account" newuser1
```

By convention, a user's login name is also the user's home directory name.

Modifying User Accounts

You can use the `usermod` command to modify the components existing in a user account.

Command Format

```
usermod [ -u uid [ -o ] ] [ -g group ] [ -G group [ , group . . . ] ] [
-d dir ] [ -m ] [ -s shell ] [ -c comment ] [ -l newlogname ] [ -f
inactive ] [ -e expire ] login
```

Options

In general, the options for the `usermod` command function the same as for the `useradd` command, with the exception of the following options:

- `-l newlogname` – Changes a user's login name for the specified user account.
- `-m` – Moves the user's home directory to the new location specified with the `-d` option.

Example

The following example changes the login name and home directory for `newuser1` to `guest1`:

```
# usermod -d /export/home/guest1 -m -l guest1 newuser1
```

Deleting User Accounts

You can use the `userdel` command to delete a user's login account from the system. This command also removes the user's home directory and all of its contents, if requested to do so.

Command Format

```
userdel [ -r ] login
```

Options

You can use the following option with the `userdel` command:

- `-r` – Removes the user's home directory from the local file system. This directory must exist.

Examples

The following example removes the login account for user `guest1`:

```
# userdel guest1
```

To request that both the user's login account and home directory be removed from the system at the same time, execute the following:

```
# userdel -r guest1
```

Adding Group Accounts

As `root`, you can create new group accounts on the local system using the `groupadd` command. This command adds an entry for the new group into the `/etc/group` file.

Command Format

```
groupadd [ -g gid [ -o ] ] groupname
```

Options

You can use the following options with the `groupadd` command:

- `-g gid` – Assigns the group ID *gid* for the new group.
- `-o` – Allows the *gid* to be duplicated.

Example

The following `groupadd` command creates the new account `class1` on the local system:

```
# groupadd -g 301 class1
```

Modifying Group Accounts

You can use the `groupmod` command to modify the definitions of the specified group by modifying the appropriate entry in the `/etc/group` file.

Command Format

```
groupmod [ -g gid [ -o ]] [ -n name ] groupname
```

Options

You can use the following options with the `groupmod` command:

- `-g gid` – Specifies the new GID for the group.
- `-o` – Allows the GID to be duplicated.
- `-n name` – Specifies the new name for the group.

Example

The following example changes the `class` account group GID to 400:

```
# groupmod -g 400 class
```

Deleting Group Accounts

You can use the `groupdel` command to delete a group account from the system. It deletes the appropriate entry from the `/etc/group` file.

Command Format

```
groupdel groupname
```

Example

The following example removes the group account `class1` from the local system.

```
# groupdel class1
```


Exercise: Adding Users and Groups



Exercise objective – In this exercise you use `admintool`, `usermod`, `userdel`, `groupadd`, `groupmod`, and `groupdel` to create, modify, and delete multiple user logins and groups.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

- Use `admintool` to create the list of groups described in step 2 of the Tasks. Add the users described in step 3 of the Tasks. Verify the shells you specify in `admintool` are set in `/etc/passwd`. In `/etc/shadow`, are the password strings for users with the same password also the same? What are the password strings for the users `locked1`, `cleared1`, and `nopass1`? Verify the users `user3` and `user4` are secondary members of the `class1` group.
- Can you log in as the user `locked1`? What happens when you try to log in as the user `cleared1`? Record the password requirements indicated. Can the user `root` use `su` to become the user `cleared1`?
- Establish password aging for the user `user5` as indicated in step 10. What happens when you attempt to log in as that user? When logged in as `user5`, can you change the password from the command line? Log in as `root` when finished.
- Use `groupadd` to add a group called `class3`. Use `usermod` to change the UID number, group, and user name for `locked1`. Verify that the changes exist in `/etc/passwd`. Use `userdel` to delete the user `cleared1`. Verify that the home directory has been deleted. Use `groupmod` to rename `class1` to `group1`. Use `userdel` to remove the group `class2`. Verify the changes to `/etc/group`.

Tasks

1. Log in as the user `root` and open a terminal window. Run `admintool`. It automatically displays the list of users.

admintool &

2. From the Browse menu, select Groups. From the Edit menu, select Add. Create two new groups with the following names and GID numbers. Click on Apply after specifying the information for the first group. Click on OK when you have entered the information for the last group.

Group Name	GID
class1	101
class2	102

3. From the Browse menu, select Users. From the Edit menu, select Add. Use this panel to create the following list of users. Click on Apply after specifying the information for each user. Click on OK when you have entered the information for the last user.

For all users, use `/export/home` as the root portion of the home directory. Use the user name as the last part of the path; for example: `/export/home/user3`, or `/export/home/nopass1`. Choose to create the home directory. Do not use password aging. Exit `admintool` when finished.

User Name	Password	Shell	UID	Primary Group	Secondary Group
user3	cangetin	Korn	1003	10	101
user4	cangetin	C	1004	10	class1
user5	cangetin	Bourne	1005	10	
locked1	Select "Account is Locked"	Korn	2001	10	
cleared1	Select "Cleared until first login"	Korn	2002	10	
nopass1	Select "No Password"	Korn	2003	10	

- Examine the content of the `/etc/passwd` file. What are the full pathnames of the shells used by user3 through user5?
user3 _____
user4 _____
user5 _____
- Examine the content of the `/etc/shadow` file. What text is found in the password field for the users `locked1`, `cleared1`, and `nopass1`?
locked1 _____
cleared1 _____
nopass1 _____
- You used the same password for user3 through user5. Are the password strings the same in `/etc/shadow`?

7. Examine the content of the `/etc/group` file. Verify `user3` and `user4` are both listed as secondary members of the `class1` group. Are they?

8. Log out of CDE and attempt to log in as `locked1`. Are you able to log in?

9. 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` once you establish a password.

10. Log out of CDE and attempt to log in as `nopass1`. Are you able to log in?

11. Log in as `root`. Open a terminal window. Can you change your user identity to `nopass1` using `su`? Exit the `su` session if it is successful.

su nopass1

\$ exit

#

12. Run `admintool`. Select `user5` from the list of users. Select the Modify item from the Edit menu. Change the password aging information for `user5` so that it matches the information below. Click on OK when complete and exit `admintool`.

Min Change: 1 day

Max Change 2 days

Max Inactive: 1 day

Expiration Date: (tomorrow's date)

Warning: 2 days

13. Log out of your `root` login session. Attempt to log in as `user5`. What happens? Supply a new password if required.
-

14. Complete the login as `user5`. Open a terminal window and attempt to change the password you just set. What happens?
-

15. Log out and log in again as `root`.

16. Use `groupadd` to create a new group called `class3` that uses GID number 103. For example:

```
# groupadd -g 103 class3
```

17. Use `usermod` to change the UID number, group, and login name of `locked1` as follows. Verify the changes you request are recorded in `/etc/passwd`.

```
# usermod -u 3001 -g 103 -l test1 locked1
```

18. Use `userdel` to delete `cleared1` and their home directory. Verify `/export/home/cleared1` no longer exists.

```
# userdel -r cleared1
```

19. Use `groupmod` to change the group name of `class1` to `group1`.

```
# groupmod -n group1 class1
```

20. Use `groupdel` to remove the group `class2`.

```
# groupdel class2
```

21. Verify the commands in Steps 19 and 20 have correctly modified the `/etc/group` file.

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

4. Examine the content of the `/etc/passwd` file. What are the full pathnames of the shells used by `user3` through `user5`?

```
user3          /bin/ksh
user4          /bin/csh
user5          /bin/sh
```

5. Examine the content of the `/etc/shadow` file. What text is found in the password field for `locked1`, `cleared1`, and `nopass1`?

```
locked1        *LK*
cleared1       none
nopass1        NP
```

6. You used the same password for `user3` through `user5`. Are the password strings the same in `/etc/shadow`?

No

7. Examine the content of the `/etc/group` file. Verify `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.

8. Log out of CDE and attempt to log in as `locked1`. Are you able to log in?

No

9. Attempt to log in as `cleared1`. What happens? Attempt to use the password `abcdefg`. What are the system requirements for the 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 and at least one numeric or special characters.

10. Log out of CDE and attempt to log in as `nopass1`. Are you able to log in?

No

11. Log in as `root`. Open a terminal window. Can you change your user identity to `nopass1` using `su`? Exit the `su` session if it is successful.

Yes

13. Log out of your `root` login session. Attempt to log in as `user5`. What happens?

You must supply a new password before you can log in.

14. 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 your password will expire in 2 days.

When you try to change your password, the following error message displays:

`passwd(SYSTEM): Sorry: less than 1 days since the last change. Permission denied`

17. Use `usermod` to change the UID number, group, and login name of `locked1` as follows. Verify the changes you request are recorded in `/etc/passwd`.

`/etc/passwd` should reflect the new UID number, group, and user name.

18. Use `userdel` to delete the user `cleared1` and their home directory. Verify `/export/home/cleared1` no longer exists.

`/export/home/cleared1` should no longer exist.

21. Verify the commands in steps 19 and 20 have correctly modified the `/etc/group` file.

The group `group1` should exist, `class1` and `class2` should not.

Understanding 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.

System-Wide Initialization Files

You maintain the system initialization files to provide an environment for the entire community of users who log in to the system. These files are provided by the Solaris Operating Environment and reside in the `/etc` directory.

The two main system initialization files are called `/etc/profile` and `/etc/.login`.

The Bourne and Korn 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 file `.hushlogin` exists in the user's home directory.

User Initialization Files

You set up the user's initialization files and place them in each user's home directory.

The primary job of a user initialization file is to define the characteristics of a user's work environment, such as a user's search path, environment variables, and windowing environment.

The owner(s) of the file(s) or `root` can change or customize the content of these files.

Table 2-4 defines the initialization files for the six possible shells in the Solaris 8 Operating Environment.

Table 2-4 Initialization Files for the Six Shells

Shells	System-wide Initialization Files	User Initialization Files Read at Login	User Initialization Files Read When a New Shell Is Started After Login	Shell Pathname
Bourne	/etc/profile	\$HOME/.profile		/bin/sh
Korn	/etc/profile	\$HOME/.profile \$HOME/.kshrc	\$HOME/.kshrc	/bin/ksh
C	/etc/.login	\$HOME/.cshrc \$HOME/.login	\$HOME/.cshrc	/bin/csh
Z	/etc/zshenv /etc/zprofile /etc/zshrc /etc/zlogin	\$HOME/.zshenv \$HOME/.zprofile \$HOME/.zlogin	\$HOME/.zshrc	/bin/zsh
BASH	/etc/profile	\$HOME/.bash_profile \$HOME/.bash_login \$HOME/.profile	\$HOME/.bashrc	/bin/bash
TC	/etc/csh.cshrc /etc/csh.login	\$HOME/.cshrc or \$HOME/.login	\$HOME/.tcshrc or \$HOME/.cshrc	/bin/tcsh

Note – The root user's login shell by default is the Bourne shell, and root's shell entry in the /etc/passwd file appears as /sbin/sh.

When a user logs in to the system, the user's login shell is invoked. The shell program looks for its initialization files, in a specific order; executes the commands contained in each file, and when finished, displays the shell prompt on the user's screen.

Customizing the Work Environment

The shells all provide basic features and a set of variables that determine what `root` or a regular user can do when customizing user initialization files for each shell.

Shell Variables

The environment maintained by the shell includes variables that are defined by the login program, system initialization file, and the user initialization files.

The shells support two types of variables:

- Environment variables – Every shell program started receives its information about the user's environment from these variables.
- Local variables – This affects only the current shell. Any subshell started would not have knowledge of these variables.

Table 2-5 lists some of the variables available for customizing a user's shell environment.

Table 2-5 Shell Variables

Variable Name	Set By	Description
LOGNAME	Set by login	Defines the user's login name.
HOME	Set by login	Sets the path to the user's home directory. Default argument for <code>cd</code> .
SHELL	Set by login	Sets the path to the default shell.
PATH	Set by login	Sets the default path the shell searches to find commands.
MAIL	Set by login	Sets the path to the user's mailbox.
TERM	Not set by default	Defines the terminal.
LPDEST	Not set by default	Sets the user's default printer.

Table 2-5 Shell Variables (Continued)

Variable Name	Set By	Description
PWD	Set by shell	Defines the current working directory.
PS1	Set by shell	Defines the shell prompt for the Bourne or Korn shell.
prompt	Set by 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).

Setting Environment Variables in User Initialization Files

A user can change the values of the predefined variables and specify additional variables.

Table 2-6 demonstrates how to set environment variables in user initialization files.

Table 2-6 Setting Environment Variables

Shell	User's Initialization File
Bourne or Korn Shell	VARIABLE=value ; export VARIABLE For example: PS1="\$HOSTNAME ! \$ " ; export PS1
C Shell	setenv variable value For example: setenv prompt "\! 'uname -n' % "

Using the Initialization File Templates

The Solaris Operating Environment provides you with a set of initialization file templates.

The initialization file templates are located in the `/etc/skel` directory and are defined in Table 2-7.

Table 2-7 Default User Initialization Files

Shell	Initialization File Templates	User's Initialization Files
Bourne	<code>/etc/skel/local.profile</code>	<code>\$HOME/.profile</code>
Korn	<code>/etc/skel/local.profile</code>	<code>\$HOME/.profile</code>
C	<code>/etc/skel/local.login</code>	<code>\$HOME/.login</code>
	<code>/etc/skel/local.cshrc</code>	<code>\$HOME/.cshrc</code>

The `root` user can customize these templates to create a standard set of user initialization files to provide a common work environment for each user.

User's can then edit their initialization files to further customize their environments for each shell.

When new user accounts are created by `root`, these initialization files are automatically copied to each new user's home directory.

Exercise: Modifying Initialization Files



Exercise objective – In this exercise you modify templates for initialization files in `/etc/skel` and create users who use them.

Preparation

This exercise requires the skills practiced in the previous exercise. The users 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.

Task Summary

- Edit `/etc/skel/local.profile` so that it sets the `PATH` variable to the same paths as used by the `root` user. Set the `EDITOR`, `LPDEST`, `EXINIT`, and `ENV` variables to appropriate values.
- Use `admintool` to create a new user called `user9` who uses the Korn shell. Log in as the new user and verify all the variables you set in `local.profile` are set correctly in the user's environment.
- Create a `.kshrc` file for the new user 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 `.kshrc` works. Log out and log in again as `root`.
- Use `useradd` to create a new user called `user10` that uses the Korn shell. Log in as this user and record the list of initialization files in your `home` directory. Copy the appropriate file to `.profile`. Test the login to verify the same list of variables is set as with the first user you created. Log out and log in as `root` when finished.

Tasks

1. Log in as `root` and open a terminal window.

2. Change directory to `/etc/skel`.

```
# cd /etc/skel
```

3. Use `vi` to edit the `local.profile` file and make the following changes.

```
# vi local.profile
```

- a. Edit the line that declares the `PATH` variable so it reads as follows. Enter this text as one line, (no spaces).

```
PATH=/usr/sbin:/sbin:/usr/dt/bin:/usr/openwin/bin:/bin:/usr/b  
in:/usr/ucb:/etc:.
```

- 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 `admintool` to create a new user with the following characteristics. Exit `admintool` when finished.

User Name:	user9
User ID:	1009
Primary Group:	10
Login Shell:	Korn
Password:	cangetin
Home directory (create it);	/export/home/user9

5. Log out and log in again as `user9`. Select CDE. Open a terminal window.

6. Verify the PATH, LPDEST, EDITOR, EXINIT, and ENV variables are set according to the changes you made in /etc/skel/local.profile.

```
$ echo $PATH
$ echo $LPDEST
$ echo $EDITOR
$ echo $EXINIT
$ echo $ENV
```

Do they match? _____

7. Create a file called .kshrc in your 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='$PWD$ '
```

8. Log out and then log in again as user9. Open a terminal window and verify your new variables work.

```
$ cd /tmp
$ cd
$ c
$ h
```

Do they work? _____

9. Log out and log in again as root. Use useradd to create a new user called user10. Assign user10 the password cangetin.

```
# useradd -u 1010 -g 10 -d /export/home/user10 -m -s
/bin/ksh -c "SA-238 Student" user10
6 blocks
# passwd user10
New password: cangetin
Re-enter new password: cangetin
```


10. Log out and log in again as `user10`. Select CDE. Open a terminal window. What shell initialization files exist in your home directory?

```
$ ls -la
```

Which of these are the same as `/etc/skel/local.profile`?

11. Copy `local.profile` to `.profile`.

```
$ cp local.profile .profile
```

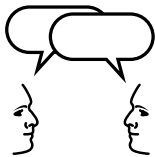
12. Log out and log in again as `user10`. Verify the variables that were set for the `user9` login are also set for this login.

```
$ echo $PATH
$ echo $LPDEST
$ echo $EDITOR
$ echo $XINIT
$ echo $ENV
```

Do they match? _____

13. Log out and log in again as `root`.

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

1. Verify the PATH, EDITOR, LPDEST, EXINIT, and ENV variables are set according to the changes you made in the file `/etc/skel/local.profile`.

Do they match?

These variables should match the settings made in the file `local.profile`.

8. Log out and then log in again as `user9`. Open a terminal window and verify that your new variables work.

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.

10. Log out and log in again as `user10`. Select CDE. Open a terminal window. What shell initialization files exist in your home directory?

`.profile`, `local.profile`, `local.login`, `local.cshrc`

Which of these is the same as `/etc/skel/local.profile`?

`local.profile`

12. Log out and log in again as `user10`. Verify the variables set in the login for `user9` are also set in this login.

Do they match?

These variables should match the settings made in the file `local.profile`.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Create and manage user accounts on the local system using the `admintool` utility
- ☐ Describe the format of the files `/etc/passwd` and `/etc/shadow` for securing login access
- ☐ Describe the format of the `/etc/group` file for maintaining shared and restricted access to files and directories
- ☐ Add, modify, and delete user accounts on the local system with the commands `useradd`, `usermod`, and `userdel`
- ☐ Add, modify, and delete group accounts for the local system with the commands `groupadd`, `groupmod`, and `groupdel`
- ☐ Define the two different types of shell initialization files
- ☐ Describe the shell startup activities during login for the three main Solaris Operating Environment shells
- ☐ List the shell initialization files used to set up a user's work environment at login
- ☐ Describe the purpose of the `/etc/skel` directory
- ☐ Modify initialization files to customize a user's work environment

Objectives

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

- Create the `/var/adm/loginlog` file to save failed login attempts
- Monitor system usage with the commands `finger`, `last`, and `rusers`
- Use the `su` command to become the `root` user or another user on the system
- Modify the `/etc/default/login` file to restrict `root` access
- Use the commands `id` and `groups` to identify users and their group memberships
- Change a file's owner or a file's group using the commands `chown` and `chgrp`, respectively
- Explain how the special permissions `setuid`, `setgid`, and the Sticky Bit can affect system security
- Create, modify, and delete access control lists (ACLs) on files
- Control remote login access by maintaining three basic network files: `/etc/hosts.equiv`, `$HOME/.rhosts`, and `/etc/ftpusers`

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10

Managing System Security Overview

Two important responsibilities of the system administrator are controlling access and securing data on a system. The Solaris Operating Environment provides some standard security features for controlling access by unauthorized users and for protecting files on local and remote systems.

Some basic steps that you should take to manage security at the user, file, system, and network level include:

- Maintaining password and login control
- Monitoring system usage
- Restricting access to data contained in files
- Tracking root logins
- Monitoring `setuid` programs
- Controlling remote access on the network

Managing Login and Access Control

All accounts on the system must have a password. Any account without a password allows unauthorized access to the local host and to the entire network.

The pwconv Command

The `pwconv` command creates and updates the `/etc/shadow` file with information from the `/etc/passwd` file.

It is the `pwconv` command that relies on the special value of 'x' in the password field of `/etc/passwd`. The 'x' indicates that the password for the user already exists in the `/etc/shadow` file.

If the `/etc/shadow` file does not exist, `pwconv` creates it with the information from `/etc/passwd`.

If the `/etc/shadow` file does exist, the following tasks are performed:

- Entries that are in the `/etc/passwd` file and not in the `/etc/shadow` file are added to the shadow file.
- Entries that are in the `/etc/shadow` file and not in the `/etc/passwd` file are removed from the shadow file.

Recording Failed Login Attempts

When a user logs in to a system, locally or remotely, from the command line only, the `login` program consults the `/etc/passwd` and `/etc/shadow` file to authenticate the user by verifying the user name and password entered.

If the user provides a login ID name from the `/etc/passwd` file and the correct password for that login name, the `login` program grants access to the system.

If the user name is not in the `/etc/passwd` file or the password is not correct for the user name, the `login` program denies access to the system.

You can save failed login attempts to a file, which is a useful tool for determining if attempts are being made to break into a system.

You can record failed login attempts can be recorded in the file `/var/adm/loginlog`.

By default, the `loginlog` file does not exist. To enable logging, you must create this file with read and write permissions for `root` only.

```
# touch /var/adm/loginlog
```

All failed login activity is written to this file automatically after five failed attempts.

The `loginlog` contains one entry for each of the failed attempts. Each entry contains the user's login name, TTY device, and time of the failed attempt.

If there are fewer than five failed attempts, no activity is logged to this file.

Monitoring System Access

All systems should be monitored routinely for unauthorized user access. Use the `who` command to see who is on the system. It looks in the `/var/adm/utmpx` file to obtain this information.

The `who` command displays a list of users currently logged on to the local system, with their login name, login device (TTY port), login date and time, and the elapsed time since last activity. If a user is logged on remotely, the remote hostname for that user is displayed.

Displaying Users on the System

To display the users who are currently on the system, execute the `who` command:

```
# who
user2    console    May 24    10:17    (:0)
user5    pts/3        May 24    17:36    (:0.0)
user9    pts/7        May 24    08:21    (:0.0)
#
```

Login Device Types

The second field displayed by the `who` command defines the user's login device, which can be 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. Remote logins are represented by this type of device.
- `term` – A device physically connected to a serial port, such as a terminal or a modem.

Displaying User Information

To display detailed information about users either locally or remotely, use the `finger` command.

Command Format

```
finger -m username
```

```
finger -m username@remotehostname
```

`-m` – Match arguments only on *username* (not first or last name).

The `finger` command displays the user's login name, home directory path, login time, login device name, data contained in the comment field of the `/etc/passwd` file (usually the user's full name), login shell, and the name of the host if logged in remotely.

Displaying User Information

To display user information, execute the following:

```
# finger user9  
Login name: user9           In real life: user9's Account  
Directory: /home/user9      Shell: /bin/ksh  
On since Apr 14 08:57:37 on console from :0  
No unread mail  
No Plan.
```

If a user creates the standard ASCII files `.plan` or `.projects` 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--`).

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. It looks in the `/var/adm/wtmpx` file, which records all logins and logouts.

Each entry includes user name, the login device, host logged in from, date and time logged in, time of log out, and total login time in hours and minutes, including entries for system reboot times.

The following is an example of the `last` command:

```
# last
user1 pts/4 host1 Fri Dec 18 10:24 - 11:00 (00:36)
user9 pts/7 host1 Tue Dec 8 09:39 - 09:49 (00:10)
user5 pts/12 host1 Thu Dec 3 15:16 - 15:18 (00:02)
reboot system boot Wed Dec 2 08:44
root console :0 Tue Dec 1 15:12 - 15:12 (00:00)
user8 pts/3 host1 Tue Dec 1 16:13 - 16:39 (00:26)
```

The `last` command can also display information about an individual user, for example:

```
# last user9
user9 pts/7 host1 Tue Dec 8 09:39 - 09:49 (00:10)
```

To view system reboot times only, execute the following command:

```
# last reboot
reboot system boot Fri Feb 11 10:15
reboot system boot Wed Jan 26 14:58
reboot system boot Mon Jan 3 16:30
```

Displaying Users on Remote Systems

The *rusers* command produces output similar to the *who* command, but displays users logged in on remote hosts. The list is displayed in the order the responses are received from the hosts — displaying the user's name and the host's name.

A remote host responds only to the *rusers* command, if its *rpc.rusersd* daemon is enabled. It is the network server daemon that returns the list of users on the remote hosts.

Command Format

```
rusers [ -l ]
```

The *rusers -l* command displays a list of login names of users who are logged in on remote systems, along with the name of the system a user is logged into, the TTY port (login device), the month, date, login time, and idle time. If the user is not idle, no time is displayed in the last field.

For example:

```
# rusers -l
user8      remotehost1:pts/4      Feb 22 11:48      27  (:0)
root       remotehost1:console    Feb 22 09:31      28:10 (:0)
user4      remotehost5:pts/12     Feb 22 8:00       1:43 (:0)
user6      remotehost2:console    Feb 22 13:41      9  (:0)
```

Accessing root Privileges

As the system administrator you should log in only to the `root` account to perform administration tasks. You should avoid performing routine work as `root`.

This helps protect the system from unauthorized access, as it reduces the likelihood that the system will be left unattended with `root` logged in. Also, critical mistakes are less likely to occur if routine work is done as a regular system user.

You can become `root` on a system by either:

- Logging in directly as `root`, and supplying the `root` password.
- Logging in as an regular user, then invoke the `su` command and supply the `root` password.

You should log in under a regular user account, then become `root` by using the `su` command, to access system files or run administration commands.

Using the su Command to Become Another User

The `su` command allows a user to become another user without logging off the system.

Command Format

```
su [ - ] [ username ]
```

To use `su`, you must supply the appropriate password unless the user is already `root`. The `root` user can run `su` without passwords.

If the password is correct, `su` creates a new shell process, as specified in the shell field of that user's `/etc/passwd` file entry.

The `su -` (dash) option specifies a complete login. It changes the user's work environment to what would be expected if the user had logged in directly as that specified user.

Effective User ID and Effective Group ID

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 switched user, rather than the UID and GID of the user who originally logged in to the system.

Note – This is important because file and directory access is determined based on the value of the EUID and EGID of the user that you have become.

Using the `whoami` Command

The `whoami` command displays the switched user's effective current user ID.

Displaying the Effective Current Username

For example, `user1` is logged into the system under that login name. This user then runs the `su` command to become `root` and enters the `root` password. The `whoami` command displays the user's effective user ID.

```
$ su
password:                (type in the root password)
# whoami
root
#
```

Using the su Command to Become root

To use the `su` command to become `root`:

1. Log in directly (from the login window) as a regular user. For example: `user1`
2. At the shell prompt, in a terminal window, type `su` and press Return. Type the `root` password and press Return.

```
$ su
Password:
```

3. To display the original login, type the command `who am i` and press Return.

```
# who am i
user1          pts/11          Apr 25 15:45      (:0.0)
```

4. To determine the login name of the user switched to, type `whoami` and press Return.

```
# whoami
root
```

5. To determine where the user is currently located, type `pwd` and press Return. The location is the original user's home directory.

```
# pwd
```

6. To exit the `root` session and return to the original user, type `exit` and press Return.

```
# exit
$
```

In the default system configuration, `root` login is restricted to the console. This means that you cannot remotely log in to a system as `root`. To remotely log in to a host, you must log in as a regular user and then run the `su` command to become `root`.

Using the su Command to Become Another Regular User

To switch to another user and have that user's environment:

1. At the shell prompt, type `su` with the dash (-) option, the name of the user to become, and press Return. Type the password for the user account and press Return. For example:

```
$ su - user2  
Password:
```

2. Determine the login name of the user switched to by typing `whoami` and pressing Return.

```
$ whoami  
user2
```

3. Determine where the user is located, type `pwd` and press Return. The location is the new user's home directory.

```
$ pwd
```

4. Display the login name of the user originally logged in as by typing `who am i` and pressing Return.

```
$ who am i  
user1 pts/4 Apr 25 15:55 (:0.0)
```

5. To return to the original user status and home directory, type the following command and press Return.

```
$ exit  
#
```

The sysadmin Group

Any user who is a member of the `sysadmin` group (GID 14) can run `admintool` for the purpose of managing local system files and functions, such as adding and removing users, groups, software, printers, and serial devices.

If you have not added any user to this group then only `root` can run the `admintool` utility.

Note – Members of the `sysadmin` group can also invoke Solstice Adminsuite™, a Solaris Operating Environment server product used to locally or remotely manage important system files and functions.

Managing User Access

Located in the `/etc/default` directory are three system files `root` can modify to monitor who is using the `su` command; restrict `root` access; and set up system-wide password aging for every user who logs in to the system.

- The `/etc/default/su` file controls how `su` attempts are logged.
- The `/etc/default/login` file can be set to restrict `root` access.
- The `/etc/default/passwd` file can be set up to enforce system-wide password aging.

Monitoring su Attempts

For security reasons, you must monitor who has been using the su command, especially those user's who are trying to gain root access on the system. You can set this using the /etc/default/su file.

The following is the content of the /etc/default/su file.

```
#ident  "@(#)su.dfl      1.6      93/08/14 SMI"    /* SVr4.0 1.2 */

# SLOG determines the location of the file used to log all su attempts
#
SLOG=/var/adm/sulog

# CONSOLE determines whether attempts to su to root should be logged
# to the named device
#
#CONSOLE=/dev/console

# PATH sets the initial shell PATH variable
#
#PATH=/usr/bin:

# SUPATH sets the initial shell PATH variable for root
#
#SUPATH=/usr/sbin:/usr/bin

# SYSLOG determines whether the syslog(3) LOG_AUTH facility should be
# used to log all su attempts. LOG_NOTICE messages are generated for
# su's to root, LOG_INFO messages are generated for su's to other
# users, and LOG_CRIT messages are generated for failed su attempts.
#
SYSLOG=YES
```

The CONSOLE Variable

The CONSOLE variable, by default, is ignored because of the preceding comment (#) symbol. Therefore, all su attempts are logged to the console regardless of success or failure.

```
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
```

By removing the comment symbol, the value of the `CONSOLE` variable is defined for `/dev/console` and all successful `SU` attempts to become `root` are logged to the console. The `/var/adm/su log` file contains only unsuccessful attempts.

```
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 SLOG Variable

The `SLOG` variable specifies the name of the file in which all `SU` attempts to switch to another user are logged. If undefined, `SU` logging is turned off.

The entries in this file include the date and time the command was issued, whether it was successful (shown by the `+` symbol for success or the `-` symbol for failure), the device from which the command was issued, and finally the name of the user and the switched identity.

For example:

```
# more /var/adm/su log
SU 10/20 14:50 + console root-sys
SU 10/20 16:55 + pts/2 user3-root
SU 11/05 11:21 - pts/3 root-user1
```

Restricting root Access

The `/etc/default/login` file gives you the ability to protect the root account on a system by restricting root access to a specific device.

The following shows the content of the `/etc/default/login` file.

```
#ident "@(#)login.dfl 1.8      96/10/18 SMI"    /* SVr4.0 1.1.1.1 */
#
# Set the TZ environment variable of the shell.
#TIMEZONE=EST5EDT
#
# ULIMIT sets the file size limit for the login.  Units are disk blocks.
# The default of zero means no limit.
#ULIMIT=0
#
# 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
#
# ALTSHELL determines if the SHELL environment variable should be set
ALTSHELL=YES
#
# PATH sets the initial shell PATH variable
#PATH=/usr/bin:
#
# SUPATH sets the initial shell PATH variable for root
#SUPATH=/usr/sbin:/usr/bin
#
# TIMEOUT sets the number of seconds (between 0 and 900) to wait before
# abandoning a login session.
#TIMEOUT=300
#
# UMASK sets the initial shell file creation mode mask. See umask(1).
#UMASK=022
#
# SYSLOG determines whether the syslog(3) LOG_AUTH facility should be
# used to log all root logins at level LOG_NOTICE and multiple failed
# login attempts at LOG_CRIT.
SYSLOG=YES
```

The CONSOLE Variable

You can set the `CONSOLE` variable to specify one of three possible conditions for restricting `root` logins:

- If the variable is defined as `CONSOLE=/dev/console`, `root` can login only at the system console. Any attempt to login as `root` from any other device generates the error message:


```
# rlogin host1
Not on system console
Connection closed.
```
- If the variable is not defined, `root` can log in to the system from any device either across the network, through a modem, or using an attached terminal.
- If the variable does not have a value assigned to it (for example `CONSOLE=`) then `root` cannot log in from anywhere, not even the console. The only way to become `root` on the system is to log in as a regular user and become `root` using the `su` command.

Implementing System-Wide Password Aging

You can force every user on the system to change their password on a regular basis, without having to set up individual password aging for each user in the `/etc/shadow` file.

This is done by modifying the `/etc/default/passwd` file. There are three different variables in the file: `MAXWEEKS`, `MINWEEKS`, and `PASSLENGTH`, as shown in the following sample file.

```
# cat passwd
#ident "@(#)passwd.dfl      1.3      92/07/14 SMI"
MAXWEEKS=
MINWEEKS=
PASSLENGTH=6
```

The /etc/default/passwd File Variables

The following sections describe the `/etc/default/passwd` file variables.

The MAXWEEKS Variable

The value set for the `MAXWEEKS` variable specifies the maximum number of weeks (seven-day weeks) a password is valid before it must be changed for all regular users.

If there is no value set for this variable, which is the default setting, only users who have a value for `Max Change` specified in the fourth field of the `/etc/shadow` file must change their passwords at the specified number of days.

The MINWEEKS Variable

The value set for the `MINWEEKS` variable specifies the minimum number of weeks between password changes for all regular users.

If there is no value set for this variable, which is the default setting, only users who have a value for `Min Change` specified in the fifth field of the `/etc/shadow` file are limited as to when they can change their passwords.

Note – The password aging entries in the `/etc/shadow` file take precedence over the `/etc/default/passwd` file entries for individual users.

The PASSLENGTH Variable

The `PASSLENGTH` variable specifies a minimum password length for all regular users between the six and eight values. Numbers below six default to six character passwords, and numbers above eight default to eight character passwords.

Exercise: User Access



Exercise objective – In this lab you will log failed login attempts; use the commands `finger`, `last`, `rusers`, `su`, and `whoami`; examine the `su` log file; and change the file `/etc/default/login` to allow `root` logins from any terminal.

Preparation

This lab requires two systems that list each other in their `/etc/inet/hosts` files. It also requires a user called `user9` and `user3` on both systems. Both users and `root` should use the password `cangetin`. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

- Create the file `/var/adm/loginlog`. Use the command line `login` to make five failed login attempts. List the contents of `/var/adm/loginlog`. Use `finger` to display information for `user9` on your system and your partner's system.
- Use `last` to identify when the first `root` login session on your system occurred and how long the session lasted. Use `last` to learn when your system last booted. Use `rusers` to list users logged in on all systems on your network, and just on your partner's system.
- Use `su` to change your user identity from `root` to `user9`, both with and without the dash (-) option. Record differences. Use `whoami` and `who am i` to list effective and real user identity during your `su` sessions. Locate the `su` log declared in `/etc/default/su` and identify which user initiated your `su` attempts.
- As `root`, attempt a `telnet` session to your partner's system. Record error messages. Change the `CONSOLE` variable on your partner's system to allow `root` logins from any terminal. Attempt the `telnet` session again.

Tasks

1. Log in as the user `root` and open a terminal window. Change directory to `/var/adm`.

cd /var/adm
2. Use `touch` to create a file called `loginlog`.

touch loginlog
chgrp sys loginlog
3. Log out. From the CDE Options menu, select the Command Line Login item. Press Return when the CDE login screen clears 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 displays again. Log in as `root` and open a terminal window.
5. Examine `/var/adm/loginlog`. What does it contain?:

6. Use `finger` to display information for the user called `user9`. What is the difference in output between `finger -m` and `finger` with no option?

```
# finger user9  
# finger -m user9
```

7. Use `finger` to display information for the same user on your partner's system. Try this with and without the `-m` option. Replace host with the name of your partner's system. Does the `-m` option change the output `finger` displays?

```
# finger user9@hostname  
# finger -m user9@hostname
```

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
```

9. Use `last` to display only system boot activity. When did the system last reboot?

```
# last reboot
```

10. Use `rusers` to list information for users on all systems on your network segment. (Note: to terminate the process press Control-c.)

```
# rusers -l
```

11. Use `rusers` to list information for users on your partner's system. When and on what terminal did the first user listed login?

```
# rusers -l hostname
```

12. Switch your user identity to `user9`.

```
# 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`?

14. Exit the su session and try 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`?

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?

16. Use `su` to change user identity from `user9` to `user3`.

```
$ su user3
Password: cangetin
$
```

Exit the both su sessions when finished.

```
$ exit
$ exit
#
```

17. Change directory to `/etc/default`. Examine `/etc/default/su` and record the value of the `SULOG` variable.

```
# cd /etc/default
# more su
```

18. Display the file named by the `SULOG` variable, and identify the entry that relates to your last `su` command. Is `user9` or `root` identified as the user who became `user3`?

```
# cat /var/adm/sulog
```

19. As the user `root`, attempt to log in to your partner's system using `telnet`. Was your attempt successful? What message displays?

```
# telnet hostname
(telnet connection messages)
```

SunOS 5.8

```
login: root
Password: cangetin
```

20. On your partner's system, edit `/etc/default/login` and change the line that reads:

```
CONSOLE=/dev/console
```

so that it reads:

```
#CONSOLE=/dev/console
```

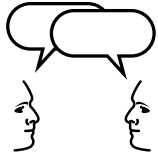
21. As the user `root`, again attempt to log in to your partner's system using `telnet`. If your log in attempt is successful, exit the `telnet` session. If not, check the change you made in Step 20 and try again.

```
# telnet host
(telnet connection messages)
```

SunOS 5.8

```
login: root
Password: cangetin
(telnet login messages)
Sun Microsystems Inc.   SunOS 5.8           Generic
February 2000
# exit
Connection closed by foreign host.
#
```

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

5. Examine `/var/adm/loginlog`. What does it contain?:

This file should contain a list of four failed login attempts.

6. Use `finger` to display information for the user called `user9`. What is the difference in output between `finger -m` and `finger` with no option?

`finger` with no option lists all users that have the string "user" in their names and comment fields. `finger -m` lists only the entry for the user named `user9`.

7. Use `finger` to display information for the same user on your partner's system. Try this with and without the `-m` option. Replace host with the name of your partner's system. Does the `-m` option change the output `finger` displays?

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?

This information depends on activity on your particular system.

9. Use `last` to display only system boot activity. When did the system last reboot?

This information depends on activity on your particular system.

11. Use `wusers` to list information for users on your partner's system. When and on what terminal did the first user listed login?

This information depends on activity on your particular system.

13. Display some of the variables that define your environment.

Are the values reported correct for the user `root` or for `user9`?

`root`

14. Exit the su session and try again, this time using the dash option.

Are the values reported now correct for the user `root` or for `user9`?

user9

15. Use the `whoami` and `who am i` commands to list your effective and real user identity.

What do these commands report?

/usr/ucb/whoami reports your effective UID, user9. The who am i command displays your real UID, root.

17. Change directory to `/etc/default`. Examine `/etc/default/su` and record the value of the `SULOG` variable.

/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 `root` identified as the user who became `user3`?

root

19. As the user `root`, attempt to log in to your partner's system using `telnet`. Was your attempt successful? What message displays?

The login attempt should not succeed. It fails with the messages:

Not on system console

Connection closed by foreign host.

Restricting Access to Data in Files

When 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 user should be able to access.

Users who need to share files should be put in a group.

Note – In general, you use file access permissions to determine what users or groups have permission to read, modify, or delete files.

Determining a User's Group Membership

The `groups` command display group memberships for the user.

For example, to see what groups you belong to, type the following command:

```
# groups  
staff class
```

To list the groups to which a specific user belongs, use the `groups` command with the user's name as an argument.

For example:

```
# groups user5  
staff class sysadmin
```

Identifying a User Account

You use the `id` command to further identify users by listing their UID, username, group ID, and group name. This is useful information when troubleshooting file access problems for users.

The `id` command returns the effective user ID and name. For example, if you logged in as `user1` and then used `su` to become `user4`, the `id` command reports information for the `user4` account.

Command Format

```
id [ options ] [ username ]
```

For example, to view your user account information:

```
$ id
uid=101(user1) gid=300(class)
```

To view all the account information for a specific user, use the `-a` option:

```
$ id -a user1
uid=101(user1) gid=300(class) groups=14(sysadmin)
```

Changing a File's Ownership with the `chown` Command

You might need to use the `chown` command to change the original owner of a file or directory to another user on the system. By default, only `root` can change the ownership of a file or directory.

Command Format

```
chown [ option(s) ] user_name filename(s)
```

or

```
chown [ option(s) ] UID filename(s)
```

Note – The *username* and the *UID* must exist in the `/etc/passwd` file.

Changing File Ownership

In this example, a user named `user1` created a file called `file7`.

```
# cd /export/home/user1
# ls -l file7
-rw-r--r--  1 user1    staff          672 Jun 1 15:11    file7
#
```

Use the `chown` command to give this file to a new user named `user2` and verify the new ownership.

```
# chown user2 file7
# ls -l file7
-rw-r--r--  1 user2    staff          672 Jun 1 15:12    file7
#
```

The file is now owned by `user2`. This file is still in the `home` directory of `user1`. The users need to determine if the file should be moved to a new directory location.

Changing Directory Ownership

In the next example, user1 owns a directory called dir4.

```
# ls -ld dir4
drwxr-xr-x  8 user1  staff          512 Apr 22 12:51  dir4
#
```

Use the `chown` command to give this directory and all of its contents (files and subdirectories) to user2.

```
# chown -R user2 dir4
# ls -ld dir4
drwxr-xr-x  8 user2  staff          512 Jun 1 15:14  dir4
#
```

The `-R` option makes the `chown` command recursive. It descends through the directory and any subdirectories setting the ownership UID as it moves through the directory hierarchy.

Changing User and Group Ownership Simultaneously

The `chown` command also gives the owner the ability to change both the ownership and group membership of a file or directory at the same time.

```
# chown user3:class file2
```

Additionally, you can use the `-R` option to recursively descend a directory hierarchy, changing ownership and group membership of the directory and its contents, simultaneously.

```
# chown -R user3:class dir1
```

Changing a File's Ownership With the `chgrp` Command

The `chgrp` command can be used by `root`, 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 that new group.

Command Format

```
chgrp groupname filename(s)
```

```
chgrp GID filename(s)
```

Note – The *groupname* and *GID* must exist in the `/etc/group` file.

For example, the file called `file4` currently belongs to a group named `staff`.

```
# ls -l file4
-rw-r--r--  1 user1  staff          874 Jun 1 15:08  file4
#
```

Use the `chgrp` command to give this file to a new group named `class`, and verify the new group ownership.

```
# chgrp class file4
# ls -l file4
-rw-r--r--  1 user1  class          874 Jun 1 15:09  file4
#
```

Now all users who are members of the group called `class` have shared access to this file.

Special File Permissions

Three types of special permissions are available for executable files and public directories. These include:

- `setuid` Permission
- `setgid` Permission
- Sticky Bit Permission

The *setuid* Permission

When 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 *root*) instead of the user who started the executable.

This allows a user to access files and directories that are normally accessible only by the owner. Plus many executable programs must be run as *root*, *sys*, or *bin* to work properly.

For example:

```
-r-sr-xr-x  1 root    sys          17156 Jan  5 17:03  /usr/bin/su
```

The *setuid* permission displays as an “S” in the owner’s execute field.

Note – If a capital “S” appears, it simply indicates that the *setuid* bit is on and the execute bit “x” is off or denied.

The *root* user and the owner can set the *setuid* permissions on an executable file using the *chmod* command and the octal value 4000.

For example:

```
# chmod 4555 executable_file
```

Except for those *setuid* executable files that exist by default in the Solaris Operating Environment, the system administrator 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 pathname, execute the following command:

```
# find / -perm -4000
```

The setgid Permission

The set-group identification (setgid) permission is similar to setuid, except that the effective group ID of the user or the process is changed to the group owner of the file. Also, access is granted based on the permissions assigned to that group.

For example, the mail program has a setgid permission used to read mail, or send mail to other users.

```
-r-x--s--x  1 root    mail          61288 Jan  5 16:57    /usr/bin/mail
```

The setgid permission displays as an “s” in the group execute field.

Note – If a lowercase letter “l” appears, it indicates that the setgid bit is on, and the execute bit is off or denied. This indicates that mandatory file and record locking occurs during access.

The root user and the owner can set setgid permissions on an executable file using the chmod command and the octal value 2000.

For example:

```
# chmod 2555 executable_file
```

Shared 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 to which the directory belongs.

For example, if a user has write permission in the directory and creates a file there, that file belongs to 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:

```
# chmod g+s shared_directory
```

Searching for setgid Files and Directories

To search for files with `setgid` permissions and display their full pathname, execute the following command:

```
# find / -perm -2000
```

The Sticky Bit Permission

The Sticky Bit is a special permission that protects the files within a publically writable directory.

If the directory has the Sticky Bit set, a file can be deleted only by the owner of the file, the owner of the directory, or by `root`. This prevents a user from deleting other users' files from publicly writable directories. For example:

```
# ls -ld /tmp
drwxrwxrwt 6 root sys      719 May 31 03:30  /tmp
```

The Sticky Bit is displayed as the letter "t" in the execute field for other.

Note – If a capital "T" appears, 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 using the `chmod` command and the octal value `1000`.

For example:

```
# chmod 1777 public_directory
```

Searching for Directories with a Sticky Bit Permission

To search for directories with Sticky Bit permissions and display their full pathname, execute the following command:

```
# find / -type d -perm -1000
```

Note – For more detailed information on the Sticky Bit, execute the following command: `man sticky`

Exercise: File Owners, Groups, and Special Permissions



Exercise objective – In this lab you will practice using commands related to user identity and file ownership, assign a user to the sysadmin group, and make use of special file permissions.

Preparation

Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

- Using `groups`, `id`, and `id -a`, identify the groups to which `root` belongs. Compare the output from these commands. Add a user called `user11` as described in step 3. Verify the list of groups to which `user11` belongs. Log in as `user11`. Use `admintool` to attempt to create a new user called `user12`. Record if this succeeds.
- Log in again as `root`, and add `user11` to the `sysadmin` group. Log in again as `user11` and attempt to create a new user called `user12` using `useradd`. Record if this succeeds. Use `admintool` to attempt to create a new user called `user12`. Record if this succeeds.
- As `user11`, create a new file called `file1`. Attempt to change its user ownership. Record error messages. Change the group ownership of `file1` to `sysadmin`. Switch user identity to `root` and change ownership of `file1` to `user12`.
- As `user11`, create a new file called `file2`. Use `chmod` to set `setuid` and `setgid` permissions on `file2`. Use `chmod` 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 `/tmp`. 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 `dir1` to `777`. Create a file called `test2` in `dir1`. As `user12` attempt to remove this file. Record the result. Log in again as `root`.

Tasks

1. Log in as `root` and open a terminal window. Use the `groups` command to display the groups to which `root` belongs. Record the list that `groups` displays.

```
# groups
```

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?

```
# id -a
```

Compare the `id -a` output with that from the `groups` command in step 1. What additional information does `id -a` provide?

3. Use `useradd` to create a new user called `user11`. Set the password for `user11` to `cangetin`.

```
# useradd -u 1011 -g 10 -d /export/home/user11 -m -s  
/bin/ksh -c "SA238 Admin User" user11
```

```
6 blocks
```

```
# passwd user11
```

```
New password: cangetin
```

```
Re-enter new password: cangetin
```

```
passwd (SYSTEM): passwd successfully changed for  
user11
```

```
#
```

-
4. Verify the list of groups to which `user11` belongs. Does `user11` belong to group 14?

```
# id -a user11
```

5. Log out and log in again as `user11`. Open a terminal window and run `admintool`. From the Edit menu select Add.

```
$ admintool &
```

6. Enter the following information in the Add User form. Click OK when finished. What message displays? Exit `admintool` when finished.

User Name:	user12
User ID:	1012
Primary Group:	10
Login Shell:	Korn
Password:	cangetin
Home directory (create it);	/export/home/user12

7. Log out and log in again as `root`. Open a terminal window. Use `usermod` to add `user11` to group 14. Verify the change took place.

```
# usermod -G 14 user11
# id -a user11
```

8. Switch your user identity to `user11`, and attempt to add a new user using `useradd`. What error message displays?

```
# su - user11
$ /usr/sbin/useradd -u 1012 -g 10 -d
/export/home/user12 -m -s /bin/ksh -c "Test User"
user12
```

9. Log out and log in again as `user11`. Open a terminal window and run `admintool`. From the **Edit** menu select **Add**. Try to create the user defined in step 6. Were you successful? Exit `admintool` when finished.

```
$ admintool &
```

10. Change directory to your home directory as `user11`. Use the `touch` command to create a file called `file1`. Verify that `user11` and the group `staff` own `file1`.

```
$ cd
$ touch file1
$ ls -l file1
```

11. Attempt to change the owner of `file1` from `user11` to `user12`. What error message displays?

```
$ chown user12 file1
```

12. Attempt to change the group ownership of `file1` from `staff` to `sysadmin`. Verify the change. Did it work?

```
$ chgrp sysadmin file1
$ ls -l file1
```

13. Switch your user identity to `root` and change 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 finished.

```
$ su -
Password: cangetin
# pwd
# cd /export/home/user11
# chown user12 file1
# ls -l
# exit
$
```

-
14. In the home directory for `user11`, use `touch` to create a file called `file2`. Display and record the permissions associated with `file2`.

```
$ touch file2
$ ls -l file2
```

15. Use `chmod` to add setuid permissions to `file2`. Display and record the permissions associated with `file2`. What changed?

```
$ chmod 4555 file2
$ ls -l file2
```

16. Use `chmod` to add setuid and setgid permissions to `file2`. Display and record the permissions associated with `file2`. What changed?

```
$ chmod 6555 file2
$ ls -l file2
```

17. Use `chmod` to remove all execute permissions from `file2`. Display and record the permissions associated with `file2`. What changed?

```
$ chmod 6444 file2
$ ls -l file2
```

18. Change 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 `/tmp`?

```
$ cd /
$ ls -ld tmp
```

19. Change directory to /tmp. Create a file called `test1` in /tmp. 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
```

20. Switch user to `user12`. In /tmp, attempt to remove `test1`. What messages display? Exit your `su` session when finished.

```
$ su user12
Password: cangetin
$ rm test1
$ exit
$
```

21. In the home directory for `user11`, create a directory called `dir1`. Change permissions for `dir1` to 777. Create a file called `test2` below `dir1`.

```
$ mkdir dir1
$ chmod 777 dir1
$ touch dir1/test2
```

22. Switch your identity to `user12`. Attempt to remove the file `test2` from `dir1`. Verify `test2` no longer exists. Exit your `su` session when finished.

```
$ su user12
Password: cangetin
$ rm dir1/test2
$ ls -l dir1
$ exit
$
```

23. Log out and log in again as `root`.

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

1. Log in as `root` and open a terminal window. Use the `groups` command to display the groups to which `root` belongs. Record the list that `groups` displays.

other root bin sys adm uucp mail tty lp nuucp daemon

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?

id reports the primary group.

Compare the `id -a` output with that from the `groups` command in step 1. What additional information does `id -a` provide?

id -a reports group ID numbers in addition to group names.

4. Verify the list of groups to which `user11` belongs. Does `user11` belong to group 14?

No.

6. Enter the following information in the Add User form. Click OK when finished. What message displays? Exit `admintool` when finished.

Security exception on <hostname> USER ACCESS DENIED. The user identity <UID#> ("<username>") was received, but that user is not authorized to execute the requested functionality on this system. Is this user a member of an appropriate security group (14) on this system?

8. Switch your user identity to `user11`, and attempt to add a new user using `useradd`. What error message displays?

UX: /usr/sbin/useradd: ERROR: Permission denied.

9. Log out and log in again as `user11`. Open a terminal window and run `admintool`. From the **Edit** menu select **Add**. Try to create the user defined in step 6. Were you successful? Exit `admintool` when finished.

Yes.

11. Attempt to change the owner of `file1` from `user11` to `user12`. What error message displays?

chown: file1: Not owner

12. Attempt to change the group ownership of `file1` from `staff` to `sysadmin`. Verify the change. Did it work?

Yes.

13. Switch your user identity to `root` and change 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 finished.

Yes.

14. In the home directory for `user11`, use `touch` to create a file called `file2`. Display and record the permissions associated with `file2`.

The permissions for file2 should read: -rw-r--r

15. Use `chmod` to add `setuid` permissions to `file2`. Display and record the permissions associated with `file2`. What changed?

The permissions for file2 should read: -r-sr-xr-x

16. Use `chmod` to add `setuid` and `setgid` permissions to `file2`. Display and record the permissions associated with `file2`. What changed?

The permissions for file2 should read: -r-sr-sr-x

17. Use `chmod` to remove all execute permissions from `file2`. Display and record the permissions associated with `file2`. What changed?

The permissions for file2 should read: -r-Sr-lr--

18. Change directory to / and list the permissions associated with the /tmp directory. Is the sticky bit set on /tmp? Do all users have write permission in /tmp?

Yes.

19. Change directory to /tmp. Create a file called test1 in /tmp. Verify that user11 and the group staff own test1, and that 644 permissions apply. Do they?

Yes.

20. Switch user to user12. In /tmp, attempt to remove test1. What messages display? Exit your su session when finished.

```
rm: test1: override protection 644 (yes/no)? y
```

```
rm: test1 not removed: Permission denied
```

Access Control Lists

Access Control Lists (ACLs) can provide greater control over file access permissions when traditional file protection is not enough.

An ACL provides better file security by enabling you to define file permissions for the file owner, file group, other, specific users and groups. ACLs also enables you to set default permissions for each of these categories.

For example, if the system administrator wanted everyone in a particular group to be able to read a file, you would simply give the group read permissions on that file.

However, what if the system administrator wanted only one person in that group to be able to write to that file? ACLs can provide that level of file security, where traditional UNIX file access protection cannot.

You should view ACLs as extensions to the standard UNIX file permissions.

The ACL information is stored and associated with each file or directory individually.

ACLs for a file or directory are set or viewed using the commands and options described in Table 3-1.

Table 3-1 ACL Commands and Options

Command/Option	Description
<code>getfacl filename(s)</code>	Displays ACL entries on a file(s).
<code>setfacl options filename</code>	Sets, adds, modifies, and deletes ACL entries on a file(s).
<code>setfacl -m acl_entries</code>	Creates or modifies ACL entries on files
<code>setfacl -s acl_entries</code>	Removes old ACL entries on a file(s) and replaces with new ACL entries.
<code>setfacl -d acl_entries</code>	Deletes one or more ACL entries on a file(s).

Table 3-1 ACL Commands and Options (Continued)

Command/Option	Description
<code>setfacl -f <i>acl_file</i></code>	Specify an ACL configuration file containing list of permissions to be set on other files. <i>acl_file</i> is used as an argument with this command only.
<code>setfacl -r</code>	Recalculates permissions for the ACL mask. ¹

1. Permissions specified in the ACL mask are ignored and replaced by the maximum permissions needed to give access to any additional user, owner group, and additional group entries in the ACL.

ACL Entries

Each ACL entry consists of the fields described in Table 3-2, which are separated by colons.

Table 3-2 ACL Entries

ACL Fields	Description
<code>entry-type</code>	Type of entry to set file permissions for owner, owner's group, specific users, additional groups, or the ACL mask.
<code>UID or GID</code>	The user's name or identification number (UID). The group's name or identification number (GID).
<code>perm</code>	Permissions set for entry-type. You can set permissions symbolically using <code>r</code> , <code>w</code> , <code>x</code> , and <code>-</code> or by using octal values from 0 to 7.

The `setfacl` command uses these ACL entries to set permissions on files, for example:

- `u[ser]::perm` – Sets the permissions for the file owner.
- `g[roup]::perm` – Sets the permissions for the owner's group.
- `o[ther]:perm` – Sets the permissions for users other than the owner or members of the owner's group.
- `u[ser]:UID:perm` or `u[ser]:username:perm` – Sets the permissions for a specific user. The `username` must exist in the `/etc/passwd` file.
- `g[roup]:GID:perm` or `g[roup]:groupname:perm` – Sets the permissions for a specific group. The `groupname` must exist in the `/etc/group` file.
- `m[ask]:perm` – Sets the ACL mask. The mask entry indicates the maximum permissions allowed for all users, except the owner, and for all groups. The mask is a quick way to change permissions for all the users and groups.

Adding and Modifying ACL Permissions on a File

You can use the `setfacl -m` command to add or modify ACL permissions on one or more of the file's ACL entries.

Command Format

```
setfacl -m acl_entry,acl_entry filename1 [filename2 ...]
```

Examples of Modifying ACL Entries on a File

The following example creates an ACL entry on `file.txt` for `user8` with permissions to read and write the file.

```
# setfacl -m user:user8:6 file.txt

# getfacl file.txt
# file: file.txt
# owner: user1
# group: class
user::rwx
user::user8:rw-      #effective:r--
group::r-             #effective:r--
mask:r--
other:---
```

The next example modifies the permissions of the ACL mask to read and write.

```
# setfacl -m m:6 file.txt

# getfacl file.txt
# file: file.txt
# owner: user1
# group: class
user::rwx
user::user8:rw-      #effective:rw-
group::r-             #effective:r--
mask:rw-
other:---
```

Determining if a File Has an ACL

There are two ways to determine if a file has an ACL

- Using the `getfacl` command
- Using the `ls -l` command

Using the `ls -l` command on any file that has an ACL displays a plus (+) sign at the end of the permission mode field. For example:

```
# ls -l file.txt
-rwxr-----+ 1 user1  class          167 Apr 18 11:13  file.txt
```

Note – If a file has no ACL entries for additional users or groups, the file is considered to be a *trivial* ACL file and the + symbol is not displayed.

Deleting an ACL Entry on a File

To delete an ACL entry from a file, use the `setfacl -d` command. An ACL entry can be one or more comma-separated ACL entries without permissions. To delete an ACL, specify the entry type and the UID (user name) or GID (group name).

You cannot delete the ACL entries for the file owner, file group owner, other, and the ACL mask.

Command Format

```
setfacl -d ACL_entry filename(s)
```

or

```
setfacl -d ACL_entry,ACL_entry filename(s)
```

The following is an example of deleting an ACL entry.

```
# setfacl -d u:user8 file.txt
```

Replacing an Entire ACL on a File

To replace the entire ACL on a file, from the command line, you must specify at least the basic set of user, group, other, and mask permissions and file name(s).

Command Format

```
setfacl -s u::perm,g::perm,o:perm,m:perm,[u:UID:perm],[g:GID:perm]  
filename(s)
```

An Example of Setting an ACL on a File

The following example sets the file owner permissions to read and write, group permissions to read only, and other permissions to none on `file.txt`.

In addition, `user8` is given read/write permissions on the file, and the ACL mask is set to read/write, which indicates that no user or group can have execute permissions on the file.

```
# setfacl -s user::rw-,group::r--,other:---,mask:rw-,user:user8:rw- file.txt
```

To verify which ACL entries were set on the file, use the `getfacl` command.

```
# getfacl file.txt  
# file: file.txt  
# owner: user1  
# group: class  
user::rw-  
user:user8:rw-          #effective:rw-  
group::r--              #effective:r--  
mask:rw-  
other:---
```

Another Example of Setting an ACL on a File

This next example sets the file owner permissions to read, write, and execute, group permissions to read only, other permissions to none, and the ACL mask to read.

In addition, user8 is given read and write permissions; however, due to the ACL mask, the effective permissions for user8 are read only.

```
# setfacl -s u::7,g::4,o:0,m:4,u:user8:7 file.txt
```

Verify which ACL entries were set on the file with the `getfacl` command.

```
# getfacl file.txt
# file: file.txt
# owner: user1
# group: class
user::rwx
user:user8:rwx      #effective:r--
group::r--          #effective:r--
mask:r--
other:---
```

Exercise: Using Access Control Lists



Exercise objective – In this exercise you create two files and manipulate their associated access control lists.

Preparation

This exercise requires a user called `user10`, and a group called `group1`. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

- Create the user called `user10` if required. Create a directory called `/var/test`. In this directory, create two files called `file1` and `file2`. Add a line of text to `file1`. Record the permissions applied to each file. Verify that the permissions and ACL information for `file1` agree.
- Set 440 permissions on `file1`. Switch user to `user10` and attempt to read the file. Record the result. Exit your `su` session. Create an ACL entry that grants `user10` read permission. Verify the new ACL entry exists and record how the entry's presence is indicated in the permissions list. Switch user to `user10` and again attempt to read the file. Record the result.
- Display the ACL for `file2`. Verify that the group and mask permissions match. Use `chmod` to grant full permissions to the group that owns `file2`. Verify that the mask and group permissions match. Set the mask permissions to read only for `file2`. Verify that the group and mask permissions match.
- Add the group called `group1` if it doesn't exist. Add ACL entries for `group1` and `user10` that grant read and execute permissions for `group1` and only execute permissions for `user10`. Record the effective permissions for `user10` and `group1`. Set the mask to

grant read, write, and execute permissions. Record the effective permissions for `user10` and `group1`. Record the permissions for the group that owns `file2`.

Tasks

1. Log in as `root` and open a terminal window.
2. If `user10` does not exist on your system, create it.

```
# useradd -u 1010 -g 10 -d /export/home/user10 -m -s  
/bin/ksh -c "SA-238 Student" user10
```

3. Create the directory `/var/test` and change directory to that location.

```
# mkdir /var/test  
# cd /var/test
```

4. Create two new files. Record the permissions applied to each.

```
# echo "Success for file1!" > file1  
# touch file2  
# ls -l
```

5. Display the Access Control List for `file1`. Do the permissions in the ACL match the permissions reported by `ls`?

```
# getfacl file1
```

6. Change permissions on `file1` so that only the owner (`root`) and group (`other`) have read access.

```
# chmod 440 file1
```

7. Switch your user identity to `user10`.

```
# su user10  
$
```


8. Attempt to display the content of `file1`. What is the result?

```
$ cat file1
```

9. Exit your `su` session. Use `setfacl` to add an ACL entry that allows read access for `user10` to the ACL for `file1`. Verify that the new ACL entry exists. Switch your user identity back to `user10`.

```
$ exit
# setfacl -m user:user10:4 file1
# getfacl file1
# su user10
$
```

10. Use `ls` to display the permissions applied to `file1`. According to these permissions does `user10` have read access?

```
$ ls -l file1
```

What indicates that an additional ACL entry exists for `file1`?

11. Attempt to display the content of `file1`. What is the result? Exit your `su` session when finished.

```
$ cat file1
$ exit
#
```

12. Display the Access Control List for `file2`. Do the group permissions match the permissions associated with the `mask` entry?

```
# getfacl file2
```

13. Grant read write and execute permissions to the group that owns `file2`. Display the ACL and a long listing for `file2`.

```
# chmod g=rwx file2
# getfacl file2
# ls -l file2
```

Do the mask permissions match the group permissions?

14. Set the mask permissions for `file2` to read only. Display the ACL and a long listing for `file2`.

```
# setfacl -m mask:r-- file2
# getfacl file2
# ls -l file2
```

Do the mask permissions match the group permissions?

In the long listing output, do you find an indication that `file2` has additional ACL entries?

15. If `group1` does not exist on your system, create it.

```
# groupadd -g 101 group1
```

16. Add an ACL entry for the group called `group1` to `file2`. Grant only read and execute permissions for this group.

```
# setfacl -m group:group1:5 file2
```

17. Add an ACL entry for the user called `user10` to `file2`. Grant only execute permissions for this user.

```
# setfacl -m user:user10:1 file2
```

Verify the current ACL permissions for `file2`.

```
# getfacl file2
```

What are the effective permissions for `user10` and `group1`?

18. Set the `mask` value to read, write, and execute.

```
# setfacl -m mask:rwx file2
```

19. Again verify the effective permissions for `user10` and `group1`. Do their effective permissions match the `mask` or what they were specifically granted?

```
# getfacl file2
```

Did changing the mask permissions affect the permissions for the group that owns the file?

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

4. Create two new files. Record the permissions applied to each.

Both files use -rw-r--r-- (644) permissions.

5. Display the Access Control List for `file1`. Do the permissions in the ACL match the permissions reported by `ls`?

Yes, they should.

8. Attempt to display the content of `file1`. What is the result?

The following error message displays: `cat: cannot open file1`

10. Use `ls` to display the permissions applied to `file1`. According to these permissions does `user10` have read access?

No.

What indicates that an additional ACL entry exists for `file1`?

The "+" symbol at the end of the permissions string.

11. Attempt to display the content of `file1`. What is the result? Exit your `SU` session when finished.

The file content displays.

12. Display the Access Control List for `file2`. Do the group permissions match the permissions associated with the `mask` entry?

Yes.

13. Grant read write and execute permissions to the group that owns `file2`. Display the ACL and a long listing for `file2`.

Do the `mask` permissions match the group permissions?

Yes.

14. Set the mask permissions for `file2` to read only. Display the ACL and a long listing for `file2`.

Do the mask permissions match the group permissions?

Yes.

In the long listing output, do you find an indication that `file2` has additional ACL entries?

No.

17. Add an ACL entry for the user called `user10` to `file2`. Grant only execute permissions for this user.

Verify the current ACL permissions for `file2`.

What are the effective permissions for `user10` and `group1`?

`user10` has no permissions, `group1` has read-only permission.

19. Again verify the effective permissions for `user10` and `group1`. Do their effective permissions match the mask or what they were specifically granted?

The permissions should match what you specifically granted.

Did changing the mask permissions affect the permissions for the group that owns the file?

No. The group permissions remain read-only.

Managing Remote Access Issues

The more access that is available over the network, the more beneficial it is for remote system users. However, unrestrained access and sharing of data and resources will create security problems.

A local host's remote security measures are generally based on being able to validate, limit, or block operations from remote system users.

The three network files listed here provide certain schemes for handling basic security issues involving remote user access of a local system.

- The `/etc/hosts.equiv` file
- The `$HOME/.rhosts` file
- The `/etc/ftpusers` file

The /etc/hosts.equiv and \$HOME/.rhosts Files

Typically, when a remote user requests login 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 login to gain system access.

When 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 should be 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. Based on the information contained in these two files, (if they exist), determines if access is granted or denied.

The `/etc/hosts.equiv` file applies to the entire system, while individual users can maintain their own `$HOME/.rhosts` files in their home directories.

Remote Access Authentication

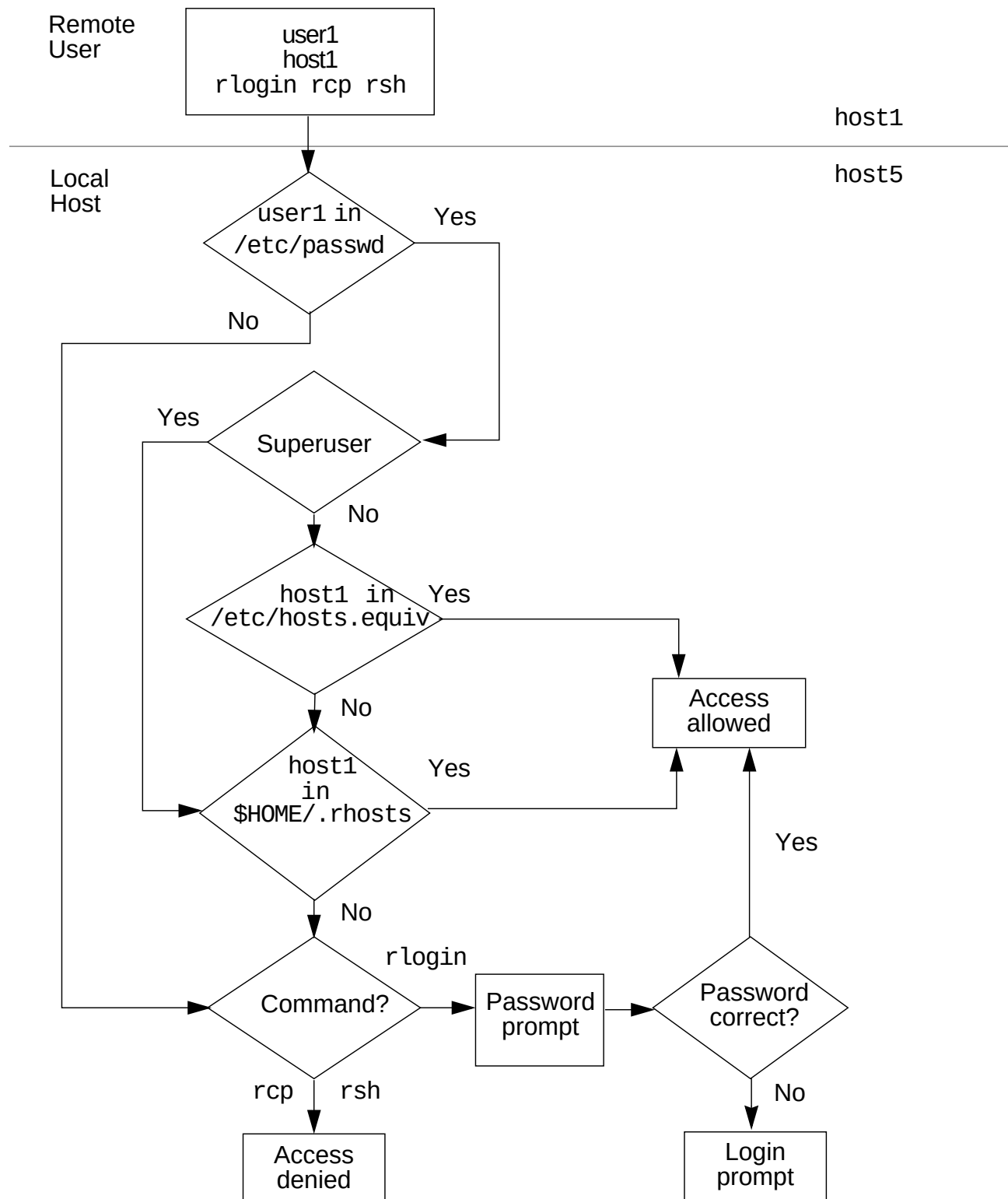


Figure 3-1 Example of Remote Access Authentication

Entries in `/etc/hosts.equiv` and `$HOME/.rhosts`

While the `/etc/hosts.equiv` and `$HOME/.rhosts` files have the same format, the same entries in each file have different effects.

The general format is presented here. Explanations and examples of the meanings of each type of entry are presented on the following pages.

- Both files are formatted as a list of one-line entries, which can contain the following types of entries:

hostname

hostname username

+

Note – The host name(s) in the `/etc/hosts.equiv` and `$HOME/.rhosts` files must be the official name of the host, not one of its alias name(s).

- If only the *hostname* is used, then all users from the named host are trusted, provided they are known to the local host.
- If both *hostname* and *username* are used, then only the named remote user from the named remote host can access the local host.
- A single plus sign (+) character placed in the file indicates that every remote host on the network is trusted by the local host. Enabling remote users to login from anywhere on the network, with no passwords required.

The /etc/hosts.equiv File

For regular users, the `/etc/hosts.equiv` file is used to identify 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 `root`.

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. Provided 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 is particularly useful for sites where it is common for regular users to 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 remote user access is required on the local host.

The \$HOME/.rhosts File

While the `/etc/hosts.equiv` file applies system-wide for non-root users, the `.rhosts` file applies to a specific user.

All users, including `root`, can create and maintain their own `.rhosts` files in their home directory.

For example, if you run an `rlogin` process from a remote host to gain `root` access to a local host, it checks for a `.rhosts` file in the `root` home directory on the local host.

If the remote host name is listed in the file, it is considered to be a trusted host and remote user access, in this case `root` access, is granted on the local host.

The `$HOME/.rhosts` file does not exist by default, you must create it in the user's home directory.

Restricting FTP Logins

The Solaris Operating Environment provides an ASCII file named `/etc/ftpusers`. The `ftpusers` file is used to list the names of users who are prohibited from running an ftp login on the system.

Each line entry in this file contains a login name for each restricted user, for example:

```
username
```

The FTP server `in.ftpd` daemon reads the `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 `ftpusers` file has the following system account entries:

```
root
daemon
bin
sys
adm
lp
uucp
nuucp
listen
nobody
noaccess
nobody4
```

As with any user name that you can add, these entries must match the user account names located in the `/etc/passwd` file.

Because the new default security policy in the Solaris 8 Operating Environment is to disallow remote root logins, the `root` entry is included in `/etc/ftpusers`.

If root login privileges are allowed by deleting the `root` entry in `/etc/ftpusers`, ensure the `/etc/default/login` file reflects remote root login privileges.

The /etc/shells File

The `/etc/shells` file contains a list of the shells on the system. Applications, such as `sendmail` and `ftp`, can use this file to determine whether a shell is valid.

This file does not exist by default.

Note – If this file does not exist, then `getuser shells(3c)` uses its own list of shells.

By creating this file, each shell that you want to be recognized by the system, must have a single line entry, consisting of the shell's path, relative to `/` (root).

For example:

```
# touch /etc/shells
/sbin/sh
/bin/sh
/bin/ksh
```

While the `/etc/ftpusers` file prohibits `ftp` connections for a specific user, you can create an `/etc/shells` file to allow `ftp` connections only to those users running shells that you have defined in this file.

If an entry for a shell does not exist in this file, any user running the undefined shell is not allowed `ftp` connections to the system.

Exercise: Managing Remote Security Issues



Exercise objective – In this exercise you will configure systems to use `rlogin`, `rcp` and `rsh`, and enable `ftp` transfers as the user `root`.

Preparation

This exercise requires you to work with a partner. Select one system to act as the "*local host*" and the other as the "*remote host*". Be sure to execute each step on the appropriate host. Some steps require execution on *both* hosts. The `root` password on both systems should be `cangetin`. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

- On both hosts, log in as `root`. Create a user called `user14` on both the local and remote host. Set its password to `cangetin`. On the local host, switch user identity to `user14`. Connect to the remote host using `rlogin`. Record what `rlogin` requires to complete the login. Exit your `rlogin` session.
- On the local host, attempt to copy the `/etc/system` file from the remote host to a file called `testfile`. Record any error messages. On the remote host, create `/etc/hosts.equiv` so it contains the name of the local host. On the local host use `rsh` to run an `ls` command on the remote host. Create a file called `testfile2` on the local host. Use `rcp` to copy `testfile2` to the `/tmp` directory on the remote host. Verify the transfer.
- On the local host, create a directory called `newdir` and create two files in it. Use `rcp` to copy `newdir` and its contents to the remote host. Verify the transfer.

- On the local host, exit your su session. As root, from /export/home/user14, use ftp to connect to the remote host. What happens? On the remote host, edit /etc/ftpusers and remove the entry for root. Create a file in /tmp called ftpfile. On the local host, use ftp to connect to the remote host and transfer /tmp/ftpfile to the local host. Verify the transfer.

Tasks

On Both Hosts:

1. Log in as root and open a terminal window. *If /export/home doesn't exist, create it.*

```
# mkdir /export/home
```

2. Use useradd to create a new user called user14. Set the password for user14 to cangetin.

```
# useradd -u 1014 -g 10 -d /export/home/user14 -m -s  
/bin/ksh user14  
6 blocks  
# passwd user14  
New password: cangetin  
Re-enter new password: cangetin  
passwd (SYSTEM): passwd successfully changed for  
user14
```

On the Local Host:

3. Switch your user identity to user14.

```
# su - user14
```

4. Use rlogin to login to the *remote host* as user14. What information did you have to provide?

```
$ rlogin remote_host
```

5. Exit your rlogin session.

```
$ exit
```

6. Use `rcp` to copy the `/etc/system` file from the *remote host* to a file called `testfile1` on the *local host*. What error message displays?

```
$ rcp remote_host:/etc/system testfile1
```

On the Remote Host

7. As the `root` user, use `vi` to create a file called `/etc/hosts.equiv`. Add one line that contains the name of the system you're using as the *local host*.

On the Local Host

8. As `user14`, use `rsh` to execute the `ls` command on the remote system.

```
$ rsh remote_host ls /tmp
```

9. Create a file called `testfile2` in the home directory of `user14`. Use `rcp` to copy `testfile2` to the `/tmp` directory on the *remote host*:

```
$ touch testfile2
$ rcp testfile2 remote_host:/tmp
```

On the Remote Host

10. Verify that `testfile2` exists in `/tmp`.

```
$ ls /tmp
```

On the Local Host

11. Create a directory called `newdir` in the home directory of `user14`. Use `touch` to create two files called `file1` and `file2` in `newdir`. Use `rcp` to copy `newdir` and its contents to the *remote host*, and place them in `/tmp`.

```
$ cd
$ mkdir newdir
$ cd newdir
$ touch file1 file2
$ cd ..
$ rcp -r newdir remote_host:/tmp
```

On the Remote Host

12. Verify that `newdir` and its contents exists in `/tmp`.

```
$ ls -R /tmp/newdir
```

On the Local Host

13. Exit your `su` session.

```
$ exit  
#
```

14. Change directory to `/export/home/user14`. Attempt to use `ftp` to connect to the remote host as the user `root`. What happens?

```
# ftp remote_host  
Connected to remote_host.  
220 host2 FTP server (SunOS 5.8) ready.  
Name (remote_host:root): root  
331 Password required for root.  
Password: cangetin
```

15. Quit the `ftp` session.

```
ftp> bye  
221 Goodbye.  
#
```

On the Remote Host

16. Use `vi` to edit the `/etc/ftpusers` file. Delete the line that lists the `root` user. Save the file and quit `vi`.

```
# vi /etc/ftpusers
```

17. Use `touch` to create a file called `ftpfile` in `/tmp`.

```
# touch /tmp/ftpfile
```

On the Local Host

18. Again attempt to use `ftp` to connect to the remote host as the user `root`. Can you connect?

```
# ftp remote_host
Connected to remote_host.
220 host2 FTP server (SunOS 5.8) ready.
Name (remote_host:root): root
331 Password required for root.
Password: cangetin
```

19. If your `ftp` login was successful, use `pwd` to identify your current directory on the remote host. Change directory to `/tmp`. Use the `get` command in `ftp` to retrieve the file called `ftpfile`. Quit your `ftp` session and verify that `ftpfile` exists in `/export/home/user14`.

```
ftp> pwd
257 "/" is current directory.
ftp> cd /tmp
250 CWD command successful.
ftp> get ftpfile
200 PORT command successful.
150 ASCII data connection for ftpfile
(192.9.200.1,32998) (0 bytes).
226 ASCII Transfer complete.
ftp> bye
221 Goodbye.
# ls -l ftpfile
```

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

4. Use `rlogin` to login to the remote host as `user14`. What information did you have to provide?

rlogin requires the password for user14.

6. Use `rcp` to copy the `/etc/system` file from the *remote host* to a file called `testfile1` on the *local host*. What error message displays?

permission denied

14. Change directory to `/export/home/user14`. Attempt to use `ftp` to connect to the remote host as the user `root`. What happens?

530 Login incorrect.

Login failed.

18. Again attempt to use `ftp` to connect to the remote host as the user `root`. Can you connect?

Yes.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Create the `/var/adm/loginlog` file to save failed login attempts
- ☐ Monitor system usage with the commands `finger`, `last`, and `rusers`
- ☐ Use the `su` command to become `root` or another user on the system
- ☐ Modify the `/etc/default/login` file to restrict `root` access
- ☐ Use the commands `id` and `groups` to identify users and their group memberships
- ☐ Change a file's owner or a file's group using the commands `chown` and `chgrp`, respectively
- ☐ Explain how the special permissions `setuid`, `setgid`, and the Sticky Bit can affect system security
- ☐ Create, modify, and delete access control lists (ACLs) on files
- ☐ Control remote login access by maintaining three basic network files: `/etc/hosts.equiv`, `$HOME/.rhosts`, and `/etc/ftpusers`

Objectives

Upon completing this module you should be able to:

- Identify the four main file types in the Solaris Operating Environment
- Describe the functions provided by regular files, directories, symbolic links, device files, and hard links
- Define the function of each subdirectory found directly within the root directory

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10

The Solaris Operating Environment File Types

The Solaris Operating Environment supports a standard set of files 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, four could be described as the main file types in the Solaris Operating Environment, which include:

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

Regular files, directories, and symbolic links all store one or more kind of data. Device files differ from the other three because they do not store data; instead, they provide access to devices.

Files that provide inter-process communication include sockets, named pipes, and doors. These last three types of files are not described in this module.

Identifying File Types

Using the `ls` command, you can easily distinguish different file types from one another. In the following example, the first column of information the `ls -l` command displays indicates the file type.

The following examples show partial listings on an Ultra 5 system from directories that contain a mix of different file types:

```
# cd /etc
# ls -l
total 428
drwxr-xr-x  2 adm      adm      512 Apr  3 10:42 acct
lrwxrwxrwx  1 root     root     14 Apr  3 11:05 aliases ->
./mail/aliases
drwxr-xr-x  2 root     sys      512 Apr  3 10:44 ami
drwxr-xr-x  2 root     bin      512 Apr  3 10:45 apache
-rwxr--r--  1 root     sys      360 Apr  3 10:45 asppp.cf
-rw-r--r--  1 root     bin       50 Apr  3 10:45 auto_home
-rw-r--r--  1 root     bin     113 Apr  3 10:45 auto_master
(output truncated)

# cd /devices/pci@1f,0/pci@1,1/ide@3
# ls -l
total 0
brw-----  1 root     sys      136,  0 Apr  3 11:11 dad@0,0:a
crw-----  1 root     sys      136,  0 Apr  3 11:11 dad@0,0:a,raw
brw-----  1 root     sys      136,  1 Apr  4 11:06 dad@0,0:b
crw-----  1 root     sys      136,  1 Apr  3 11:11 dad@0,0:b,raw
(output truncated)
```

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

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

File Names, Inodes, and Data Blocks

All files in the Solaris Operating Environment make use of a file name and a record called an inode. Most files also make use of data blocks.

File names are the objects most often used to access and manipulate files.

Inodes are the objects the system uses to record information about a file.

Data blocks are units of disk space used to store data.

To exist, a file must have a name that is associated with an inode. In general, inodes contain two parts. First, they contain information about the file, including who owns it, its permissions and size. Second, they contain pointers to data blocks associated with the file.

Subsequent modules that describe the ufs file system describe the content of inode records in detail. However, in general, a file name is associated with an inode, and an inode provides access to data blocks.

For the purpose of understanding file types, use Figure 4-1 to visualize these relationships.

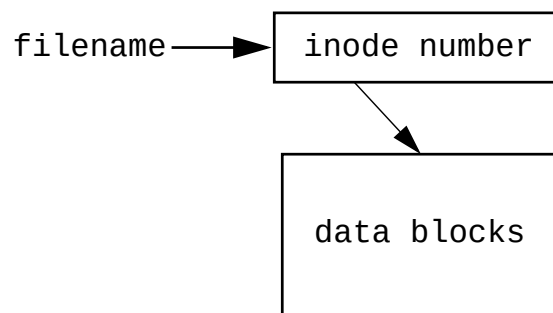


Figure 4-1 File Names, Inodes, and Data Blocks

Inodes are numbered, and each file system contains its own separate list of inodes. When you create a new file system, it generates a complete list of inodes found in that file system.

Regular Files

A regular file simply holds data. Perhaps the most common file type found in the Solaris Operating Environment are regular files, which allow you to store many different kinds of data. Regular files can hold ASCII text, binary data, image data, databases, application-related data, and more.

You can create regular files in many ways. For example, you could use `vi` to create an ASCII text file, or you could use a compiler to create a file that contains binary data. The `touch` command creates a new, empty regular file.

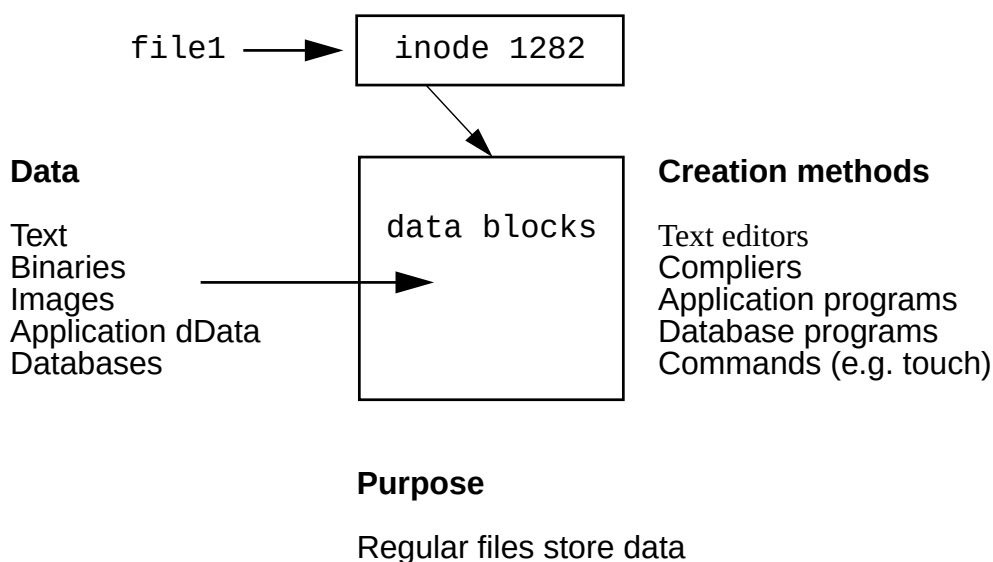


Figure 4-2 Regular Files

Figure 4-2 describes 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 kinds of data, and the file could have been created in one of many different ways.

Directories

Directories store information that associates file names with inode numbers. Unlike regular files that can hold many different kinds of data, directories can hold only one kind.

You must understand that directories themselves do not contain other files. A directory contains entries for files of all types logically found within that directory.

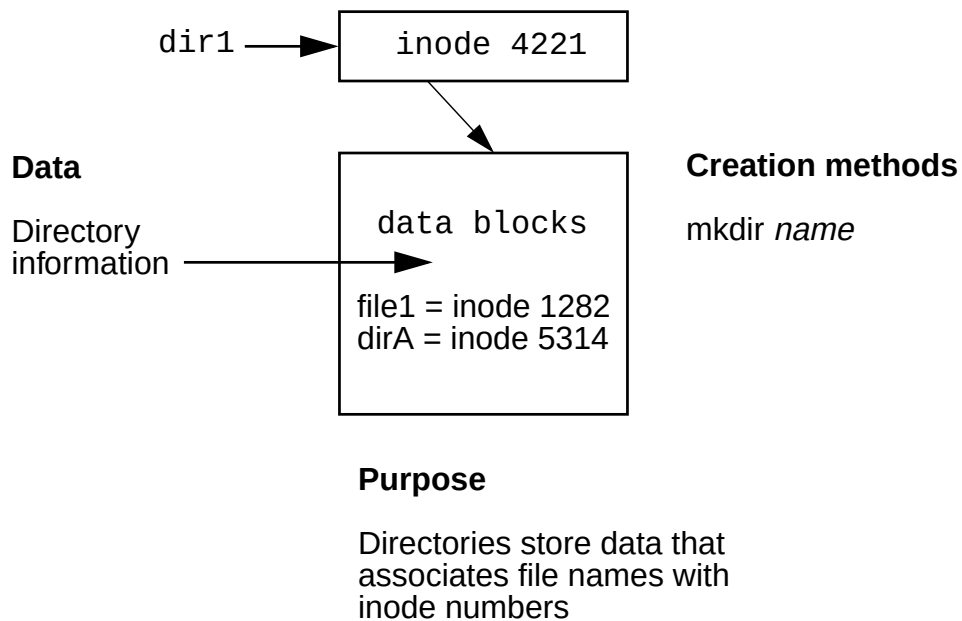


Figure 4-3 Directories

Figure 4-3 describes a directory file called `dir1`. As illustrated, the name `dir1` is associated with inode number 4221. The data blocks associated with `dir1` hold a list of file names and their associated inode numbers. The `mkdir` command creates new 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` was logically located in the directory called `dir1`, then `dir1` 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, symbolic links contain only one kind of data.

A symbolic link contains the pathname of the file to which it points. Because symbolic links use pathnames to point to other files, they can point to files found in other file systems. Also, the size of a symbolic link always matches the number of characters found in the pathname it contains.

For example, the symbolic link called `/bin` points to the directory `./usr/bin`. Its size is 9 bytes because the pathname `./usr/bin` contains nine characters.

```
# cd /
# ls -l
total 135
lrwxrwxrwx  1 root    root          9 Apr  3 10:39 bin -> ./usr/bin
(output truncated)
```

Symbolic links can point to regular files, directories, other symbolic links, and device files. And they can use absolute or relative pathnames.

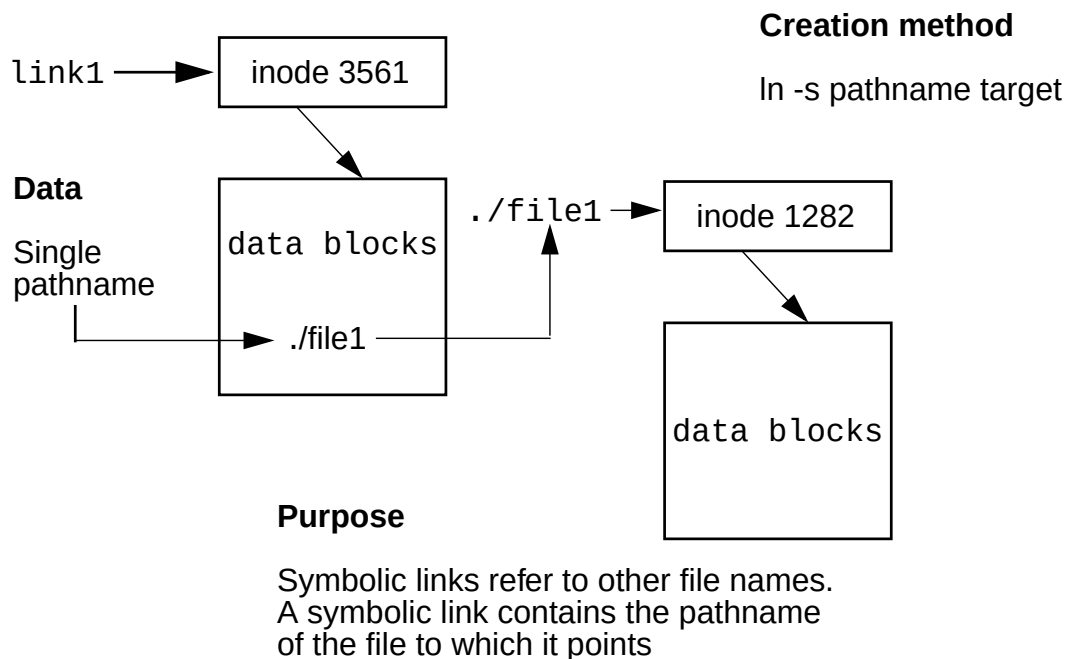


Figure 4-4 Symbolic Links

Figure 4-4 describes a symbolic link file called `link1`. As illustrated, the name `link1` is associated with inode number 3561. The data blocks associated with `link1` contain the pathname of the file to which `link1` points.

Depending on the length of the pathname the link contains, it can either reside directly in the link's inode record or in data blocks.

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

Symbolic links direct read and write operations to the file to which they point. In the example above it shows how using `link1` as a command's argument would cause that command to refer to the file called `file1`.

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, in their inode information, they hold numbers that refer to devices. Where the file size displays for other file types, listings of device files display two numbers, separated by a comma.

These two numbers are called major and minor device numbers. In the example below, the device file `dad@0,0a` refers to major device number 136 and minor device number 0.

```
# cd /devices/pci@1f,0/pci@1,1/ide@3
# ls -l
total 0
brw----- 1 root    sys      136,  0 Apr  3 11:11 dad@0,0:a
crw----- 1 root    sys      136,  0 Apr  3 11:11 dad@0,0:a,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.

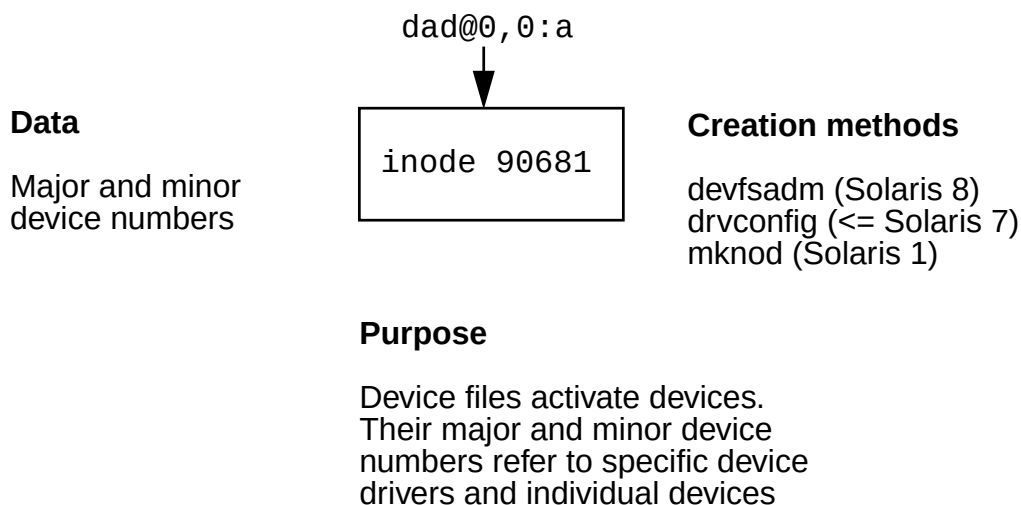


Figure 4-5 Device Files

The device file `dad@0,0:a` described in Figure 4-5 occupies inode number 90681. That inode contains the major and minor device numbers that refer to a specific device, in this case, a slice on a disk.

In general, device files are created automatically when you perform a reconfiguration reboot. In the Solaris 8 Operating Environment, you can use the `devfsadm` command to create new device files manually. Before the Solaris 8 Operating Environment you used `drvconfig`.

Information about interpreting device file names and procedures for creating device files manually and automatically are described in later modules.

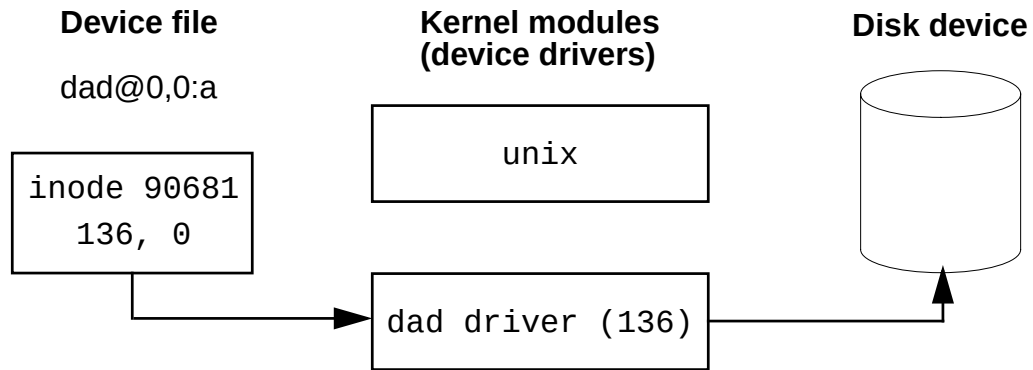


Figure 4-6 Device File Example

Figure 4-6 illustrates 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 0. Major device number 136 identifies the `dad` device driver. The `dad` device driver controls IDE disk drives. Minor number 0 identifies slice 0 of the master disk on the first IDE bus.

Device files fall into two categories: character-special devices and block-special devices. Character-special devices are also called simply *character* or *raw* devices. Block-special devices are often called simply *block* devices. These two categories of device files interact with devices differently.

Character Device Files

The file type “c” identifies character device files. For disk devices, character device files call for I/O operations based on the disks smallest addressable unit, or sectors. Each sector is 512 bytes in size.

```
crw----- 1 root    sys      136,  0 Apr  3 11:11 dad@0,0:a,raw
```


Block Device Files

The file type “b” identifies block device files. For disk devices, block device files call for I/O operations based on a defined block size. The block size depends on the particular device, but for UFS file systems, the default block size is 8 Kbytes.

```
brw----- 1 root    sys      136,  0 Apr  3 11:11 dad@0,0:a
```

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 `touch` for example, you create a new directory entry that links the file name you specify with a particular inode.

In Figure 4-7, the file called `file1` is listed in the directory `dir1`. In `dir1`, the name `file1` is associated with inode number 1282. In this way, simply creating a new file creates a hard link.

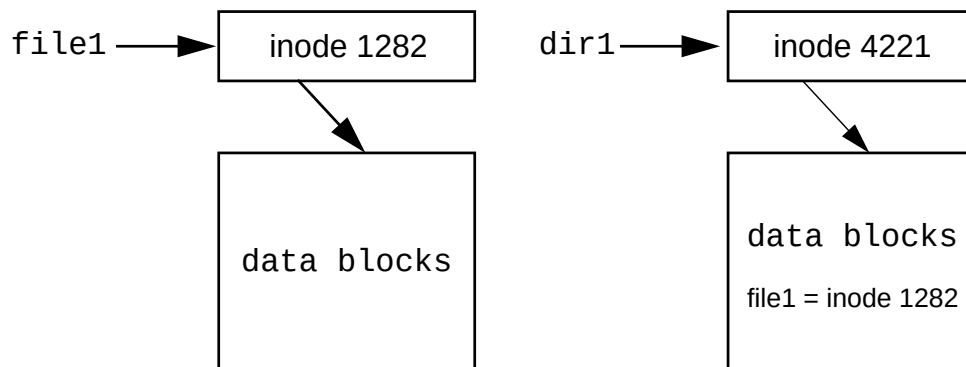


Figure 4-7 Hard Links

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 `ls -l`, the link count displays between the file permissions and the owner column. In the following example, `file1` uses one hard link.

```
# touch file1
# ls -l
total 0
-rw-r--r--  1 root    other      0 Apr  7 15:26 file1
```

Using the `ln` command, you can create new hard links to regular files. The command `ln file1 file2` creates a new directory entry called `file2`, associated with the same inode associated with `file1`.

Figure 4-8 illustrates the result, where two file names are associated with inode number 1282. These file names are functionally identical. Unlike symbolic links, hard links cannot span file systems.

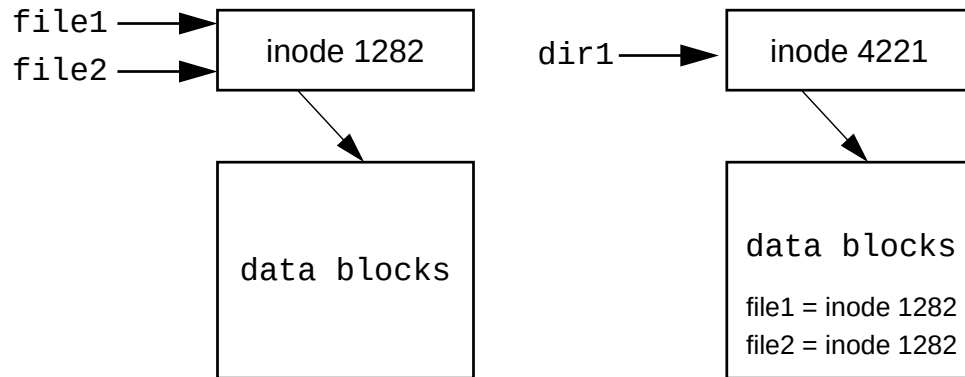


Figure 4-8 File Names Associated With an Inode Number

Creating the new hard link increments the link count. In the example below, 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.

```
# ln file1 file2
# ls -l
total 0
-rw-r--r--  2 root    other      0 Apr  7 15:26 file1
-rw-r--r--  2 root    other      0 Apr  7 15:26 file2
# ls -li
total 0
 1282 -rw-r--r--  2 root    other      0 Apr  7 15:26 file1
 1282 -rw-r--r--  2 root    other      0 Apr  7 15:26 file2
```

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

```
# rm file1
# ls -li
total 0
1282 -rw-r--r--  1 root    other      0 Apr  7 15:26 file2
```

The root Subdirectories

The directory tree is organized for administrative convenience. Branches within this tree segregate directories 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, all directories can be located on one file system or divided among more than one file system. Every Solaris Operating Environment has a root file system and can also have other file systems attached at points within the directory tree. File systems are structures created on disk slices, and they contain or hold files and directories.

The terms file systems and disk slices are only briefly explained here because they are described in detail in subsequent modules.

Note – file systems are described in Module 7. Disk slices are described in Module 6. See also, `man -s5 filesystem` for information on file system organization.

The Solaris Operating Environment is comprised of a hierarchy of critical system directories and files that are necessary for the operating system to function properly.

- / – Root of the overall file system name space.
- /bin – This directory is a symbolic link to the /usr/bin directory. It is the directory location for standard system commands, or binary files.

- `/dev` — Primary location for logical device names. These are symbolic links that point to device files in the `/devices` directory. Table 4-1 describes the contents of the `/dev` directory.

Table 4-1 The `/dev` Directory Contents

Directory	Description
<code>/dev/cua</code>	Dial out device files for uucp
<code>/dev/dsk</code>	Block disk devices
<code>/dev/fbs</code>	Frame buffer for device files
<code>/dev/fd</code>	File descriptors
<code>/dev/md</code>	Logical volume management meta-disk devices
<code>/dev/pts</code>	Pseudo terminal devices
<code>/dev/rdisk</code>	Raw disk devices
<code>/dev/rmt</code>	Raw magnetic tape devices
<code>/dev/sound</code>	Audio device and audio device control files
<code>/dev/term</code>	Serial devices

- `/devices` – Primary location for physical device names. These are device files.

- `/etc` – Host-specific system administrative configuration files and databases. Table 4-2 describes the contents of the `/etc` directory.

Table 4-2 The `/etc` Directory Contents

Directory	Description
<code>/etc/acct</code>	Accounting configuration information
<code>/etc/cron.d</code>	Configuration information for <code>cron</code>
<code>/etc/default</code>	Defaults information for various programs
<code>/etc/inet</code>	Configuration files for network services
<code>/etc/init.d</code>	Scripts for changing between run levels
<code>/etc/lib</code>	Dynamic linking libraries needed when <code>/usr</code> is not available
<code>/etc/lp</code>	Configuration information for the printer subsystem
<code>/etc/mail</code>	Mail subsystem configuration information
<code>/etc/nfs</code>	NFS server logging configuration file
<code>/etc/openwin</code>	OpenWindows™ configuration files
<code>/etc/opt</code>	Configuration information for optional packages
<code>/etc/rc#.d</code>	Scripts for entering/leaving run level #
<code>/etc/skel</code>	Default profile scripts for new user accounts

- `/export` – Default directory for commonly shared file systems, such as users home directories, client file systems, or other shared file systems.
- `/home` – Default directory or mount point for users home directories. When AutoFS is running, you cannot create any new entries in this directory.
- `/kernel` – Directory of platform-independent loadable kernel modules required as part of the boot process. It includes the generic part of the core kernel that is platform independent, `/kernel/genunix`.
- `/mnt` – Convenient, temporary mount point for file systems.

- /opt – Default directory or mount point for add-on application packages.
- /sbin – Essential executables used in the booting process and in manual system failure recovery.
- /tmp – Temporary files; cleared during boot sequence.
- /usr – Mount point for the /usr file system. This directory name is an acronym for UNIX System Resources. Table 4-3 describes the contents of the /usr directory.

Table 4-3 The /usr Directory Contents

Directory	Description
/usr/bin	Location for standard system commands
/usr/ccs	C compilation programs and libraries
/usr/demo	Demonstration programs and data
/usr/dt	Directory or mount point for CDE software
/usr/include	Header files (for C programs, and so on)
/usr/java	Directories containing Java™ technology programs and libraries
/usr/lib	Various program libraries, architecture-dependent databases, and binaries not invoked directly by the user
/usr/openwin	Directories containing OpenWindows programs
/usr/opt	Configuration information for optional packages
/usr/pub	Files for online man page and character processing
/usr/spool	Symbolic link to the /var/spool directory

- /var – Directory for varying files, which usually includes temporary, logging, or status files.

Exercise: Identifying File Types



Exercise objective – In this lab you will navigate within the directory tree and identify different types of files.

Preparation

Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

- Identify the first symbolic link listed in the `root (/)` directory. Record its size and the name of the file it references. Identify the types of files found in `/dev/dsk`, and the types of files they reference if any. Identify the types of files found in `/dev/pts`, and the types of files they reference if any.
- Identify the types of files found in `/etc/init.d`. Record the inode number and link count for the `nfs.server` file. Use the `find` command to locate all other files below `/etc` that use the same inode as `nfs.server`.
- Create a directory called `/testdir`. In this directory, create a file and a symbolic link that points to it. Determine if they use the same or different inode. Create a directory called `newdir` within `/testdir`. Identify the inode it uses, its link count, and the name of any other file that uses the same inode as `newdir`. Create another directory below `newdir`. Determine how the link count for `newdir` changes, and find any new file that uses the same inode as `newdir`.

Tasks

1. Log in as `root` 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
```

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 directory to `/dev/dsk`. Record what file types you find in this directory.

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

4. Use `ls -lL` to display information for the files referenced by the files in `/dev/dsk`. Record the file types reported by `ls -lL`

```
# ls -lL
```

5. Change directory to `/dev/pts` and use `ls` as you did in `/dev/dsk`. Record the file types you find.

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

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

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

7. In `/etc/init.d` display a long listing of the file `nfs.server`. What is the number of hard links associated with this file? What is the inode number associated with this file?

```
# ls -li nfs.server
```

8. Use the `find` command to identify all the file names found below `/etc` that use the same inode number as the file `nfs.server`. Substitute the inode number you recorded in the previous step for `<inode_number>`. How many files use this inode?

```
# find /etc -inum <inode_number>
```

9. Create a new directory called `/testdir`. Change directory to `/testdir` and create a file 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 `link1`. Do these files use the same or different inodes?

```
# ls -li
```

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

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

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

```
# ls -lia newdir
```

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

```
# mkdir newdir/dir2
# ls -ldi newdir
```

14. Use `ls` to find the new file name that uses the same inode as `newdir`. Record its name.

```
# ls -laRi newdir
```

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

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

`/bin` 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?

`/bin` contains 9 bytes of data, and points to `./usr/bin`.

3. Change directory to `/dev/dsk`. Record what file types you find in this directory.

`/dev/dsk` contains symbolic links.

4. Use `ls -lL` to display information for the files referenced by the files in `/dev/dsk`. Record the file types reported by `ls -lL`.

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

5. Change directory to `/dev/pts` and use `ls` as you did in `/dev/dsk`. Record the file types you find.

`/dev/pts` contains symbolic links.

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

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

`/etc/init.d` contains regular files.

7. In `/etc/init.d` display a long listing of the file `nfs.server`. What is the number of hard links associated with this file? What is the inode number associated with this file?

`/etc/init.d/nfs.server` has 6 hard links associated with it. The inode number will vary among different systems.

8. Use the `find` command to identify all the file names found below `/etc` that use the same inode number as the file `nfs.server`. Substitute the inode number you recorded in the previous step for the example listed. How many files use this inode?

Six files, including `nfs.server` use the same inode number. They are:

```
/etc/init.d/nfs.server
/etc/rc0.d/K28nfs.server
/etc/rc1.d/K28nfs.server
/etc/rc2.d/K28nfs.server
/etc/rc3.d/S15nfs.server
/etc/rcS.d/K28nfs.server
```

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

These two files use two separate inodes.

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

The link count for `newdir` is 2. The inode number will vary among different systems.

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

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

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

The link count increases from 2 to 3.

14. Use `ls` to find the new file name that uses the same inode as `newdir`. Record its name.

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

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Identify the four main file types in the Solaris Operating Environment
- ☐ Describe the functions provided by regular files, directories, symbolic links, device files, and hard links
- ☐ Define the function of each subdirectory found directly within the root directory

Objectives

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

- Describe the disk components: sectors, tracks, and cylinders
- Define the term disk slice
- Identify a disk device by its logical device name, physical device name, and instance name
- Describe the purpose of the `/etc/path_to_inst` file
- List a system's device configuration information using the `prtconf` command
- Display the system's current disk configuration using the `format` commands
- Show how to invoke a reconfiguration boot after adding a peripheral device to the system
- Describe how devices are reconfigured using the `devfsadm` command

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10

Basic Architecture of a Disk

The following sections describe the architecture of a disk.

Physical Disk Structure

A disk is physically composed of a series of flat, magnetically coated platters stacked on a spindle. The spindle turns while the read/write heads move between platters, in unison, radially reading and writing data on the platters.

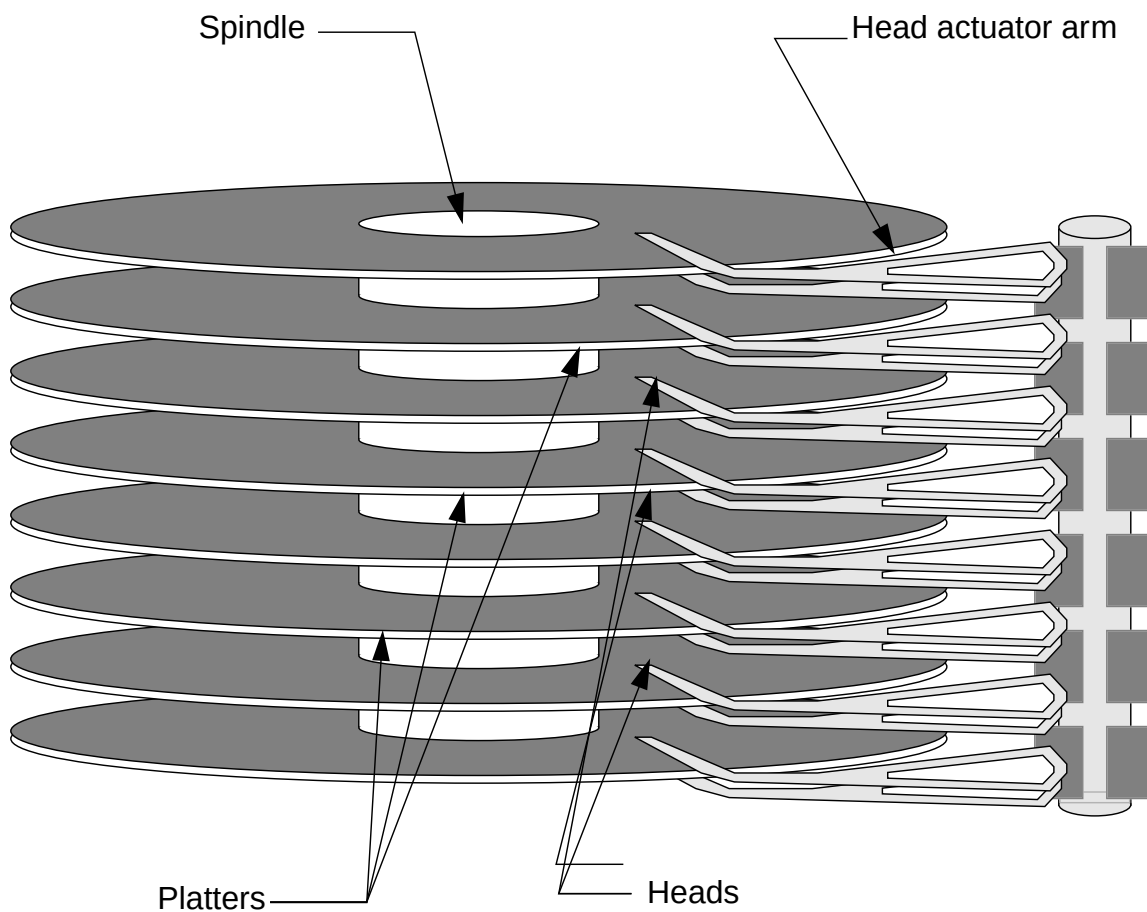


Figure 5-1 Components of a Disk

The following describes the components of a disk:

- One or more *platters*.
- Platters rotate around the *spindle*.
- *Head actuator arm* moves the *read/write heads* as a unit above and below each platter.

Components of a Disk Platter

A disk is divided into the following components: 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.

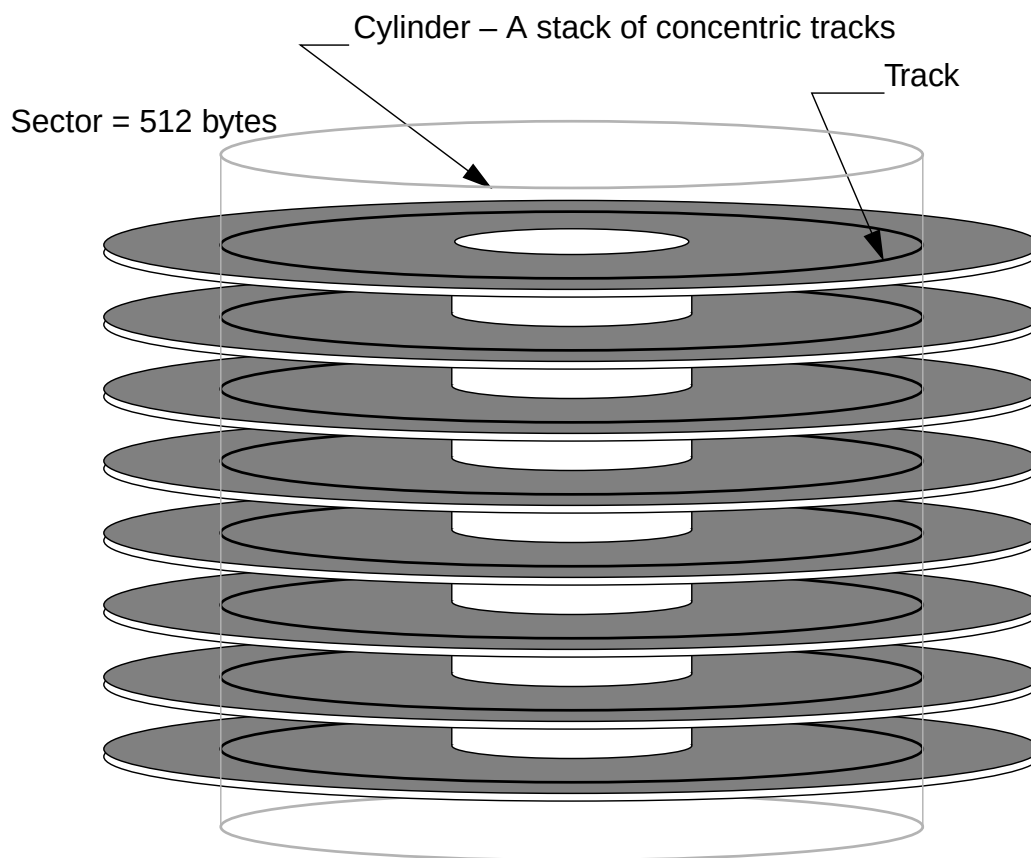


Figure 5-2 Components of a Disk Platter

Note – 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 or written to are located in a single cylinder.

Defining Disk Slices

Disks can be divided into individual partitions, known as *slices*. Slices are groupings of cylinders commonly used to organize data by function.

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

Note – By grouping cylinders in this way, the amount of movement required by the read/write heads to access a file is reduced, which improves disk I/O performance.

A disk under SunOS can be divided into eight slices, labeled slice 0 through slice 7.

By convention, slice 2 is used to represent the entire disk. It records items, such as the size of the actual disk, and the total number of cylinders available for the storage of files and directories.

The Boot Disk

The slices shown in Figure 5-3 are a possible configuration convention for logically organizing data that is to be stored on the boot disk. Not all slices have to be defined on a disk.

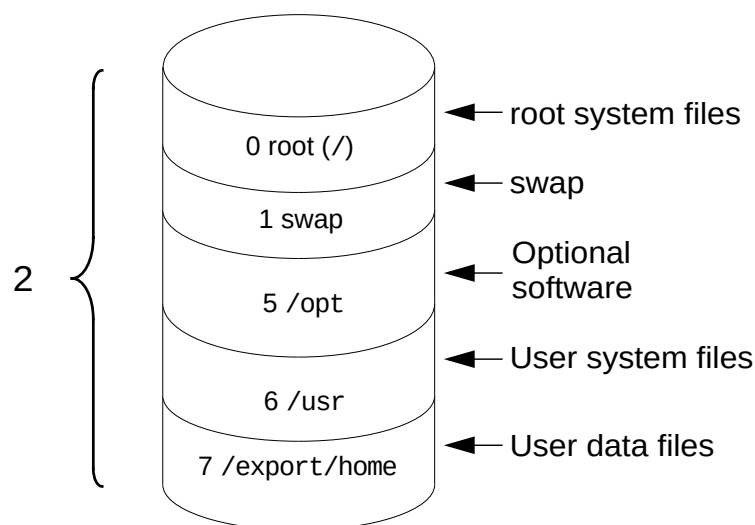


Figure 5-3 Disk Slices on a Single Disk System

Table 5-1 identifies the disk slices.

Table 5-1 Disk Slices

Slice	Name	Function
0	/	root'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 5-4 illustrates how the above slices reside on the disk. Each slice is defined by a starting cylinder and an ending cylinder. These cylinder boundaries determine the size of a slice.

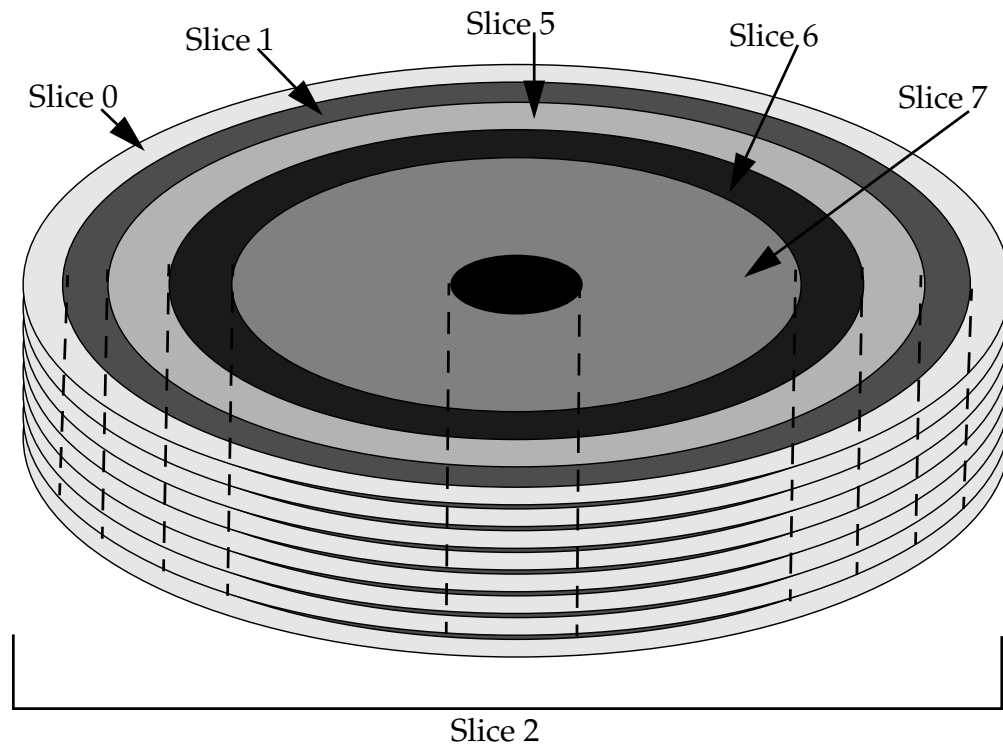


Figure 5-4 Top View of Five Disk Slices

Disk Slice Naming Convention

The full name of a slice is represented by an eight-character string which includes the controller number, the target number, the disk number, and the slice number.

- Controller number – Identifies the host bus adapter, which controls communications between the system and disk unit. It takes care of moving disk heads, data transfer, and location of data on 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 address switch setting that is selected for each disk, tape, or CD-ROM. An external disk drive has an address switch located on the rear panel. An internal disk has address pins which are jumpered to assign its 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. The disk number is always set to d0 with embedded SCSI disks.
- Slice number — A slice number ranging from 0 to 7.

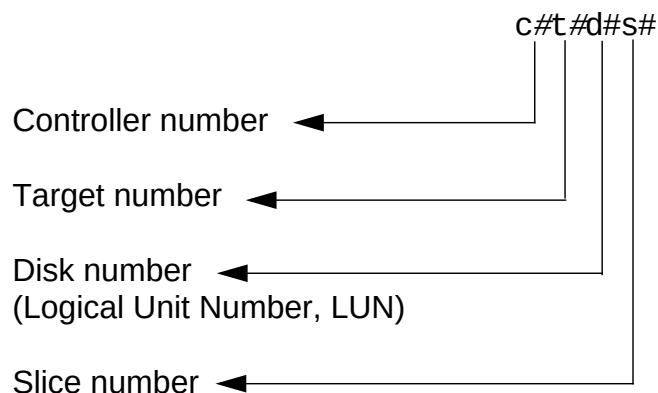


Figure 5-5 Disk Slice Naming Conventions

Figure 5-6 illustrates an embedded SCSI configuration.

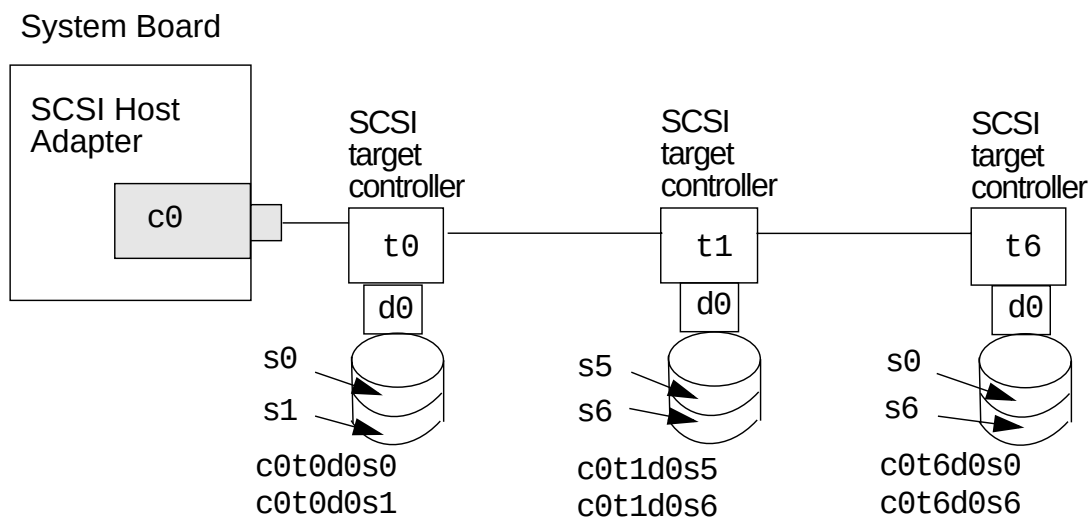


Figure 5-6 Embedded SCSI Configuration

Figure 5-7 illustrates an IDE configuration.

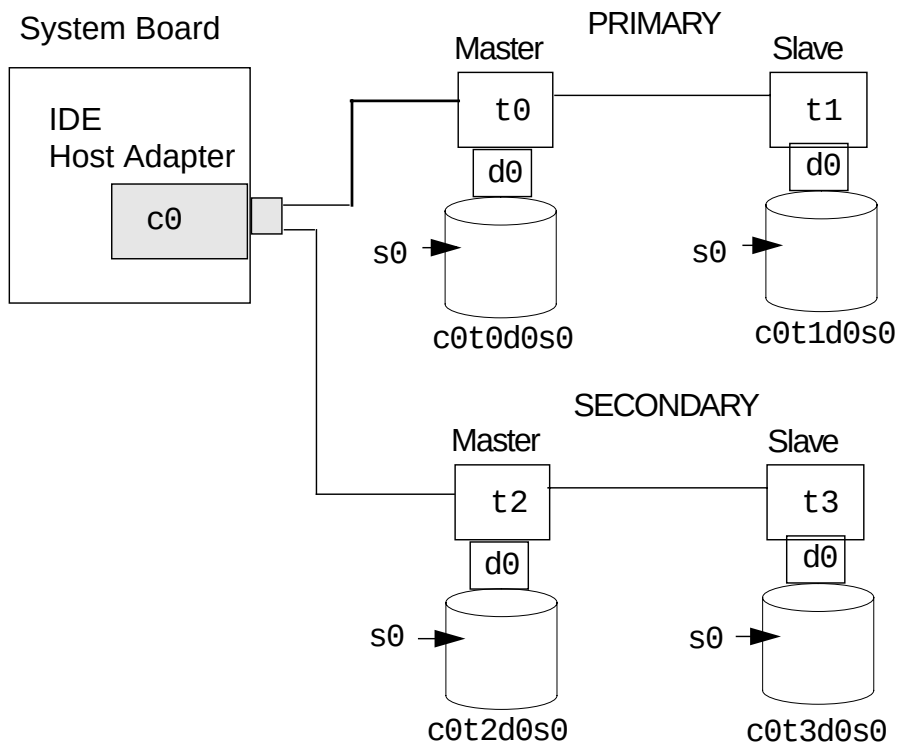


Figure 5-7 IDE Configuration

Device Naming Conventions

In the Solaris Operating Environment, all devices have three different types of names, depending on how the device is being referenced.

- Logical device names
- Physical device names
- Instance names

Note – BSD device names also exist in the Solaris Operating Environment if the BSD compatibility packages are installed with either the Developer, Entire Distribution, or Entire Distribution plus OEM Solaris Software Group. The BSD device names are typically used for backwards compatibility with old scripts, (for example, /dev/sd0a).

Logical Device Names

You use logical device names, and in some cases by regular users, primarily to refer to a device on the command line.

All logical device names are kept in the /dev directory.

Logical device names are symbolic links to the physical device names kept in the /devices directory.

The logical disk 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/rdisk directories, for the block and character (raw) disk devices respectively. For example:

```
# ls /dev/dsk
c0t0d0s0  c0t0d0s4  c0t3d0s0  c0t3d0s4  c0t6d0s0  c0t6d0s4
c0t0d0s1  c0t0d0s5  c0t3d0s1  c0t3d0s5  c0t6d0s1  c0t6d0s5
c0t0d0s2  c0t0d0s6  c0t3d0s2  c0t3d0s6  c0t6d0s2  c0t6d0s6
c0t0d0s3  c0t0d0s7  c0t3d0s3  c0t3d0s7  c0t6d0s3  c0t6d0s7
```

- `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.
- `c0t3d0s0` through `c0t3d0s7` — Identifies the device names for disk slices 0 through 7, for a disk that is attached to controller 0, at target 3, on disk unit 0.
- `c0t6d0s0` through `c0t6d0s7` — Identifies the device names for disk slices 0 through 7. Normally, CD-ROM devices are treated the same as disks. This indicates a device on controller 0, at target 6, and 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.

Note – Various hardware platforms have different device trees.

A physical device name uniquely identifies the location of the device. It contains the hardware information, represented as a series of node names, separated by slashes, to indicate the path to the device that reflects hardware connectivity. For example:

```
# ls -l /dev/dsk/c0t0d0s0
lrwxrwxrwx  1 root      root 46 Jun 16 19:07 /dev/dsk/c0t0d0s0 ->
../../../../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a
```

For example, an Ultra 5 system has the device configuration tree structure shown in Figure 5-8 (not all possible devices are included).

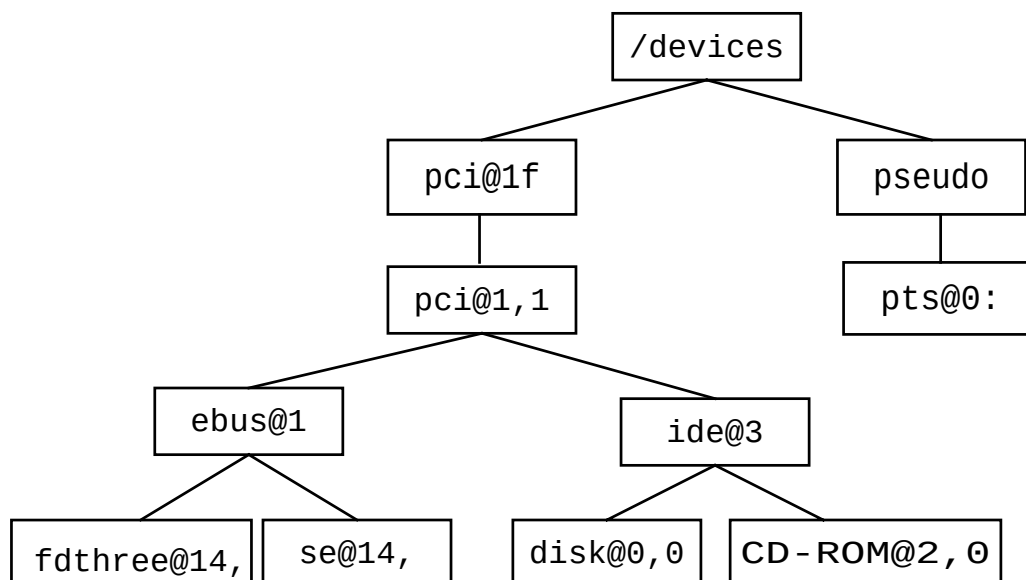


Figure 5-8 The /devices Directory Structure

The top-most directory in the hierarchy is called the *root node* of the device tree. An object below the root node has a device driver associated with it, which is called a *leaf*, or *bus nexus* node.

Note – A device driver is the software that communicates with the device. This software must be available to the kernel to use the device.

The kernel identifies the physical location of a device by associating a node with an address, *nodename@address*, which is called the physical device name, for example, *dad@0*.

Instance Names

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

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

`sdn`

where `sd` is the disk name and *n* is the disk number, such as `sd0`, for the first SCSI (small computer system interface) disk device:

`dadn`

where `dad` (direct access device) is the disk name and *n* is the disk number, such as `dad0`, for the first `ide` (integrated drive electronics) disk device.

Listing a System's Devices

The following sections describe how to list a system's devices.

The /etc/path_to_inst File

In the Solaris Operating Environment, the system records, for each device, its instance name and number along with its physical 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.

Note – The device instance number, shown in bold below, appears to the right of the device instance name when recorded in this file.

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 ever change this file.

```
# more /etc/path_to_inst
#
#      Caution! This file contains critical kernel state
#
"/pci@1f,0" 0 "pci"
"/pci@1f,0/pci@1,1/ide@3/sd@2,0" 2 "sd" (CD-ROM)
"/pci@1f,0/pci@1,1/ide@3/dad@0,0" 0 "dad" (disk)
"/pci@1f,0/pci@1,1/ebus@1" 0 "ebus" (extended bus)
"/pci@1f,0/pci@1,1/ebus@1/fdthree@14,3023f0" 0 "fd" (floppy disk)
"/pci@1f,0/pci@1,1/ebus@1/su@14,3062f8" 1 "su" (mouse)
"/pci@1f,0/pci@1,1/ebus@1/se@14,400000" 0 "se" (serial ports A and B)
"/pci@1f,0/pci@1,1/ebus@1/su@14,3083f8" 0 "su" (keyboard)
"/pci@1f,0/pci@1,1/ebus@1/ecpp@14,3043bc" 0 "ecpp" (extended
capability parallel port)
"/pci@1f,0/pci@1,1/ebus@1/SUNW,CS4231@14,200000" 0 "audiocs" (crystal
semiconductor)
"/pci@1f,0/pci@1,1/ebus@1/power@14,724000" 0 "power" (power management
bus)
"/pci@1f,0/pci@1,1/network@1,1" 0 "hme" (Fast-Ethernet)
"/pci@1f,0/pci@1,1/SUNW,m64B@2" 0 "m64" (color memory frame buffer)
"/pci@1f,0/pci@1" 1 "simba" (pci bus A controller)
"/options" 0 "options"
"/pseudo" 0 "pseudo"
```

Note – Different systems have different physical device paths. This example shows an onboard peripheral component interconnect (PCI) bus configuration.

Sample /etc/path_to_inst File

The following is a `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 onboard Sun system bus (Sbus).

```
# more /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 prtconf Command

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

The `prtconf` command lists all instances of devices, whether the device is attached or not attached to the system.

To view only a list of attached devices on the system, execute the following commands.

```
# prtconf | grep -v not
```

System Configuration: Sun Microsystems sun4u

Memory size: 64 Megabytes

System Peripherals (Software Nodes):

SUNW,Ultra-5_10

options, instance #0

pci, instance #0

pci, instance #0

ebus, instance #0

power, instance #0

se, instance #0

su, instance #0

su, instance #1

fdthree, instance #0

network, instance #0

SUNW,m64B, instance #0

ide, instance #0

dad, instance #0

sd, instance #2

pci, instance #1

pseudo, instance #0

Note – The command `grep -v not` is used to omit all lines containing the word “not” from the output.

The format Command

You use the `format` command to display both logical and physical device names for all currently available disks. For example:

```
# format
```

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

```
AVAILABLE DISK SELECTIONS:
```

```
0. c0t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
```

```
   /pci@1f,0/pci@1,1/ide@e/dad@0,0
```

```
1. c1t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
```

```
   /pci@if,0/pci@1/isptwo@4/sd@3,0
```

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

Note – Press `Control+d` to exit the `format` command.

Reconfiguring Devices

The system recognizes a newly added peripheral device if a *reconfiguration boot* is invoked. This particular boot process adds the new device to a newly generated device tree and to the `/dev` and `/devices` directories.

The following steps reconfigure a system to recognize a newly attached 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. This command brings the system to an appropriate state for turning the system power off to safely allow for adding or removing devices.

```
# init 5
```

3. Turn off the power to all external devices.
4. Install the peripheral device, making sure the device being added has no conflicting address with other devices on the system.
5. Turn on the power to all external devices.
6. Turn on the power to the system. The system boots to the login screen.
7. Verify that the peripheral device has been added by issuing one of the following commands: `prtconf` or `format`.

Once the disk is recognized by the system, you can 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 PROM level command: `boot -r`.

Configuring the Solaris 8 Operating Environment Devices

Before the Solaris 8 Operating Environment release, you used the `drvconfig` command to configure devices. This command managed the physical device entries in the `/devices` directory. The commands `disks`, `tapes`, `devlinks`, and `ports` manage the logical device entries in the `/dev` directory.

Note – The `ports` command creates `/dev` entries for serial lines.

Now, both the reconfiguration boot process and the updating of the `/dev` and `/devices` directories for dynamic reconfiguration events are handled by the `devfsadm` command.

For compatibility purposes, `drvconfig` and the other commands are symbolic links to `devfsadm`.

The `devfsadm` command attempts to load every driver in the system and attach to 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, `devfsadm` also maintains the `/etc/path_to_inst` file.

`devfsadm` Options

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

```
devfsadm -c device_class
```

where the values to *device_class* include: `disk`, `tape`, `port`, `audio`, and `pseudo`. For example:

```
# devfsadm -c disk
```

You can use the `-c` option more than once on the command line to specify multiple device classes. For example:

```
# 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. For example:

```
devfsadm -i driver_name
```

Some examples of using the `-i` option include:

- To configure only those disks supported by the `dad` driver:

```
# devfsadm -i dad
```

- To configure only those disks support by the `sd` driver:

```
# devfsadm -i sd
```

- To configure devices supported by the `st` driver:

```
# devfsadm -i st
```

Configuring a Device Before the Solaris 8 Operating Environment

You can also use the `drvconfig` command to reconfigure the system to recognize new devices without rebooting.

By default, this command configures the `/devices` directory with the physical device name(s) of the newly attached device(s) and updates the `/etc/path_to_inst` file.

Adding a New Disk or Tape Drive

Commonly, the types of peripheral devices added to a workstation are disks and tape drives.

- When adding a new disk, you must issue the `disk` command to create the `/dev` entries for the newly attached disk(s).
- When adding a tape drive, you must issue the `tape` command to create the `/dev` entries for the newly attached tape drive(s).

Note – If adding miscellaneous devices or pseudo-devices, you use the `devlinks` command to add `/dev` entries for the new devices.

Adding a New Disk Device

The following steps illustrate how to add a new disk device:

1. Invoke the `drvconfig` command.

```
# drvconfig -i dad
```

or

```
# drvconfig -i sd
```

2. Invoke the `disks` command.

disks

This command creates symbolic links in the `/dev/dsk` and `/dev/rdisk` directories pointing to the actual disk device files located in the `/devices` directory.

Adding a New Tape Drive

The following steps illustrate how to add a new tape drive:

1. Invoke the `drvconfig` command.

drvconfig -i st

2. Invoke the `tapes` command.

tapes

This command creates symbolic links in the `/dev/rmt` directory to the actual tape device files located in the `/devices` directory.

Exercise: Configuring and Naming Disks



Exercise objective – In this lab you will identify logical, physical, and instance names for disk devices, add a new disk or tape to a system, and create new device files for it.

Preparation

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

Task Summary

- Identify the logical device name of your boot disk. Locate the logical device files in `/dev/dsk` and `/dev/rdisk` 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 you've found. 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 you want to connect. Halt the system and power on the device. Boot the system to its default run state. Run `devfsadm` in verbose mode to create new links and device files and confirm they exist.

Tasks

Identifying Device Files

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

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

2. List the files in this directory. Identify the files related to the boot disk of your system. Most systems will use either `c0t0d0` or `c0t3d0`, depending on their type and configuration. Locate the item related to slice 0 on this disk, and display a long listing of it. Example:

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

What type of file did you just locate? The file type indicator is the first character (left side) found in the long listing.

Record the full pathname to which this file points.

3. Highlight the pathname you recorded above. Double-click on the pathname using the left mouse button to do this. Use the Copy and Paste keys to paste this pathname into a long listing command. If you're not using CDE, you'll need to type in the pathname.

```
# ls -l <pathname>
```

What type of file is this?

The command `ls -lL c0t0d0s0` displays the same information, but only shows the link file name (for example, `c0t0d0s0`) rather than the real device file name.

4. Change directory to `/dev/rdisk`. Display a long listing of the same file name you selected in step 2 (either `c0t0d0s0` or `c0t3d0s0`). Example:

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

What type of file is this?

Record the full pathname to which this file points.

5. Highlight the pathname you recorded in step 4. Use the copy and paste keys to paste this pathname into a long listing command. If you're not using CDE, you'll need to type in the pathname.

```
# ls -l <pathname>
```

What type of file is this?

The command `ls -lL c0t3d0s0` displays the same information, but only shows the link file name (e.g., `c0t3d0s0`) rather than the real device file name.

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

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

7. Locate and record the entry for your boot disk. Use the information from the previous steps to know what to look for. For example, a `sun4u` system would use `c0t0d0` as its boot disk. This relates to the device file called `dad@0,0`, and is listed in `/etc/path_to_inst`.
-

The instance name is composed of the `dad` or `sd` tag and the number that precedes it in `/etc/path_to_inst`. What is the instance name for the device listed in step 7?

Adding a New Disk or Tape Device

8. In `/dev/dsk` and `/dev/rdisk`, or in `/dev/rmt`, confirm that no files exist for your external disk or tape device (for example, `/dev/dsk/c1t3d0s0` or `/dev/rmt/0`). If files for the external device do exist, your instructor will provide directions to remove them.
9. Shut down your system to run state 0.
init 0
10. Power on the external disk or tape attached to your system.
11. Boot the system to its default run state.
ok boot
12. Log in in as `root` and open a terminal window. Run `devfsadm` to create new links and device files for the new disk. Observe the messages `devfsadm` displays.
devfsadm -v
13. Confirm that new links and device files exist in `/dev/dsk` and `/dev/rdisk` for disks, `/dev/rmt` for tapes, and below `/devices` for both.

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

2. What type of file did you just locate? The file type indicator is the first character (left-hand side) found in the long listing.

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

Record the full pathname to which this file points.

Systems using PCI bus architectures will list pathnames similar to the following:

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

Systems using sbus architectures will list pathnames similar to the following:

```
../../../../devices/iommu@0,10000000/sbus@0,10001000/es  
pdma@5,8400000/esp@5,8800000/sd@3,0:a
```

3. Highlight the pathname you recorded above. Double-click on the pathname using the left mouse button to do this. Use the Copy and Paste keys to paste this pathname into a long listing command. If you're not using CDE, you'll need to type in the pathname.

What 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.*

4. Change directory to `/dev/rdisk`. Display a long listing of the same file name you selected in step 2 (either `c0t3d0s0` or `c0t0d0s0`). Example:

What type of file is this?

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

Record the full pathname to which this file points.

Systems using PCI bus architectures will list pathnames similar to the following:

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

Systems using sbus architectures will list pathnames similar to the following:

```
../../../../devices/iommu@0,100000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/sd@3,0:a,raw
```

5. Highlight the pathname you recorded in step 4. Use the copy and paste keys to paste this pathname into a long listing command. If you're not using CDE, you'll need to type in the pathname.

What 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.

7. Locate and record the entry for your boot disk. Use the information from the previous steps to know what to look for. For example, a sun4u system would use c0t0d0 as its boot disk. This relates to the device file called dad@0,0, and is listed in /etc/path_to_inst

Systems using PCI bus architectures will list pathnames similar to the following:

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

Systems using sbus architectures will list pathnames similar to the following:

```
/iommu@0,100000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/sd@3,0
```

The instance name is composed of the dad or sd tag and the number that precedes it in /etc/path_to_inst. What is the instance name for the device listed in step 7?

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

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Describe the disk components: sectors, tracks, and cylinders
- ☐ Define the term disk slice
- ☐ Identify a disk device by its logical device name, physical device name, and instance name
- ☐ Describe the purpose of the `/etc/path_to_inst` file
- ☐ List a system's device configuration information using the `prtconf` command
- ☐ Display the system's current disk configuration using the `format` commands
- ☐ Show how to invoke a reconfiguration boot after adding a peripheral device to the system
- ☐ Describe how devices are reconfigured using the `devfsadm` command

Objectives

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

- Explain the term disk slice
- Describe and create a disk label
- Define and modify a partition table using the `format` utility
- Describe the purpose of the `/etc/format.dat` file
- Use the `format` utility to save and retrieve customized partition tables
- Demonstrate how to view the disk's VTOC using two different commands: `verify` and `prtvtoc`
- Use the `fmthard` command to update the VTOC on a disk

Additional Resources



Additional resources – The following references can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10
- *Solaris 8 System Administration Guide, Volume II*, Part Number 805-7229-10

Disk Slices and the format Utility

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

Though you can use the `format` utility to perform a variety of disk management activities, the main reason you use the `format` utility is to divide a disk into disk slices.

Note – The Solaris Operating Environment installation program also divides disks into disk slices as part of installing the Solaris Operating Environment release.

To divide a disk into slices, the system administrator will need to:

- Identify the correct disk
- Plan the layout of the disk
- Use the `format` utility to divide into slices
- Label the disk with new slice information

Only the `root` user can use the `format` utility. If `format` is run by a regular user, the following error message is displayed:

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

Disk Labels and Partition Tables

Every disk in the Solaris Operating Environment has a special area set aside for storing information about the disk's controller, geometry, and slices.

This information is called the disk's *label*. 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 the system administrator fails 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.

Disk Partition Table

A disk's partition table can be displayed using the `format` utility.

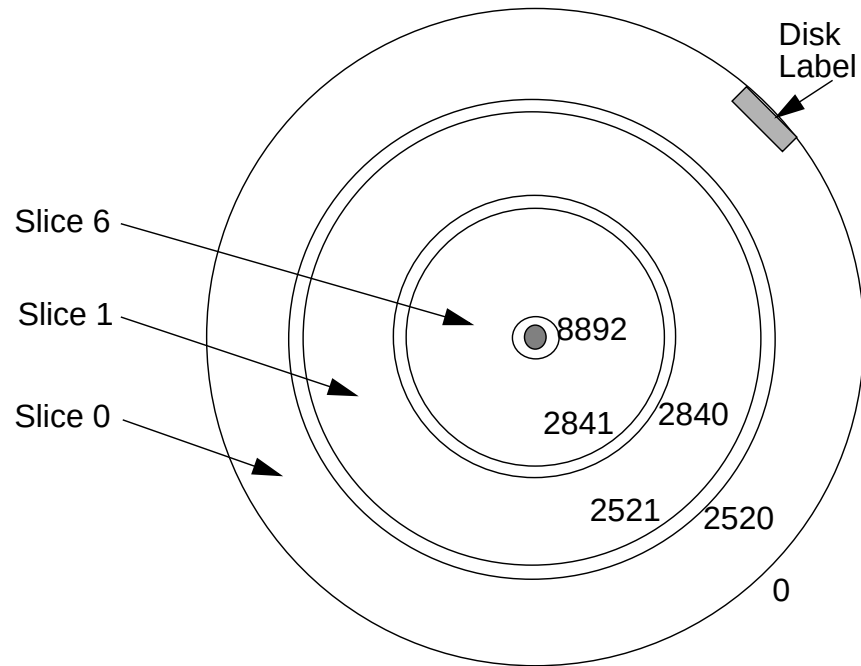


Figure 6-1 A Partitioned Disk

The partition table primarily defines partition boundaries and the number of cylinders in a partition. For example:

Current partition table (original):

Total disk cylinders available 8892 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 2520	1.14GB	(2521/0/0) 2382345
1	swap	wu	2521 - 2840	147.66MB	(320/0/0) 302400
2	backup	wm	0 - 8892	4.01GB	(8892/0/0) 8402940
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	2841 - 8892	2.73GB	(6051/0/0) 5718195
7	unassigned	wm	0	0	(0/0/0) 0

Partition boundaries must begin and end with entire cylinders.

Table 6-1 describes the fields contained in a disk's partition table.

Table 6-1 Partition Table Terms and Usage

Field	Description
Part	Slice number. Valid slice numbers include 0 through 7.
Tag	A value used to indicate how the slice is being used. 0 = unassigned 1 = boot 2 = root 3 = swap 4 = usr 5 = backup 6 = stand 8 = home StorEdge Volume Manager tags: 14 = public (region) 15 = private (region)
Flag	wm = disk slice is writable and mountable. wu = disk slice is writable and unmountable. <i>This is the default state of slices dedicated for swap areas.</i> rm = disk slice is read only and mountable.
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.

Note – Partition tags and flags are assigned by convention. They are ignored by the SunOS and require no maintenance.

Defining Disk Slices

Disk slices are defined by an offset and a size in cylinders. The offset is the distance from cylinder 0. For example:

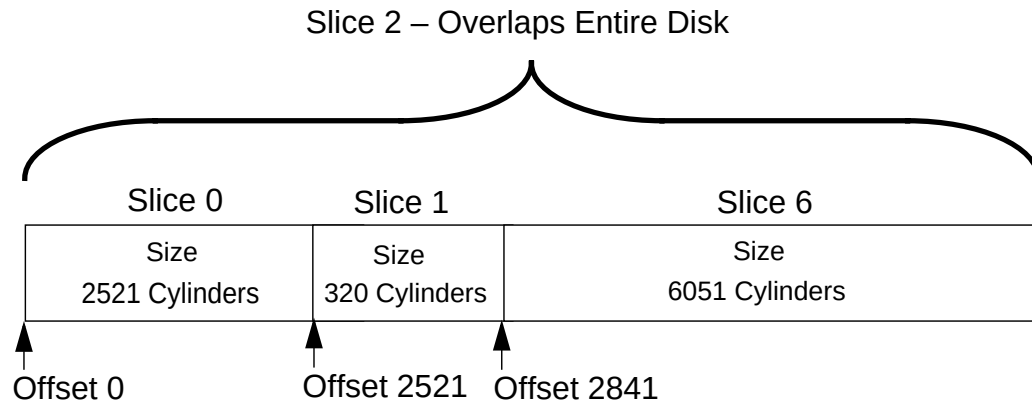


Figure 6-2 Offsets and Sizes for Disk Partitions

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 8892.

Defining Disk Partitions

The following sections describe conditions that can occur when you are defining disk partitions.

Undesirable Conditions

When creating or changing disk slices, two types of undesirable conditions can occur: wasted disk space and overlapping disk space.

Wasted Disk Space

Wasted disk space occurs when one or more cylinders are not allocated to a disk slice.

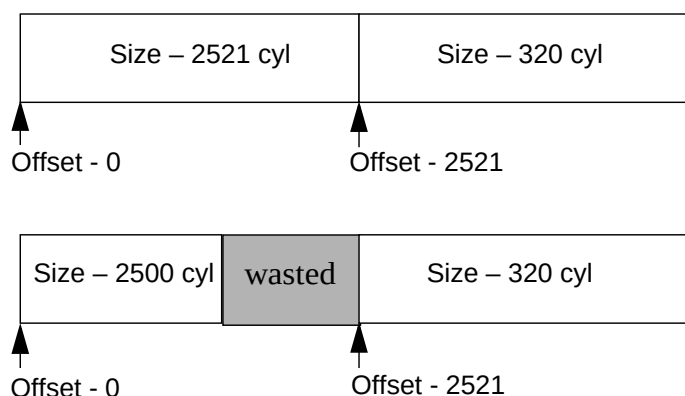


Figure 6-3 Disk Slice With Wasted Space

The wasted disk space condition can occur when you decrease the size of one slice, and do not adjust the starting cylinder number of the next disk slice. (*In the example above, cylinders 2501 through 2520 are unusable.*)

Overlapping Disk Slices

Overlapping disk slices occurs when one or more cylinders are allocated to more than one disk slice.



Figure 6-4 Disk Slices With Overlapping Cylinders

This type of condition can occur when you increase the size of one slice and do not adjust the starting cylinder number of the next disk slice. In the example above, cylinders 2521 through 2590 are overlapping two disk slices. The `format` utility does not warn you of overlapping disk slices.

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



Caution – When a disk with existing slices is repartitioned and relabeled, any existing data will be inaccessible. Existing data must be copied to backup media before the disk is repartitioned and restored after the disk is relabeled.

Locations of Disk Partition Tables

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

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

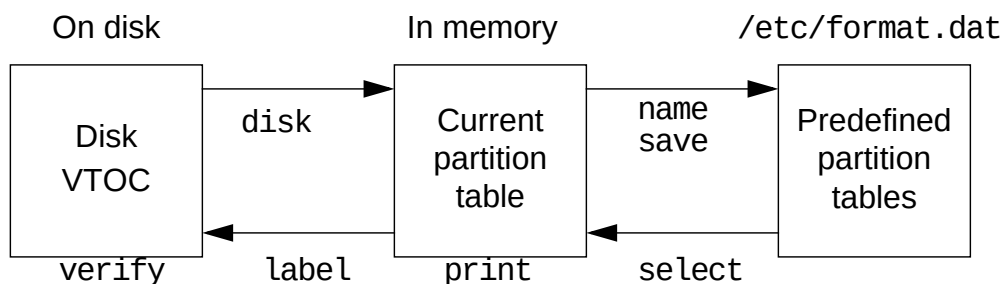


Figure 6-5 Partition Table Locations

You can select a predefined partition table from `/etc/format.dat` to be read in as the disk's current label by using the following commands within the `format` utility.

- `select` – Selects a predefined partition table from the list of tables stored in `/etc/format.dat`.
- `print` – Displays the selected partition table.
- `label` – Writes the selected partition table to the disk's label.
- `verify` – Locates the disk's label and displays the new information.

You can also save a modified partition table to the `/etc/format.dat` file for later use on other disks by using the commands within `format`.

- `disk` – Selects a disk
- `name` – Creates a name for the modified partition table
- `save` – Saves the named table to `./format.dat` for future use

The `format` utility, by default, saves disk labels in `./format.dat`.

Disk Partitioning

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

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

format

Searching for disks...done

AVAILABLE DISK SELECTIONS:

- 0. **c0t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>**
/pci@if,4000/pci@1,1/ide@3/dad@0,0
- 1. **c1t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>**
/pci@if,4000/pci@1/isptwo@4/sd@3,0

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

The `format` utility searches for all attached disks. For each disk found, `format` displays its logical device name, hardware 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 (for example, 1).

The `format` utility's main menu is displayed.

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
show	- translate a disk address
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
volname	- set 8-character volume name
!<cmd>	- execute <cmd>, then return
quit	

The specific menu selections that are used to divide a disk into slices include:

- ▼ `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.

```
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 is displayed. This menu enables you to perform the following functions:

- ▼ `0-7` – Specifies the offset and size of up to eight partitions
- ▼ `select` – Chooses a predefined partition table from `/etc/format.dat`
- ▼ `modify` – Changes a predefined partition table
- ▼ `name` – Identifies the current partition table
- ▼ `print` – Displays the current partition table
- ▼ `label` – Writes the current partition table to the disk label

4. Type `print` at the partition prompt to display the disk label that was copied to RAM when the `format` utility was started.

Current partition table (original):

Total disk cylinders available: 2036 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks	
0	root	wm	0 - 2520	1.14GB	(2521/0/0)	2382345
1	swap	wu	2521 - 2840	147.66MB	(320/0/0)	302400
2	backup	wm	0 - 8892	4.01GB	(8892/0/0)	8402940
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	2841 - 8892	2.73GB	(6051/0/0)	5718195
7	unassigned	wm	0	0	(0/0/0)	0

The name of the partition table is displayed 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 (*Mbytes, Gbytes, Blocks, or Cylinders*)
- ▼ Blocks – The total number of cylinders and the total number of sectors per slice

5. Type **0** (zero) to select slice 0.

```
partition> 0
Part      Tag  Flag  Cylinders      Size      Blocks
0         root  wm    0 - 1830      901.20MB  (1831/0/0)  1845640
Enter partition id tag[root]: <press Return>
Enter partition permission flags[wm]: <press Return>
Enter new starting cyl[0]: <press Return>
Enter partition size[1845648b, 1831c, 901.20mb]: 400mb
```

6. When prompted for the ID tag, enter a question mark (?) and press Return, to list the available choices. *A tag can be changed by typing a new tag name.*

Enter partition id tag[root]: ?

Expecting one of the following: (abbreviations ok):

unassigned	boot	root	swap
usr	backup	stand	var
home	alternates		

Enter partition id tag[root]:

7. Press the Return key to except the default tag.
8. When prompted for the permission flags, enter a question mark (?) and press Return, to list the available choices. *A flag can be changed by typing 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]:

9. Press the Return key to except the default flags.
10. Press the Return key to except the starting cylinder of 0 (zero).
11. Enter the new partition size for slice 0.
12. Type **print**.

partition> **print**

Current partition table (unnamed):

Total disk cylinders available: 2036 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks	
0	root	wm	0 - 2520	1.14GB	(2521/0/0)	2382345
1	swap	wu	2521 - 2840	147.66MB	(320/0/0)	302400
2	backup	wm	0 - 8892	4.01GB	(8892/0/0)	8402940
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	2841 - 8892	2.73GB	(6051/0/0)	5718195
7	unassigned	wm	0	0	(0/0/0)	0

The current partition table shows the change to slice 0.

This change has resulted in wasted disk space between slice 0 and slice 1. To remove this undesirable condition, adjust the starting cylinder for the next slice.

13. Type **1** to select slice number 1.

```
partition> 1
Part      Tag  Flag  Cylinders      Size      Blocks
0        swap  wu    1831 - 1983    75.30MB   (153/0/0)   154213
Enter partition id tag[swap]:
Enter partition permission flags[wu]:
Enter new starting cyl[1831]: 813
Enter partition size[154224b, 153c, 75.30mb]: 60mb
```

14. Press the Return key to select the default tag and the default flags.

15. Enter the new starting cylinder for slice 1.

16. Enter the new partition size for slice 1.

17. Type **print**.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 2036 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 2520	1.14GB	(2521/0/0) 2382345
1	swap	wu	2521 - 2840	147.66MB	(320/0/0) 302400
2	backup	wm	0 - 8892	4.01GB	(8892/0/0) 8402940
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	2841 - 8892	2.73GB	(6051/0/0) 5718195
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 partition 0.

This change has resulted in wasted disk space between slice 1 and slice 7. To remove this undesirable condition adjust the starting cylinder for the next slice.

18. Type **7** to select slice number 7.

```
partition> 7
Part      Tag  Flag  Cylinders      Size      Blocks
7        home  wm    1984 - 2034    25.10MB    (51/0/0)    51404
Enter partition id tag[home]:
Enter partition permission flags[wm]:
Enter new starting cyl[1831]: 935
Enter partition size[154224b, 153c, 75.30mb]: $
```

19. Press the Return key to select the default tag and the default flags.

20. Enter the new starting cylinder for slice 7.

21. Enter the new partition size for slice 7, by typing a \$ sign.

Note – Entering a \$ sign as a value for the last partition size automatically assigns the ending cylinder boundary for the last slice.

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

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 2036 + 2 (reserved cylinders)

Part      Tag  Flag  Cylinders      Size      Blocks
0        root  wm     0 - 2520     1.14GB    (2521/0/0)    2382345
1        swap  wu   2521 - 2840   147.66MB    (320/0/0)     302400
2    backup  wm     0 - 8892     4.01GB    (8892/0/0)   8402940
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   2841 - 8892     2.73GB    (6051/0/0)   5718195
7 unassigned  wm     0              0          (0/0/0)         0
```

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.

23. After checking the partition table to ensure there are no errors, label the disk.

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

```
partition>
```

Saving a Partition Table to the /etc/format.dat File

You can use this optional procedure to add the newly created partition table to the `/etc/format.dat` file. You save a customized partition table so you can use it to quickly partition other disks of the same type on the system.

To save a customized partition table, at the `partition` menu:

1. Type **name** to enter a unique name for the current partition table. (*Frequently the disk manufactures name is used.*)

```
partition> name
Enter table name (remember quotes): SUN4.2
```

2. Exit the `partition` menu.

```
partition> quit
```

3. Type **save** to save the new partition table information. Enter the full pathname for the `/etc/format.dat` file.

```
format> save
Saving new partition definition
Enter file name["./format.dat"]: /etc/format.dat
```

Locating and Using the Customized Partition Table

To retrieve a customized partition table, at the `format` menu:

1. Type **partition**.

```
format> partition
```

2. Locate and select the customized partition table from the list, using its assigned number.

```
partition> select
    0. original
    1. unnamed
    2. SUN4.2
Specify table (enter its number)[0]: 2
```

3. Label the disk with the selected partition table.

```
partition> label
Ready to label disk, continue? yes
```


4. Exit the partition menu.

```
partition> quit
```

5. Read the new disk label.

```
format> verify
```

6. Exit the format utility.

```
format> quit
```

Repartitioning a Disk with the `modify` Command

You will need to change the size of slices on a disk, as storage requirements grow, or diminish. The easiest way to accomplish this is using the **modify** command from the **partition** menu.

Warning – When a disk with existing slices is repartitioned and relabeled, any existing data is inaccessible. Existing data must be copied to backup media before the disk is repartitioned and restored after the disk is relabeled.

The **modify** command allows `root` to create slices by specifying the size of each slice without having to keep track of starting cylinder boundaries. It also keeps track of any disk space remainder in the free hog slice.

The free hog slice is used as a disk space accumulator that expands and contracts as other slice sizes are changed.

Using the `modify` Command

The following steps describe how to change the size of a disk slice.

In this procedure slice 0 is increased from 128Mbytes to 200Mbytes.

1. Type **format** at the prompt and press Return.
2. Select a disk by typing the appropriate number.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /pci@if,4000/pci@1,1/ide@3/dad@0,0
  1. c1t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /pci@if,4000/pci@1/isptwo@4/sd@3,0
Specify disk (enter its number): 1
```

The format utility's main menu is displayed.

```
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
show	- translate a disk address
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
volname	- set 8-character volume name
!<cmd>	- execute <cmd>, then return
quit	

3. Type **partition**.

The partition menu is displayed.

```
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	

```
partition>
```

4. Type **modify** and press Return.

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

5. Press the Return key to accept the default selection.

The current partition table is displayed.

Part	Tag	Flag	Cylinders	Size	Blocks	
0	root	wm	0 - 189	200.39MB	(190/0/0)	410400
1	swap	wu	190 - 311	128.67MB	(122/0/0)	263520
2	backup	wm	0 - 8892	4.00GB	(8892/0/0)	8402940
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	312 - 8892	3.67GB	(3568/0/0)	7853760
7	unassigned	wm	0	0	(0/0/0)	0

```
Do you wish to continue creating a new partition
table based on above table[yes]? <Return>
```

6. Select the default option by pressing the Return key, or typing **yes**
7. Press the Return key to accept slice 6 (the default) as the Free Hog partition. If slice 6 does not have space allocated to it, then you must specify another slice.

```
Free Hog partition [6] ? <Return>
```

Using the Free Hog Slice

When `root` invokes the `format` utility to change the size of one or more disk slices, a “temporary” slice must be designated that expands and shrinks to accommodate the resizing operations.

This temporary slice is used to donate space when another slice is expanded, and it receives, or hogs, the discarded space when a slice is shrunk. For this reason, the designated temporary slice is sometimes called the free hog.

The free hog slice exists only during installation, or when you run **format**. There is no permanent free hog slice during normal operations.

8. Enter the size of slice 0 as **200mb** and press Return.

```
Enter size of partition '0' [263520b, 122c, 128.67mb, 0.13gb]: 200mb
Enter size of partition '1' [263520b, 122c, 128.67mb, 0.13gb]: <Return>
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]: <Return>
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]: <Return>
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]: <Return>
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]: <Return>
```

9. Press the Return key through the remaining slices (1, 3, 4, 5, 7) to default to their current sizes. Slice 6 is skipped because it has been designated as the Free Hog partition.

In the partition table, slice 6 has decreased in size as the size of slice 0 increased.

10. Press Return to confirm using this modified partition table.

```
Okay to make this the current partition table[yes]?
<Return>
```

11. Name the modified partition table and press Return.

```
Enter table name (remember quotes): c1t3d0.4gb
```

12. Write the modified partition table to the disk by typing **yes** and pressing Return.

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

13. Type **quit** (or **q**) and press Return to exit the partition menu.

```
partition> quit
```

The main format menu is displayed.

Viewing the Disk's VTOC

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

The first method is to use the `verify` command from the `format` utility.

The second method is to invoke the `prtvtoc` command from the command line.

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

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

```
format> verify
Primary label contents:
ascii name  = <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
pcyl        = 3882
ncyl        = 3880
acyl        = 2
nhead       = 16
nsect       = 135
Part        Tag    Flag  Cylinders      Size      Blocks
  0         root    wm     0 - 189      200.39MB   (190/0/0)    410400
  1         swap    wu    190 - 311    128.67MB   (122/0/0)    263520
  2         backup   wm     0 - 8892     4.00GB     (8892/0/0)   8402940
  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    312 - 8892    3.67GB     (3568/0/0)   7853760
  7 unassigned    wm     0              0          (0/0/0)       0
format> quit
```

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 gives you the ability to view a disk's VTOC from the command line. For example,

```
# prtvtoc /dev/rdisk/c1t3d0s2
* /dev/rdisk/c0t0d0s2 partition map
* Dimensions:
*   512 bytes/sector
*   135 sectors/track
*   16 tracks/cylinder
* 2160 sectors/cylinder
* 3882 cylinders
* 3880 accessible cylinders
* Flags:
*  1: unmountable
* 10: read-only
*
* Partition  Tag  Flags      First      Sector      Last
*           Tag  Flags      Sector      Count      Sector  Mount Directory
*     0      2    00          0      408240      408239      /
*     1      3    01     410400      671760     1082159
*     2      5    00          0     8380800     8380799
*     6      4    00     673920     7706880     8380799      /usr
```

The disk label information includes the following fields:

- **Dimensions** – Describes the physical dimensions of the disk.
- **Flags** – Describes the flags listed in the partition table.
- **Partition (or slice)** – Described in Table 6-1 on page 6-5.
- **Tag** – Described in Table 6-1 on page 6-5.
- **Flags** – Described in Table 6-1 on page 6-5: 00=rw, mountable; 01=rw, unmountable; 10=read only.
- **First Sector** – Defines the first sector (disk block) 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** – Indicates if it is a file system currently in use. If the field is empty the slice is currently not being used. If a directory name appears in this field, the slice is currently being used to store data.

The `fmthard` Command

You should save a disk's VTOC to a file, using the `prtvtoc` command. This allows you to relabel the disk using the `fmthard` command, should one of the following situations occur.

- 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.

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

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

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

If one of the situations described above has occurred, and the VTOC was previously saved to a file, you can:

1. Run `format`, select the disk, and label it with the default partition table.
2. Use the `fmthard` command to write the desired label information, save to a *datafile* back to the disk. For example:

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


Exercise: Disks, Slices, and Format



Exercise objective – In this lab you use the `format` utility to partition a disk and use `prtvto` and `fmthard` to repair a corrupted disk label.

Preparation

This exercise requires an unused disk. Refer to the lecture notes as necessary to perform the tasks listed. The disk configuration you create in this exercise will be used in later sections of the class.

Task Summary

- Use `format` to list the disks currently attached to your system. Use `prtvto` to identify a disk that is not currently used to hold any mounted filesystems. Examine the `Mount Directory` field in the information `prtvto` displays. Record the name of a disk that has no mount directory listed.
- Use `format` to manually 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 it ends 25 megabytes into the space assigned to slice 1.
- Attempt to correct the overlap using option 0 found in the `modify` menu. Record the message that displays. Use the All Free Hog method from the `modify` menu to set the sizes of slice 0, 1, 3, and 4 so they are again approximately equal. Use slice 4 as the free hog partition. Verify your disk label with `prtvto`.
- Create a directory called `/vtoc`. Run `prtvto` to read the label of the disk you modified, and save its output in a file in `/vtoc`. Use `dd` to destroy the label on the same disk. Attempt to read the disk label using `prtvto` and record the result. If required, use `format` to write a default label to the disk. Use `fmthard` to restore the label using the output from `prtvto` you saved earlier. Verify the new label exists.

Tasks

1. Log in as `root` and open a terminal window. Run `format`.

```
# format
```

2. Record the list of disks presented by `format` (e.g., `c0t0d0`, `c1t3d0`).

Press the Control-d keys to exit the `format` utility.

```
format> Control-d
#
```

3. Use `prtvtoc` to list the VTOC for each of the disks you found in the previous step. Examine the `Mount Directory` field in the information `prtvtoc` displays. Record the name of a disk that has no mount directory listed. This will be an unused disk. For example:

```
# prtvtoc /dev/rdisk/c1t3d0s2
```

Unused disk: _____

4. Run `format` again. Select the unused disk from the list of disks presented. For example:

```
# format
(list of disks)
Specify disk (enter its number): x
```

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 4 gigabytes, record 4000 megabytes.

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

Mbytes: _____

6. Divide the number of megabytes by four. Use the result for the number of megabytes to assign 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 partition 0. Accept the defaults for tags and flags. (A question mark displays the list of available tags and flags.) Start this first partition on cylinder 0. Enter the resulting number of Mbytes from the previous step for the partition size. Print the partition table again to verify the change. For example:

```
partition> ?
(partition menu)
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, 0.00mb, 0.00gb]: 1000m
partition> print
(partition table)
```

8. Set the sizes of partitions 1, 3, and 4 so they are the same as partition 0. Begin each successive partition on the cylinder that follows the ending cylinder of the previous partition. Example:

```
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]: 949
Enter partition size[0b, 0c, 0.00mb, 0.00gb]: 1000m
partition> print
(partition table)
```

9. Set partitions 5, 6, and 7 to start at cylinder 0, and assign them 0 megabytes. For example:

```
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, 0.00mb, 0.00gb]: 0m
partition>
```

10. Print the partition table. Is there any overlap of ending and beginning cylinders for any of the partitions listed? If so, re-do the steps above to correct the problem. If not, proceed to the following steps to introduce this problem.

```
partition> print
```

11. Add 25 to the number Mbytes/4 value listed in step 6 above.

(Mbytes/4) + 25: _____

Change partition 0 so it uses the new size listed above. For example:

```
14 partition> ?
15 (partition menu)
16 partition> 0
17 Part      Tag      Flag      Cylinders      Size      Blocks
18  0 unassigned    wm          0 - 948      1000.90MB      (949/0/0)
2049840
```

```
Enter partition id tag[unassigned]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 0
Enter partition size[2049840b, 949c, 1000.90mb, 0.98gb]: 1025m
partition> print
(partition table)
```

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

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
```

What warning displays?

13. Use the `modify` command from the `partition` menu to attempt to fix the problem. Select item 1 to use the All Free Hog method.

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

The partition table is displayed. What size has been assigned to all partitions except 2?

14. Respond to the prompts to continue the process. Select slice 4 as the Free Hog partition. Use the size listed in step 6 for partitions 0, 1, and 3. Set the other partitions to size 0. `format` will not ask for a value for partition 2 or 4. Example:

```
Do you wish to continue creating a new partition  
table based on above table[yes]? y  
Free Hog partition[6]? 4  
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 1000m  
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 1000m  
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]: 1000m  
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:  
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
```

Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

(partition table)

Okay to make this the current partition table[yes]? **y**

Enter table name (remember quotes): **test**

Ready to label disk, continue? **y**
partition>

15. At the end of this process you should have three partitions of equal size, where slice 4 takes up any extra room if it exists. Quit the partition menu.

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

16. Save your new partition table to `/etc/format.dat`. Carefully read the message that is displayed by `format` utility, and enter the correct file name. Quit `format` when finished.

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

17. Verify your new partition table with `prtvtoc`. Example:

```
# prtvtoc /dev/rdisk/c1t3d0s2
```

18. Create a directory called `/vtoc`.

```
# mkdir /vtoc
```

19. Use `prtvtoc` to print the partition table you just created, and save its output to a file in `/vtoc`. Name the file so it corresponds with the disk you're examining. Verify that valid information exists in the file you create. For example:

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

20. Use the `dd` command below 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/rdisk/c1t3d0s2 bs=512 count=1
1+0 records in
1+0 records out
#
```

21. Attempt to read the label from the same disk. For example:

```
# prtvtoc /dev/rdisk/c1t3d0s2
```

What happens?

-
22. If `prtvtoc` reported an "Unable to read Disk geometry" message, use `format` to place a default label on the disk whose label you destroyed earlier. If `prtvtoc` reports that only slice 2 exists on the disk, skip to Step 23. For example:

```
# format
```

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

```
c1t3d0: configured with capacity of 4.00GB
```

```
AVAILABLE DISK SELECTIONS:
```

```
0. c0t0d0 <Seagate Medalist 34342A cyl 8892 alt 2 hd 15 sec 63>
   /pci@1f,0/pci@1,1/ide@3/dad@0,0
```

```
1. c1t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
   /pci@1f,0/pci@1/pci@2/SUNW,isp2004/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/rdisk/c1t3d0s2
```

23. Use `fmthard` to write to the disk the label information you saved earlier. For example:

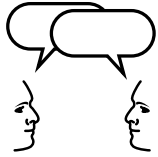
```
# fmthard -s /vtoc/c1t3d0 /dev/rdisk/c1t3d0s2
fmthard:  New volume table of contents now in place.
#
```

24. Attempt to read the label from the same disk. For example:

```
# prtvto /dev/rdisk/c1t3d0s2
```

Was this successful?

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

12. Use the `modify` command from the `partition` menu to attempt to fix this problem. Select item 0 to modify the current partition table. What warning displays?

Warning: Overlapping partition (1) in table.

Warning: Fix, or select a different partition table.

13. Use the `modify` command from the `partition` menu to attempt to fix the problem. Select item 1 to use the All Free Hog method.

The partition table displays. What size has been assigned to all partitions except 2?

Zero.

21. Attempt to read the partition table from the same disk. For example:

What happens?

Different disk types present different results. SCSI disks may report messages that indicate that the disk label is unreadable. For example:

```
prtvto: /dev/rdisk/c1t3d0s2: Unable to read Disk
geometry
```

IDE disks may simply 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

24. Attempt to read the label from the same disk. For example:

Was this successful?

This command should successfully read the disk label.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Explain the term disk slice
- ☐ Describe and create a disk label
- ☐ Define and modify a partition table using the `format` utility
- ☐ Describe the purpose of the `/etc/format.dat` file
- ☐ Use the `format` utility to save and retrieve customized partition tables
- ☐ Demonstrate how to view the disk's VTOC using two different commands: `verify` and `prtvtoc`
- ☐ Use the `fmthard` command to update the VTOC on a disk

The Solaris Operating Environment

ufs File System

7 

Objectives

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

- Describe the three different types of file systems in the Solaris Operating Environment
- Define the term file system
- List the components that are contained in the structure of a file system
- Create a new ufs file system using the `newfs` command

Additional Resources



Additional resources – The following references can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10
- *Solaris 8 System Administration Guide, Volume II*, Part Number 805-7229-10

File System Types Supported by the Solaris Operating Environment

The Solaris Operating Environment supports three different types of file systems:

- Disk-based file systems
- Distributed file systems
- Pseudo file systems

Disk-Based File System

Disk-based file systems include hard disks, CD-ROMs, diskettes, and DVD.

- `ufs` – The standard UNIX file system. Under the Solaris Operating Environment, the `ufs` file system is based on the Berkeley fast file system.
- `hfs` – 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 DOS file attribute table (FAT32) file system. It allows the Solaris Operating Environment to access PC-DOS formatted file systems, giving users direct read/write access to PC-DOS files using UNIX commands.
- `udf` – The Universal Disk Format file system for optical storage targeted at DVD and CD-ROM media. Provides for universal data exchange and supports read-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 between many types of systems on the network. It provides a method of making a disk on one system appear as though it was connected to another system.

Pseudo File System

Pseudo file systems are memory-based. These file system types provide access to kernel information and facilities.

- **tmpfs** – The Temporary file system for file storage in memory without the overhead of writing to a disk-based file system. It is created and destroyed every time the system is rebooted.
- **swapfs** – The Swap file system used by the kernel to manage swap space on disk(s).
- **fdfs** – The File Descriptor file system provides explicit names for opening files 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, by process number, in the `/proc` directory. Information in this directory is used by commands such as the `ps` command.

Introducing the Solaris Operating Environment ufs File System

To a user in the Solaris Operating Environment, a file system is a collection of files and directories used to store and organize data for access by the system and 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 and allow for the storage and management of data.

The Solaris Operating Environment stores data in a logical file hierarchy. This file hierarchy is referred to as the Solaris directory tree, which is formed by a number of file systems.

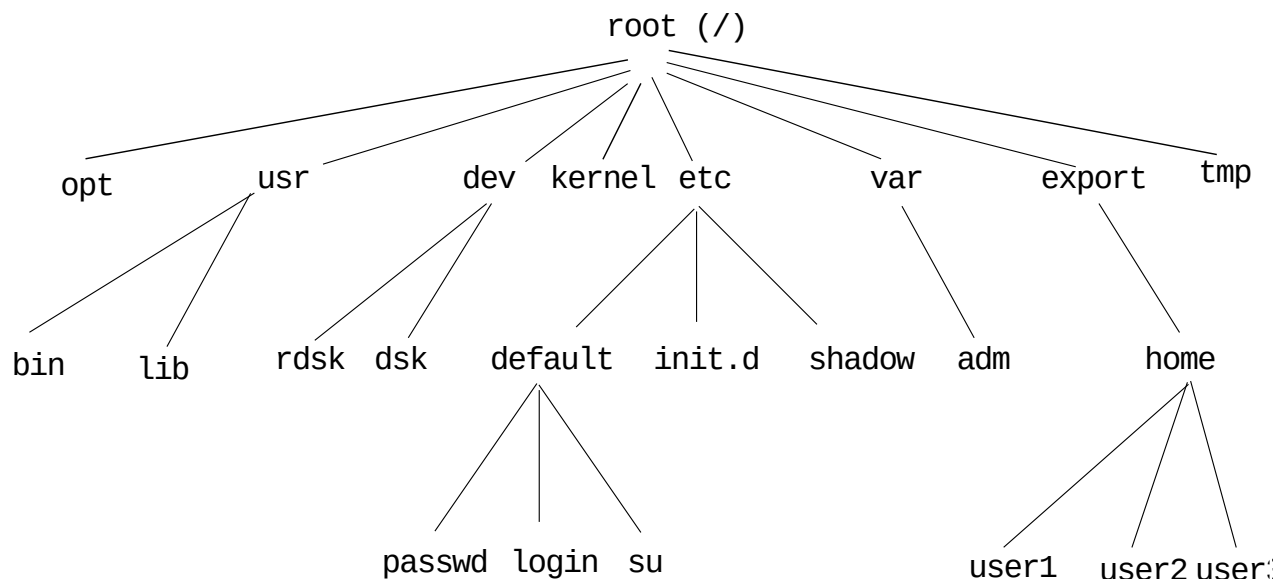


Figure 7-1 Solaris Directory Tree

Note – This is not a complete representation of a Solaris directory tree.

Every `ufs` file system must be created on a disk slice before it can be used in the Solaris Operating Environment. Creating a file system on a disk slice enables the Solaris Operating Environment to store UNIX directories and files.

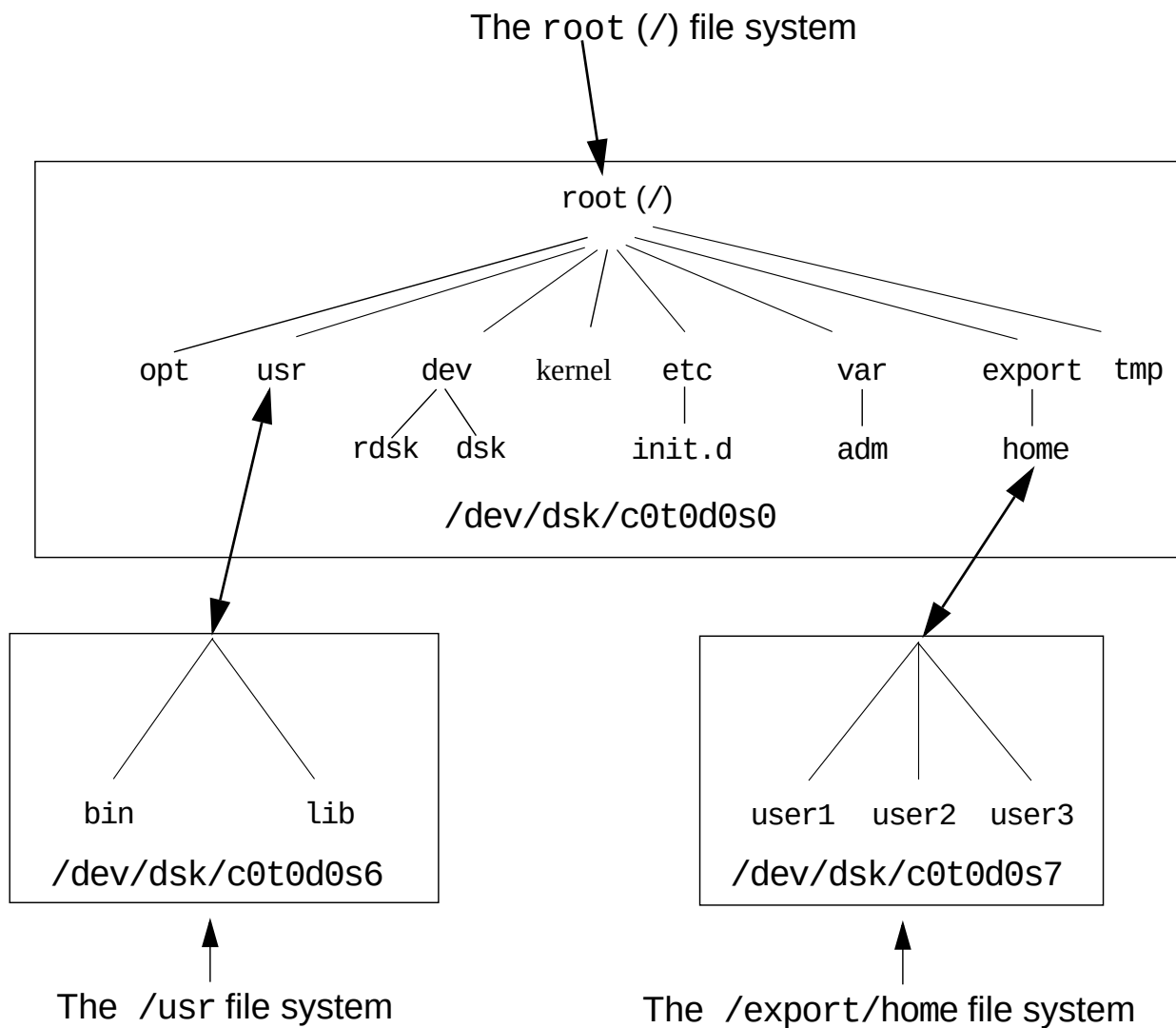


Figure 7-2 Solaris `ufs` File Systems Residing on Disk Slices

Basic Disk Structures

The Disk Label (VTOC)

The disk label (VTOC) contains the partition table for the disk, and is located in the first disk sector (512-byte blocks). A disk partition can contain a file system that the Solaris Operating Environment interprets as an organization of directories and files.

The Boot Block

The bootstrap program (`bootblk`) is found in the next 15 disk sectors. Only the `root` file system has an active boot block, although space is allocated for a boot block at the beginning of each file system.

The Superblock

The file system is described by its superblock. The superblock is contained in the 16 disk sectors following the boot block. The superblock is a table of information about 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

Because the superblock contains critical data, it is replicated in each cylinder group to protect against catastrophic loss. This is done when the file system is created.

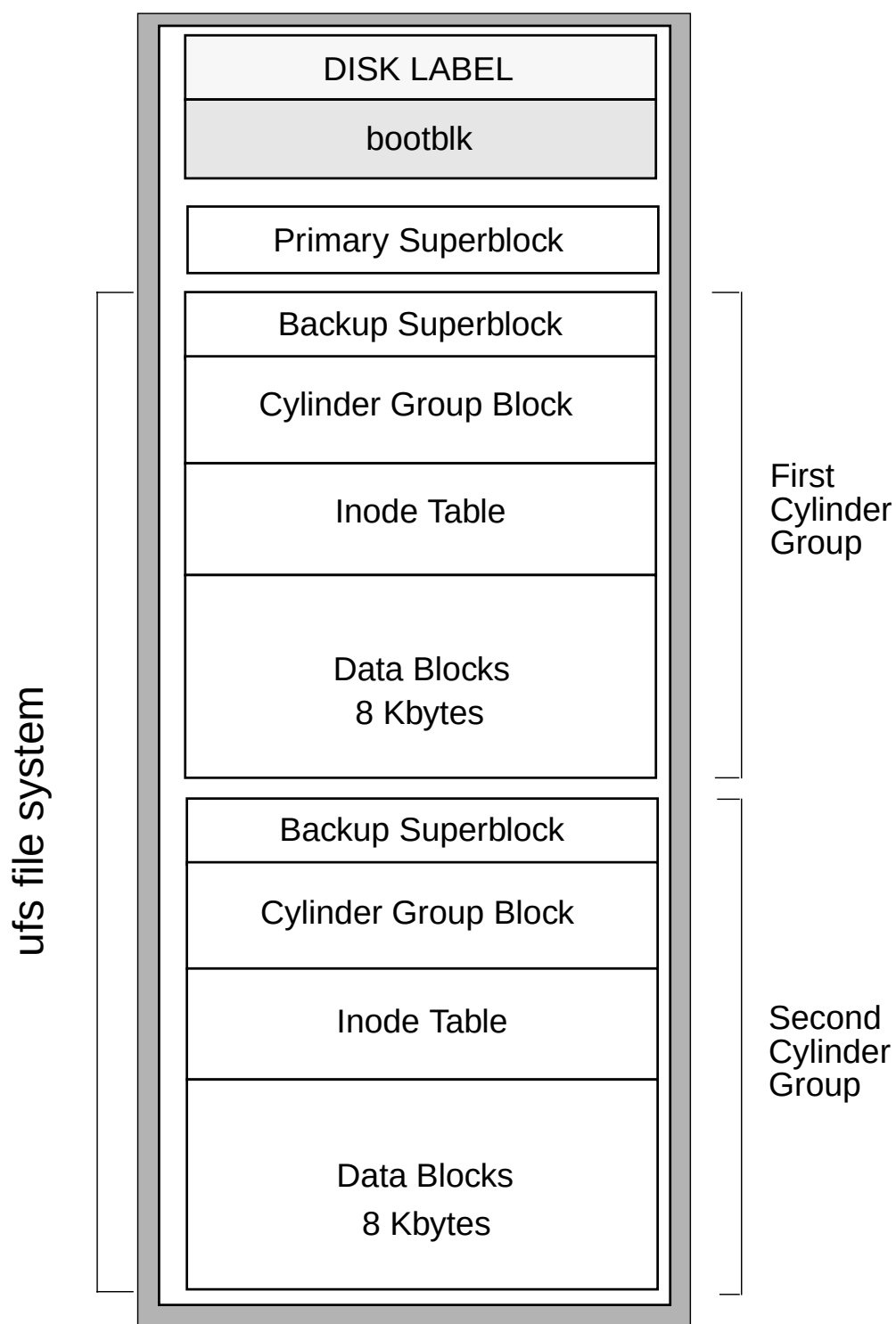


Figure 7-3 ufs File System Structure

Figure 7-3 shows a series of cylinder groups in a ufs file system.

Cylinder Groups

By dividing the partition into cylinder groups (the minimum default size is 16 cylinders per group), disk access is improved. The file system constantly optimizes the disk by placing file data in one cylinder group, thus reducing head travel. The file system stores files across several cylinder groups if needed.

Cylinder Group Blocks

The cylinder group block is a table 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

Inode Table

The inode table contains the inodes for the cylinder group. An inode (from the term *index node*) is the internal description of a file and the location of its data blocks. Each cylinder group contains a portion of the total number of inodes.

Data Blocks

A data block is the unit of storage for data in the Solaris 8 Operating Environment file system. The data block is 8192 bytes in size by default.

Inodes

An inode contains the following information about a file:

- The type of file and the access modes
- The UID and GID numbers of the file's owner and group
- The size of the file
- The time the file was last accessed or modified, and the inode changed
- The total number of data blocks used by, or allocated to the file

The inode contains two types of pointers: direct pointers and indirect pointers.

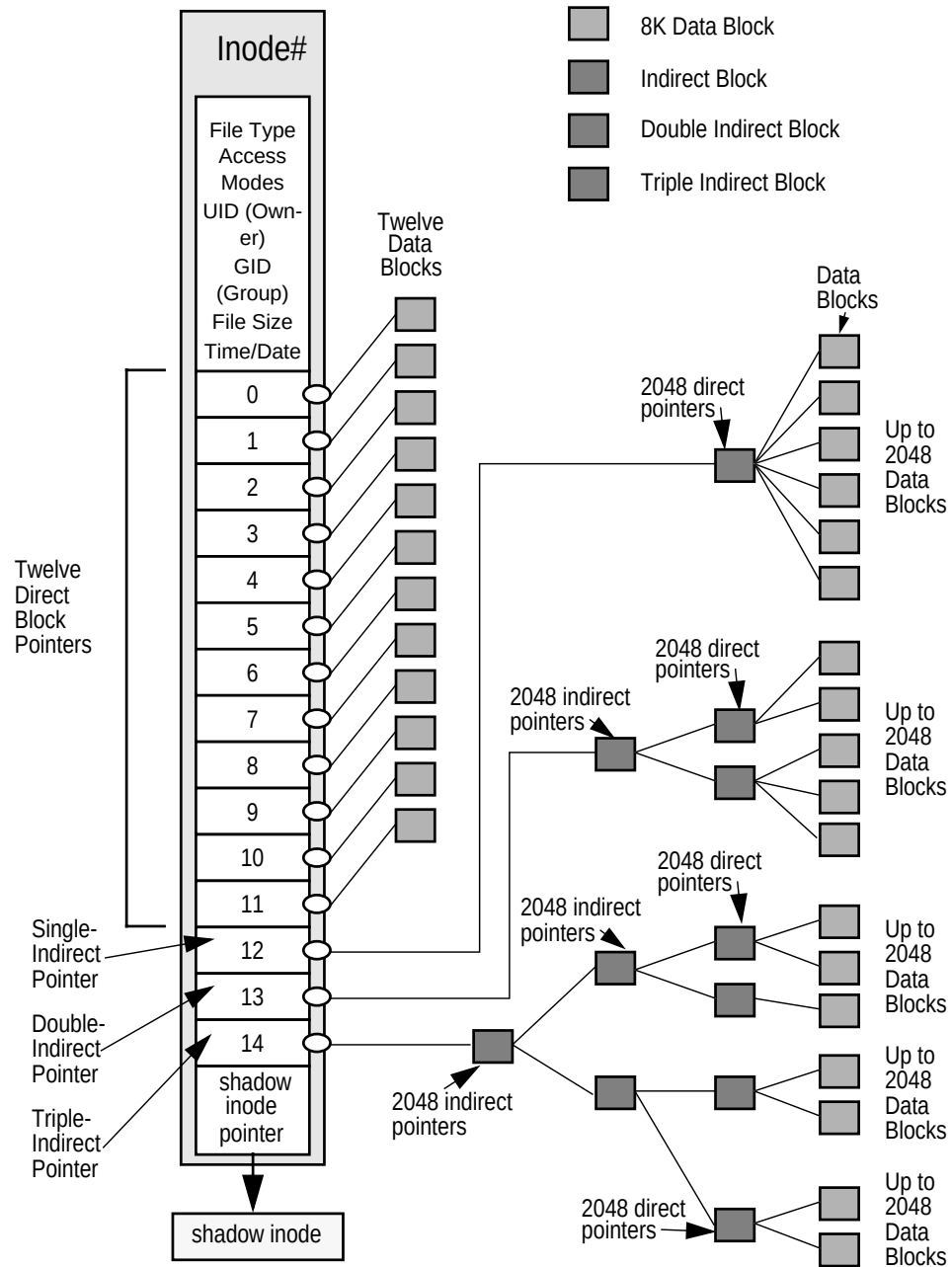


Figure 7-4 Structure of an Inode

Direct Pointers

There are 12 direct pointers, which refer directly to data blocks. The 12 direct pointers can directly reference the data blocks for a file up to 96 Kbytes.

Indirect Pointers

The three types of indirect pointers are:

- Single indirect pointer – A single indirect pointers refers to a file system block containing 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 – A double indirect pointer refers to a file system block containing single indirect pointers. Each indirect pointer refers to a file system block containing the data block pointers. Double indirect pointers points to an additional 32 Gbytes of data.
- Triple indirect pointer – A triple indirect pointer can reference up to an additional 70 Tbytes of data. However, the maximum file size is limited to 1 Tbyte in a `ufs` file system.

Data Blocks

The rest of the space allocated to the file system is occupied by data blocks, also called storage blocks.

Data blocks are allocated, by default, in 8-Kbyte logical block sizes, and further divided into a 1-Kbyte fragment.

For a regular file, the data blocks contain the contents of the file.

For a directory, the data blocks contain entries that give the inode number and the file name of those files contained in that directory.

Free Blocks

Those blocks that are currently not being used as ACL lists, indirect address blocks, or storage blocks are marked as free in the cylinder group map. This map also keeps track of fragments to prevent fragmentation from degrading disk performance.

Data Blocks and Fragmentation

The method used by the ufs file system to store the contents of a file which is not large enough to fill one data block is called *fragmentation*.

Data blocks can be divided into eight fragments of 1024 bytes each, for the storage of small files.

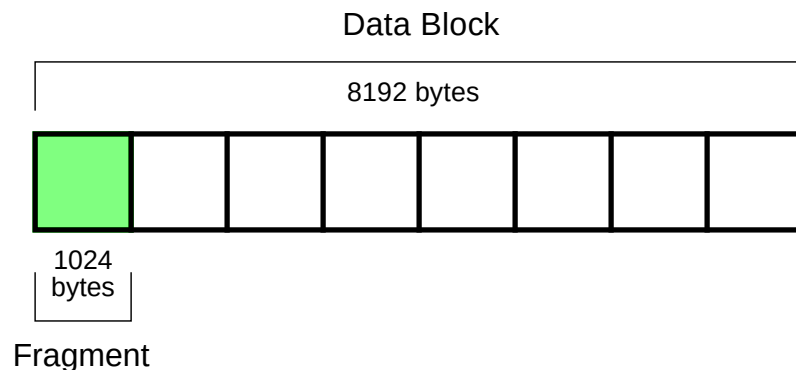


Figure 7-5 Example of a Divided Data Block

If a file, contained in a fragment, grows and requires more space, it is allocated one or more fragments in the same data block.

The content of two different files can be stored in fragments in the same data block. For example:

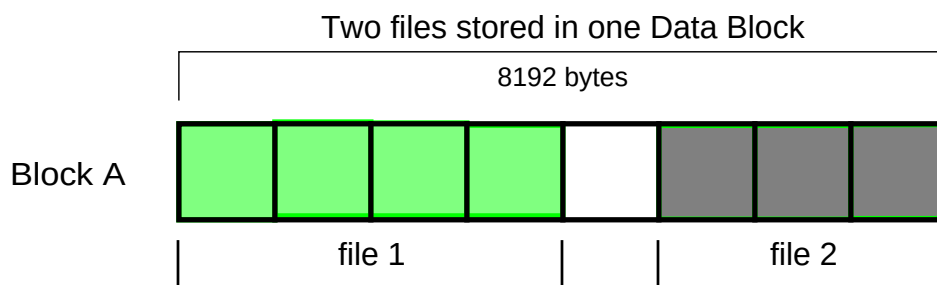


Figure 7-6 Example of Two Files Stored in One Data Block

If file 1 requires more space than is currently available in the shared data block, then the entire contents of that expanding file are moved by `ufs` into a free data block. This is a requirement of `ufs` to assure that all the same file fragments are contained in a whole data block.

The `ufs` file system will not allow *fragments* of the same file to be stored in two different data blocks.

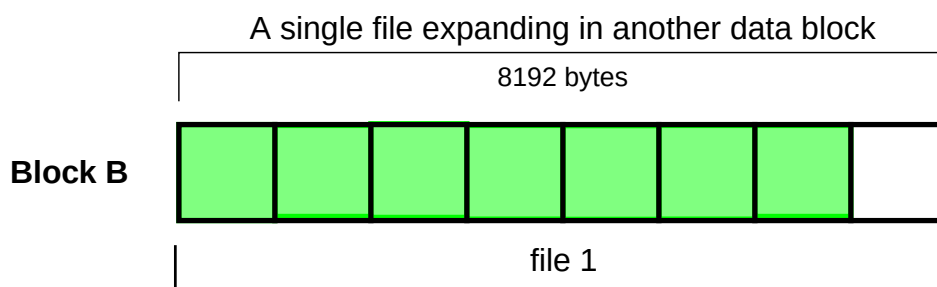


Figure 7-7 Example of an Expanded File

Shadow Inode

Files with an ACL list have two inodes, a ufs inode and a shadow inode.

On disk, the ACL lists are stored the same way as the file data, and is referred to through the direct block pointers in the inode.

The shadow inode points to the data block that contains the actual ACL list.

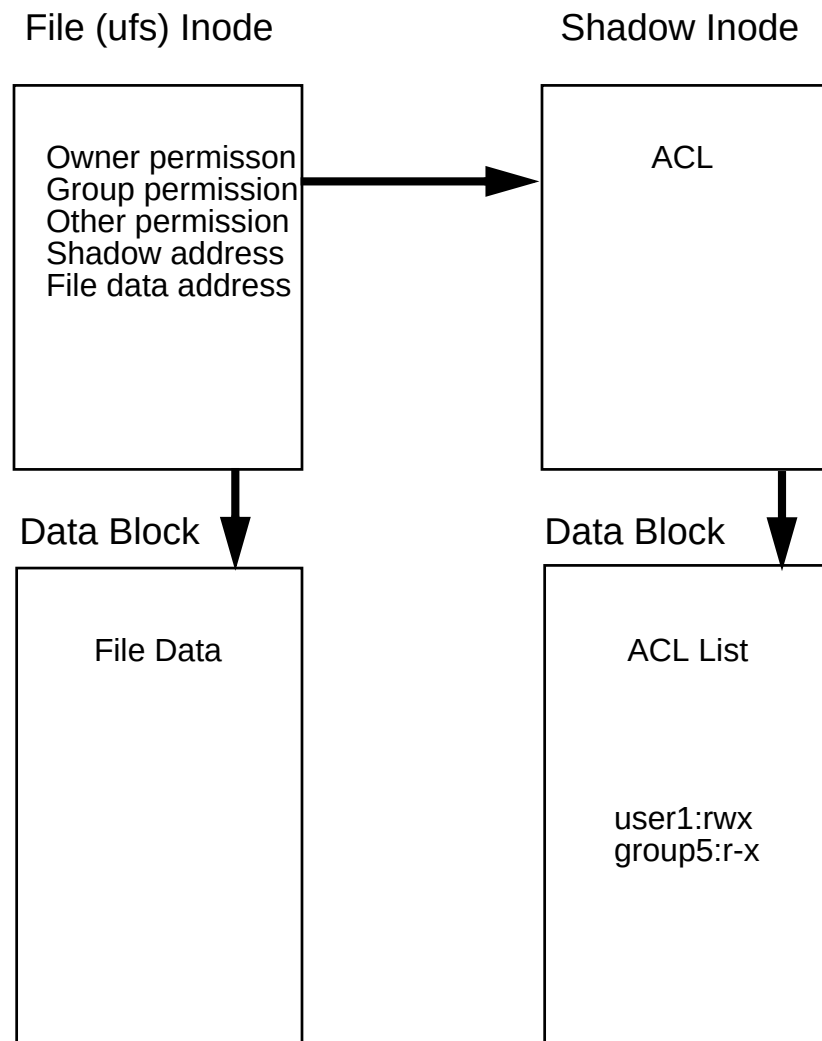


Figure 7-8 Shadow Inode

Creating ufs File Systems

Every disk slice on a newly partitioned disk that is used to store directories or files must have a file system created on it first.

As root, you can construct a ufs file system on a disk slice using the `newfs` command.

The `newfs` command is a front-end to the `mkfs` command used to create file system file systems. The `newfs` command is located in the `/usr/sbin` directory.



Caution – The `newfs` command is destructive; it overwrites any data that resides on the selected disk slice.

Creating a ufs File System

1. As root, create a file system on the first slice of a newly partitioned disk. For example:

```
# newfs /dev/rdisk/c1t3d0s0
newfs: construct a new file system /dev/rdisk/c1t3d0s0: (y/n)? y
/dev/rdisk/c1t3d0s0: 410720 sectors in 302 cylinders 17 tracks 80 sectors
      200.5MB in 19 cyl groups (16 c/g, 10.62MB/g, 5120 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
  32, 21872, 43712, 65552, 87392, 109232, 131072, 152912, 174752, 196592,
  218432, 240272, 262112, 283952, 305792, 327632, 349472, 371312, 393152
```

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**

The `newfs` command displays information about the new file system being created.

The first line printed by `newfs` describes the basic disk geometry. The second line describes the ufs file system created in this slice. The third and remaining lines list the locations of the backup superblocks.

Note – This process also creates a `lost+found` directory for the `ufs` file system. A directory that is used by the file system check and repair (`fsck`) utility.

3. Steps 1 and 2 above are repeated for every disk slice (on any newly partitioned disk) that needs to contain a file system.

The `newfs` command uses a minimum percentage of free space to be maintained in the new file system. This free space in the file system is referred to as *minfree*. It 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.

To change the minimum percentage value of free space on an existing file system, the system administrator can use the command:

```
tunefs -m %free
```

For example:

```
# tunefs -m 1 /dev/rdisk/c1t3d0s0
```

Exercise: Creating UFS File Systems



Exercise objective – In this exercise you create UFS file systems, and calculate and adjust `minfree` values.

Preparation

This exercise requires an unused disk, divided into four slices, where slices 0, 1, and 3 are equal size, and slice 4 takes up the remaining space. If it is necessary to partition this disk, this exercise requires an understanding of using the `format` utility. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

- Find a disk that is not in use and that is partitioned as specified in the preparation description above. If necessary, partition a disk accordingly. Use `newfs` with no options to create a new file system on slice 0. Use `newfs` to create a file system on slice 1 with an inode ratio of 1 per 16384 bytes of data space. Compare how quickly `newfs` 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?
- Use `df -k` to display the number of kilobytes used, available and 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 avail values, and express the result as a percentage. Use `fstyp` to verify the result. Use `tunefs` to adjust the `minfree` value up or down by 3%. Record the message `tunefs` displays. Use `df` to find the change made by `tunefs`.
- Create new file systems on slices 3 and 4 of your spare disk.

Tasks

1. Log in as `root` and open a terminal window. Change directory to `/dev/rdisk`.

```
# cd /dev/rdisk
```

2. To find a spare disk, use `ls` to display a list of possible disks, and `prtvtoc` to display the VTOC for each disk you find. Examine the partition list and the Mount Directory field that `prtvtoc` displays. Disks that are not in use have no mount directory listed. Record the name of the unused disk. For example:

```
# ls *s2
```

```
# prtvtoc /dev/rdisk/c1t3d0s2
```

Unused disk: _____

3. If a spare disk exists, but it is not divided into four partitions, use `format` to partition the disk accordingly. Use the All Free Hog method to set partition 0, 1, and 3 to exactly the same size. Pick a value that is roughly 25% of the total disk space. Use slice 4 as the free hog slice.

4. Use `newfs` without options to create a new file system on slice 0 on the spare disk. Observe how quickly `newfs` 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 /dev/rdisk/c1t3d0s0
```

Cylinder groups: _____

Cylinders per group: _____

Inodes per group: _____

5. Use `newfs` to create a new file system on slice 1 on the spare disk. Use the `-i` option to create one inode per 16384 bytes of data space. Observe how quickly `newfs` 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/rdisk/c1t3d0s1
```

Cylinder groups: _____

Cylinders per group: _____

Inodes per group: _____

6. According to the statistics you've gathered, how do the file systems on slices 0 and 1 differ?

7. Use `df -k` to display statistics for the file systems on slice 0 and 1 that you used in the previous steps. Record the values listed in the `kbytes`, `used`, and `avail` columns.

```
# df -k /dev/dsk/c1t3d0s0
```

```
# df -k /dev/dsk/c1t3d0s1
```

c1t3d0s0: used: _____ avail: _____ kbytes: _____

c1t3d0s1: used: _____ avail: _____ kbytes: _____

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.

✓ *For example: $(used+avail) / kbytes / * 100 = \% ; 100 - \% = minfree$*

c1t3d0s0: Sum of used + avail = _____ kbytes: _____ % _____

c1t3d0s1: Sum of used + avail = _____ kbytes: _____ % _____

Use `fstyp` to verify your result.

```
# fstyp -v /dev/dsk/c1t3d0s0 | grep minfree
# fstyp -v /dev/dsk/c1t3d0s1 | grep minfree
```

9. Use `tunefs` 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%, reduce it by 3%. If it is less than or equal to 5% add 3%. For example:

```
# tunefs -m 8 /dev/dsk/c1t3d0s0
```

What message does `tunefs` display?

10. Use `df -k` to verify that the `minfree` value has changed. Record the values listed in the `kbytes`, `used`, and `avail` columns.

```
# df -k /dev/dsk/c1t3d0s0
```

c1t3d0s0: used: _____ avail: _____ kbytes: _____

Which of the values has changed from the information you gathered in step 7?

11. To prepare for later lab exercises, create new file systems on slices 3 and 4 of your spare disk. For example:

```
# newfs /dev/dsk/c1t3d0s3
# newfs /dev/dsk/c1t3d0s4
```


Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

6. According to the statistics you've 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 `df -k` to display statistics for the file systems on slice 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?

file system 1 has the larger amount of available data space because it holds fewer inode records.

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 `fstyp` to verify your result. For example:

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:

$$1926808 / 1986439 = .969981$$

- c. Multiply the result of step b by 100. For example:

$$.969981 * 100 = 96.9981$$

- 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:

$$3.0019 = 3\%$$

9. Use `tunefs` 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%, reduce it by 3%. If it is less than or equal to 5% add 3%. For example:

What message does `tunefs` display?

minimum percentage of free space changes from 3% to 8%

10. Use `df -k` to verify that the `minfree` value has changed. Record the values listed in the `kbytes`, `used`, and `avail` columns.

Which of the values has changed from the information you gathered in step 7?

The `avail` column changes but not the `kbytes` or `used` columns.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Describe the three different types of file systems in the Solaris Operating Environment
- ☐ Define the term file system
- ☐ List the components that are contained in the structure of a file system
- ☐ Create a new ufs file system using the `newfs` command

Objectives

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

- Define the term mount point
- Identify mounted and unmounted file systems
- Mount file systems using the commands `mount` and `mountall`
- Describe some of the commonly used options of the `mount` command: `noatime`, `nolargefiles`, and `logging`
- Describe the purpose and format of the `/etc/mnttab` and `/etc/vfstab` files
- Define the procedure for mounting different types of file systems
- List the system files used to determine a file system's type
- Unmount local and remote file systems using the commands `umount` and `umountall`
- Forcibly unmount a busy file system
- Describe how to mount and access file systems residing on removable media devices, such as diskettes and CD-ROMs

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10
- *Solaris 8 System Administration Guide, Volume II*, Part Number 805-7229-10

Working With File Systems

Once you have created a file system, you must attach it to the Solaris Operating Environment directory tree, 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.

In the Solaris Operating Environment, you use the *mounting* process to attach individual file systems to their mount points on the directory tree. 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 tree. This action makes a file system unavailable to the system or users.

Figure 8-1 illustrates how the directory tree spans from one file system to the next. File systems do not contain their own mount point directories.

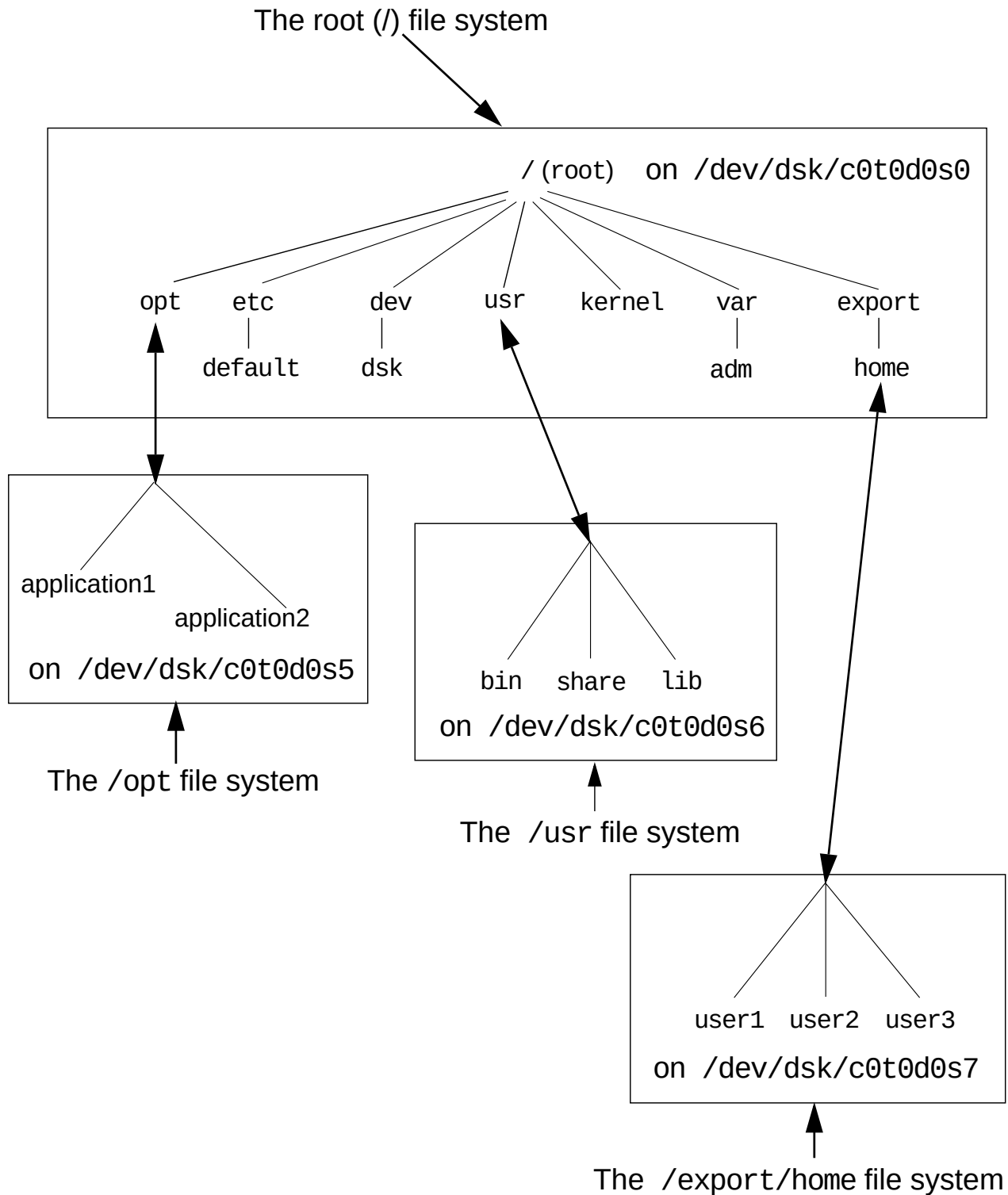


Figure 8-1 File Systems and Mount Points

Identifying Mounted File Systems

The mount Command

All users can determine which file systems are currently mounted by running the `mount` command, which is located in the `/sbin` directory.

The /etc/mnttab File

The `mount` command maintains the `/etc/mnttab` file, mounted file system table.

Each time a file system is mounted, an entry is added to this file by `mount`. Whenever a file system is unmounted, its entry is removed from the `mnttab` file.

A typical `/etc/mnttab` file is shown on the following page.

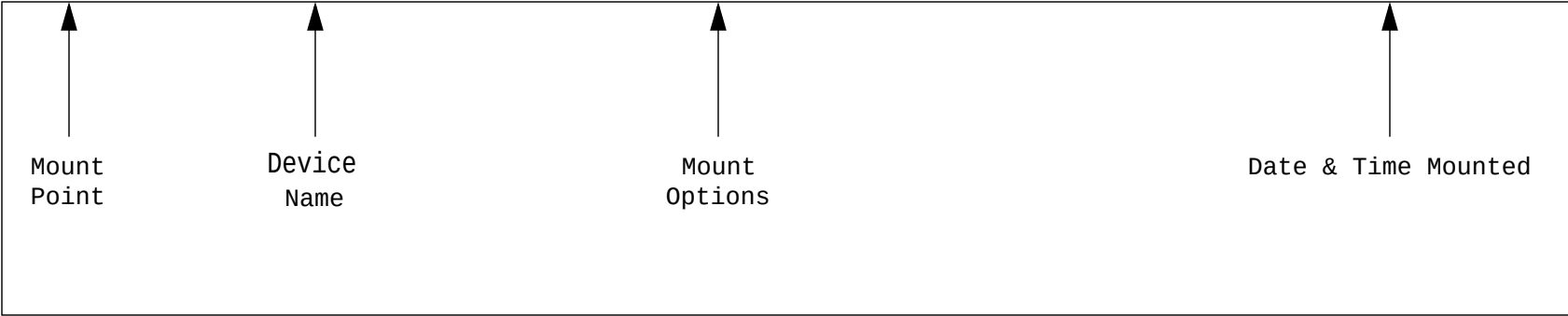
- **Mount Point** – The mount point, or directory name where the file system is to be attached to within the `root` file system, (for example: `/usr`, `/opt`).
- **Device Name** – The name of the device that is mounted at the mount point. This block device is where the file system is physically located.
- **Mount Options** – The list of mount options in effect for the file system.
- **Date and Time Mounted** – The date and time the file system was mounted to the directory tree.

mount

```
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic/dev=22000000 on Thu Apr 13 17:26:29 2000
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/onerror=panic/dev=22000006 on Thu Apr 13 17:26:30 2000
/proc on /proc read/write/setuid/dev=2d800000 on Thu Apr 13 17:26:28 2000
/dev/fd on fd read/write/setuid/dev=2e800000 on Thu Apr 13 17:26:31 2000
/etc/mnttab on mnttab read/write/setuid/dev=2f400000 on Thu Apr 13 17:26:34 2000
/var/run on swap read/write/setuid/dev=1 on Thu Apr 13 17:26:34 2000
/tmp on swap read/write/setuid/dev=2 on Thu Apr 13 17:26:38 2000
/export/home on /dev/dsk/c0t0d0s7 /read/write/setuid/largefiles/onerror=panic/dev=22000007 on Mon Mar 6 17:58:21 2000
```

The fields in the mount output are described in the example below.

```
/export/home on /dev/dsk/c0t0d0s7 /read/write/setuid/largefiles/onerror=panic/dev=22000007 on Mon Mar 6 17:58:21 2000
```



Mount Table Changes in /etc/mnttab

In previous Solaris Operating Environment releases, `/etc/mnttab` was a text file that stored information about mounted file systems. In Solaris 8 this file is an `mntfs` file system that provides read-only information directly from the kernel about mounted files systems for the local host.

Note – No administration is required for the `/etc/mnttab` mount table.

The /var/run File System

The `/var/run` file system is a new `tmpfs` mounted file system in the Solaris 8 Operating Environment. It is the repository for temporary system files that are not needed across system reboots in this Solaris Operating Environment 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 `root`.

The `/tmp` directory continues to be repository for temporary files.

Mounting File Systems

The /usr/sbin/mount Command

The `mount` command not only lists which file systems are currently mounted, it also provides the `root` user with a method for mounting file systems.

You can mount file systems manually by `root` running the `mount` command, or the system can automatically mount file systems at boot time after consulting the `/etc/vfstab` file.

Note – The `/etc/vfstab` file lists file systems to be mounted when the system is booted. This file is covered in detail later in this module.

Command Format

```
mount [ option(s)] device_name mount_point
```

Mounting a Local File System Manually

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 directory name. For example:

```
# mount /dev/dsk/c0t0d0s7 /export/home
```

In this example, the default action is to mount the file system with the following preferences: `read/write`, `setuid`, `nologging`, and `largefiles`.

- `read/write` – Indicates the file permissions. Access is based on the permissions of the files and directories in the file system. (The default for `hsfs` file systems is `ro`.)
- `setuid` – Permits the execution of `setuid` programs in the file system.

- `nologging` – Disables logging for the `ufs` file system.
- `largefiles` – Allows for the creation of files larger than 2 gigabytes. A file system mounted with this option may contain large size files.
- `onerror=action` – Specifies the action that `ufs` 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.

Note – Due to file system overhead, the largest file size that can be created is 866 Gbytes.

Using Options With the mount Command

When using `mount` options on the command line, the options are preceded by the `-o` flag. When multiple options are used, they are entered as a comma separated list following the `-o` flag.

```
mount -o options,option,... device_name mount_point
```

Some options used to mount local file systems include: `ro`, `nosetuid`, `noatime`, `nolargefiles`, and `logging`.

- `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 files, reducing 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/c1t0d0s7 /export/home
```

- **nolargefiles** – Prevents a file system containing one or more “large files” from being mounted. For example:

```
# mount -o nolargefiles /dev/dsk/c0t0d0s7 /export/home
```

Using 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 **root** needs to mount it with this option, then the large file(s) must be located, and moved or removed from the file system. Then you must run the file system check program manually to update the superblock information.

The mount will also fail if the file system at one time contained a large file, even though it was moved or removed. You must run the file system check program to clear the old information and allow the file system to be mounted.

Note – Module 9, “Maintaining File Systems” describes the file system check program (**fsck**).

- **logging** – Enables logging for a **ufs** file system.

For example:

```
# mount -o logging /dev/dsk/c0t0d0s7 /export/home
```

UFS file system logging is a process of storing file system transactions, or changes that make up a complete file or directory operation, into a log before they are applied to the file system. Once a transaction is stored, the complete transaction can be applied or reapplied to the file system later.

The **ufs** log is allocated from free blocks in the file system. It is sized approximately 1 Mbyte per 1 Gbyte, up to a maximum of 64 Mbytes.

As a ufs log reaches its maximum size, it begins to write transactions to the file system (for example, disk). When the file system is unmounted the entire ufs log is emptied and all transactions are written to disk.

UFS logging offers two advantages. First, it prevents file systems from becoming inconsistent; therefore, eliminating the need to run lengthy fsck scans. Secondly, you can bypass fsck scanning, which reduces the time required to reboot a system if it was stopped by a method other than an orderly shutdown.

Automatic Mounting of File Systems

The Virtual File System Table: /etc/vfstab

The Solaris Operating Environment provides several methods for automating file system mounts.

One method is to add the file system(s) to the `/etc/vfstab` file. This file lists all the file systems that are to be automatically mounted at system boot time.

The `/etc/vfstab` file provides you with another important feature.

If the `/etc/vfstab` file contains the mapping between the mount point and the actual device name, `root` can manually mount a file system specifying only the mount point on the `mount` command-line.

For example:

```
# mount /export/home
```

The /etc/vfstab File

A default `/etc/vfstab` file is created during the Solaris Operating Environment software installation, based on your selections.

However, the system administrator can edit the `/etc/vfstab` file whenever file entries need to be added or modified.

The following is an example of an `/etc/vfstab` file, on a system with one disk (`c0t0d0`).

The file format includes seven fields per line entry, each field is separated by a Tab. A - (dash) character indicates an empty field. Commented lines begin with the `#` symbol.

cat /etc/vfstab

#device #to mount	device to fsck	mount point	FS type	fsck pass	mount at boot	mount options
#/dev/dsk/c1d0s2	/dev/rdisk/c1d0s2	/usr	ufs	1	yes	-
fd	-	/dev/fd	fdfs	-	no	-
/proc	-	/proc	proc	-	no	-
/dev/dsk/c0t0d0s1	-	-	swap	-	no	-
dev/dsk/c0t0d0s0	/dev/rdisk/c0t0d0s0	/	ufs	1	no	-
/dev/dsk/c0t0d0s6	/dev/rdisk/c0t0d0s6	/usr	ufs	1	no	-
/dev/dsk/c0t0d0s3	/dev/rdisk/c0t0d0s3	/opt	ufs	1	yes	noatime
/dev/dsk/c0t0d0s7	/dev/rdisk/c0t0d0s7	/export/home	ufs	1	yes	logging
swap	-	/tmp	tmpfs	-	yes	-

To add a line entry, you need the following information: the device where the file system resides; the name of the mount point; the type of file system; whether it is to be mounted automatically during a system boot; and any mount options. For example:

device to mount — The block 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 *character* device to be checked by the file system check program (fsck).

mount point — The name of the directory where the device should be added to the Solaris Operating Environment directory tree.

FS type — The type of file system to be mounted.

fsck pass — Indicates whether the file system is to be checked by fsck at boot time. A whole number placed in this field indicates a *yes*. A - (dash) or a 0 (zero) indicates a *no*.

mount at boot — Enter a *yes* to enable the mountall command to mount the file systems at boot time. Enter a *no* to prevent a file system mount at boot time.

Note – For / (root) and /usr, the mount at boot field value is specified as *no*. These file systems are mounted by the kernel as part of the boot sequence before the mountall command is run.

mount options — A comma-separated list of options to be passed to the **mount** command.

The /usr/sbin/mountall Command

The **/etc/vfstab** file is read by the **/usr/sbin/mountall** command during the system boot sequence; and mounts all file systems specified in **vfstab** that have a **yes** in the **mount at boot** field.

The **root** user can use this command to manually mount every file system in **/etc/vfstab** that has a **yes** in the **mount at boot** field. For example:

```
# mountall
```

To mount only the local file systems specified in the **/etc/vfstab** file:

```
# mountall -l
```

Checking File Systems Before Mounting

Each local file system in the **vfstab** file that has a **device** to **fsck** entry and a **fsck pass** number is checked by **fsck** to determine if the file system is in a usable state to be safely mounted.

If the file system is found to be in an unusable state (for example, corrupted), it is repaired by **fsck** before the mount is attempted. Any local file systems with a **'-'** or **'0'** (zero) entry in the **fsck pass** field will attempt to be mounted without being checked.

Unmounting File Systems

The /usr/sbin/umount Command

Unmounting a file system using the `umount` command removes it from the file system mount point and deletes the entry from the `/etc/mnttab` file.

Some file system administration tasks cannot be performed on mounted file systems.

A file system is commonly unmounted if it is no longer needed, if it needs to be checked and repaired by `fsck`, or if it needs to be backed up completely.

Note – Notify users before unmounting a file system they are currently accessing.

To manually unmount a file system using the mount point or directory name:

```
# umount /export/home
```

or

```
# umount /dev/dsk/c0t0d0s7
```

Automatic Unmounting of File Systems

The /usr/sbin/umountall Command

The `/etc/mnttab` file is also read by the `/usr/sbin/umountall` command during the system shutdown sequence and unmounts all file systems specified in `vfstab` except `/` (root), `/usr`, `/proc`, `/dev/fd`, `/var`, `/var/run`, and `/tmp`.

Manually Unmounting all File Systems

This command can be run by `root` to manually unmount all the file systems listed in `/etc/mnttab`. For example:

```
# umountall
```

To unmount all local file systems specified in the `/etc/mnttab` file:

```
# 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.

Commands to Unmount a Busy File System

Any file system that is busy is not available for unmounting. Both the `umount` and `umountall` command 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 directory in the file system; a user is in the file system mount point directory; a program has a file open in that file system, or it is being shared.

There are two methods to make a file system available for unmounting if it is busy.

- `fuser` command – To list all the processes accessing the file system, and kill them if necessary.
- `umount -f` command – To force the unmount of a file system.

Note – The `umount -f` command is new in the Solaris 8 Operating Environment.

Using the `fuser` Command

To stop all processes from accessing a file system:

1. As `root`, list all the processes accessing the file system. Use the following command to identify which processes need to be terminated.

```
# fuser -cu mount_point
```

This 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` is sent to each process using the file system.

3. Verify 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 **root**, you can unmount a file system even if it is busy using the **-f** (force) option with **umount**. This is a new option in the Solaris 8 Operating Environment.

```
# umount -f mount_point
```

The file system is unmounted even if there are open files. A forced unmount can result in loss of data. However, it is particularly useful for unmounting a shared file system if the remote file server is non-functional.

Procedure for Mounting a New File System

The general procedure outlined below briefly describes the steps for adding a new disk to the system, preparing the disk to hold a file system, and mounting the file system.

1. Set up the disk hardware. Includes setting address switches and connecting cables.
2. Perform a reconfiguration boot to add support for the new device.
3. Use the `format` utility to partition the disk into one or more slices.
4. Create a new file system structure on one slice using the `newfs` command.
5. Create a mount point for the file system by creating a new directory in the `root` file system using the `mkdir` command. For example:

```
# mkdir /database
```

6. Mount the new file system manually using the `mount` command, For example:

```
# mount /dev/dsk/c1t3d0s3 /database
```

7. Check to see if the file system is mounted with the `mount` command.
8. Edit the `/etc/vfstab` file to add a line entry for the new file system. The file system will automatically be mounted whenever the system boots.

#device #to mount	device to fsck	mount point	FS type	fsck pass	mount at boot	mount options
<code>/dev/dsk/c1d0s2</code>	<code>/dev/rdisk/c1d0s2</code>	<code>/usr</code>	ufs	1	yes	-
<code>fd</code>	-	<code>/dev/fd</code>	fdfs	-	no	-
<code>/proc</code>	-	<code>/proc</code>	procfs	-	no	-
<code>/dev/dsk/c0t0d0s1</code>	-	-	swapfs	-	no	-
<code>dev/dsk/c0t0d0s0</code>	<code>/dev/rdisk/c0t0d0s0</code>	<code>/</code>	ufs	1	no	logging
<code>/dev/dsk/c0t0d0s6</code>	<code>/dev/rdisk/c0t0d0s6</code>	<code>/usr</code>	ufs	1	no	-
<code>/dev/dsk/c0t0d0s3</code>	<code>/dev/rdisk/c0t0d0s3</code>	<code>/opt</code>	ufs	2	yes	noatime
<code>/dev/dsk/c0t0d0s7</code>	<code>/dev/rdisk/c0t0d0s7</code>	<code>/export/home</code>	ufs	2	yes	logging
<code>swap</code>	-	<code>/tmp</code>	tmpfs	-	yes	-
<code>/dev/dsk/c1t3d0s3</code>	<code>/dev/rdisk/c1t3d0s3</code>	<code>/database</code>	ufs	2	yes	-

Removable Media Device Management

To access file systems on diskettes and CD-ROMS, the Solaris Operating Environment gives users a standard interface referred to as Volume Management.

Volume Management provides three major benefits:

- It automatically mounts diskettes and CD-ROMs for users.
- It allows access to diskettes and CD-ROMs without having to become `root`.
- It can give other systems on the network automatic access to any diskettes and CD-ROMs currently inserted in the local system.

The volume management service is controlled by the `/usr/sbin/vold` daemon. By default, this service is always running on the system to automatically manage diskettes and CD-ROMs for regular users.

Volume management provides automatic detection of CD-ROMs. However, it does not detect the presence of a diskette that has been inserted in the drive until it is informed, by the `volcheck` command. You run this command to instruct `vold` to check the diskette drive for installed media.

Note – Automatic detection of diskettes would cause excessive reads, which would quickly wear out the drive.

Accessing Mounted Diskettes and CD-ROMs

To make working with diskettes and CD-ROMs simple for your users, each device is mounted in an easy-to-remember location by `vold`.

- For diskettes, `vold` automatically mounts the device after you insert the diskette and run the `volcheck` command.
- For CD-ROMs, `vold` automatically mounts the device when you insert the CD into the drive.

If `vol`d detects that the mounted diskette or CD-ROM contains a file system, then the device is mounted at the directory location described in Table 8-1.

Table 8-1 Directory Locations

Media Device	Access file systems On
First diskette drive	/floppy/floppy0
First CD-ROM drive	/cdrom/cdrom0

If `vol`d detects the mounted diskette or CD-ROM does not contain a file system, the raw device is accessible using the following paths described in Table 8-2.

Table 8-2 Paths for Accessing Raw Devices

Media Device	Access Raw Device On
First Diskette Drive	/vol/dev/aliases/floppy0
First CD-ROM Drive	/vol/dev/aliases/cdrom0

When volume management is running on the system, a regular user can easily access a diskette or CD-ROM following these basic steps:

1. Insert the media.
2. For diskettes only, use the `volcheck` command.
3. Work with files on the media.
4. Eject the media.

Administering Volume Management

To restrict regular users from accessing diskettes or CD-ROMs on the system, `root` can terminate the volume management service.

To stop volume management from running on a system temporarily, the following command would be run by `root`.

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

To restart the volume management service, the following command is invoked by root.

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

Administering Volume Management

Two configuration files are used by volume management.

File	Description
/etc/vold.conf	The volume management configuration file. This defines items such as what action should be taken when media is inserted or ejected, what devices are used, and what file system types are unsafe to eject.
/etc/rmmount.conf	The rmmount command configuration file. The rmmount command is a removable media mounter that is executed by the volume management daemon whenever a CD-ROM or diskette is inserted.

Accessing a Diskette or CD-ROM Without Volume Management

When volume management is not running, then only `root` can mount and access a diskette or CD-ROM, using the following:

1. Insert the media device.
2. Become `root`.
3. Create a mount point, if necessary.
4. Determine the file system type.
5. Mount the device using the proper mount options.
6. Work with files on the media device.
7. Unmount the media device.
8. Eject the media device.
9. Exit the `root` session.

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 mounting a different type of file system, its type may have to be specified on the command line.

You use `-F` option on the `mount` command to specify the type of file system to be mounted.

Specifying a `hsfs` File System Type

As `root`, to mount a file system that resides on a CD-ROM, when the volume management services are stopped:

```
# 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 used, `/cdrom` is a pre-existing directory in the Solaris Operating Environment.

Specifying a `pcfs` File System Type

As `root`, to mount a file system that resides on a diskette, when the volume management services are stopped:

```
# mkdir /pcfs
# mount -F pcfs /dev/diskette /pcfs
```

In this example, the file system type is `pcfs`, the file system resides on the device `/dev/diskette`, and the mount point used, `/pcfs` had to be created.

Determining a File System's Type

Because the `mount` command needs the file system type to be specified to function properly, it must be explicitly specified, or it will have to be determined by searching the following files.

- The `/etc/vfstab` for FS type field.
- The `/etc/default/fs` file for local file system type.
- The `/etc/dfs/fstypes` file for remote file system type.

If the file system's type has not been explicitly specified on the command line using `mount -F FStype` option, `mount` looks in `/etc/vfstab` to determine the file system's type, using its block device name, raw device name, or mount point directory name.

If you cannot determine the file system's type by searching `/etc/vfstab`, `mount` uses the default file system type specified in either `/etc/default/fs` or `/etc/dfs/fstypes`, depending on whether the file system is local or remote.

The default local file system type is specified in `/etc/default/fs` by the line entry `LOCAL=fstype`. For example:

```
LOCAL=ufs
```

The default remote file system type is determined by the line entry in the `/etc/dfs/dfstypes` file. For example:

```
nfs NFS Utilities
```

Finding a File System's Type

To determine a file system's type to use with the `-F` option of the `mount` command, run the following `grep` command to display the information:

```
# grep mount-point fs-table
```

`mount-point` — Specifies the mount point directory name of the file system. For example, the `/var` directory.

fs-table — Specifies the absolute path to the file system table used to search for the file system's type.

If the file system is mounted, *fs-table* should be `/etc/mnttab`. If the file system is not mounted, *fs-table* should be `/etc/vfstab`.

The following example uses the `/etc/vfstab` to determine the type of the `/export/home` file system.

```
# grep /export/home /etc/vfstab  
/dev/dsk/c0t0d0s7  dev/rdisk/c0t0d0s7  /export/home  ufs  1  yes  -  
#
```

The fstyp Command

The `fstyp` command can also be used with the raw device name of the disk slice to determine a file system's type. For example:

```
# fstyp /dev/rdisk/c0t0d0s7  
ufs
```

Exercise: Mounting File Systems



Exercise objective – In this lab you will create mount points, mount file systems, and 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.

Task Summary

- Record the default mount options used by the / (root) file system mounted on your system. Mount the file system found on slice 4 of your spare disk as the directory /morespace. Verify the mount options applied to /morespace.
- Create a new file in /morespace that contains one line of text. Record the modify time for this file. Use `ls` 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 /morespace. 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 into /etc/vfstab that causes /morespace to mount on reboot. Reboot the system and verify that /morespace 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 directory to /dir0/dir1. In the original terminal window, attempt to unmount /opt/dir1. Record error messages. Attempt to forcibly unmount /dir0/dir1. Record the result. Attempt to use `pwd` in the second terminal window. Record what happens.

Tasks

1. Log in as `root` and open a terminal window. Use `mount` 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
```

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/c1t3d0s4 /morespace
# mount
```

4. Change directory to `/morespace` and create a new file that has one line of content. Example:

```
# cd /morespace
# cat > testfile
This is a test.
<ctrl> d
#
```

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 `cat` to display the file. Again check and record the access time. It should differ from the access time indicated in the previous step.

```
# cat testfile
This is a test
# ls -lu
```

8. Change directory to `/`. Unmount `/morespace`. Re-mount the same file system as `/morespace`, but add the option that prevents update of access time values. Verify the options applied to the mount. Example:

```
# cd /
# umount /morespace
# mount -o noatime /dev/dsk/c1t3d0s4 /morespace
# mount
```

9. Return to `/morespace` and use `cat` to display your test file. Again check and record the access time. It should match the last access time that existed prior to unmounting and mounting `/morespace`.

```
# cd /morespace
# cat testfile
This is a test
# ls -lu
```

10. Add a line to `/etc/vfstab` to make the mount for `/morespace` happen when you boot the system. For example:

```
/dev/dsk/c1t3d0s4  /dev/rdisk/c1t3d0s4  /morespace  ufs  2  yes  noatime
```

11. Reboot your system. Login as `root` and open a terminal window. Use `mount` to verify that `/morespace` is mounted.

```
# reboot
(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`. For example:

```
# mkdir /dir0
# mount /dev/dsk/c1t3d0s0 /dir0
```

13. Create a directory called `/dir0/dir1`. Mount the file system that resides on slice 1 of your spare disk as `/dir0/dir1`. For example:

```
# mkdir /dir0/dir1
# mount /dev/dsk/c1t3d0s1 /dir0/dir1
```

14. Open a second terminal window. In this new window, change directory to `/dir0/dir1`.

```
# cd /dir0/dir1
```

15. In your original terminal window, attempt to unmount the file system mounted below `/dir0/dir1`. What message displays? Does the file system unmount?

```
# umount /dev/dsk/c1t3d0s1
# mount
```

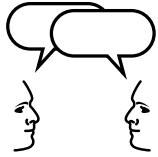
16. In your original terminal window, again attempt to unmount the file system mounted below `/dir0/dir1`. Add the `-f` option to the `umount` command. What message displays? Does the file system unmount?

```
# umount -f /dev/dsk/c1t3d0s1
# mount
```

17. In the second terminal window, attempt to determine your current working directory. What message displays? Change directory to `/` (root) and verify that `pwd` works.

```
# pwd
# cd /
# pwd
```

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

1. Log in as `root` and open a terminal window. Use `mount` to list the file systems that are currently mounted on your system. What are the default mount options applied to the `/` (root) file system?

```
read/write/setuid/intr/largefiles/onerror=panic/d
ev=2200000
```

2. 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.

```
read/write/setuid/intr/largefiles/onerror=panic/d
ev=80001c
```

15. In your original terminal window, attempt to unmount the file system mounted below `/dir0/dir1`. What message displays? Does the file system unmount?

```
umount: /dir0/dir1 busy
```

The file system does not unmount.

16. In your original terminal window, again attempt to unmount the file system mounted below `/dir0/dir1`. Add the `-f` option to the `umount` command. What message displays? Does the file system unmount?

No messages displays. The file system unmounts.

17. In the second terminal window, attempt to determine your current working directory. What message displays? Change directory to `/` (root) and verify that `pwd` works.

Cannot determine current directory

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Define the term mount point
- ☐ Identify mounted and unmounted file systems
- ☐ Mount file systems using the commands `mount` and `mountall`
- ☐ Describe some of the commonly used options of the `mount` command: `noatime`, `nolargefiles`, and `logging`
- ☐ Describe the purpose and format of the `/etc/mnttab` and `/etc/vfstab` files
- ☐ Define the procedure for mounting different types of file systems
- ☐ List the system files used to determine a file system's type
- ☐ Unmount local and remote file systems using the commands `umount` and `umountall`
- ☐ Forcibly unmount a busy file system
- ☐ Describe how to mount and access file systems residing on removable media devices, such as diskettes and CD-ROMs

Objectives

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

- Describe why fsck is necessary
- Describe how to check and repair a file system
- Display disk space usage by file systems
- Display disk usage of a directory
- Display disk usage by user name
- Demonstrate how to repair the `/etc/vfstab` file when the system fails to boot completely

Additional Resources



Additional resources – The following references can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10
- *Solaris 8 System Administration Guide, Volume II*, Part Number 805-7229-10

The File System Check Program

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 corrects or repairs any inconsistencies or damage found.



Caution – Never run `fsck` on a mounted file system. It could leave the file system in an unusable state and delete data. Always run `fsck` on unmounted file systems *only*.

Every time a system boots, `fsck` automatically performs a file system consistency check. `fsck` checks and repairs any problems encountered in file systems before they are mounted.

When a file system is mounted with the `ufs` logging option, it eliminates the need to run `fsck` because logging prevents the file system from becoming inconsistent.

Note – The status of a file system's state flag determines whether the file system needs to be checked by `fsck`. When the file system is "clean," "stable," or "logging," file system checks are not run.

Data Inconsistencies Checked by fsck

The `fsck` command makes several passes through a file system, each time it scans to check the following types of file system inconsistencies.

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 its name. If the `lost+found` directory does not exist `fsck` creates it, and if there is not enough space in the `lost+found` directory `fsck` increases its size.

Superblock Consistency

The file system superblock is checked for inconsistencies involving file system size, free block count, and free inode count.

Inode Consistency

The `fsck` command checks for the allocation state (inodes allocated or unallocated), the type, link count, duplicate blocks (blocks already claimed by another inode), bad blocks, inode size, and block count for each inode. Any unreferenced inode with a non-zero link count is linked to the file systems `lost+found` directory.

Data Block Consistency

The `fsck` command cannot check ordinary data blocks, but it can check directory data blocks. In these data blocks, it checks for inodes pointing 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.

Cylinder Group Block Consistency

The `fsck` command checks unallocated data blocks claimed by inodes, unallocated data block count, and unallocated inode count.

Phases of fsck

The `fsck` command runs through five phases for each file system in the `/etc/vfstab` file that has a device to `fsck` and `fsck pass` entry. The five phases are:

- Phase 1: Check Blocks and Sizes – Checks inodes for inconsistencies.
- Phase 2: Check Pathnames – Checks directory inode consistencies.
- Phase 3: Check Connectivity – Checks that all directories are connected to the file system.

- Phase 4: Check Reference Counts – Compares link count information from Phases 2 and 3, correcting discrepancies.
- Phase 5: Check Cylinder Groups – Checks free blocks and the used inode maps for consistency.

The following example shows `fsck` running through its five phases on the `/export/home` file system.

```
# fsck /dev/rdisk/c0t0d0s7
** /dev/rdisk/c0t3d0s7
** Last Mounted on /export/home
** 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)
#
```

The last line displayed by `fsck` contains the following information about the file system

- The number of files is 7
- The number of Kbytes used is 14
- The number of Kbytes free is 2798265
- The number of free block fragments is 17
- The number of free blocks is 347891
- The ratio of free block fragments to total Kbytes is 0.0%

The file system check program can operate in two modes: *non-interactive* and *interactive*.

Non-Interactive Mode

During a normal system boot, `fsck` operates in non-interactive mode, often referred to as *preen*, or silent mode.

During this process, `fsck` repairs only minor inconsistency problems that can be corrected.

However, if a more serious inconsistency is found, and a decision has to be made, the `fsck` program terminates and leaves the system in single-user mode. You must run `fsck` interactively to continue.

Interactive Mode

During this process, `fsck` lists each problem it encountered, followed by a suggested corrective action, in the form of a question that requires a yes or no response.

By responding yes, `fsck` applies the corrective action and moves on. By responding no, `fsck` will often simply repeat the original problem and suggest corrective action, and not move forward until you respond with a yes.

For example:

```
# fsck /export/home
** /dev/rdisk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
INCORRECT BLOCK COUNT I=743 (5 should be 2)
CORRECT?
. . .
```

Using the fsck Command

The following examples demonstrate how the system `root` can use the `fsck` command to check the integrity of file systems.

- To check a single unmounted file system, execute the following command.

```
# fsck /dev/rdisk/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, execute the following command.

```
# fsck /opt
```

The following example has `fsck` check and repair the file system in non-interactive mode and exit if a serious problem requiring intervention is encountered.

```
# fsck -o f,p /dev/rdisk/c0t0d0s5
/dev/rdisk/c0t0d0s5: 77 files, 9621 used, 46089 free
/dev/rdisk/c0t0d0s5: (4 frags, 57 blocks, 0.0% fragmentation)
```

The `f` option forces checking of the file system regardless of the state of its superblock clean flag.

The `p` option checks and fixes the file system non-interactively (preen). The program exits immediately if a problem requiring intervention is found. This option is required to enable parallel file system checking.

Troubleshooting with fsck

If problems occur in a file system, you are alerted by fsck. Some of the more common file system errors that require interactive intervention are described in the following sections.

Reconnecting an Allocated Unreferenced File

In this example, the fsck program discovers an inode that is allocated but unreferenced or not linked in any directory.

A yes response to the RECONNECT? question causes fsck to save the file to the lost+found directory and names it using the inode number.

```
** Phase 3 - Check Connectivity
UNREF FILE I=788 OWNER=root MODE=100644
SIZE=19994 MTIME=Jan 18 10:49 1999
RECONNECT? y
```

To determine what type of file had to moved to the lost+found directory by fsck:

1. List the contents of the file system's lost+found directory, for example:

```
# ls /export/home/lost+found
#788
```

2. Determine the file type, using the file command, for example:

```
# file /export/home/lost+found/#788
/export/home/lost+found/#788: ascii text
```

3. To view the contents of the 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, (e.g. a word processing document), it would be necessary to 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, the file can be copied back to its original location in the file system. For example:

```
# cp /export/home/lost+found/#788 /export/home/user1/report
```

Adjusting a Link Counter

In this example, the `fsck` program discovers that the value of a directory inode link counter and the actual number of directory links are inconsistent.

A yes response to the `ADJUST?` question causes `fsck` to correct the directory inode link counter from 4 to 3.

```
** Phase 4 - Check Reference Counts
LINK COUNT DIR I=2 OWNER=root MODE=40755
SIZE=512 MTIME=Jan 18 15:59 1999 COUNT 4 SHOULD BE 3
ADJUST? y
```

Salvaging the Free List

In this example, the `fsck` program discovers that the unallocated block count and the free block number listed in the superblock are inconsistent.

A yes response to the `SALVAGE?` question causes `fsck` to update the information in the file system superblock.

```
** Phase 5 - Check Cyl groups
CG 0: BAD MAGIC NUMBER
FREE BLK COUNT(S) WRONG IN SUPERBLK
SALVAGE? y
```

Using Backup Superblocks

Superblock corruption can cause a file system to be unmountable.

You know when a file system is unusable when the message “Can’t mount *file system name*” appears. For example:

```
Can't mount /dev/dsk/c0t0d0s7
```

which can appear during a system boot or when manually mounting the file system.

If `fsck` fails because of a corrupted superblock it returns an error message informing you that it must be run using an alternative superblock backup to recover the file system.

The corrective action is to run `fsck` using the `-o` option with the `b` flag. The `b` flag is followed by a backup superblock number.

Every file system always has an alternate backup superblock at block number 32, which can be given to `fsck` to repair the main superblock. For example:

```
# fsck -o b=32 /dev/rdisk/c1t3d0s0
Alternate super block location: 32.
** /dev/rdisk/c1t3d0s0
** Currently 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
171 files, 3762 used, 5984 free (79 frags, 748 blocks, 0.1% fragmentation)
#
```

The `fsck` program takes the information in the backup superblock, compares it with the actual file system and attempts to rebuild the main superblock.

If however, this block is part of the file system that was damaged it is unusable. You must select another backup superblock for `fsck` to continue.

To list the locations of all the alternate backup superblocks in the file system, run the `newfs -N` command. For example:

```
# newfs -N /dev/rdisk/c#t#d#s#
```



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, then you must run `newfs -N` using the identical parameters to generate identical superblock locations.

You use the `-N` option to print out file system parameters *that would be used* to create a new file system without actually creating the file system.

A portion of that print out is a list of all the backup superblock locations that can be used with `fsck -o b#`. For example:

```
# newfs -N /dev/rdisk/c0t0d0s7
newfs -N /dev/rdisk/c0t0d0s7
/dev/rdisk/c0t0d0s7: 3537040 sectors in 2327 cylinders of 19 tracks, 80 sectors
      1727.1MB in 73 cyl groups (32 c/g, 23.75MB/g, 5888 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 48752, 97472, 146192, 194912, 243632, 292352, 341072, 389792,
487232, 535952, 584672, 633392, 730832, 779552, 828272, 876992,
925712, 974432, 1023152, 1071872, 1169312, 1218032, 1266752, 1315472,
1364192, 1412912, 1461632, 1510352, 1556512, 1605232, 1653952, 1702672,
1751392, 1800112, 1848832, 1897552, 1946272, 1994992, 2043712, 2092432,
2141152, 2189872, 2238592, 2287312, 2336032, 2384752, 2433472, 2482192,
2530912, 2579632, 2628352, 2677072, 2725792, 2774512, 2823232, 2871952,
#
```

You could use any other alternative superblock number in the list with `fsck`. For example:

```
# fsck -o b=535952 /dev/rdisk/c0t0d0s7
Alternate super block location: 535952.
** /dev/rdisk/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)
#
```


Monitoring File System Usages

An important activity of a system administrator is to monitor file system usage on a regular basis. There are four useful commands available for this task, which include `df`, `du`, `ff` and `quot`.

- `df` – Display the number of free disk blocks and files.
- `du` – Summarize disk usage.
- `ff` – List files names and statistics for a file system.
- `quot` – Summarize file system ownership.

The `df` Command

You use the `df` command to display the amount of disk space occupied by mounted file systems. It lists the amount of used and available space, and how much of the file system's total capacity is used.

Command Format

`df [-k] [directory]`

Options

`-k` Displays usage in Kbytes and subtracts the space reserved by the operating system from the amount of available space.

To display the capacity of file systems, use the following command:

```
# df -k
file system      kbytes    used    avail capacity  Mounted on
/dev/dsk/c0t3d0s0 38111    19196   18877    51%           /
/dev/dsk/c0t3d0s6 565503   361529  203409   64%          /usr
/proc              0         0        0        0%          /proc
fd                 0         0        0        0%          /dev/fd
/dev/dsk/c0t3d0s1 25159     4886   20248    20%          /var
/dev/dsk/c0t3d0s5 27439    20362   7050     75%          /opt
swap              45980      12   45968     1%          /tmp
```

The amount of space that is reported as `used` and `avail` is less than the amount of total space in the file system. A fraction of space, from 1 percent to 10 percent, is reserved in each file system.

When all the reported space on the file system is in use, its capacity is displayed as 100 percent. Regular users receive the message “File System Full” and cannot continue working. The reserved space is still available to `root`, who can delete or back up files to free space in the file system.

The following lists the fields displayed by `df -k`

<code>file system</code>	Mounted file system
<code>kbytes</code>	Size of the file system in Kbytes (1024 bytes)
<code>used</code>	Number of Kbytes used
<code>avail</code>	Number of Kbytes available
<code>capacity</code>	Percentage of file system capacity used
<code>Mounted on</code>	Mount point

The du Command

You use the `du` command to display the number of disk blocks (512 bytes) used by directories and files.

Command Format

```
du [-a] [-s] [-k] [directory]
```

Options

- k Displays in Kbytes.
- s Displays only the summary in 512-byte blocks. Using the s and k options together will show the summary in Kbytes.
- a Displays the number of blocks used by all files and directories within the specified directory hierarchy.

To display disk usage in kilobytes, execute the following:

```
# cd /opt
# du -k
8      ./lost+found
3      ./SUNWits/Graphics-sw/xil/lib
4      ./SUNWits/Graphics-sw/xil
16     ./SUNWits/Graphics-sw/xgl/demo
...

38     ./netscape/movemail-src
11392  ./netscape
20362  .
```

To display disk usage including files, execute the following:

```
# du -ak /usr
16     /usr/lost+found
2      /usr/X
2      /usr/lib/libICE.so
2      /usr/lib/libICE.so.6
2      /usr/lib/libMrm.so
...
6      /usr/kvm
...
723057 /usr
```

To display a summary of disk usage, execute the following:

```
# du -sk /usr
723057 /usr
```

The ff Command

The `ff` command provides a list of pathnames and inode numbers of files in the file system.

The command output is sorted in ascending inode number order. For example:

```
$ ff /dev/dsk/c1t3d0s5
/dev/dsk/c1t3d0s5:
inode# pathname
inode# pathname
inode# pathname
inode# pathname
inode# pathname
```

Some Options

- l – Prints a list of all pathnames for multiple-linked files.
- s – Prints the file size, in bytes, with each pathname.
- u – Prints the owner's login name with each pathname.
- i – Prints the inode number with each pathname.

The quot Command

The `quot` command displays how much disk space (in Kbytes) is being used by users.

Note – The `quot` command can only be run by `root`.

Command Format

```
quot [-af] [file system...]
```

Options

- a Reports on all mounted file systems
- f Includes number of files

To display disk space being used by users on all mounted file systems, execute the following:

```
# quot -af
/dev/rdisk/c0t0d0s0 (/):
    14326    1284    root
     4792     37    bin
      31     27    lp
       1      1    sys
/dev/rdisk/c0t0d0s6 (/usr):
    197394    6962    root
    161203   11884    bin
     2140    232    lp
       1      1    adm
```

The columns represent Kbytes 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, execute the following:

```
# quot -f /dev/dsk/c1t0d0s5
/dev/dsk/c1t0d0s5:
    134     62    root
    103     84   user1
    140     32   user9
```

Troubleshooting

Repairing Important Files if Boot Fails

The `/etc/vfstab` file is an important system file. If it becomes corrupted or contain editing errors, it can cause the system boot to fail.

The following procedure describes how to boot from the Solaris Operating Environment software CD-ROM to edit the `/etc/vfstab` file.

1. Insert the Solaris 8 Operating Environment software CD-ROM 1 of 2 into the CD-ROM drive.
2. Run a single-user boot from the CD-ROM.

```
ok boot cdrom -s
Boot device: /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f File and args -s
SunOS Release 5.8 Version Generic_106541-02 [UNIX(R) System V
Copyright (c) 1983-1999 by Sun Microsystems, Inc.
Configuring the /dev and /devices directories
-
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 `/ (root)` partition to check and repair any potential problems in the file system.

```
# fsck /dev/rdisk/c0t0d0s0
```
4. If `fsck` completed successfully, 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!
```

7. Unmount the file system.

```
# cd /
```

```
# umount /a
```

8. Reboot the system.

```
# reboot
```

Exercise: Maintaining File Systems



Exercise objective – In this exercise you create a new file system on an unused disk slice, destroy its superblock, and repair it using `fsck` and alternative superblocks. You also use `ff` and `fsck` to identify unreferenced files.

Preparation

This exercise requires a spare disk with a defined but unused slice 3. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

- Create a new file system on slice 3 of the spare disk. Check the file system with `fsck` and record if it reports any errors. Use the `dd` command as described in step 3 below to destroy the primary superblock of the new file system. Use `fsck` and the backup superblock found at sector 32 to repair the file system and main superblock. Verify the repair by running `fsck` again.
- Create a directory called `/dir3` and mount the same file system below it. Recursively copy the `/usr/lib/locale/iso_8859_1` directory to `/dir3`. Record the inode number of the `/dir3/iso_8859_1` directory. Use `ff` to list the files and their inode numbers on this file system, and save the output to a file. Unmount the file system. Use `clri` to clear the inode associated with `/dir3/iso_8859_1`.
- Run `fsck` on the file system and respond `y` to all questions. Mount the file system below `/dir3` and check for files and directories. In `/dir3/lost+found`, identify the files you find using the saved output from `ff`. List the steps required to reconstruct the original directory structure you made below `/dir3`. Unmount `/dir3` when finished.

Tasks

1. Create a new file system on slice 3 of the spare disk. For example:

```
# newfs /dev/rdisk/c1t3d0s3
```

2. Run fsck interactively to check the new file system.

```
# fsck /dev/rdisk/c1t3d0s3
```

Did fsck report errors?

3. Use the dd command to destroy the main superblock of the file system on slice 3. The count= option indicates the number of output blocks to write, of the size specified by the bs= option.

Note – For this exercise, only use 32 and 512 as the values for the count= and bs= arguments.

```
# dd if=/dev/zero of=/dev/rdisk/c1t3d0s3 count=32 bs=512
32+0 records in
32+0 records out
```

4. Run fsck interactively to check the new file system.

```
# fsck /dev/rdisk/c1t3d0s3
```

Did fsck report errors? If so, what corrective action does fsck suggest?

5. Run fsck and specify an alternate superblock. Block 32 is always one of the alternates available.

```
# fsck -o b=32 /dev/rdisk/c1t3d0s3
```

6. Run fsck again to verify that the file system was repaired.

```
# fsck /dev/rdisk/c1t3d0s3
```

7. Create a directory called `/dir3`. Mount the file system on slice 3 as `/dir3`.

```
# mkdir /dir3
# mount /dev/dsk/c1t3d0s3 /dir3
```

8. Copy the directory structure found below `/usr/lib/locale/iso_8859_1` to `/dir3`. Change directory to `/dir3` and list the inode number of the `iso_8859_1` directory. Record the inode number you find.

```
# cp -r /usr/lib/locale/iso_8859_1 /dir3
# cd /dir3
# ls -di iso_8859_1
```

9. Use `ff` to create a list of inodes and their associated file names found in this file system. Save the output of `ff` in a file below `/var/tmp`. For example:

```
# ff /dev/dsk/c1t3d0s3 > /var/tmp/c1t3d0s3_log
```

10. Change directory to `/` (root) and unmount `/dir3`.

```
# cd /
# umount /dir3
```

11. Use `clri` to clear the inode associated with the `iso_8859_1` directory on the unmounted file system. Use the inode number you recorded in step 8 for the second argument to `clri`.

```
# clri /dev/rdisk/c1t3d0s3 inode#
```

12. Run `fsck` to check the file system on slice 3. Note the messages `fsck` reports and respond `y` to all questions. Record the number of the inode(s) that `fsck` asks to remove.

```
# fsck /dev/rdisk/c1t3d0s3
```

13. Mount the same file system again as `/dir3`. Does the directory `iso_8859_1` exist? If not, why not?

```
# mount /dev/dsk/c1t3d0s3 /dir3
# cd /dir3
# ls
```

14. Change directory to `/dir3/lost+found`. Record the names of the items you find in `lost+found`. Use the saved `ff` command output to match the files in `lost+found` to their original file names.

```
# cd /dir3/lost+found
# ls -l
# cat /var/tmp/c1t3d0s3_log
```

File name in lost+found	Original file name
_____	_____
_____	_____
_____	_____

15. What steps would be required to reconstruct the original directory structure you created in `/dir3`?

16. Change directory to `/` (root) and unmount `/dir3` when finished.

```
# cd /
# umount /dir3
```

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

2. Run `fsck` interactively to check the new file system.

Did `fsck` report errors?

No.

4. Run `fsck` interactively to check the new file system.

Did `fsck` report errors? If so, what corrective action does `fsck` suggest?

`fsck` indicates that the magic number in the superblock is wrong, and suggests repairing it by using an alternate superblock. For example:

```
** /dev/rdisk/c1t3d0s3
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).
```

13. Mount the same file system again as `/dir3`. Does the directory `iso_8859_1` exist? If not, why not?

This directory does not exist because it was removed by `fsck`.

15. What steps would be required to reconstruct the original directory structure you created in `/dir3`?

It would be necessary to first create *the directory* `/dir3/iso_8859_1`, then move the files and directories *from* *lost+found* into `/dir3/iso_8859_1` using their names as they are listed in the file `/var/tmp/c1t3d0s3_log`.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Describe why fsck is necessary
- ☐ Describe how to check and repair a file system
- ☐ Display disk space usage by file systems
- ☐ Display disk usage of a directory
- ☐ Display disk usage by user name
- ☐ Demonstrate how to repair the /etc/vfstab file when the system fails to boot completely

Objectives

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

- Start the CDE Process Manager to monitor and control active processes
- Report active process statistics using the `prstat` command
- Schedule the automatic execution of commands, programs, or scripts using the commands `at` and `crontab`
- Define the files used to control user access to the commands `at` and `crontab`
- Create and execute an `at` job
- Describe the location and format of a `crontab` file
- Demonstrate the steps to create, view, edit, and remove a `crontab` file

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Part Number 805-7228-10

Processes Running on the System

A process is any program that is running on the system. All processes are assigned a unique *process identification number* (PID), which is used by the kernel to track and manage the process. The PID numbers are used by root and regular users to identify and control their processes.

Viewing Processes and PIDs

The `ps` (process status) command is commonly the method used for viewing a list of processes currently running on a system. However, there are two other methods for managing processes, which include:

- The CDE Process Manager
- The `prstat` command

Note – The `prstat` command is new with the Solaris 8 Operating Environment.

CDE Process Manager

The Solaris Operating Environment CDE provides a Process Manager window to monitor and control processes running on the local system.

To view the Process Manager, go through the Workspace Manager to locate and select the **Find Process** tool, as shown below.

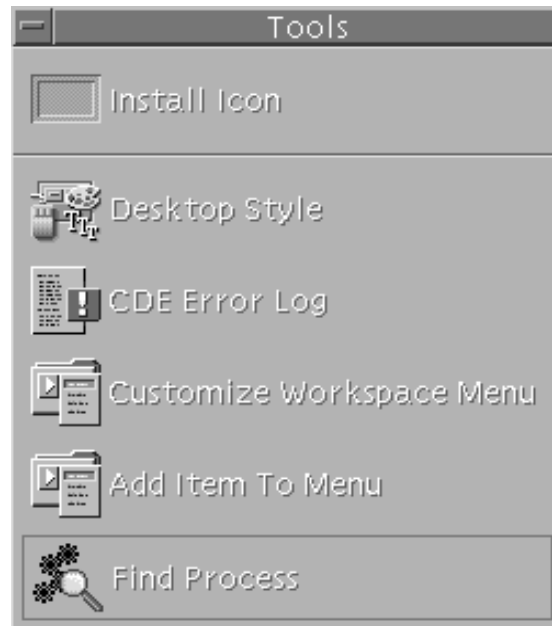


Figure 10-1 The Tools Menu

You can also start the CDE Process Manager on the command line by typing the command:

```
# /usr/dt/bin/sdtprocess &
```

ID	Name	Owner	CPU%	RAM	Swap	Started	Parent	Command
144	automoun	root	0.0	646	2584	Dec_06	1	/usr/lib/autofs/automountd
585	cat	root	0.0	210	840	Dec_09	566	/bin/cat /tmp/.removable/notify0
162	cron	root	0.0	416	1664	Dec_06	1	/usr/sbin/cron
231	dmispd	root	0.0	646	2584	Dec_06	1	/usr/lib/dm/dmispd
538	dsm	root	0.0	518	2072	Dec_09	1	/usr/dt/bin/dsm
5816	dtexec	root	0.0	697	2788	08:15:19	560	/usr/dt/bin/dtexec -open 0 -ttprocid 2.sRc82 01 552 128
5819	dtexec	root	0.0	697	2788	08:17:42	560	/usr/dt/bin/dtexec -open 0 -ttprocid 2.sRc82 01 552 128
5846	dtexec	root	0.1	697	2788	08:20:25	560	/usr/dt/bin/dtexec -open 0 -ttprocid 2.sRc82 01 552 128
562	dtfile	root	0.0	1946	7784	Dec_09	553	dtfile -session dtN0IRj_
235	dtlogin	root	0.0	1618	6472	Dec_06	1	/usr/dt/bin/dtlogin -daemon
245	dtlogin	root	0.0	1659	6536	Dec_06	235	/usr/dt/bin/dtlogin -daemon
5848	dtmail	root	0.9	2382	9528	08:20:25	5847	/usr/dt/bin/dtmail
553	dtsession	root	0.0	1847	7388	Dec_09	539	/usr/dt/bin/dtsession
561	dtterm	root	0.0	1649	6596	Dec_09	553	/usr/dt/bin/dtterm -session dt0KVI_z
563	dtterm	root	0.0	1650	6600	Dec_09	553	/usr/dt/bin/dtterm -session dtYenQG_
560	dtwm	root	0.4	2042	8168	Dec_09	553	dtwm
5796	fa.html	root	0.0	486	1944	07:56:36	1	/export/home/framemaker,v5.1/bin/sunxm.s5.sparc/fa.htm
247	fbconsole	root	0.0	513	2052	Dec_06	1	/usr/openwin/bin/fbconsole -d :0
501	fbconsole	root	0.0	513	2052	Dec_09	491	/usr/openwin/bin/fbconsole

Figure 10-2 The CDE Process Manager Window

The Process Manager can sort processes alphabetically or numerically depending on the column that is selected.

You can initiate a search using the Find command.

To terminate a process, highlight it and then either press Control+c or select kill from the PROCESS menu.

The `prstat` Command

The `prstat` command interactively examines and displays information about active processes on the system.

This command enables you to view information by specific processes, UIDs, CPU IDs, or processor sets. By default, `prstat` displays information about all processes sorted by CPU usage.

prstat

To quit `prstat` type: **q**

Table 10-1 Column Headings for the `prstat` Command

Column Heading	Description
PID	The process identification number of the process.
USERNAME	The login ID name 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: cpu – process is running on the CPU. sleep – process is waiting for an event to complete. run – process is in run queue. zombie – process terminated and parent not waiting. stop – 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 LWPs in the process.

Note – Lightweight process (LWP) is a virtual CPU or execution resource. LWPs are scheduled by the kernel to use available CPU resources based on their scheduling class and priority.

Table 10-2 describes some options for the `prstat` command.

Table 10-2 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: <code>cpu</code> , <code>time</code> , <code>size</code> , <code>rss</code> , and <code>pri</code> . 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 in the given list.
-U <i>euidlist</i>	Reports only processes that have an effective user ID is in the given list.

Scheduling the Automatic Execution of Commands

Users can schedule a job for a one-time execution at a specified time by using the `at` command.

Users can schedule a job to be executed repetitively, at regular intervals, by using a `crontab` file.

The `cron` daemon is responsible for scheduling and running these jobs.

Note – The `cron` daemon is started at system boot and runs continuously in the background.

The crontab Command

A `crontab` file is used to automatically execute commands or scripts repetitively, at regularly scheduled intervals. All `crontab` files are maintained in `/var/spool/cron/crontabs/username(s)`.

The `crontab` command enables the user to view, edit or remove a `crontab` file.

The crontab File Format

A crontab file consists of commands, one per line, that will be executed at regular intervals.

The beginning of each line contains date and time information that tells the cron daemon when to execute the command.

These first five fields are separated by spaces, and indicate when the command will be executed.

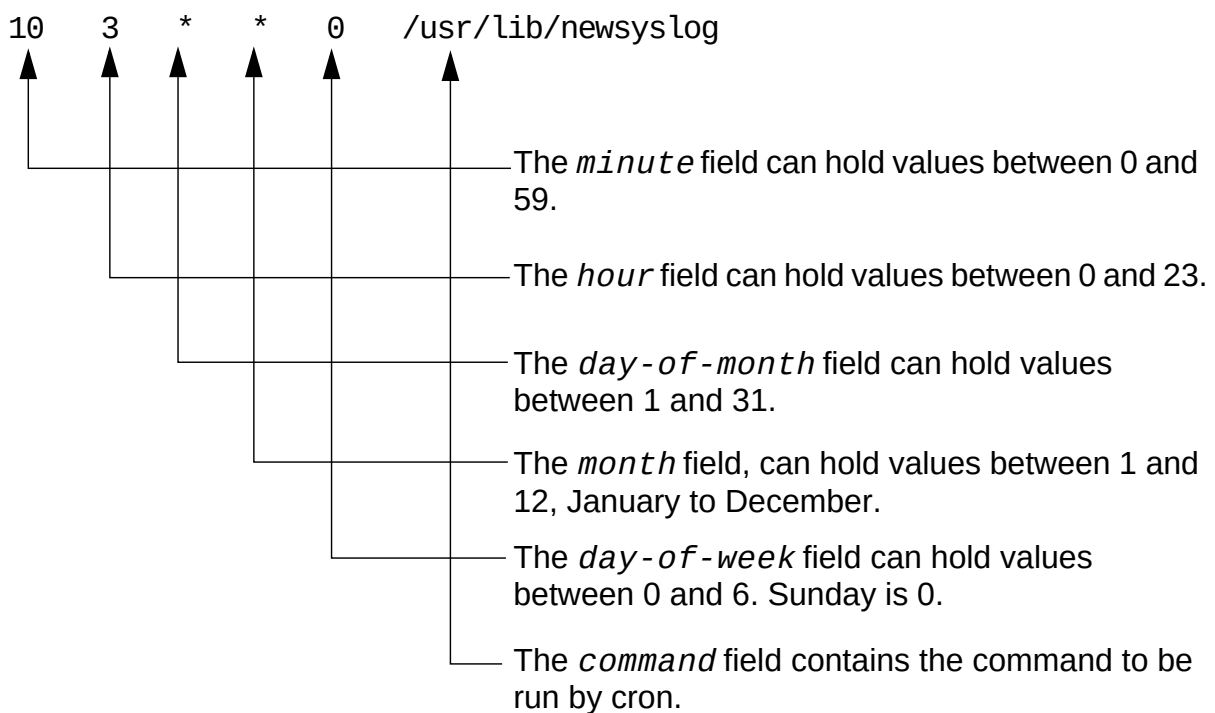


Figure 10-3 Fields in a crontab File

The first five fields can follow these format rules:

<i>n</i>	Matches if field value is <i>n</i>
<i>n,p,q</i>	Matches if field value is <i>n</i> , <i>p</i> , or <i>q</i>
<i>n-p</i>	Matches if field has values between <i>n</i> and <i>p</i> inclusive
<i>*</i>	Matches any value (or can be used as a placeholder)

crontab for the root User

A crontab file, `/var/spool/cron/crontabs/root` is provided in the Solaris Operating Environment for the `root` user. By default, regular users do not have crontab files.

The `root` crontab file contains the following command lines by default:

```
#ident "@(#)root      1.19      98/07/06 SMI"      /* SVr4.0 1.1.3.1      */

# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] &&
/usr/lib/gss/gsscred_clean
```

The first line instructs `cron` to run `logchecker` at 3:10AM on Sunday and Thursday.

The second line instructs `cron` to run `newsyslog` at 3:10AM every Sunday.

The third line instructs `cron` to execute `nfsfind` every Sunday at 3:15AM.

The fourth line instructs `cron` to check daily for Daylight Savings Time and make corrections if necessary.

The fifth line instructs `cron` to check for and remove duplicate entries in the Generic Security Service table, `/etc/gss/gsscred_db`.

Using crontab -l to View a Crontab File

To view the contents of the root crontab run the following command, as root:

```
# crontab -l
```

This is the same command regular users would run to view the contents of their own crontab file.

As root, you can view the contents of any regular user's crontab by running the command:

```
# crontab -l username
```

Editing a crontab 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 cron on which editor to use to open the file. For example:

```
# EDITOR=vi  
# export EDITOR
```

2. Run the following crontab command to open your crontab file, and add the following entry.

```
# crontab -e  
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console  
:wq
```

Controlling crontab Access

Control access to crontab with two files in the `/etc/cron.d` directory:

- `/etc/cron.d/cron.deny`
- `/etc/cron.d/cron.allow`

These files permit only specified users to perform crontab tasks such as creating, editing, displaying, or removing their own crontab files.

A default `cron.deny` file is provided in the Solaris Operating Environment.

The `cron.allow` file does not exist by default, so all users (except those listed in the `cron.deny` file) can access `crontab`. By creating a `cron.allow` file, you can list those users who can access `crontab` commands.

These two files consist of a list of user names, one per line. You must use the following rules:

- If `cron.allow` exists, only the users listed in this file can create, edit, display, or remove `crontab` files.
- If `cron.allow` does not exist, all users, except for users listed in `cron.deny`, can create, edit, display, or remove `crontab` files.
- If neither file exists, `root` privileges are required to run `crontab`.

Removing a crontab File

The correct way to remove a `crontab` file is to invoke the command:

```
# crontab -r username
```

- Regular users can remove only their own `crontab` file; however, `root` can delete any user's `crontab` file.



Caution – If the `crontab` command is accidentally entered 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 will overwrite the existing `crontab` file with an empty file.

The at Command

The `at` command is used to automatically execute a job at a specified time just once.

Command Format

```
at [-m] [-r job] [-q queueName] [-t time] [date]
```

Options

The options that can be used to instruct `crontab` on how to execute an `at` job include:

<code>-t time</code>	Specifies a time for the command to execute. Includes the following formats: h, hh, hh:mm now noon midnight A 24-hour clock is assumed unless you use am/AM or pm/PM on the command line.
<code>date</code>	Specifies a date for the command to execute. Includes formats, such as: month followed by a day number, (e.g. Jun 6) name of a day, (e.g. Friday) today tomorrow
<code>-m</code>	Sends mail to the user after the job has finished. This is the default for <code>root</code> .
<code>-r</code>	Removes a scheduled <code>at</code> job from the queue.
<code>-q queueName</code>	Specify a specific queue.

Executing the at Command

To create an at job to run at a specified time to locate and delete core files:

```
# at 8:45 pm
at>find /export/home/user2 -name core -exec rm {} \;
at><Press Control-d here>
commands will be executed using /bin/ksh
job 891550468.a at Thu Apr  2 14:45:00 2000
```

To display information about execution times of jobs:

```
# at -l [ job_id ]
897543900.a Thu Apr  2 14:45:00 2000
```

To display the jobs queued to run at specified times by ranking order:

```
# atq
```

Rank	Execution Date	Owner	Job	Queue	JobName
1st	Apr 2, 2000 14:45	user2	891550468.a	a	stdin

To remove a job from the at queue:

```
# at -r 891550468.a
```

To view all the at jobs currently scheduled in the queue:

```
# ls -l /var/spool/cron/atjobs
-r-S----- 1 user2 staff 634 Apr 2 14:45 891550468.a
-r-S----- 1 user1 staff 321 Apr 2 21:02 952725600.a
```

Denying at Access

By default, the Solaris Operating Environment includes the file `/etc/cron.d/at.deny`. This file identifies users who are prohibited from using the at command. The file format is one user name per line.

A user who is denied access to `at` receives the following message when attempting to use this command:

```
at: you are not authorized to use at. Sorry.
```

If the `/etc/cron.d/at.deny` file exists, but is empty, then all logged in users can access the `at` command.

Allowing at Access

As `root`, you can create the file `/etc/cron.d/at.allow` to list the names of users who are permitted to use the `at` command. When this file exists, it is read before the `/etc/cron.d/at.deny` file. If a user name exists in both files then that user will be denied access to the `at` command.

When neither the `at.deny` or the `at.allow` files exists, only `root` can use this command.

Exercise: Process Control



Exercise objective – In this lab you will use the Process Manager and `prstat` to monitor and kill processes. You will 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.

Task Summary

- Start the Process Manager. Run `prstat` in a window. In a separate window run the command `find /`. Make note of the CPU percentages for `find` displayed by `prstat` and the Process Manager. Open a third window and identify the PID of the shell running in it. Use the Process Manager to show the ancestry of the shell process. Use Process Manager to kill the shell process. Use the Process Manager to send the `TERM` signal to the `prstat` process. Exit the Process Manager when 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 5 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 `crontab` to determine when the logchecker process is scheduled to run. Use the `crontab` command to edit the `crontab` file for the user `root`. Add an entry that will send the message "It works!" to your current window 5 minutes from the current time.

Tasks

1. Login as `root` and open a terminal window. Start the Process Manager either by selecting **Find Process** from the **Tools** menu in CDE, or by using the following command:

```
# /usr/dt/bin/sdtprocess &
```

In the Process Manager display, sort the listing according to CPU%.

2. Open a second terminal window and run `prstat`.

```
# prstat
```

3. Position the Process Manager and the window where `prstat` is running so you can observe both simultaneously. In an available window, run the `find` command to list all files on your system. Observe how the Process Manager and `prstat` display statistics for `find`.

```
# find /
```

What is the maximum percentage of recent CPU time used by `find` as it executes?

4. Open a third terminal window and run `ps` to determine the PID of the shell. Record the PID you find.

```
# ps
```

5. In the Process Manager, locate and select the shell process you identified in the previous step. Select **Show Ancestry** from the **Process** menu in the Process Manager. 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 Manager, select **Kill**. What happens?

7. In the Process Manager, use the Find function to locate the `prstat` process. Select **Signal** from the **Process** menu. In the **Signal** fill-in field, enter the **TERM** signal and click on **OK**. What happens to the `prstat` process? Close the Process Manager when finished.
-

8. Identify the device associated with your current terminal and display the current time of day.

```
# tty
# date
```

9. Submit an `at` job that echoes "Test Complete" to your current window. Have the job run 5 minutes from the current time, and submit it to the queue called "x". For example:

```
# at -q x 13:30
at> echo "Test Complete" > /dev/pts/6
at> <Ctrl> d
commands will be executed using /sbin/sh
job 958163400.x at Fri May 12 13:30:00 2000
#
```

10. Display the `at` job in the queue.

```
# atq
```

11. Set and export the `EDITOR` environment variable in order to use `vi` to edit `crontab` files.

If you are using the Bourne or Korn shell, enter the following:

```
# EDITOR=vi
# export EDITOR
```

If you are using the C shell, enter the following:

```
# setenv EDITOR vi
```

12. Use the `crontab` command to view the current `crontab` file for the user `root`.

```
# crontab -l
```

13. When is the `logchecker` process scheduled to run?
-

14. Use the `crontab` command to edit the crontab file for the user `root`. Add an entry that will send the message “It works!” to your current window 5 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:

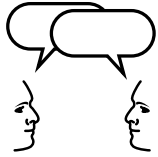
```
# tty
/dev/pts/#
# date
Thu May 11 10:25:14 PDT 2000
# crontab -e
```

Add the following line, substituting 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 5 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 exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

3. Position the Process Manager and the window where `prstat` is running so you can observe both simultaneously. In an available window, run the `find` command to list all files on your system. Observe how the Process Manager and `prstat` display statistics for `find`.

What is the maximum percentage of recent CPU time used by `find` as it executes?

This varies according to your system configuration. Some systems may display values in the 20% range.

5. In the Process Manager, locate and select the shell process you identified in the previous step. Select **Show Ancestry** from the **Process** menu in the Process Manager. What is the name and PID of the first process listed?

The PID will vary. On systems running 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 Manager, select **Kill**. What happens?

The process stops and the window no longer displays.

7. In the Process Manager, use the **Find** function to locate the `prstat` process. Select **Signal** from the **Process** menu. In the **Signal** fill-in field, enter the **TERM** signal and click on **OK**. What happens to the `prstat` process? Close the Process Manager when finished.

The `prstat` process terminates and the prompt displays in the window in which it ran.

13. When is the `logchecker` process scheduled to run?

Ten minutes after 3 AM, Sundays and Thursdays.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Start the CDE Process Manager to monitor and control active processes
- ☐ Report active process statistics using the `prstat` command
- ☐ Schedule the automatic execution of commands, programs, or scripts using the commands `at` and `crontab`
- ☐ Define the files used to control user access to the commands `at` and `crontab`
- ☐ Create and execute an `at` job
- ☐ Describe the location and format of a `crontab` file
- ☐ Demonstrate the steps to create, view, edit, and remove a `crontab` file

Objectives

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

- Describe the basic functions of the Solaris Operating Environment LP print service
- Define the important LP print service directories, files, and daemons
- Describe the function of a print server and a print client
- Define the terms local printer, network printer, and remote printer
- Use the Solaris Operating Environment 8 Print Manager to configure a network printer
- List the resources used by the print service to locate the destination printer
- Discuss the differences between the local printing process and a remote printing process
- Use the print service administration commands: `accept`, `reject`, `enable`, `disable`, and `lpmove`
- Configure the LP print services from the command line using `lpadmin`

Additional Resources



Additional resources – The following references can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume II*, Part Number 805-7229-10
- *Solaris 8 System Administration Guide, Volume III*, Part Number 806-0916-10

Solaris Operating Environment LP Print Service

The Solaris Operating Environment 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 continuing to work on other tasks.

Print Management Tools

The LP print service software contains the following three components for setting up and administering printers in the Solaris Operating Environment.

- Solaris Operating Environment Print Manager – A graphical user interface that provides the ability to configure and manage printers.

Note – Solaris Print Manager is new to Solaris 8 Operating Environment and is preferred over `admintool` as the method for installing and modifying printers and adding access to remote printers.

- `admintool` – A graphical user interface that is used to set up and manage printers on a local system.
- LP print service commands – A command-line interface that is used to configure and manage printers. These commands also provide functionality not available in the other print management tools.

Client-Server Model

The Solaris Operating Environment print service is implemented in a client-server model.

- A *print server* – Any system configured to manage a printer directly connected to it, or that is attached to the network, the print server makes the printer(s) available to other systems on the network.
- A *print client* – A system that sends print requests to a print server.

Types of Printer Configurations

As a system administrator, it is important to set up printers so that users have access to one or more printers.

You should distribute printers 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.

You can set up and access the following types of printer configurations in the Solaris Operating Environment:

- Local printer – A local printer is physically connected to the system, and is accessed from that system.
- Network printer – A network printer is physically attached to the network and has its own hostname and IP address. A network printer provides print services to clients without being directly connected to a print server.
- Remote printer – A remote printer is one that users access over the network; that is either a printer physically connected to a remote system or physically attached to the network.

Figure 11-1 illustrates the concept of local, remote, and network printers.

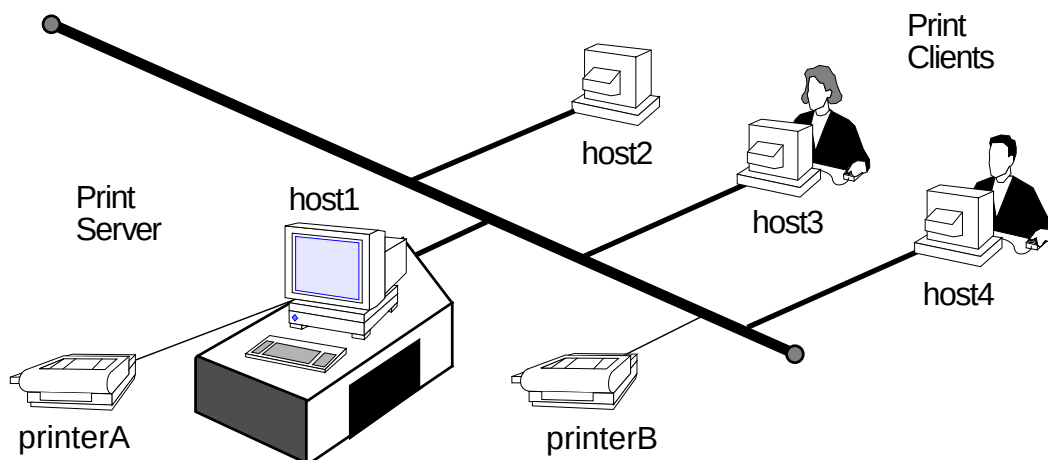


Figure 11-1 Local, Remote, and Network Printers

In Figure 11-1, the printer named `printerA` connected to the system named `host1` is a local printer for any user logged in on that system.

The printer named `printerB` is a network printer controlled by the print server, `host1`. This is a network printer for any users logged in on `host1`.

For users who are logged in on `host2`, `host3`, or `host4`, both `printerA` and `printerB` are accessed as remote printers.

LP Print Service Functions

Some basic functions of the Solaris LP print service include:

Initialization

The print service initializes a printer prior to sending it a print request to ensure the printer is in a known state.

Queuing

When print requests are spooled, the requests are scheduled with other print requests waiting to be sent to the printer. This process is called *queuing*.

Tracking

The print service tracks the status of every print request to enable `root` to manage all the requests, and for regular users to be able to view or cancel their own requests. It also logs any errors that may have occurred during the printing process.

Fault Notification

If a problem does occur in the print service, an error message is displayed on the console or emailed to the user.

Configuring Printer Services

Configuring printer services in the Solaris Operating Environment involves the following main tasks.

- Setting up the printer – Physically connect the printer to a system or the network.
- Setting up the print server – Configure the system that is to manage and provide access to the printer.
- Setting up the print client – Configure the system to access a remote printer.
- Verifying printer access – Check 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) each print client's host name and IP address must be entered in the `/etc/inet/hosts` file on the print server when setting up the printer services.

Print Server Requirements

Any system on the network can be a print server, if it has the resources to manage the printing load; such as spooling space and memory.

Spooling Space

The spooling space is the amount of disk space used to store and process print requests. Spooling space is the most important factor to consider when designating systems to be print servers. The recommended starting size for spooling space is from 25 to 500 Mbytes, depending on the type and the size of files being printed, and the number of users.

Note – The term `spool` is an acronym for system peripheral operation offline.

Memory

The Solaris Operating Environment itself requires 64 Mbytes of memory to run on a system. Print servers do not require additional memory, however an extra 32 Mbytes of memory can improve performance when filtering print requests.

The Solaris 8 Print Manager

The Solaris 8 Print Manager enables you to set up and manage printers.

The Solaris Print Manager is the preferred method for managing printers. It centralizes printer information when used in conjunction with a name service, such as Network Information Service (NIS), which eases printer administration.

Note – Solaris Print Manager recognizes existing printer information on print servers, print clients and in the name service databases.

Starting the Solaris Print Manager

As root, start the Solaris Print Manager with the following command:

```
# /usr/sadm/admin/bin/printmgr &
```

You can also start the Solaris Print Manager by selecting the Printer Administrator from the Tools option on the CDE Workspace menu, and entering the host name of the workstation to continue.

Using either method displays the Solaris Print Manager main window with the Select Naming Service window overlaid on top.



Figure 11-2 Select Naming Service window

1. Click on OK to select the default, (files). The Print Manager main window remains on the screen.

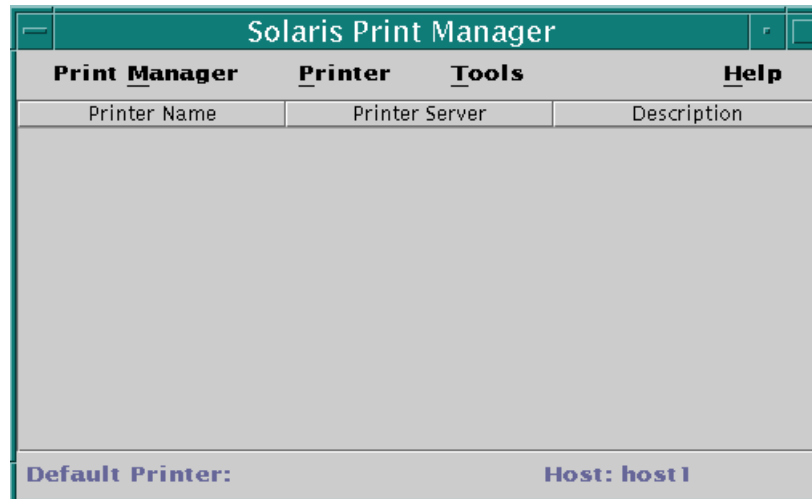


Figure 11-3 Solaris Print Manager window

2. Click on the Printer menu in this window to view the possible menu selections.

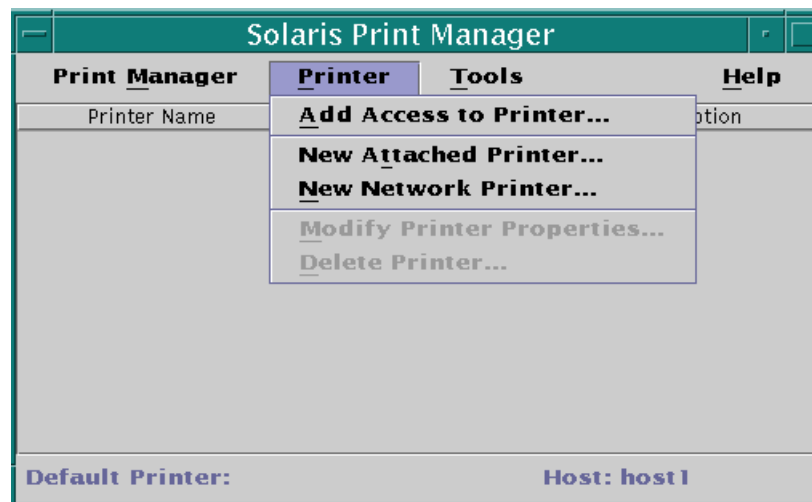


Figure 11-4 Solaris Print Manager Printer Menu

From these choices you can choose to:

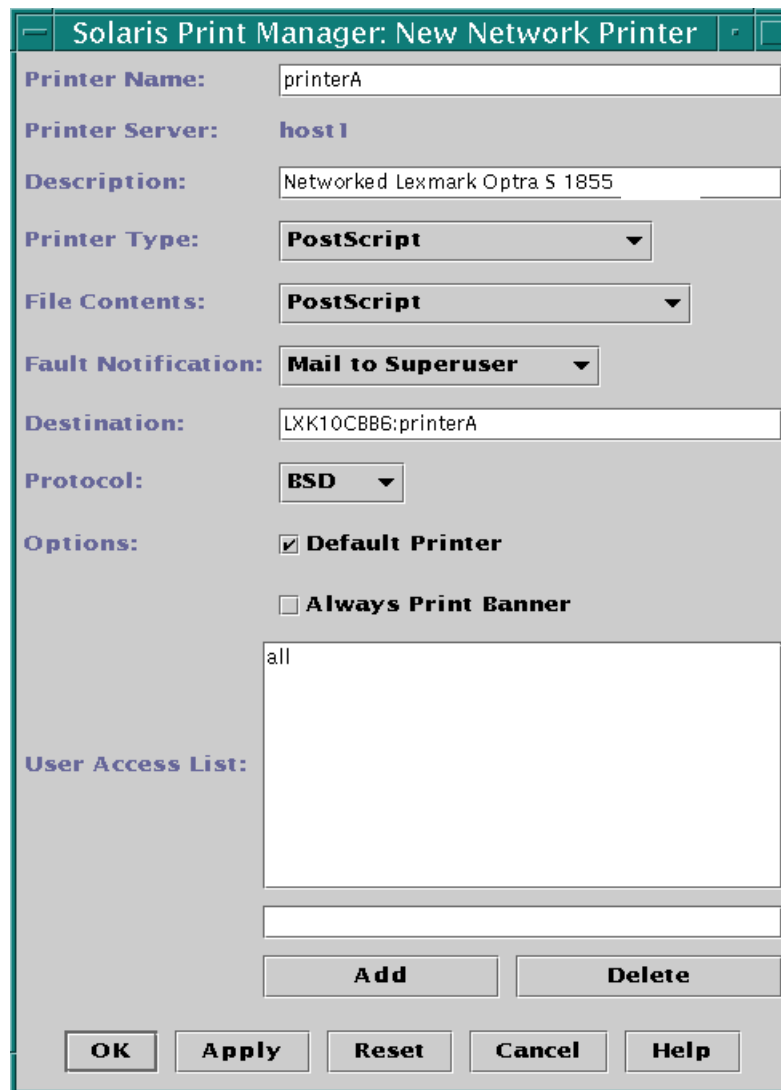
- Add Access to Printer – This is selected from a print client to set up access to printers that are physically connected to a print server, or directly attached to the network. The host name and IP address of the print server must be in the print client's `/etc/inet/hosts` file, or in a name service database, (for example, NIS).

- New Attached Printer – This is selected from a print server to configure a printer that is physically connected to it.
- New Network Printer – This is 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

From the print server, the following procedure sets up the configuration information to provide access to a new network printer.

1. From the Printer menu, select the New Network Printer option. The Solaris Print Manager: New Network Printer window is displayed.



The image shows a window titled "Solaris Print Manager: New Network Printer". It contains several fields and options for configuring a new network printer. The fields are: "Printer Name:" with the value "printerA", "Printer Server:" with the value "host1", "Description:" with the value "Networked Lexmark Optra S 1855", "Printer Type:" with a dropdown menu showing "PostScript", "File Contents:" with a dropdown menu showing "PostScript", "Fault Notification:" with a dropdown menu showing "Mail to Superuser", "Destination:" with the value "LXK10CBB6:printerA", and "Protocol:" with a dropdown menu showing "BSD". There are two checkboxes under "Options": "Default Printer" (checked) and "Always Print Banner" (unchecked). Below the checkboxes is a text area labeled "User Access List:" containing the text "all". At the bottom of the window are buttons for "Add", "Delete", "OK", "Apply", "Reset", "Cancel", and "Help".

Figure 11-5 New Network Printer Window

The information required to configure the new network printer includes:

- Printer Name – A unique name for the network printer. The name can contain a maximum of 14 alphanumeric characters, including dashes and underscores. This is the name entered in the `/etc/hosts` file, and on the command line when using print commands.
- 2. In the Printer Name field, type in the new printer name, for example: `printerA`

- Printer Server – Defaults to the name of the system you are currently logged in on and running the Solaris 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).
3. Click on the Description field and type in a printer description of your choice.
- Printer Type – The generic name for the type of printer, (e.g. PostScript, HP Printer, Diablo). The LP print service identifies each printer by its printer type which is held in the directory `/usr/share/lib/terminfo`. The Other option located at the end of the list allows for the selection of any other printer type listed in the `terminfo` database.
4. Accept the default Printer Type: PostScript

The LP print service uses information in the `terminfo` database to initialize the printer, as well as to communicate the sequence of codes to the printer. To view the contents of the `terminfo` directory, type the following command:

```
# ls /usr/share/lib/terminfo
1 3 5 7 9 B H P a c e g i k m o q s u w y
2 4 6 8 A G M S b d f h j l n p r t v x z
```

The `terminfo` directory contains many different subdirectories named with a letter or digit. The same initial letter or digit the manufacturer has assigned to the printer's generic name, (including terminals and modems).

For example, the printer type for a particular Epson printer would be located in the subdirectory `/usr/share/lib/terminfo/e`.

```
# ls /usr/share/lib/terminfo/e
emots          ep2500+high    ergo4000       exidy2500
env230         ep2500+low     epon2500       esprit
envision23     ep40           epon2500-80    ethernet
ep2500+basic   ep400          epon2500-hi    ex3000
ep2500+color   ep4080         epon2500-hi80
```

- File Content Type – Specifies the data format of files that can be printed without any special filtering by the LP print service software.

5. Accept the default File Content: PostScript

Every printer has configuration information pertaining to the content type of files that it can accept for its printer type. The LP print service depends on this configuration information to match the content type of each print request to the printer's printer type, which ensures the file is printed correctly.

By selecting a file content type, described in Table 11-1, it specifies the data format of the file that can be printed without any special filtering by the print software.

Table 11-1 Descriptions of File Content Types

File Content Type	Description
PostScript	PostScript files do not require filtering. This is the default.
ASCII	ASCII files do not require filtering.
Both PostScript and ASCII	PostScript and ASCII files do not require filtering.
None	All files require filtering, except those matching the printer's type.
Any	No filtering required. If printer cannot handle the file content type, the file will not be printed.

- 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.
- #### 6. Click on the Fault Notification button and select: Mail to Superuser
- 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 sub-system when 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.

7. Click on the Destination field and type in a Destination access name.

Should the network printer not be recognized by its name/IP address in the hosts table, you may 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 vendors documentation.

The format of a Destination entry can include one of the following:

Destination	Protocol
<i>printer_name</i>	BSD
<i>printer_name:printer</i>	BSD
<i>printer:printer_name</i>	BSD
<i>IP_ADDR</i>	BSD
<i>IP_ADDR:port_number</i>	TCP
<i>printer:port_number</i>	TCP

The port number is print server dependent. For example, Lexmark uses port 9100.

- Protocol –The internet protocol used to communicate with the printer for file transfer. The choices are BSD Printer Protocol and raw TCP. In general the TCP protocol is more generic across printers. The printer vendor documentation supplies the information regarding the protocol to select.

8. Leave this protocol set to BSD.
- Options – Identifies two options, which by default are disabled. To enable an option click in the appropriate box, (a check mark will appear).
 - ▼ Default Printer – If enabled, designates this printer as the default printer for print jobs from this system.
9. Click in the Default Printer box to enable this option.
 - ▼ Always Print Banner Page – If enabled, a banner page will always be printed between print jobs.
10. You can (optionally) click in the Always Print Banner box to enable this option.
- User Access List – Specifies print clients that can print to this printer. By default, the word `all` allows every print client access to this printer.
11. Accept the default, `all`.

To restrict user access to this printer, you can enter the following values in the text field below the User Access List window:

- ▼ `user-name` – Only the specified user can access the printer. For example: `user1`
- ▼ `system-name!login-ID` – Only the specified user from the named host can access the printer. For example: `host2!user4`
- ▼ `system-name!all` – All users from the named host only can access the printer. For example: `host5!all`
- ▼ `all!login-ID` – Only the specified user from all hosts can access the printer. For example: `all!user1`

Note – To delete an entry from the User Access List, select the entry and click Delete.

12. To accept the new network printer's configuration information, click on OK.

The Solaris Print Manager window, displaying the newly configured printer, remains on the Desktop.

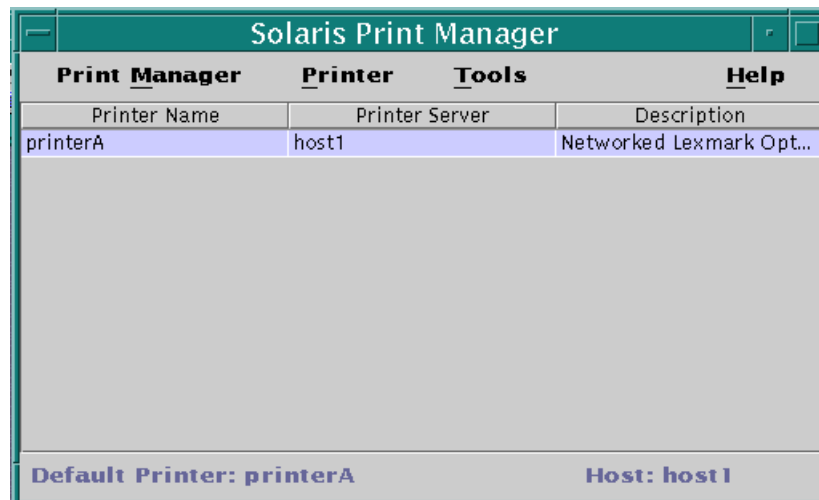


Figure 11-6 Configured Printer

13. To close the Solaris Print Manager window, select **Exit** from the **Print Manager** menu.

Printing the Solaris Operating Environment

Users submit print requests from print clients using the `lp` or `lpr` command.

Note – The Solaris Print Service accepts both the SVID (System V Interface Definition) `/bin/lp` command and the BSD `/usr/ucb/lpr` command to submit print requests.

These commands are used to print ASCII text files. They are not used to print documents created in applications (for example, FrameMaker).

The function of the `lp` or `lpr` commands is to queue print requests for printing on a destination printer.

Examples of Using the Print Command

```
$ /bin/lp filename
```

or

```
$ /usr/ucb/lpr filename
```

These two commands are the simplest methods for submitting a print request.

Examples of Specifying a Destination Printer

To specify a destination printer for a print request you can use one of the following styles:

- Atomic Style
- Portable Open Systems Interface (POSIX) Style

Submitting a Print Request Atomic Style

Submitting a print request using the atomic style includes the print command and an option, followed by a printer name. For example:

```
$ /bin/lp -d printerB filename
```

```
$ /usr/ucb/lpr -P printerB filename
```

Either of these commands submit a print request to a destination printer called printerB.

Submitting a Print Request POSIX Style

Submitting a print request using the POSIX style, includes the print command and an option, followed by the *print_servername:printername*, as shown in the following example:

```
$ /bin/lp -d host1:printerA filename
```

```
$ /usr/ucb/lpr -P host1:printerA filename
```

Either of these commands submit a print request to a destination printer called printerA managed by the print server host1.

Locating the Destination Printer

The Solaris LP print service checks the following resources to locate the destination printer from the client-side.

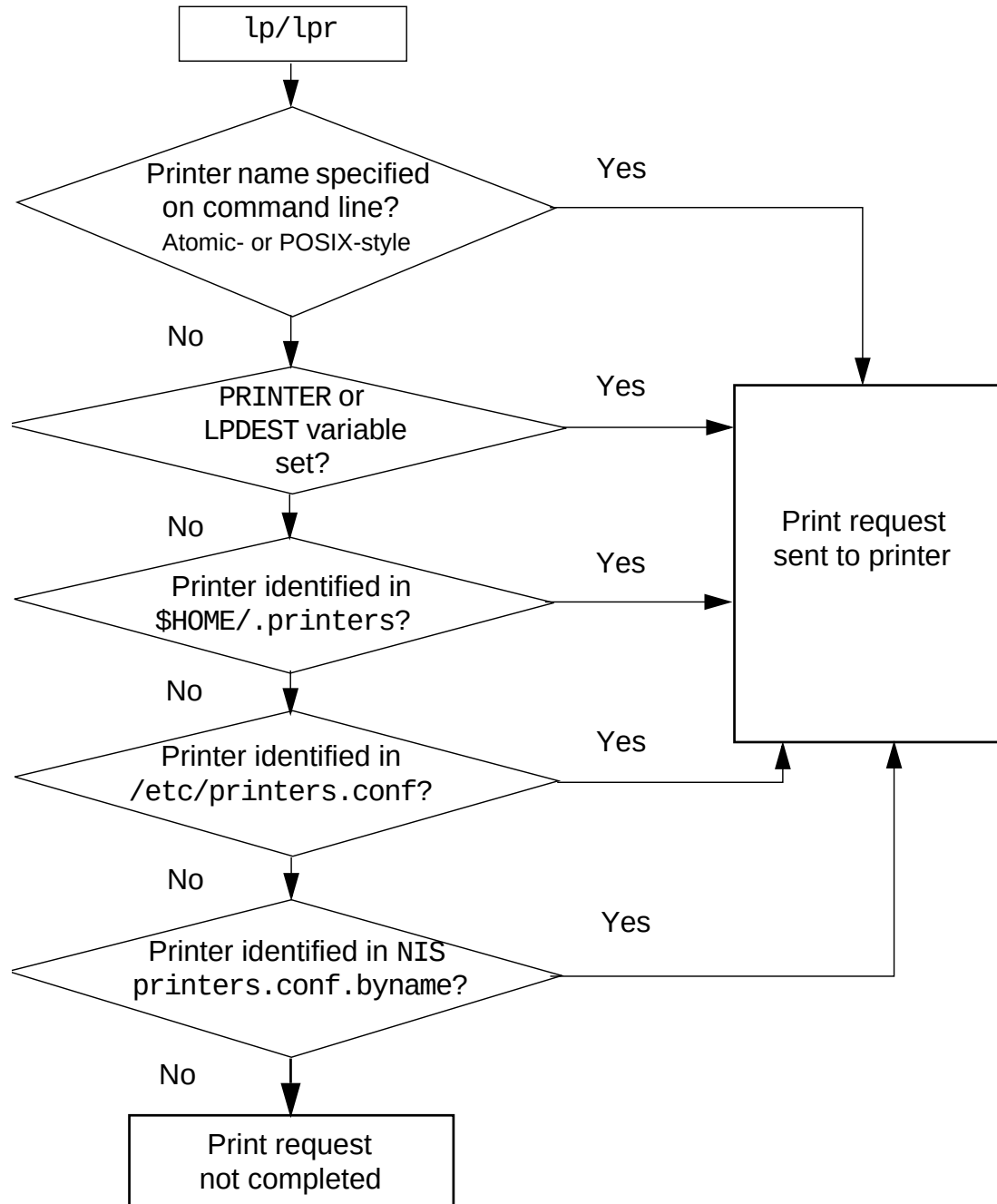


Figure 11-7 Locating the Destination Printer

If the command-line does not specify a named printer destination, the user's \$HOME environment is checked:

- The LPDEST or PRINTER environment variables can be set 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 has been set to specify a named printer destination, then the variable named `_default` is checked for in the following files:

- `$HOME/.printers`

Users can create their own `.printers` file in their home directory to set the default printer name.

`_default printer-name`

If the `$HOME/.printers` file does not exist, or does not specify a printer name destination, the `/etc/printers.conf` file is checked.

- `/etc/printers.conf`

For example, if the print server is named `host1` and the printer is named `printerA`, the entry in this file would appear as:

```
_default | lp:
      :use=host1:
      :bsdaddr=host1,printerA
```

If this `_default` variable has not been set, then the `_default` variable in the name service database (e.g. NIS) is checked.

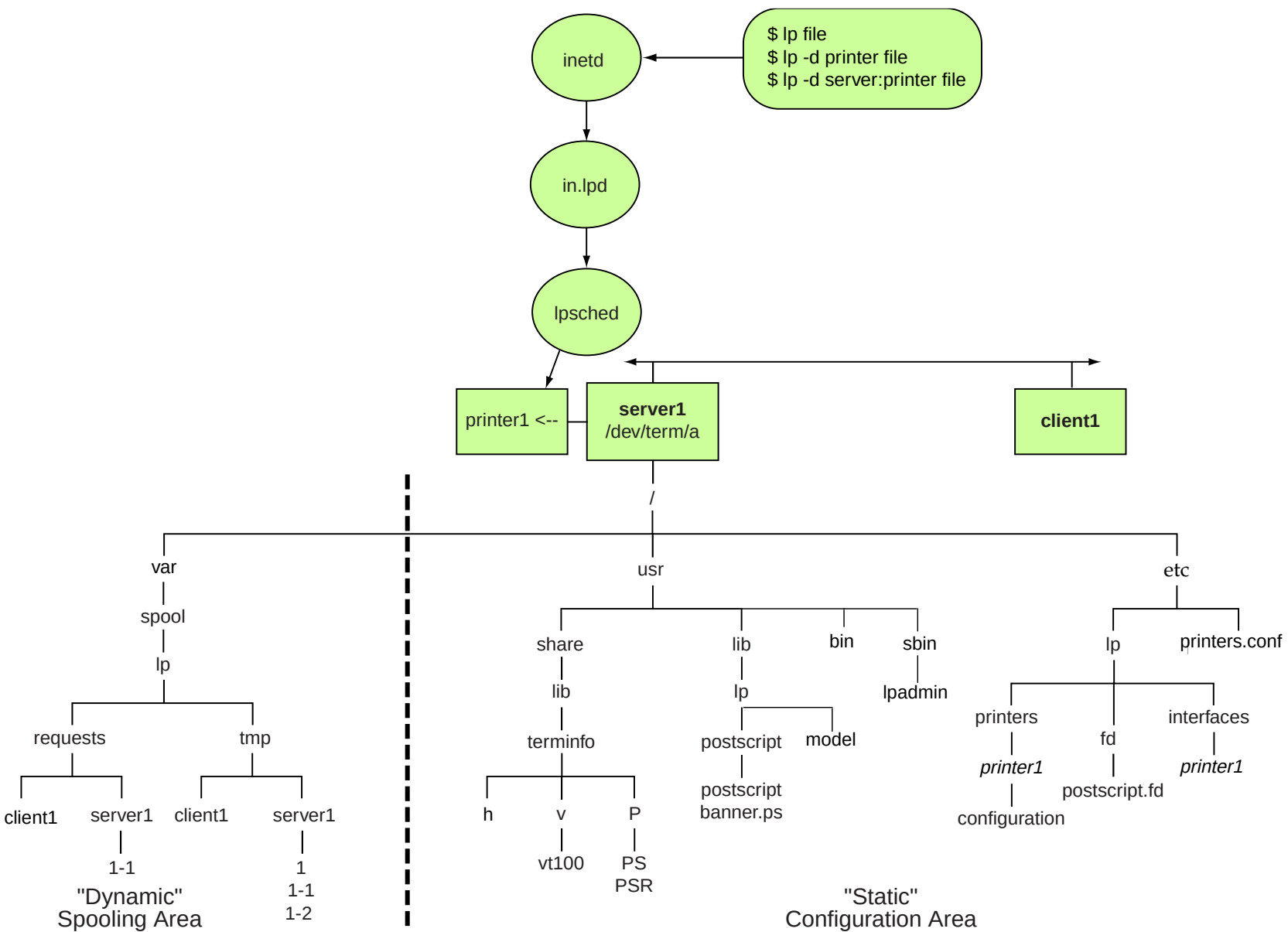
- `printers.conf.byname`

In this case, the `_default` variable entry in the name service map called `printers.conf.byname` would define 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.

The LP Print Service Directory Structure



LP Print Service Directories

The Solaris LP print service includes a directory structure, files and logs. The following sections describe some of the more important components of this structure.

The /usr/bin Directory

This directory contains the LP print service user commands, such as `lp`, `lpstat`, and `cancel`.

The /usr/sbin Directory

This directory contains the LP print service administrative commands. For example: `lpadmin`, `lpusers`, and `lpshut`.

The /usr/share/lib/terminfo Directory

This directory contains the `terminfo` database directory, which describes the capabilities of devices, such as printers and terminals.

The /usr/lib/lp Directory

This directory contains the `lpsched` daemon; binary files used by the LP print service; PostScript filters; and default printer interface programs. Two important directories include: `model` and `postscript`.

The /usr/lib/lp/model Directory

There are two default printer interface programs, (shell scripts) located in the `model` directory, called `standard` and `netstandard`.

The `standard` script is designed to support local printers. For example, when a print request is queued for printing, the print service runs the printer's interface program 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 specified by the user's print request.

The `netstandard` script is specifically designed to support network printers. It collects the spooler and print database information needed to perform network printing and passes it to a print output module. This module, `netpr` opens the network connection to the printer and sends the data to the printer.

The `root` user can modify any printer's interface script. For example, to turn off the printing of a banner page, edit `/etc/lp/interfaces/printer_name` on the print server and 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 LP print service.

Note – Print filters are programs on the print server that convert the content type of a queued print request from one format to another format accepted by the destination printer.

The Solaris LP print service provides a set of PostScript print filters in this directory to cover most situations where the printer requires the content of files to be in PostScript format.

These filters come with descriptor files in the `/etc/lp/fd` directory that tell the LP Print service the characteristics of the filters and where to locate them.

The /etc/lp Directory

This directory contains a hierarchy of LP server configuration directories and files. The `lp sched` daemon administers and updates the files located in this directory.

The contents of these configuration file can be viewed, however you should not edit these files directly. To make configuration changes, use the `lpadmin` command.

There are three subdirectories in `/etc/lp` which are important to a printer configuration. These include: `fd`, `interfaces`, and `printers`.

The /etc/lp/fd Directory

This 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 – A filter lookup table is kept in the `/etc/lp/filter.table` file.

The /etc/lp/interfaces Directory

This directory contains each printers interface program file. When a printer is configured the print service places a copy of the appropriate default `/usr/lib/lp/model` interface script in the directory `/etc/lp/interfaces/printer-name`, where *printer-name* is the directory created for the newly configured printer's own interface script.

The /etc/lp/printers Directory

This directory contains a subdirectory for each local printer known to 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: Always
Content types: PS
Device: /dev/term/a
Interface: /usr/lib/lp/model/standard
Printer type: PS

The /var/spool/lp Directory

This directory contains a list of current requests that are in the print queue.

The lpshd daemon for each system keeps a log of print requests in the directories:

- */var/spool/lp/tmp/system-name*
- */var/spool/lp/requests/system-name*

Each print request has two files, (one in each of the directories), that contains information about the print request.

The information in */var/spool/lp/requests/system-name* directory can be accessed only by root or lp.

The information in */var/spool/lp/tmp/system-name* directory can be accessed only by the user who submitted the request.

These files remain in their directories only as long as the print request is in the queue. Once the request is finished, the information in the files is combined and appended to */var/lp/logs/requests* file.

The /var/lp/logs Directory

This directory contains an ongoing history of print requests. The log file */var/lp/logs/requests* contains information about print requests that are completed and no longer in the print queue.

LP Print Service Daemons

The LP print service daemons and their responsibilities are described below.

The Internet Service Daemon/`usr/sbin/inetd`

The Internet services daemon, `inetd` is the server process for the Internet standard services. It is usually started up at system boot time. It listens for service requests on the ports associated with each of the services listed in its configuration file `/etc/inetd.conf`. When a request arrives, `inetd` executes the server program associated with the service.

The `/usr/lib/print/in.lpd` Program

The `in.lpd` program is started from `inetd`. It implements the network listening service for the print protocol. The print protocol provides a remote interface for systems to interact with a local spooling system. This protocol defines standard requests from the print client to the print server: starting queue processing, transferring print jobs, retrieving status, and canceling print jobs.

On receipt of a connect request, `in.lpd` is started to service the connection. Once the request has been serviced, `in.lpd` closes the connection and exits.

The `/usr/lib/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, and manages requests issued to the system by the `lp` commands.

The `lpsched` daemon schedules all 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, `lpsched` schedules the next request, if there is one in the queue on the print server.

Each print server must have only one `lpsched` daemon running. It is started when the system is booted, (or enters run level 2), by the control script `/etc/rc2.d/S80lp`.

The /usr/lib/saf/listen Daemon

In the Solaris 2.0-2.5.1 Operating Environments, the network listener daemon listens to a network for service requests, accepts requests when they arrive, and invokes print servers in response to the requested services.

The network listener process is no longer used in the LP print service software released with the Solaris 2.6 Operating Environment and later versions.

The lpNet Daemon

In the Solaris 2.0-2.5.1 Operating Environments, each print client and each print server must have at least one `lpNet` daemon. This daemon schedules network print requests. It needs a `listen` service to handle incoming network requests on each print server.

The `lpNet` daemon is started when a system is booted. When you stop and restart `lpsched` using the `lpshut` and `lpsched` commands, the `lpNet` daemon is also stopped and restarted.

The Solaris 2.6 Operating Environment does not use the `lpNet` daemon to schedule network requests. Instead, network scheduling is handled by the `inetd` Internet services daemon, which listens for requests.

The Solaris Operating Environment Printing Process

The following sections describe the Solaris Operating Environment printing process.

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 print scheduler, `lpsched`.

The print scheduler matches the printer type and identifies the default printer for the system, it then filters the job.

The `lpsched` daemon keeps a log of print requests in the directories:

- `/var/spool/lp/requests/system_name`
- `/var/spool/lp/tmp/system_name`

If the printer is free, `lpsched` starts the printer's interface program. The interface program initializes the printer port, initializes the actual printer, prints the banner page, prints the correct number of file copies, and catches any faults.

Figure 11-1 illustrates the local printing process.

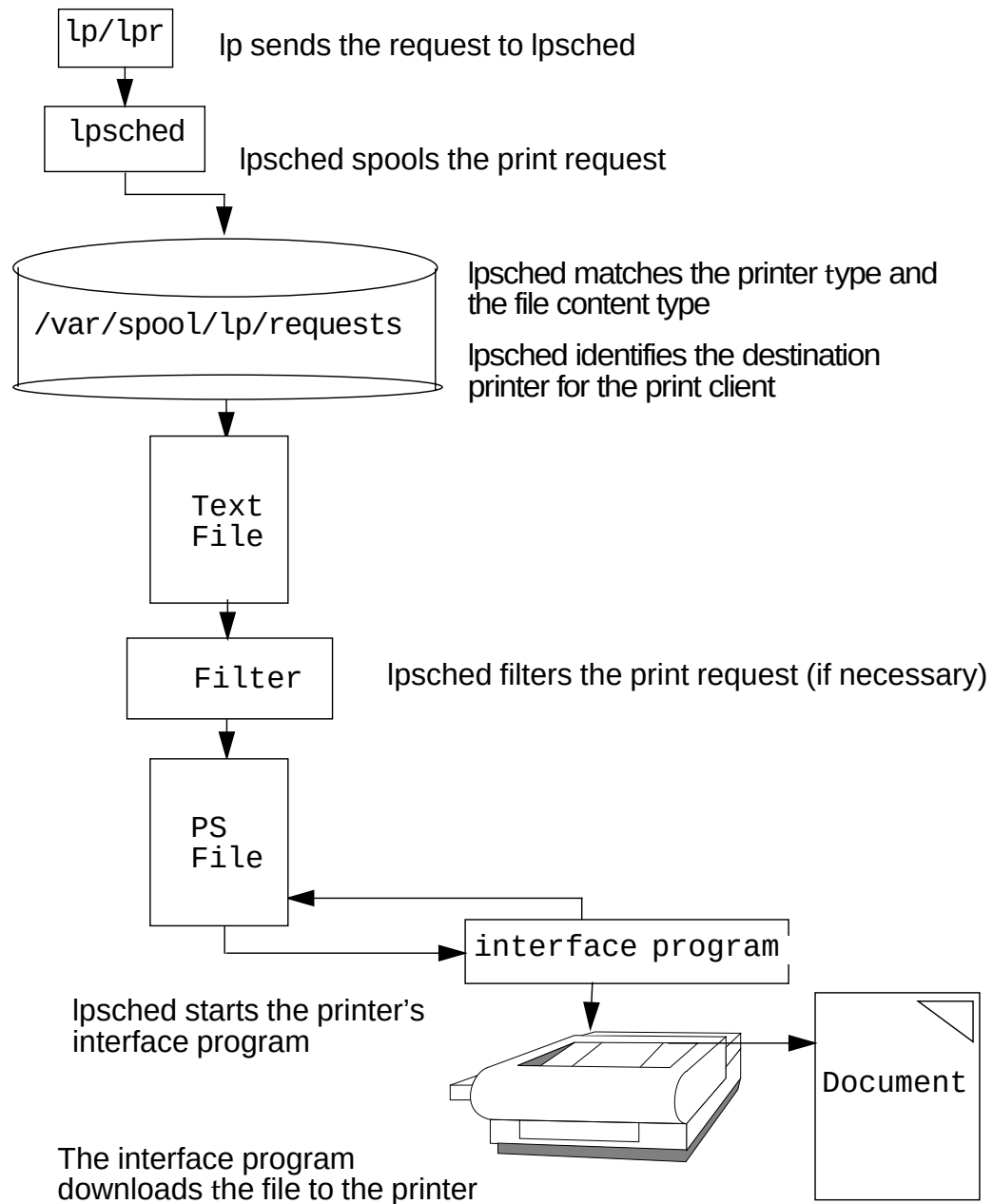


Figure 11-8 Local Printing Process

The 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 it to the destination printer to be printed.

Remote Printing in a Solaris 2.6 to Solaris 8 Operating Environment

Figure 11-9 illustrate a remote print request being submitted from a print client to a print server in a Solaris 2.6 to Solaris 8 Operating Environment.

The client's print command communicates directly with the print service on the server to transfer a print request to the printer.

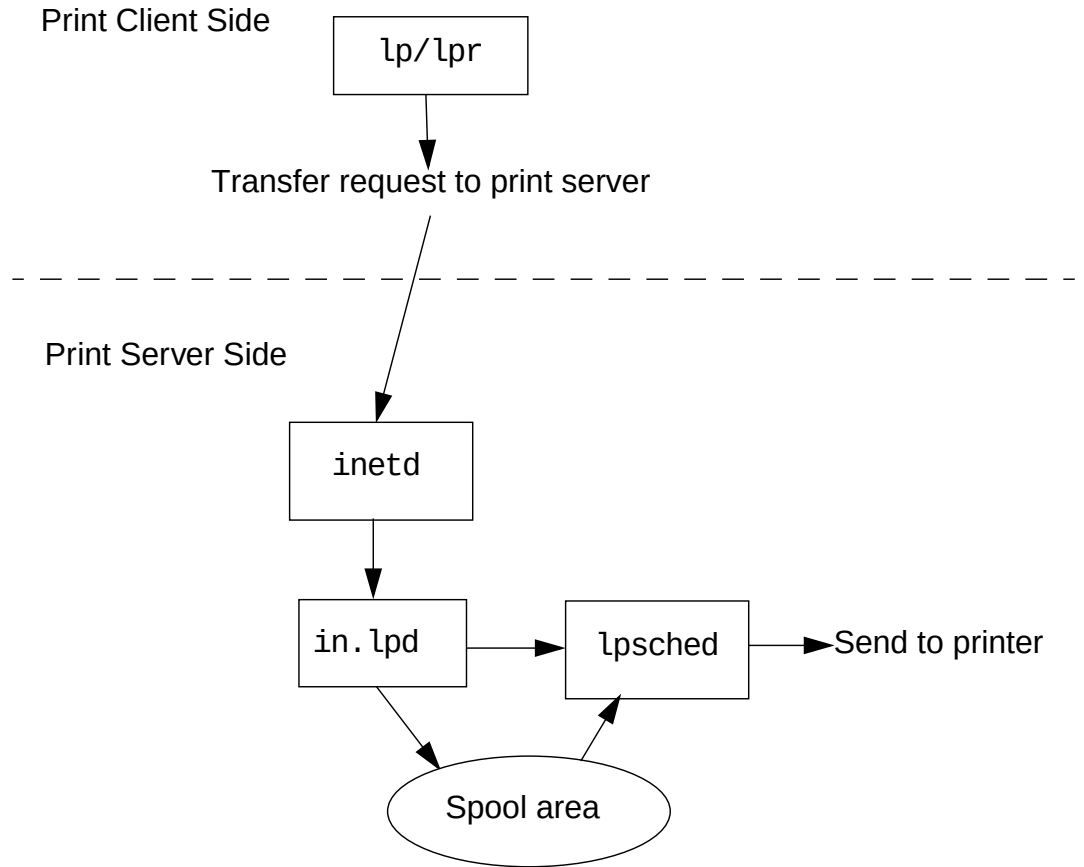


Figure 11-9 Solaris 2.6 to Solaris 8: Remote Printing

The print server listens for print requests with the Internet services daemon `inetd`. When `inetd` hears a request for a print service on the network, it starts a program called the protocol adapter, `in.lpd`.

The protocol adapter translates the print request, communicates it to the print spooler, and returns the results to the print requester.

The `in.lpd` contacts `lpsched` to start the printer's interface program and transfer the print request to the destination printer. Then `in.lpd` starts on demand and exits when the network request has been completed.

Remote Printing in a Solaris 2.0 to Solaris 2.5.1 Environment

Figure 11-10 illustrate a remote print request being submitted from a print client to a print server in a Solaris 2.0 to Solaris 2.5.1 operating environment.

The client's print command contacts `lp sched` and places the print request into the local spooling area.

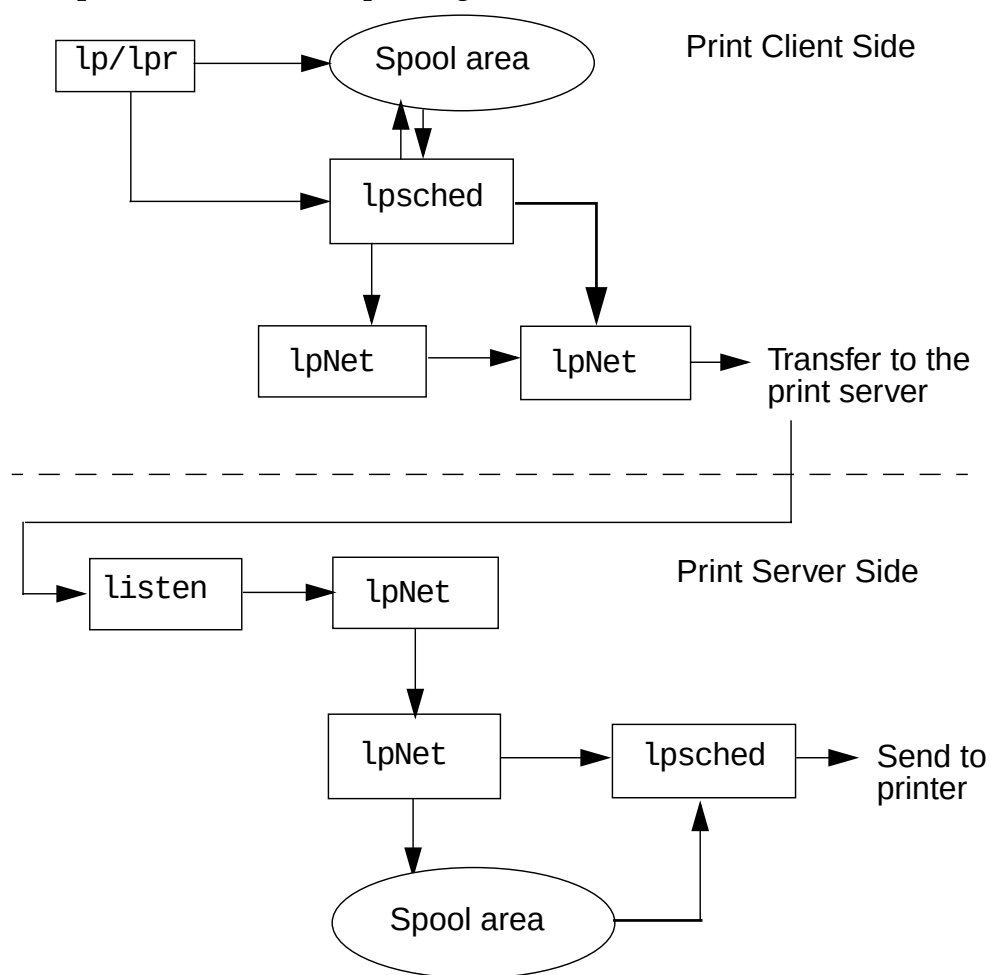


Figure 11-10 Solaris 2.0–2.5.1 Remote Printing

When `lp sched` is contacted, it contacts `lpNet` which forks a child process that transfers the print request to the print server.

On the print server, the Service Access Facility's `listen` daemon listens for network print requests. Print requests are passed to `lpNet`. It forks an `lpNet` child process for each print request, which in turn contacts `lp sched` who processes the request and sends it to the printer.

LP Print Service Commands

Table 11-2 lists some of the more frequently used print service administration commands. You must be `root` to use these commands.

Table 11-2 LP Print Service Administration Commands

Command Name	Description
<code>accept</code>	Permits print requests to be queued for a specific printer
<code>reject</code>	Prevents print requests from being queued for a specific printer
<code>enable</code>	Activates a printer
<code>disable</code>	Deactivates one or more printers
<code>lpmove</code>	Moves print requests from the destination printer to another printer.
<code>lpadmin</code>	Sets up, changes, or removes printer configurations

The accept and reject Commands

The commands `accept` and `reject` are used by `root` on the print server to permit print requests be queued for a specific printer, or to prevent print requests from being queued to a specific printer.

Using the accept Command to Allow Queuing

You use the `accept` command to allow queueing of print requests for a named destination printer. This means that users can submit print requests into the printer queue for processing.

```
# accept printer-name
```

For example:

```
# accept printerD
```

Using the reject Command to Prevent Queuing

You use the `reject` command to prevent queueing of print requests for the named destination printer. This means that users cannot submit print requests to the printer queue.

```
# reject [-r "reason" ] printer-name
```

The option `-r reason` is used for entering an explanation for the rejection of print requests for this printer.

For example:

```
# reject -r "Replacing Toner Cartridge" printerD
```

The enable and disable Commands

The commands `enable` and `disable` are used by `root` on the print server to activate a specific printer, or to deactivate one or more printers.

Using the enable Command to Activate a Printer

The `enable` command activates a printer, enabling it to print requests that have been submitted into the print queue.

```
# /usr/bin/enable printer-name
```

For example:

```
# enable printerD
printer "printerD" now enabled
```

Using the disable Command to Deactivate a Printer

The `disable` command deactivates printers, disabling them from printing user's print requests that are waiting in the print queue.

By default, any requests currently printing on the printer when the `disable` command is issued, will be reprinted in their entirety.

```
# /usr/bin/disable [-c | -w] [-r "reason"] printername
```

The following list describes the options for the `disable` command:

- `-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.

For example:

```
# disable -w -r "Printer down for maintenance" printerD
printer "printerD" now disabled
```


The `lpmove` Command

You use the `lpmove` command to move one or all print requests, from one printer to another printer.

1. Become `root` 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
```

3. Display the print queue to see how many print requests are to be moved. This step is needed to identify print request IDs if only selected print requests are going to be moved to another printer.

```
# lpstat -o
printerC-29    host7!user2    61426    Jun 07 10:30
printerC-30    host4!user1    9560     Jun 07 10:30
printerC-31    host7!user5    845      Jun 07 10:30
printerC-32    host7!user5    845      Jun 07 10:30
printerC-33    host7!user5    845      Jun 07 10:30
```

4. Verify that the destination printer is accepting print requests.

```
# lpstat -a printerA
printer printerA accepting requests since Wed May 8
```

5. To move all print requests from `printerC` over to `printerA`:

```
# lpmove printerC printerA
```

- a. To move one or more individual print requests from `printerC` to `printerA`:

```
# lpmove printerC-32 printerC-33 printerA
```

6. Once `printerC` is available again, use the `accept` command.

```
# accept printerC
destination "printerC" now accepting requests
```

Configuring the LP Print Service Using `lpadmin` Command

You can use the `lpadmin` command to configure the LP print services from the command line. For example:

- Defining printer devices and printer names
- Specifying interface programs (custom or standard) and printer options
- Defining printer types and file content types
- Creating printer classes
- Defining allow and deny user lists
- Specifying fault recovery
- Removing printers and printer classes

The `lpadmin` command is most commonly used by `root` for the purpose of:

- Creating printer classes
- Setting or changing a system's default printer destination
- Removing a printer's configuration from the LP print service

Creating Printer Classes

A printer class is a specific group of individual printers identified by a class name.

Once created, a printer class name is used on the command line as the destination for user's print requests. The LP print service automatically sends each print request to the first available printer within the class which matches the content type expected by the printer.

This is a useful feature for balancing the load of print requests among several printers.

A printer class can include:

- Specific printer types (for example, all PostScript printers)
- Printers in a specific location (for example, Building 2)
- Printers in a specific work group or department (for example, Marketing, Engineering, Accounting).

You can create a printer class using the `lpadmin` command only on the print server where the printers are configured. Printer classes cannot be defined on print clients.

Note – You cannot activate or deactivate a printer class with the `enable` and `disable` commands. You can activate or deactivate only the individual printers within a printer class.

Printer Priority Within a Class

When a printer class is created, root can control the printer access order by adding the printers to the class in a descending order. For example, by adding a high-speed printer to the printer class first, this enables it to handle as many print requests as possible, before off-loading to the printer that was added to the class next, and so on.

Creating a Printer Class

A class is created when the first printer is added to the printer class name. After a class is created, other printers can be added to it at any time.

To create a printer class called `bldg2`:

```
# lpadmin -p printerB -c bldg2
```

To add another printer to this class:

```
# lpadmin -p printerD -c bldg2
```

Once the system administrator has finished adding printers to the printer class the `accept` command is invoked to allow queueing of print requests to the new `bldg2` print queue.

```
# accept bldg2
destination "bldg2" now accepting requests
```

Use the `lpstat -t` command on the print server to check the status of the new printer class:

```
# lpstat -t
scheduler is running
system default destination: printerA
members of class bldg2:
printerB
printerD
device for printerB:
device for PrinterD:
bldg2 accepting requests since Wed Jun 07 15:27:10 MST 2000
printerB accepting requests since Wed Jun 07 15:27:10 MST 2000
printerD accepting requests since Wed Jun 07 15:27:10 MST 2000
```

To send a print request to a printer class:

```
# lp -d bldg2 myfile
request id is bldg2-0 (1 file)
```

Setting or Changing a System's Default Printer

The `root` user can run the `lpadmin` command to set or change an individual printer or a printer class to be the system's default destination for all print requests.

```
# lpadmin -d prntername
# lpadmin -d printer-classname
```

For example:

To set or change a system's default destination printer:

```
# lpadmin -d printerE
```

To verify that the default destination printer has been set or changed:

```
# lpstat -d
system default destination: printerE
```

The print request issued from this system is sent by default to `printerE`.

```
# lp myfile
```

To set or change a system's default destination printer class:

```
# lpadmin -d bldg2
# lpstat -d
system default destination: bldg2
# lp myfile
```

To remove an individual system's default destination printer or printer class, enter the following command, from that system:

```
# lpadmin -d
```

Manually Removing a Printer's Configuration

To manually remove a printer's configuration on the print client side:

1. Log in as `root` 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.

```
# lpadmin -x printer-name
```

-x — Deletes the specified printer.

For example:

```
# lpadmin -x printerD
```

Information for the specified printer is deleted from the print client's `/etc/lp/printers` directory.

Steps 1 and 2 should be repeated for each print client that has access to the printer.

To manually remove a printer's configuration on the print server side:

1. Log in as `root` on the print server that the printer is configured on.
2. Stop accepting print requests on the printer.

```
# reject printer-name
```

3. Stop the printer.

```
# disable printer-name
```

4. Delete the printer from the print server.

```
# lpadmin -x printer-name
```

Configuration information for the printer is deleted from the print server's `/etc/lp/printers` directory.

Halting and Restarting the LP Print Service

You use the `lpshut` and `lpsched` commands to temporarily halt and then restart the LP print service.

The `lpshut` command halts the LP print service. Any printers that are currently printing when the command is invoked stop printing.

```
# /usr/lib/lpshut
Print services stopped.
```

The `lpsched` command is used to start or restart the LP print service. Printers that are restarted using this command will reprint, in their entirety, the print requests that were stopped by `lpshut`.

```
# /usr/lib/lpsched
Print services started.
```

The LP print services can also be manually stopped and started on the command line using the `lp` print service script located in the `/etc/init.d` directory.

The commands `lpshut` and `lpsched` are invoked automatically from this script.

```
# /etc/init.d/lp stop
Print services stopped.
```

```
# /etc/init.d/lp start
Print services started.
```

Exercise: LP Print Service



Exercise objective – In this lab you will use the Solaris Print manager to set up a print spooler that sends output to a local terminal window, add access to a remote printer, and use print management commands.

Preparation

The host name and IP address of the system who's printer you wish to access must exist in the `/etc/hosts` file. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

- Open two terminal windows. Record the pseudo terminal device used by one of them. In the other window, run the Solaris Print Manager and define a local PostScript printer that uses the first terminal as its output device. Test the new printer.
- Use the Solaris Print Manager to gain access to a printer defined on another system. Test the new printer.
- Use the following commands to manipulate your default printer to 1) disable printer output, 2) send four files to your printer, 3) list all print jobs, 4) cancel two jobs by request ID, 5) cancel the remaining jobs by user name, 6) enable printing again, 7) reject print requests and supply a reason, 8) view the reason, 9) again accept print requests.

```
enable
disable
lp
lpstat
accept
reject
cancel
```


Tasks

Complete the following steps:

1. Log in as `root` and open two terminal windows. In one of them, use the `tty` command to identify the pseudo terminal device it uses. You'll use this device name as the port for the new printer. Example:

```
# tty
/dev/pts/5
```

Device name: _____

2. In the other terminal window, run the Solaris Print Manager

```
# /usr/sadm/admin/bin/printmgr &
```
3. In the **Select Naming Service** panel, verify that **files** is selected and click on **OK**. From the **Print Manager** menu, select **Show Command Line Console**. Position the **Command Line Console** in a convenient location.
4. From the **Printer** menu, select **New Attached Printer**.
5. Fill in the fields presented according to the table below. To name your printer, use a name different from that of your system.

Field	Selection / Entry
Printer name	(your choice)
Description	(your choice)
Printer Port	Select Other. Enter the name of the terminal window found in step 1 above.
Printer Type	PostScript
File Contents	PostScript
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 on **OK** when finished. Select **Exit** from the **Print Manager** menu to exit the Solaris Print Manager.

7. Test your printer configuration by sending a job to the default printer. Observe the output on the other terminal window. For example:

```
# lp /etc/hosts
```

8. Start the Solaris Print Manager again. In the **Select Naming Service** panel, verify that **files** is selected and click on **OK**. From the **Print Manager** menu, select **Show Command Line Console**.

```
# /usr/sadm/admin/bin/printmgr &
```

9. From the **Printer** menu, select **Add Access to Printer**.

10. Fill in the fields presented according to the table below.

Field	Selection / Entry
Printer Name	Enter the name of a printer on another system.
Print Server	Enter the name of the system where the printer above is defined.
Description	(your choice)
Default Printer	(do not select the box)

11. Click on **OK** when finished. Select **Exit** from the **Print Manager** menu to exit the Solaris Print Manager.

12. Test your new configuration by sending a job to the remote printer. Observe the output on the other system.

```
# lp -d printer /etc/hosts
```

13. In an available terminal window, use **lpstat** to display the current status information for printers defined on your system.

```
# lpstat -t
```

14. Disable print output for your default printer. Example

```
# disable printer1
```

15. Send four files to your default printer.

```
# lp /etc/hosts
# lp /etc/inittab
# lp /etc/dfs/dfstab
# lp /etc/skel/local.profile
```

16. Check the print queue to find the request ID for each job.

```
# lpstat -o
```

17. Use the request IDs to cancel two of the requests. Verify the result.
Example:

```
# cancel printer1-2 printer1-3
# lpstat -o
```

18. Cancel the other two jobs using according to the user who sent them. Verify the result.

```
# cancel -u root
# lpstat -o
```

19. Enable printing for your default printer. Example:

```
# enable printer1
```

20. Cause your default printer to reject requests and display a reason for doing so. Example:

```
# reject -r "Printer is down for maintenance" printer1
```

21. Attempt to send a job to the default printer. Observe the messages displayed. Example:

```
# lp /etc/hosts
```

22. Use lpstat to display the reason that the printer is not accepting requests. Example:

```
# lpstat -a printer1
```

23. Cause your default printer to again accept requests. Example:

```
# accept printer1
```

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Describe the basic functions of the Solaris LP print service
- ☐ Define the important LP print service directories, files, and daemons
- ☐ Describe the function of a print server and a print client
- ☐ Define the terms *local printer*, *network printer* and *remote printer*
- ☐ Use the Solaris 8 Print Manager to configure a network printer
- ☐ List the resources used by the print service to locate the destination printer
- ☐ Discuss the differences between the local printing process and a remote printing process
- ☐ Use the print service administration commands: `accept`, `reject`, `enable`, `disable`, and `lpmove`
- ☐ Configure the LP print services from the command line using `lpadmin`

Objectives

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

- Describe the main functions of the boot programmable read-only memory (PROM) chip and NVRAM
- Explain the basic elements of POST and the purpose of the Stop key to control POST
- Invoke some common boot PROM commands from the ok prompt to customize how the system boots
- Use boot command options to boot a system in different situations
- Demonstrate how to display the device tree to list all the configured devices using the `show-devs` command
- Use the `probe-` commands to identify what peripheral devices (disks, tape drives, or CDROMs) are currently connected to the system
- Determine a system's default boot device using the `devalias` command
- Create a custom device alias name for a new boot device using the `nvalias` or `nvedit` commands
- Delete a custom device alias name with the `nvunalias` command.
- Use the `eeeprom` command within the Solaris Operating Environment to view or change the values of NVRAM parameters
- Demonstrate the steps to interrupt an unresponsive system

The Boot PROM Concept

Each Sun system has a boot PROM chip. This 8-Kbyte chip is typically located on the same board as the CPU.

The main functions of the boot PROM are to test the system hardware and boot the operating system.

The boot PROM firmware, referred to as the monitor program, controls the operation of the system before the kernel is available. The boot PROM firmware has the capabilities to perform system initialization at power on and provide a user interface.

Note – The boot PROM does not understand the Solaris Operating Environment file systems or files; it deals mainly with hardware devices.

Currently there are three generations of Sun boot PROMs. Each generation has its own base revision number as described in the following list:

- 1.x – The original SPARC™ boot PROM
- 2.x – The first OpenBoot PROM (OBP)
- 3.x – The OpenBoot PROM with a flash update feature. You can update the 3.x firmware without having to replace the PROM chip.

Note – There is no OpenBoot PROM in the Intel environment.

The NVRAM Component

Another important hardware element in each Sun system is the NVRAM chip. The NVRAM is 8-Kbytes of nonvolatile random access memory. This is a pluggable chip that is often located on the main system board.

The NVRAM stores the Ethernet address, host ID, and the time-of-day (TOD) clock. A single lithium battery within the NVRAM module provides battery backup for the NVRAM and clock.

The NVRAM module also contains the EEPROM for the storage of user-configurable parameters that have been changed or customized from the boot PROM's default parameters settings. This gives you a certain level of flexibility in configuring the system to behave in a particular manner for a specific set of circumstances.

The user-interface commands and device aliases are stored in the NVRAM.

Note – The NVRAM chip has a yellow sticker with a bar code on it. Many software packages that are licensed are based on the system host ID in NVRAM. If the chip fails, Sun will replace it with a new chip containing the same host ID and Ethernet address.

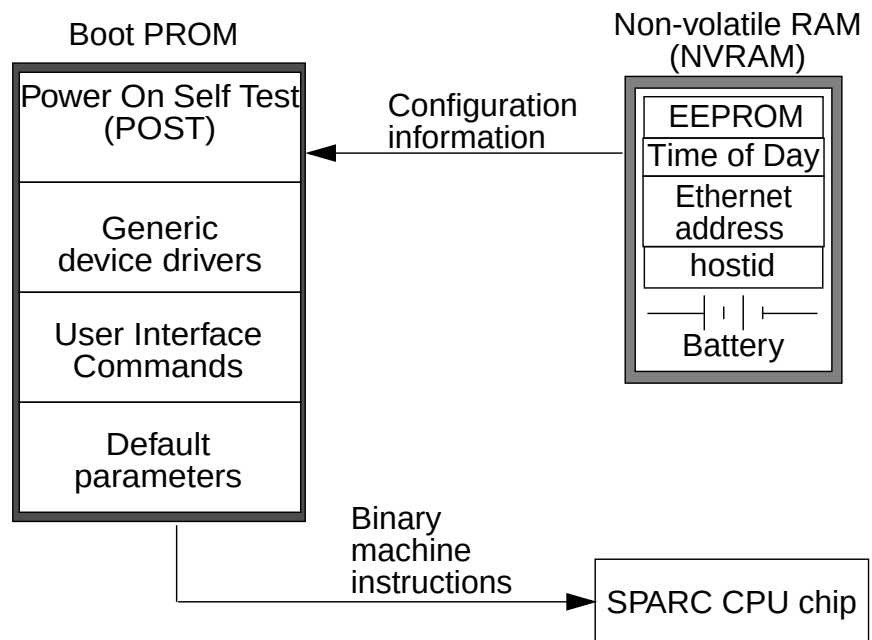


Figure 12-1 Basic Elements of the Boot PROM and NVRAM

Power On Self Test (POST)

When a system's power is turned on, a low-level power on self test (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:

- Initializes the system
- Probes the memory and then the CPU
- Probes bus devices, interprets their drivers, and builds a device tree
- Installs the console

After system initialization the banner displays on the console and the high level testing begins. When the high-level tests are finished, the system checks parameters stored in the NVRAM to determine if and how to boot the operating system.

The OpenBoot Goal

The overall goal of the OpenBoot Institute of Electrical and Electronics Engineers, (IEEE) standard is to provide the capabilities to:

- Test and initialize system hardware
- Determine the systems hardware configuration
- Boot the operating system
- Provide interactive debugging facilities
- Enable the use of third-party devices

Third-Party Device Configuration

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 OBP during the system hardware configuration phase of the boot process.

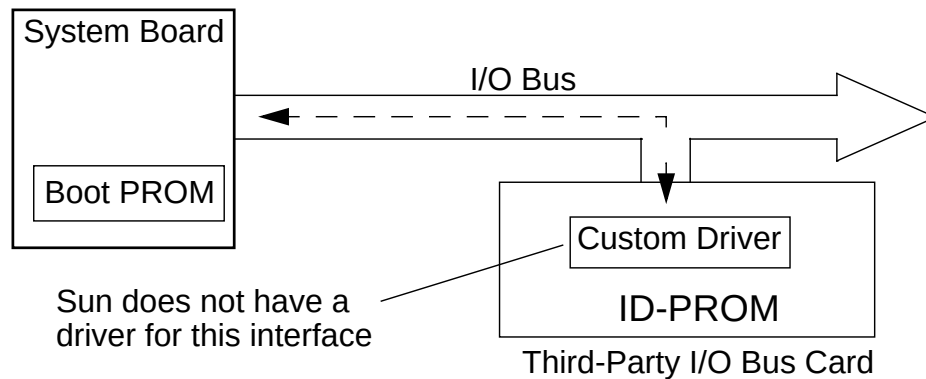


Figure 12-2 Third-Party Device Identification Process

Basic BootPROM Configurations

The following sections describe the basic BootPROM configurations.

Systems Containing a Single System Board

The following Sun systems are configured with only one system board, which holds both the boot PROM and NVRAM chip.

- SPARCstation™ 4, 5, 10, and 20
- Ultra™ 1, 2, 5, 10, 30, 60, 80, 220, 250, 420, and 450

The Ultra systems use a re-programmable boot PROM called a *flash PROM* (or FEPROM). This allows new boot program data to be loaded into the PROM via software, instead of having to replace the chip. These updates are distributed on CDROM.

Systems Containing Multiple System Boards

The following SUN systems are configured with multiple System boards.

- Enterprise 3X00
- Enterprise 4X00
- Enterprise 5X00
- Enterprise 6X00

Systems containing multiple system boards have a special boot PROM and NVRAM arrangement. These systems also have a clock board to oversee the backplane communications.

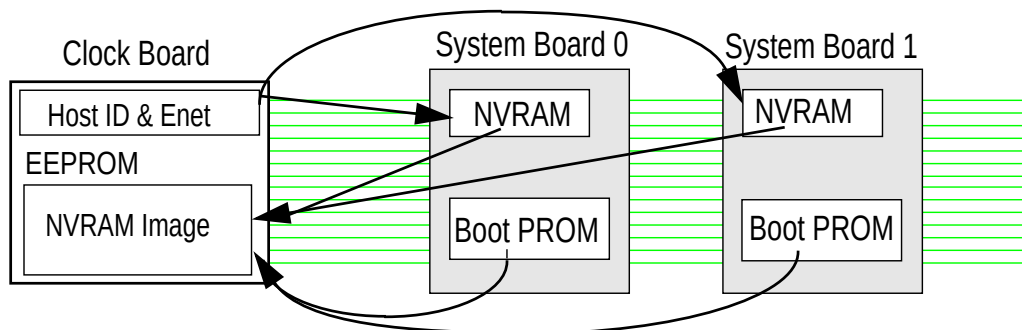


Figure 12-3 NVRAM and Boot Prom in Multi-board Systems

Some characteristics of these particular systems are:

- The CPU located in the lowest card cage slot becomes the Master CPU board.
 - ▼ Each CPU board runs its own individual POST.
 - ▼ The host ID and Ethernet address are on the Clock board and are automatically downloaded to all CPU board NVRAMs when POST is complete.
- PROM contents are verified by checksum comparisons.
 - ▼ Clock board and all system boards are compared.
 - ▼ Invalid PROM values can be manually rewritten and verified.
 - ▼ If the PROM contents on the Clock board are found to be different, it is reloaded with the contents from the Master CPU board NVRAM.
- You can update the flash PROMs (FPROMs) to newer firmware versions without replacing them. These updates are distributed on CDROM.

Controlling the POST Phase

The **Stop** key, located on the left-side of the keyboard, is used to effect the POST phase.

- To skip the POST phase at power up, power on the system while holding down the **Stop** key.
- To run extensive POST diagnostics during power up using **STOP-d**

Power on the system while holding down the **Stop** key and the “d” key simultaneously. This action sets the value of the parameter `diag-switch?` to `true`. This also forces the system to boot from the parameter `diag-device`. Its default value is usually set to `net`.

The firmware automatically switches to diagnostic mode to run extensive POST diagnostics on the system hardware.

By default, the parameter `diag-level` defaults to the maximum (`max`) setting, which instructs POST to run all available tests.

By modifying the value of `diag-level` to the minimum (`min`) setting, POST only runs an abbreviated set of tests, (in approximately half the time of the maximum setting).

- To reset the NVRAM parameter settings to the default values:

If a system does not boot and the NVRAM settings are suspect, power on the system while holding down the **Stop** key and the “n” key simultaneously. Once the keyboard LED’s (light emitting diodes) start to flash, release the keys and the system continues to boot.

Halting the Solaris Operating Environment

To halt the Solaris Operating Environment to get to the PROM monitor prompt, hold down the **Stop** key and the “a” key simultaneously. An `ok` prompt displays on the screen indicating that the monitor program is available.



Warning – You should not interrupt the Solaris Operating Environment because file systems can be corrupted. However, if a system is frozen, you can use this method to reboot the system.

If the Solaris Operating Environment had been running before the Stop-a key sequence, enter the `reset` command at the `ok` prompt to clear all buffers and registers *before entering any diagnostic commands*.

Basic Boot PROM Commands

The boot PROM monitor provides a user interface for invoking OpenBoot commands, such as those listed below.

Note – The ok prompt indicates the Solaris Operating Environment is currently not running.

The following are some commonly used commands:

- ok banner
- ok boot
- ok help
- ok printenv
- ok setenv
- ok reset
- ok set-defaults
- ok probe-ide
- ok probe-scsi
- ok probe-scsi-all

The banner Command

The banner command lists several lines of useful information about the system, such as the model name, amount of memory, host ID, Ethernet address, and the boot PROM version number, (for example, 1.x, 2.x, or 3.x).

ok **banner**

```
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), Keyboard Present
OpenBoot 3.11, 128 MB memory installed, Serial #11900965.
Ethernet address 8:0:20:b5:98:25, Host ID: 80b59825.
```


The boot Command

You use the `boot` command to boot the Solaris Operating Environment from the `ok` prompt.

This command has several options available for booting the system in different situations.

Command Format

```
ok boot [device_name] - [options]
```

Entering the `boot` command at the `ok` prompt boots the system to multiuser mode automatically. For example:

```
ok boot
```

Options

The following list describes the options for the `boot` command:

- `s` – Boots the system to a single-user mode and prompts for the `root` password. For example:

```
ok boot -s
```

Note – To continue the process and bring the system to multiuser mode, press the `Control-d` keys.

- `a` – Boots the system interactively. This is useful if you need to make a temporary change to the system file or the kernel. The boot program asks you for the following information:

```
ok boot -a
```

```
Enter filename of the kernel (kernel/unix):
```

```
Enter default directory for modules (kernel,  
/usr/kernel):
```

```
Enter name of system file (etc/system):
```

```
Enter default root file system type (ufs):
```

```
Enter physical name of root device:
```

- **r** – Performs a reconfiguration boot. Any newly attached device is found and new device entries are created in the `/devices` and `/dev` directories, and the `/etc/path_to_inst` file is updated. For example:

```
ok boot -r
```

- **V** – Boots the system while displaying more detailed device information to the console. Useful for troubleshooting problems during the boot process. Some examples include:

```
ok boot -V
```

```
ok boot -rV
```

```
ok boot -sV
```

Single-User Mode

In single user mode, the system is running only minimal processes and services, and regular users cannot log in.

Note – Single user mode is often referred to as maintenance mode. The `root` password is required to move into single-user mode on a system.

Multi-User Mode

Multi-user mode indicates the system is running all of the processes and services necessary to support multiple users who have logged in to access the system and its data.

The help Command

You use the `help` command to obtain help on the main categories contained in the OpenBoot firmware.

The `help` listing provides a number of other key words that you can use in the `help` command to provide further details.

For example:

ok **help**

Enter 'help command-name' or 'help category-name' for more help
(Use ONLY the first word of a category description)

Examples: help select -or- help line

Main categories are:

Repeated loops

Defining new commands

Numeric output

Radix (number base conversions)

Arithmetic

Memory access

Line editor

System and boot configuration parameters

Select I/O devices

Floppy eject

Power on reset

Diag (diagnostic routines)

Resume execution

File download and boot

nvrampc (making new commands permanent)

ok

Detailed Help

To view specific information for one of the main categories listed above, type the following:

- ok **help line**
- ok **help system**
- ok **help diag**
- ok **help file**

The printenv Command

You can use the `printenv` command to list all the NVRAM parameters. The name of each parameter is displayed along with the values of its default setting and current setting (if the parameter can be modified).

(The following output is edited to fit the page.)

ok **printenv**

Variable Name	Value	Default Value
tpe-link-test?	true	true
scsi-initiator-id	7	7
keyboard-click?	false	false
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
diag-level	max	max
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
security-mode	none	
security-password		
security-#badlogins	0	
diag-switch?	false	false

ok

You can also use the `printenv` command to display only a single parameter and its values. For example, to display only the `boot-device` parameter:

```
ok printenv boot-device  
boot-device = disk net
```

The possible values to `boot-device` include: `disk`, `net`, and `cdrom`.

Note – If an OBP parameter ends in a question mark (?), for example: `auto-boot?` the parameter value is either `true` or `false`.

The setenv Command

You use the `setenv` command to change the current values assigned to NVRAM parameters.

In this example, the `auto-boot?` parameter is changed from its default setting of `true` to a new current value of `false`.

```
ok printenv auto-boot?  
auto-boot? = true  
ok  
ok setenv auto-boot? false  
auto-boot? = false  
ok reset  
Resetting ...
```

The `reset` command reads the changes to the environment variables.

The reset Command

The `reset` command halts the system, clears all buffers, registers the system, and does one of the following:

- Reboots the system if the `auto-boot?` parameter is set to `true`
- Redisplays the `ok` prompt if the `auto-boot?` parameter is set to `false`

The set-defaults Command

You use the `set-defaults` command to reset all parameters to their default values. It affects only those parameters that have assigned default values.

ok **set-defaults**

Setting NVRAM parameters to default values.

ok

To reset only a specific parameter to its default value, use the `set-default` command.

ok **set-default** *parameter-name*

For example:

ok **set-default** **diag-level**

Device Tree

Sun hardware uses the concept of a device tree to organize devices that are attached to the system.

The OpenBoot firmware builds the device tree from information gathered at POST. The device tree is loaded into memory to be used by the kernel during boot to identify all configured devices.

Each node in the device tree represents a device. Nodes with children usually represent buses and their associated controllers. Their children are devices connected to the buses or controllers.

A full device path begins with a slash (/) character, the root of the tree. Each node name has the form `name@address:arguments`. Other than name, the rest are optional and the format is device-dependent.

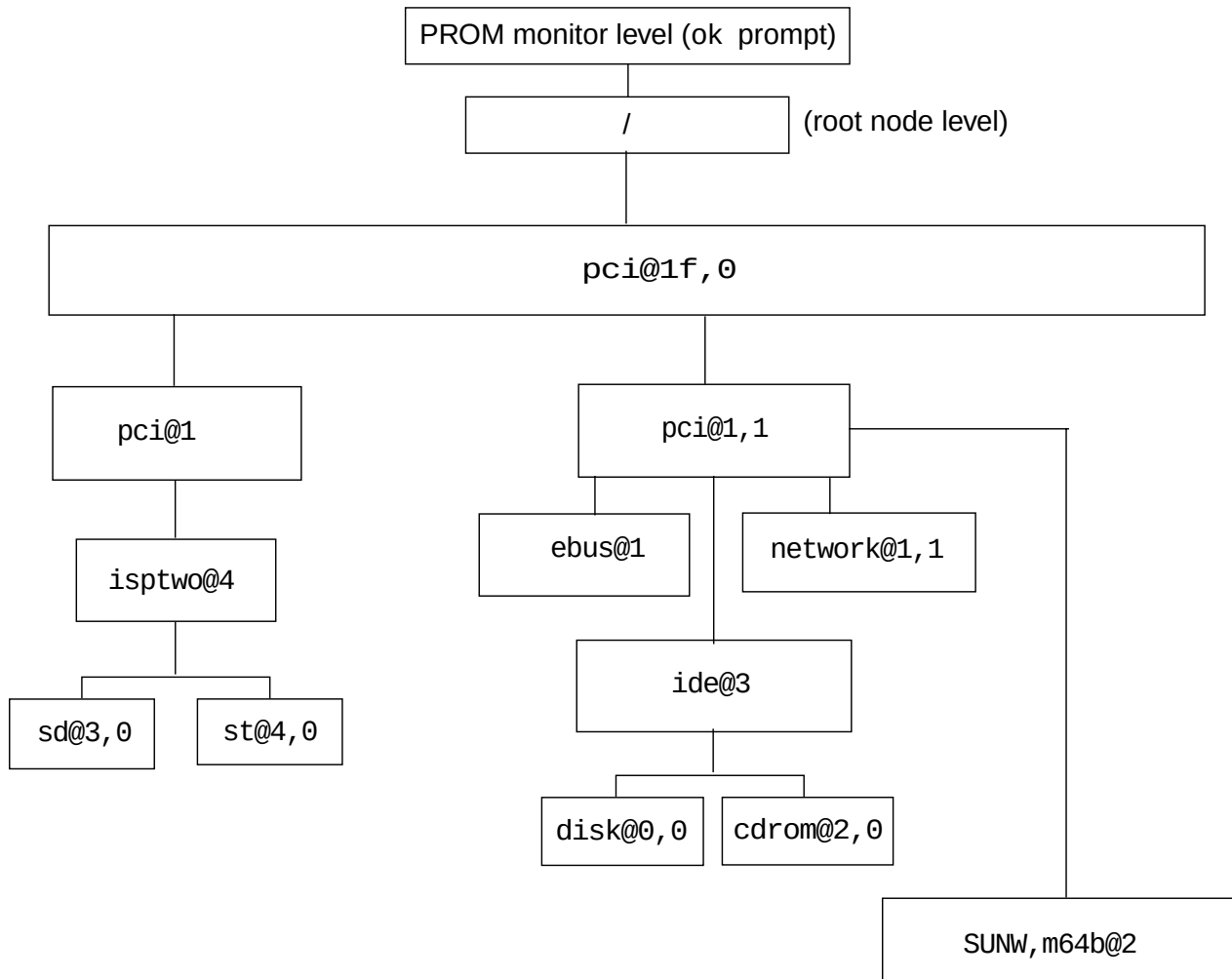


Figure 12-4 A Partial Device Tree for an Ultra 5/10

Viewing Device Path Names

To see the entire device tree, use 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/pci@1
/pci@1f,0/pci@1/pci@1/SUNW,isptwo@4
/pci@1f,0/pci@1/pci@1/SUNW,hme@0,1
/pci@1f,0/pci@1/pci@1/SUNW,isptwo@4/st
/pci@1f,0/pci@1/pci@1/SUNW,isptwo@4/sd
/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/eprom@14,0
/pci@1f,0/pci@1,1/ebus@1/fdthree@14,3023f0
/pci@1f,0/pci@1,1/ebus@1/ecpp@14,3043bc
/pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
/pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
/pci@1f,0/pci@1,1/ebus@1/se@14,400000
/pci@1f,0/pci@1,1/ebus@1/power@14,724000
/pci@1f,0/pci@1,1/ebus@1/auxio@14,726000
<output truncated>
ok
```

Boot Disk Device Path Example

The paths built in the device tree by the OpenBoot firmware will vary depending on the system type and its device configuration.

Figure 12.5 shows a sample disk device path on an Ultra system with a PCI bus.

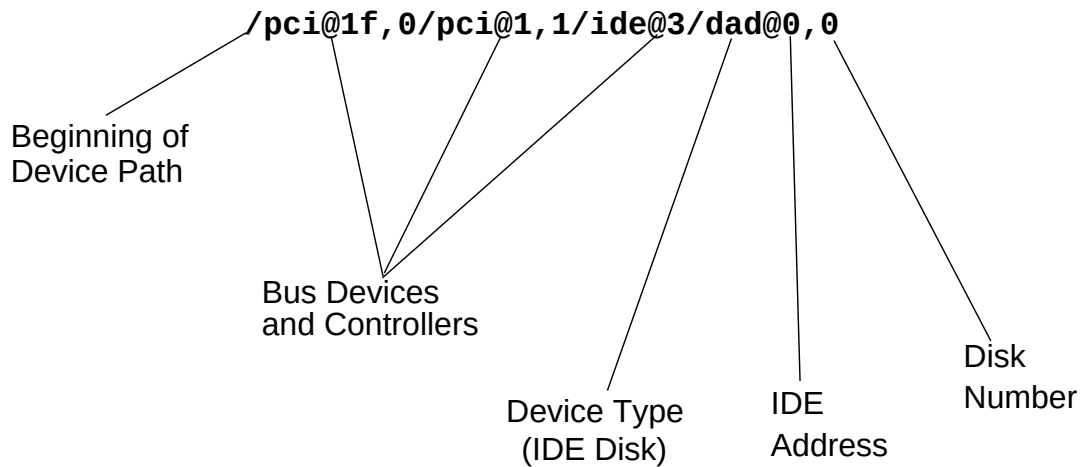


Figure 12-5 Disk Device Path - Ultra System With PCI Bus

Figure 12-6 shows a sample disk device path on an Ultra System with a PCI-SCSI bus.

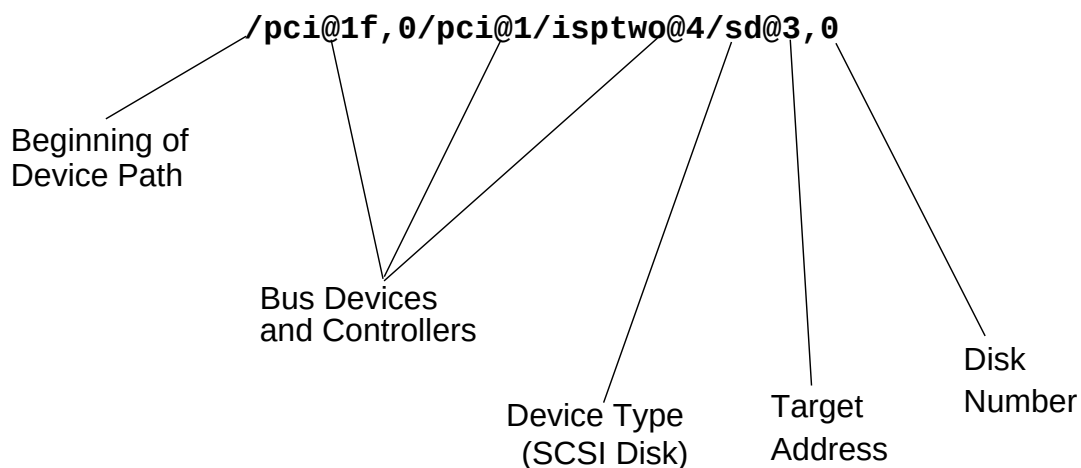


Figure 12-6 Disk Device Path - Ultra System With PCI-SCSI Bus

Using probe - Commands to Identify Devices

To identify the peripheral devices, such as disks, tape drives or CDROMs currently connected to the system, use the OBP commands:

- `probe-ide`
- `probe-scsi`
- `probe-scsi-all`

Note – Use the `probe-fcal` OBP command to identify peripheral devices on systems containing the Fiber Channel Arbitrated Loop (FC-AL) GBIC Gigabit Interface Converters.

Peripheral devices are connected to the System board by I/O (input/output) buses.

You can configure Sun systems with a small computer system interface (SCSI) bus or an integrated drive electronics (IDE) bus.

A probe - Warning Message



Warning – The following warning message is displayed if you invoke the `probe-` commands on Sun systems that contain a 3.x boot PROM.

Shutting down the Solaris operating system abruptly with the Stop-a sequence, or with the halt command, creates a condition where running the probe command hangs the system unless you run the `reset-all` command first.

When the Solaris Operating Environment has been running before the Stop-a key sequence, you must complete the following steps before using the `probe-` commands, because these commands can cause the system to freeze.

Note – If a `probe -` command causes a system to freeze, turn off the system and then turn it back on by toggling the power switch located on the back of the system unit.

1. At the `ok` prompt, set the NVRAM `auto-boot?` parameter to `false`
`ok setenv auto-boot? false`
2. At the `ok` prompt, enter the `reset` command to clear all buffers and registers before entering any diagnostic commands.
`ok reset`

The probe-scsi Command

The `probe-scsi` command identifies the peripheral devices (disks, tape drives, or CDROMs) attached to the on-board SCSI controller, by their target address. For example:

```
ok probe-scsi
Target 3
Unit 0   Disk SEAGATE ST1480 SUN0424626600190016
Target 6
Unit 0   Removable Read Only device SONY CDROM
```

The probe-scsi-all Command

The `probe-scsi-all` command identifies the peripheral devices 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,lsptwo@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 the peripheral devices, currently only disks and CDROMs, attached to the on-board `ide` controller. This command does not display target addresses, only device numbers.

For example:

```
ok probe-ide
Device 0      ( Primary Master )
               ATA Model : ST 34342A

Device 1      ( Primary Slave )
               Not Present

Device 2      ( Secondary Master )
               Removable ATAPI Model : CRD-8240B

Device 3      ( Secondary Slave )
               Not Present
```

Identifying the System's Boot Device

The system's boot device is set in the NVRAM as the `boot-device` parameter, which is by default set to `disk`.

```
ok printenv boot-device
boot-device =          disk net
```

To identify the current boot device for the system, use 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
```

Device alias names are listed on the left side of the command output, and the physical address of each device is shown on the right side of the output.

Device aliases are hard-coded into the OBP 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.

To boot the system from the default device simply type the `boot` command.

```
ok boot
```

Creating Custom Device Aliases

You can boot from an external device. External devices do not, by default, have built-in device aliases associated with them.

A portion of the NVRAM called NVRAMRC contains registers to hold parameters and is also, reserved for storing new device alias names. The NVRAMRC is effected by the commands `nvalias`, `nvunalias`, `nvedit` and the parameter `use-nvramrc?`.

The nvalias and nvunalias Commands

To create a new device alias name to access the newly attached external device, use the command `nvalias`.

To create a custom device alias name:

```
ok nvalias alias-name device-path
```

The effect of `nvalias` is to store this entire command line in the NVRAMRC.

To remove a custom device alias name:

```
ok nvunalias alias-name
```

The effect of `nvunalias` is to delete the alias name from NVRAMRC.

Using nvalias to Create Custom Device Aliases

The following procedure shows how to add a new boot device alias, called *mydisk*, and boot the system from this new boot device alias.

Using `show-disks` select the device path that relates to the disk to be used. Using `nvalias` create a new device alias called *mydisk*.

```
ok show-disks  
(select a disk from the list)  
ok nvalias mydisk /pci@1f,0/pci@1/pci@1/SUNW,isp2two@4/sd
```

To paste the device path, for the selected disk, on the command line press **Control-y**.

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.

Set the **boot-device** parameter to the new value of **mydisk**, and boot the system.

```
ok setenv boot-device mydisk
boot-device =          mydisk
ok boot
```

Removing Custom Device Aliases

You use **nvunalias** to delete the alias name **mydisk** from **NVRAMRC**, and set the **boot-device** to **disk**.

```
ok nvunalias mydisk
ok setenv boot-device disk
boot-device =          disk
ok reset
Resetting ...
```

The nvedit Command

On Sun systems with PROM versions 1.x and 2.x, the **nvalias** command might not be available to create custom device alias names.

On these systems you use the **nvedit** command to edit the **NVRAMRC** directly. The **nvedit** editor is a simple line editor that has a set of editing commands and operates in a temporary buffer.

The following is a sample **nvedit** session:

```
ok setenv use-nvramrc? true
use-nvramrc? =          true
ok nvedit
  0: devalias mydisk /pci@1f,0/pci@1,1/ide@3/disk@0,0
  1: Control-c
ok nvstore
ok reset
```


Resetting ...
ok **boot mydisk**

You use the `nvstore` command, which is invoked after exiting `nveditp` to make permanent changes to NVRAMRC.

The following lists some basic `nvedit` commands:

- ^C – Exits the editor
- ^U – Deletes the current line
- Delete – Erases the previous characters
- Return – Closes the current line, opens a new line
- ^B – Goes back one character
- ^F – Goes forward one character
- ^P – Goes back one line
- ^N – Goes forward one line

Changing NVRAM Parameters with the eeprom Command

You use the `/usr/sbin/eeprom` command to view and change the NVRAM parameters while the Solaris Operating Environment is running.

You should be aware of the following guidelines when using the `eeprom` command:

- Only `root` can change the value of a parameter.
- Parameters with a trailing question mark must be enclosed in single quotes when executed in the C shell.
- All changes are permanent. There is no `reset` command to be run.

Examples

- To list all of the parameters with default and current values, type:

```
# eeprom
```
- To list a single parameter and its value, type:

```
# eeprom boot-device
boot-device=disk
#
```
- To change the value of the default boot device, type:

```
# eeprom boot-device=disk2
#
```
- To change the value of the `auto-boot?` parameter, type:

```
# eeprom auto-boot?=true
auto-boot?=true
#
```

Interrupting an Unresponsive System

When a system freezes, or stops responding to the keyboard, you must interrupt it. Interrupting the system stops the processor immediately and does not allow for memory to be flushed, or file systems to be synchronized.

To 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 using the OBP sync command.

ok **sync**

This command causes the system to create a crash dump of memory and then reboot the system.

Exercise: OpenBoot PROM



Exercise objective – In this lab you will use the OpenBoot PROM and Solaris 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.

Task Summary

- Shut down the system to run level 0 and use the following commands to set parameters and gather basic information about your system. Set the `auto-boot?` parameter to `false`.
`banner`
`set-defaults`
`help`
`help file`
`printenv`
`setenv`
`reset`
`probe-scsi`
`probe-scsi-all`
`probe-ide`
- Create a new device alias called `mydisk` that uses the same device as the `disk` device alias. Verify the contents of `nvrामrc` and verify how `use-nvrामrc?` is set.
- Boot the system using the new alias. As `root`, use the `eepram` command to list all parameters. Set the `boot-device` parameter to `mydisk`.
- Shut down the system to run level 0 and verify the change you made using the `eepram` command. Remove the `mydisk` device alias. Reset the `boot-device` parameter to its default value and boot the system.

Tasks

1. If the Solaris Operating Environment is currently running, log in as `root` and halt your system:

\$ init 0

2. When the `ok` prompt displays, set the OPB parameters to their default values.

ok set-defaults

3. Use the `help` command to display the list of help topics.

ok help

4. Use `help` to display information about the `File download` and `boot` category.

ok help file

What does `help` list as the respective functions of the `boot`, `boot net`, and `boot cdrom` commands?

5. Use the `banner` command to obtain the following information:

ROM revision _____

Mbytes of installed memory _____

System type _____

(PROM) serial number _____

Ethernet address _____

Host ID _____

6. Use `printenv` to display the list of OBP parameters. Record the current values for the parameters listed below:

ok **printenv**

output-device_____

input-device_____

auto-boot?_____

boot-device_____

7. Prevent the system from booting automatically after a reset by setting the `auto-boot?` parameter to `false`.

ok **setenv auto-boot? false**

8. Use `reset` to verify that the new `auto-boot?` value is in effect. The system should remain at the `ok` prompt after the reset completes.

ok **reset**

9. 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**



Caution – If any of these commands returns a message warning that your system will hang if you proceed, enter `n` to avoid running the command. Run `reset` before running `probe-scsi`, `probe-scsi-all` or `probe-ide` again, and then respond `y` to this message.

10. What are the main differences in the information these commands display?

11. List the target number and device type (disk, tape, CDROM) of all the devices shown by `probe-scsi`, `probe-scsi-all`, and `probe-ide`.

Target or Device Number _____ Device type _____

Target or Device Number _____ Device type _____

Target or Device Number _____ Device type _____

Target or Device Number _____ Device type _____

Target or Device Number _____ Device type _____

Target or Device Number _____ Device type _____

Target or Device Number _____ Device type _____

12. Verify your default `boot-device` is set to: `disk net`

ok **printenv boot-device**

13. Use the `devalias` command to display the full device path for the `disk` alias.

ok **devalias disk**

Record the pathname reported:

14. Use `show-disks` to select the device path that relates to the disk recorded previously, and `nvalias` to create a new device alias called `mydisk`. Set `mydisk` to the path and disk name you recorded in the previous step.

Remember to use Control-y to paste the disk path into your `nvalias` command. You must manually complete the path in order to specify the disk you want to use.

ok **show-disks**

(select a disk from the list)

ok **nvalias mydisk *pathname***

15. Verify the new alias is correctly set.

ok **devalias mydisk**

16. Use `printenv` to display the content of `nvrsrc`.

ok **`printenv nvrsrc`**

What command does `nvrsrc` contain that creates the `mydisk` alias?

17. Use `printenv` to display the setting of the `use-nvrsrc?` parameter.

ok **`printenv use-nvrsrc?`**

What is the current setting of the `use-nvrsrc?` parameter?

18. Boot your system using the `mydisk` alias.

ok **`boot mydisk`**

19. Log in as `root` on your system. Open a new terminal window.

20. Use the `eepram` command to list all NVRAM parameters.

\$ **`eepram`**

21. Use the `eepram` command to list the setting of the `boot-device` parameter.

\$ **`eepram boot-device`**

22. Use the `eepram` command to set the `boot-device` parameter to `mydisk`.

\$ **`eepram boot-device=mydisk`**

23. Bring your system to run level 0.

\$ **`init 0`**

24. Verify the `eepram` command set the `boot-device` parameter to `mydisk`.

ok **`printenv boot-device`**

25. Set the `boot-device` parameter to its default value, and verify the setting.

```
ok set-default boot-device
ok printenv boot-device
```
26. Use `nvunalias` to remove the alias `mydisk`.

```
ok nvunalias mydisk
```
27. Verify the `mydisk` alias is no longer in `nvrnrc`.

```
ok printenv nvrnrc
```
28. Use `devalias` to see if `mydisk` has been removed from the list of device aliases.

```
ok devalias mydisk
```

Has it? _____
29. Run `reset` and then check again if `mydisk` has been removed from the list of device aliases.

```
ok reset
ok devalias mydisk
```

Has it? _____
30. Set OBP parameters back to their default values and boot the system from the default device.

```
ok set-defaults
ok boot
```
31. Log in as `root`.

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

4. What does `help` list as the respective functions of the `boot`, `boot net`, and `boot cdrom` commands?

boot - boot kernel from default device

boot net - boot kernel from network using auto-selected interface

boot cdrom - boot kernel from CDROM

5. Use the `banner` command to obtain the following information...

Each system will present its own unique information.

6. Use `printenv` to display the list of OBP parameters. Record the current values for the parameters listed below:

`output-device` - *screen*

`input-device` - *keyboard*

`auto-boot?` - *true*

`boot-device` - *disk net*

10. What are the main differences in the information these commands display?

`probe-scsi-all` lists all devices on all SCSI chains and their full device paths. `probe-scsi` only lists devices on the first SCSI chain, and does not list the full device paths. `probe-ide` reports the list of IDE devices attached to the system.

11. List the target number and device type (disk, tape, CDROM) of all the devices shown by `probe-scsi`, `probe-scsi-all`, and `probe-ide`.

Each system will present its own unique information.

13. Use the `devalias` command to display the full device path for the `disk` alias.

This may differ from system to system. On a SPARCstation 5, the alias is defined as follows:

`/iommu/sbus/espdma@5,8400000/esp@5,8800000/sd@3,0`

16. What command does `nvrarc` contain that creates the `mydisk` alias?

Systems differ according to the disk devices they use. An Ultra 5 would report the following:

`devalias mydisk /pci@1f,0/pci@1,1/ide@3/disk@0,0`

17. What is the current setting of the `use-nvrarc?` parameter?

`true`

28. Use `devalias` to see if `mydisk` has been removed from the list of device aliases. Has it?

`no`

29. Run `reset` and then check again if `mydisk` has been removed from the list of device aliases. Has it?

`yes`

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Describe the main functions of the boot PROM and NVRAM
- ☐ Explain the basic elements of POST and the purpose of the Stop key to control POST
- ☐ Invoke some common boot PROM commands from the `ok` prompt to customize how the system boots
- ☐ Use `boot` command options to boot a system in different situations
- ☐ Demonstrate how to display the device tree to list all the configured devices using the `show-devs` command
- ☐ Use the `probe-` commands to identify what peripheral devices (disks, tape drives, or CDROMs) are currently connected to the system
- ☐ Determine a system's default boot device using the `devalias` command
- ☐ Create a custom device alias name for a new boot device using the `nvalias` or `nvedit` commands
- ☐ Delete a custom device alias name with the `nvunalias` command.
- ☐ Use the `eeeprom` command within the Solaris Operating Environment to view or change the values of NVRAM parameters
- ☐ Demonstrate the steps to interrupt an unresponsive system

Objectives

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

- Describe the four phases of the boot process
- Identify the directories that contain the `kernel` and its loadable modules
- Modify the `kernel`'s configuration file
- Describe the eight Solaris Operating Environment run levels
- Define a system's current run level using the `who -r` command
- Explain the purpose of the `/etc/inittab` file
- Describe the steps in the `init` process to bring a system to multiuser mode
- List the directories that hold the run control scripts used to stop and start system processes and services
- Describe the steps to add a new run control script
- Use the following commands to shut down the system: `init`, `shutdown`, `halt`, `poweroff`, and `reboot`

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Sun Part Number 805-7228-10

The Solaris Operating Environment Run Levels

A run level is a digit or a letter representing a system state that defines what services and resources are currently available to users. The system is always running in one run level.

Run levels are sometimes referred to as `init` states because the `init` process is used to transition between run levels. You can use the `init` command to manually initiate run-level transitions.

The Solaris Operating Environment has eight run levels, which are described in the following table.

Table 13-1 Solaris Run Levels

Run Level	Function
0	Shuts down the Solaris Operating Environment and displays the boot PROM ok prompt so it is safe to turn off power to the system.
s or S	Runs as single user with all file systems mounted and accessible.
1	Indicates system is running in a single-user administrative state with access to all available file systems.
2	Indicates system is running in multi-user operations. Multiple users can access the system. All system daemons are running except for the NFS server daemons.
3	Indicates system is running in multi-user operations with NFS resource-sharing available. Specified as the default run level in the <code>/etc/inittab</code> file.
4	This level is currently not implemented.
5	Shuts down the Solaris Operating Environment and powers off the system.
6	Shuts down the system to run level 0, and then reboots to multi-user operations, (or the level set in the default in the <code>/etc/inittab</code> file).

Determining a System's Current Run Level

To determine the current run level of a system, use the following command.

```
# who -r
```

```

.   run level 3   Jun 9 08:30   3   0   S
    ^              ^             ^   ^   ^
    |              |             |   |   |
Current run level  Date and Time Current run level Number of times Previous
                  of last run   run level at this run level run level
                  level change

```

The Boot Process

In general, when a system is turned on, the PROM monitor runs a quick self-test procedure that checks the hardware and memory on the system. If no errors are found, the system begins the automatic boot process.

The entire boot process is described by four distinct phases:

- Boot PROM phase
- Boot Programs phase
- Kernel Initialization phase
- `init` phase

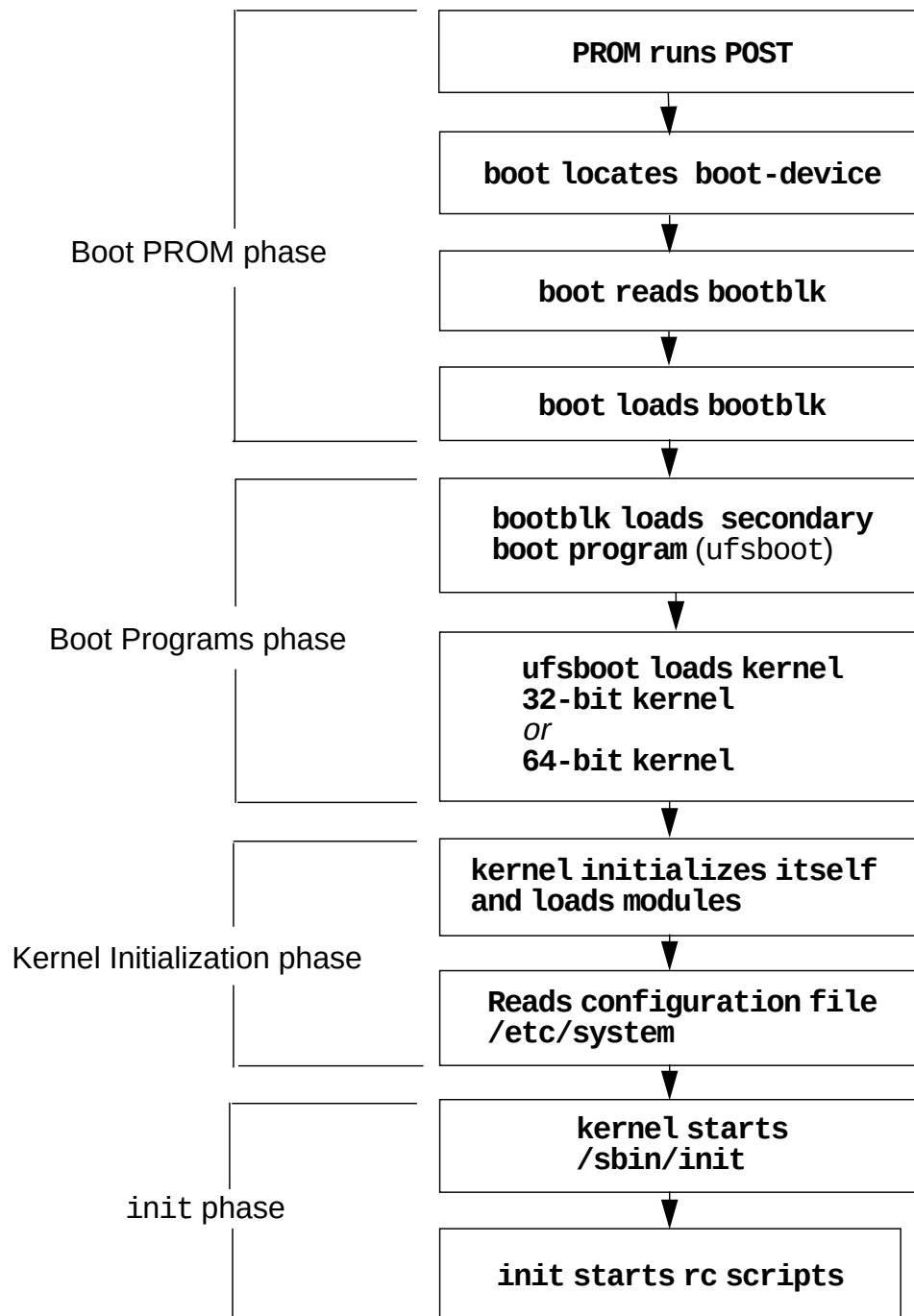


Figure 13-1 Phases of the Boot Process

Boot PROM Phase

The following describes the Boot PROM phase:

- PROM runs POST

The boot PROM firmware runs the power on self test (POST) to verify the system's hardware and memory.

The PROM displays the system identification banner, (for example, model type, amount of installed memory, PROM version number, PROM serial number, Ethernet address, and host ID).

- boot determines the boot device.
- boot locates the `bootblk` on the boot device.
- boot loads the `bootblk` from its location on the boot device into memory.

The primary boot program, `bootblk`, is located in a fixed location on the boot device in sectors 1-15.

Its purpose is to load the secondary boot program located in the `ufs` file system on the boot device.

Boot Programs Phase

The following describes the Boot Programs phase:

- `bootblk` loads the secondary boot program, `ufsboot` from the boot device into memory.

The path to `ufsboot` is recorded in the `bootblk`, which is installed by the Solaris utility `installboot`.

- `ufsboot` locates and loads the appropriate two-part kernel.

The kernel is comprised of a two-piece static core 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 32-bit system, the two-part kernel is located in the directory `/platform/'uname -m'/kernel`.

On a 64-bit system, the two-part kernel is located in the directory `/platform/'uname -m'/kernel/sparcv9`.

Note – To determine the platform name (e.g. the system hardware class), type the command `uname -m`. For example, by typing this command on a Sun Ultra10 it would display: `sun4u`

The kernel Initialization Phase

The following describes the kernel Initialization phase:

- The kernel initializes itself and begins loading modules.

The kernel uses `ufsboot` to read the files. When it has loaded enough modules to mount the root file system it unmaps the `ufsboot` program and continues on.

- The kernel reads its configuration file called `/etc/system`.
- The kernel starts the `/sbin/init` process.

The SunOS kernel consists of a small, static core (`genunix` and `unix`) and many dynamically loadable kernel modules.

Modules can consist of device drivers, file systems, streams, as well as other types used for specific tasks within the system.

The modules which comprise 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.

Each subdirectory located under these directories is a collection of similar-type modules.

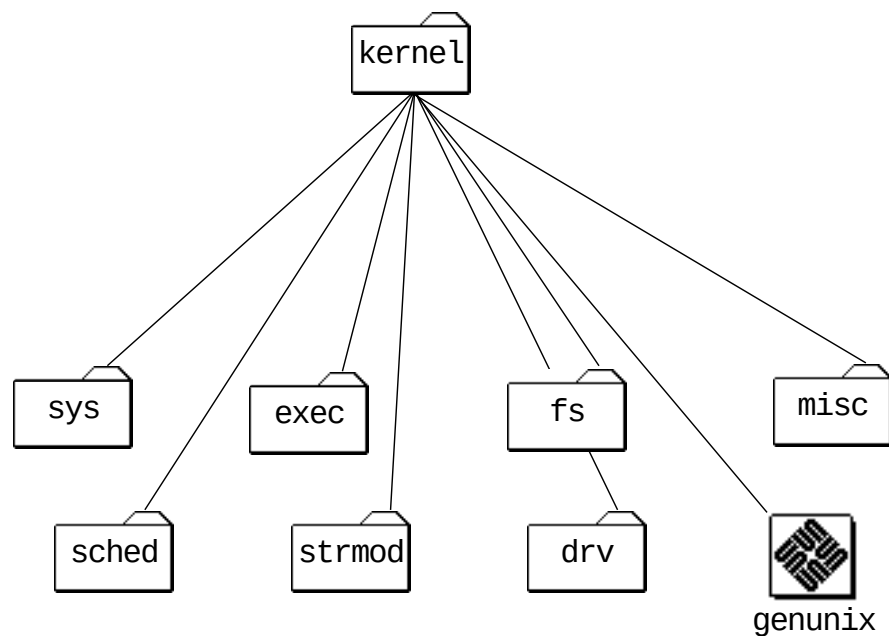


Figure 13-2 Module Subdirectories in /kernel

The following describes the types of module subdirectories contained in /kernel, /usr/kernel, /platform/'uname -m'/kernel, or /platform/'uname -i'/kernel directories:

- sys – System calls (defined interfaces for applications to use)
- exec – Executable file formats
- fs – File system types, for example, ufs, nfs, and proc
- misc – Miscellaneous modules (virtual swap)
- sched – Scheduling classes (process execution scheduling)
- strmod – Streams modules (generalized connection between users and device drivers)
- drv – Device drivers

The /kernel/drv directory contains all of the device drivers used for system boot.

The directory /usr/kernel/drv 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 is unloaded on the basis that the memory it uses up is needed for another task.

The advantages of this dynamic kernel arrangement are the overall size of the kernel is smaller making more efficient use of memory and allowing for simpler modification and tuning.

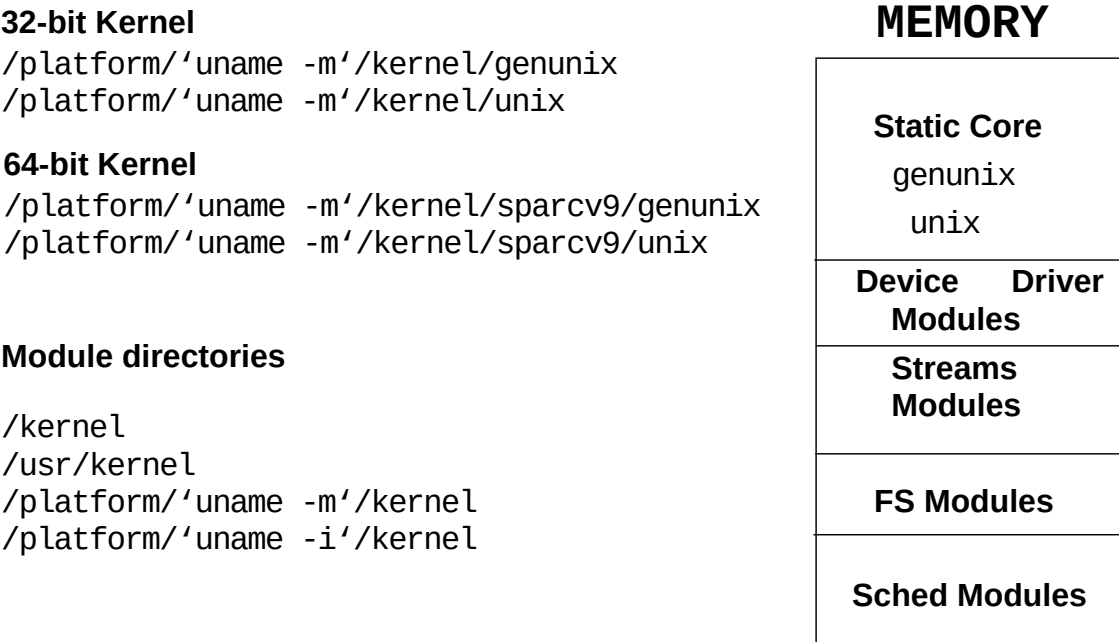


Figure 13-3 Kernel and Modules Loaded In Memory

Note – The sparcv9 is the type of CPU that supports 64-bit processing.

Configuring the kernel

The /etc/system file is the control file for specifying 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 the system administrator to modify the kernel's treatment of loadable modules, as well as kernel parameters for some performance tuning.

The `boot` program contains a list of default loadable kernel modules which are loaded at boot time. However, you can override this list by modifying the `/etc/system` file to control which modules, as well as parameters are loaded.

All changes to this file take effect after a `reboot`.

The `/etc/system` file explicitly controls:

- The search path for default modules to be loaded at boot time.
- The root type and device.
- The modules not to be loaded automatically at boot time.
- The modules to be forceable loaded automatically 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.

Sample /etc/system File

```
* ident "@(#)system      1.18      97/06/27 SMI" /* SVR4 1.5 */
* SYSTEM SPECIFICATION FILE
*
* moddir:
* Set the search path for modules. This has a format similar to the csh path
* variable. If the module isn't found in the first directory it tries the second
* and so on. The default is /kernel /usr/kernel
* Example:
*         moddir: /kernel /usr/kernel /other/modules
*
* root device and root filesystem configuration:
* The following may be used to override the defaults provided by the boot program:
* rootfs:          Set the filesystem type of the root.
*
* rootdev:         Set the root device. This should be a fully
*                  expanded physical pathname. The default is the
*                  physical pathname of the device where the boot
*                  program resides. The physical pathname is
*                  highly platform and configuration dependent.
* Example:
*         rootfs:ufs
*         rootdev:/sbus@1,f8000000/esp@0,8000000/sd@3,0:a
*         (Swap device configuration should be specified in /etc/vfstab.)
*
* exclude:
* Modules appearing in the moddir path which are NOT to be loaded, even if referenced.
* Note that 'exclude' accepts either a module name, or a filename which includes the
* directory.
* Examples:
*         exclude: win
*         exclude: sys/shmsys
*
* forceload:
* Cause these modules to be loaded at boot time, (just before mounting the root
* filesystem) rather than at first reference. Note that forceload expects a
* filename which includes the directory. Also note that loading a module does
* not necessarily imply that it will be installed.
* Example:
*         forceload: drv/foo
*
* set:
* Set an integer variable in the kernel or a module to a new value.
* This facility should be used with caution. See system(4).
*
* Examples:
* To set variables in 'unix':
*         set nautopush=32
*         set maxusers=40
* To set a variable named 'debug' in the module named 'test_module'
*         set test_module:debug = 0x13
```

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, it tries the second directory, and so on.

- `root device and root filesystem 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 pathname of the device where the boot program resides. The physical pathname is platform and configuration dependent. For example:

`rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a`

- `exclude:`

Does not allow the loadable kernel module(s) to be loaded during kernel initialization. For example: `exclude: sys/shmsys`

- `forceload:`

Forces the kernel module(s) to be loaded during kernel initialization. For example: `forceload: drv/vx`

The default action is to automatically load a kernel module when its services are first accessed during runtime, by a user or application.

- `set:`

Changes kernel parameters to modify the operation of the system. For example: `set maxusers=40`

Editing the `/etc/system` File

Before editing 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 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 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.

The init Phase

The final phase of the boot process is the `/etc/init` phase. During this phase `init` start the run control scripts which starts other processes.

The `init` process executes rc scripts which in turn execute a series of other scripts

Once the `init` phase completes successfully, the system login prompt is displayed.

The /etc/inittab File

When you boot a system, or change run levels with the `init` or `shutdown` command, the `init` daemon starts processes by reading information from the `/etc/inittab` file.

The `inittab` file defines three important items for the `init` process:

- The system's default run level.
- What processes to start, monitor, or restart if terminated.
- What actions to take when the system enters a new run level.

Each line entry in the `/etc/inittab` file has the following four fields:

`id:rstate:action:process`

The fields in an `inittab` entry are described in the following table:

id	A 1 to 4 character identifier for the entry.
rstate	One or more run levels to which this entry applies.
action	How the process (in the next field) is to be treated.
process	The command or script to execute.

s3:3:wait:/sbin/rc3	> /dev/msglog 2<> /dev/msglog </dev/console

Figure 13-4 An `/etc/inittab` File Entry

Note – Message output from system startup (`rc`) scripts is directed to `/dev/msglog`. Previously, all of these messages were written to `/dev/console`. For more information refer to `msglog(7D)`.

Some possible keywords used in the `action` field include:

`initdefault` Identifies the default run level. Read when `init` is initially invoked. Used by `init` to determine which run level to enter initially. The default is run level 3.



Caution – If the `rstate` field is empty it is interpreted as `0123456` and `init` will enter run level 6, as the default. This will cause the system to reboot continuously.

<code>sysinit</code>	Executes the process before <code>init</code> tries to access the console (for example, the console login prompt). <code>init</code> waits for its completion before it continues to read the <code>inittab</code> file.
<code>wait</code>	Starts the process and waits for it to complete before moving to the next entry containing the same run level.
<code>respawn</code>	If the process dies, <code>init</code> will restart it. If the process does not exist, <code>init</code> starts it and continues reading the <code>inittab</code> file. If the process does exist, no action required, and <code>init</code> continues reading the <code>inittab</code> file.
<code>powerfail</code>	Executes the process only if <code>init</code> receives a power fail signal.

Note – Additional `action` keywords are available and defined in the `inittab` man page.

Default /etc/inittab File

The following is an example of the default /etc/inittab file.

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
ap::sysinit:/sbin/soconfig -f /etc/sock2path
fs::sysinit:/sbin/rcS sysinit >/dev/msglog 2< >/dev/msglog </dev/console
is:3:initdefault:
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog
2< >/dev/msglog
sS:s:wait:/sbin/rcS >/dev/msglog 2< >/dev/msglog </dev/console
s0:0:wait:/sbin/rc0 >/dev/msglog 2< >/dev/msglog </dev/console
s1:1:respawn:/sbin/rc1 >/dev/msglog 2< >/dev/msglog </dev/console
s2:23:wait:/sbin/rc2 >/dev/msglog 2< >/dev/msglog </dev/console
s3:3:wait:/sbin/rc3 >/dev/msglog 2< >/dev/msglog </dev/console
s5:5:wait:/sbin/rc5 >/dev/msglog 2< >/dev/msglog </dev/console
s6:6:wait:/sbin/rc6 >/dev/msglog 2< >/dev/msglog </dev/console
fw:0:wait:/sbin/uadmin 2 0 >/dev/msglog 2< >/dev/msglog </dev/console
of:5:wait:/sbin/uadmin 2 6 >/dev/msglog 2< >/dev/msglog </dev/console
rb:6:wait:/sbin/uadmin 2 1 >/dev/msglog 2< >/dev/msglog </dev/console
sc:234:respawn:/usr/lib/saf/sac -t 300
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "'uname -n' console login: "
-T sun -d /dev/console -l console -m ldterm,ttcompat
```

The following describes each inittab line entry:

1. Initializes STREAMS modules
2. Configures socket transport providers
3. Initializes file systems
4. Defines default run level
5. Describes a power fail shutdown
6. Defines single-user mode
7. Defines run level 0
8. Defines run level 1
9. Defines run level 2
10. Defines run level 3
11. Defines run level 5

12. Defines run level 6
13. Defines transition to firmware
14. Defines transition to power off
15. Defines transition to reboot
16. Initializes Service Access Controller
17. Initializes console

The `init` Process

The following illustrates the process of bringing a system to the default run level 3.

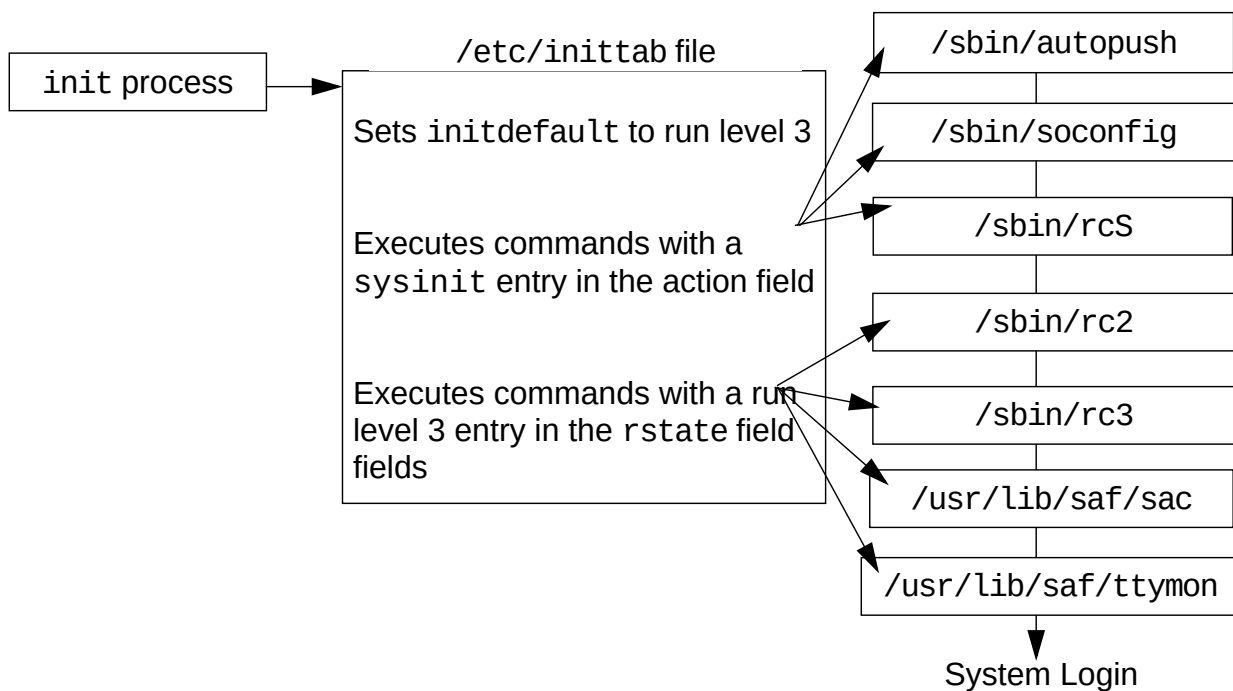


Figure 13-5 The `init` Process

The `/etc/init` process reads the `/etc/inittab` file to do the following:

1. Identify the `initdefault` entry, which defines the default run level 3.
2. Execute any process entries that have `sysinit` in the action field so that any special initialization can take place before users login.
3. Execute any process entries that have 3 in the `rstate` field, which matches the default run level, 3.

The commands executed at this run level include:

- `/usr/sbin/shutdown` – The `init` process runs the `shutdown` command only if the system has received a powerfail signal.
- `/sbin/rcS` – Mounts and checks `/` (`root`), `/usr`, `/var`, and `/var/adm` file systems.
- `/sbin/rc2` – Starts the system daemons, bringing the system up into run level 2 (multi-user mode).
- `/sbin/rc3` – Starts NFS resource sharing for run level 3.
- `/usr/lib/saf/sac` – Starts or restarts the port monitors and network access for UUCP.
- `/usr/lib/saf/ttymon` – Starts or restarts the `ttymon` process that monitors the console for login requests. The `terminal_type` on a SPARC-based system is `sun`. The `terminal_type` on an IA-based system is `AT386`.

Run Control Scripts

The Solaris Operating Environment provides a series of run control (rc) scripts to stop and start processes normally associated with run levels.

The /sbin Directory

Each run level has an associated rc script located in the /sbin directory.

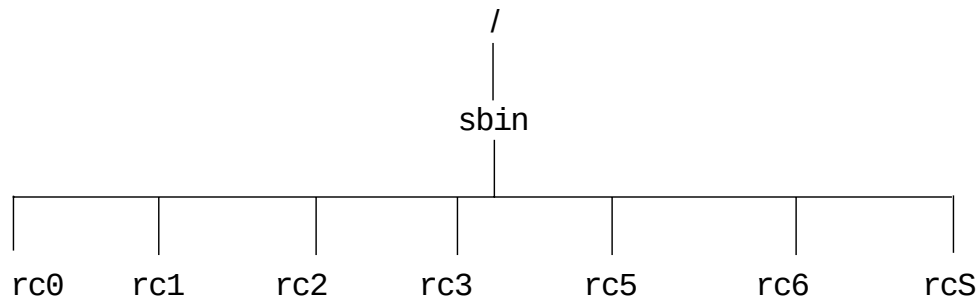


Figure 13-6 The /sbin Directory

The rc scripts are executed by `init` to set up variables, test conditions, and make calls to other scripts that start and stop processes for that run level.

The rc scripts `rc0`, `rc5` and `rc6` files are hard linked. For example:

```

# cd /sbin
# ls -i rc*
47154 rc0      47156 rc2      47154 rc5      47158 rcS
47155 rc1      47157 rc3      47154 rc6
  
```

SunOS 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.

```

# cd /etc
# ls -l rc?
  
```

```

lrwxrwxrwx 1 root    root      11 Feb 22 14:19 rc0 -> ../sbin/rc0
lrwxrwxrwx 1 root    root      11 Feb 22 14:19 rc1 -> ../sbin/rc1
lrwxrwxrwx 1 root    root      11 Feb 22 14:19 rc2 -> ../sbin/rc2
lrwxrwxrwx 1 root    root      11 Feb 22 14:19 rc3 -> ../sbin/rc3
lrwxrwxrwx 1 root    root      11 Feb 22 14:19 rc5 -> ../sbin/rc5
lrwxrwxrwx 1 root    root      11 Feb 22 14:19 rc6 -> ../sbin/rc6
lrwxrwxrwx 1 root    root      11 Feb 22 14:19 rcS -> ../sbin/rcS
#

```

The /etc/rc#.d Directories

For each /sbin/rc script, there is a corresponding directory named /etc/rc#.d.

The /etc/rc#.d directories contain additional scripts that start and stop system processes for that run level.

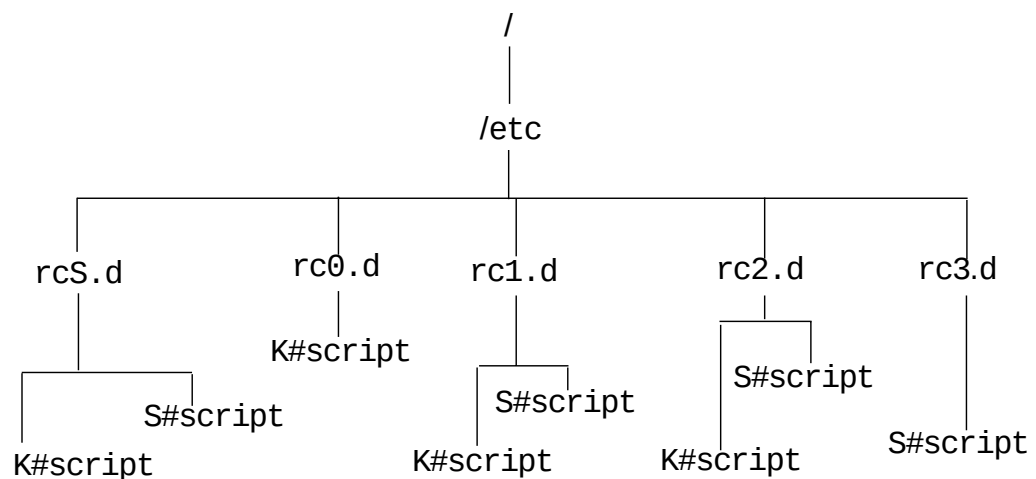


Figure 13-7 The /etc/rc#.d Directories

For example, /etc/rc2.d contains scripts used to start and stop processes for run level 2.

```
# ls /etc/rc2.d
```

The /etc/rc#.d scripts are always run in the sort order shown by the ls command. These files have names in the form of:

```
[KS][0-9][0-9]*
```

Files beginning with **K** are run to terminate (kill) a system process.
Files beginning with **S** are run to start a system process.

Note – File names that begin with a lowercase **k** or **s** are ignored by **init** and they 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. These files are hard linked to corresponding run control scripts in the `/etc/rc#.d` directories.

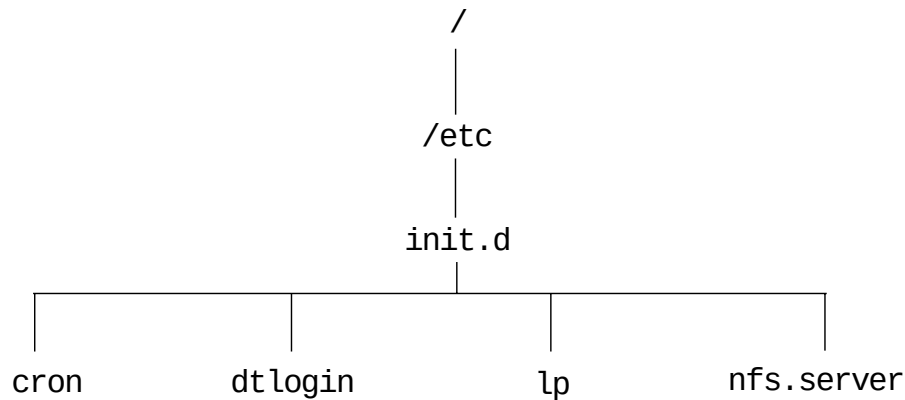


Figure 13-8 The `/etc/init.d` Directory

The benefit of having individual scripts for each run level is that you can run scripts in the `/etc/init.d` directory individually by **root**. You can turn off a process or start a process without changing the system's run level.

For example, to stop and restart the **lp** print services, run the following scripts with a **stop** or **start** command:

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

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

Summary of Run Control Scripts and Functions

The following table summarizes the tasks that are performed by each of the /sbin rc scripts.

Table 13-2 Run Control Scripts and Function

rc Script	Function
/sbin/rc0	Runs the /etc/rc0.d/K* scripts to perform the following tasks - Stops system services and daemons - Terminates all running processes - Unmounts all file systems
/sbin/rc1	Runs the /etc/rc1.d scripts to perform the following tasks: - Stops system services and daemons - Terminates all running processes - Unmounts all file systems
/sbin/rc2	Runs the /etc/rc2.d scripts to perform the following tasks: - Mounts all local file systems - Removes any files in the /tmp directory - Configures system accounting - Configures default router - Starts most of the system daemons
/sbin/rc3	Runs the /etc/rc3.d scripts to perform the following tasks: - Cleans up /etc/dfs/sharetab file - Starts nfsd and mountd
/sbin/rc5 /sbin/rc6	Runs the /etc/rc0.d/K* scripts to perform the following tasks: - Kills all active processes and unmounts the file systems
/sbin/rcS	Runs the /etc/rcS.d scripts to bring the system up to run level S. - Establishes a minimal network - Mounts /usr, if necessary - Sets the system name - Checks the / (root) and /usr file systems - Mounts pseudo file systems (/proc and /dev/fd) - Rebuilds the device entries for reconfiguration boots - Mounts other file systems to be mounted in single-user mode

Creating a New Run Control Script

You can create new scripts to start and stop additional processes or services to customize a system.

For example, to eliminate the requirement for having to manually start a database server, create a script to automatically start the database server once 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.

To add run control scripts to start and stop a service, create the script in the `/etc/init.d` directory and create links in the appropriate `/etc/rc#.d` directory for the run level the service is to be started and stopped.

See the README file in each `/etc/rc#.d` directory for more information on run control scripts.

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 0744 /etc/init.d/filename
# chown root:sys /etc/init.d/filename
```

2. Create links to the appropriate `/etc/rc#.d` directory.

```
# cd /etc/init.d
# ln filename /etc/rc#.d/S##filename
# ln filename /etc/rc#.d/K##filename
```

3. Use the `ls` command to verify that the script has links in the appropriate directories.

```
# ls /etc/init.d /etc/rc#.d /etc/rc#.d
```

4. Test the *filename* by entering the following commands:

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

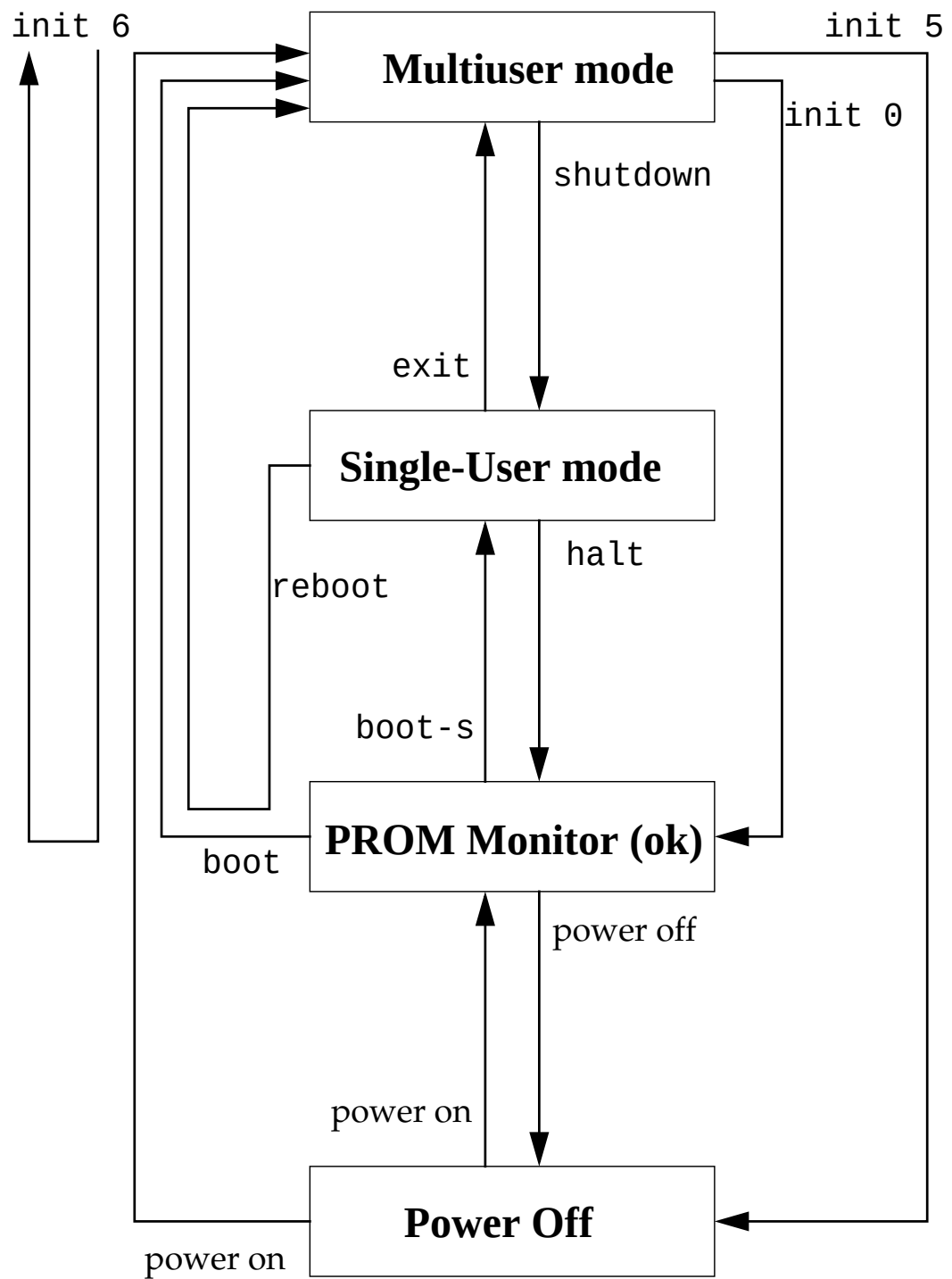


Figure 13-9 Run-Level Transitions

System Shutdown Procedures

You can shut down the Solaris Operating Environment to perform administration tasks or maintenance activities, in anticipation of a power outage, or if you need to move the system to a new location.

The Solaris Operating Environment requires a clean and orderly shutdown process, which stops processes, writes data in memory to disk(s), and unmounts file systems.

Of course, the type of work you need to do determines how the system is shut down and what command is used.

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 Operating Environment and display the `ok` prompt
- Shut down the system to turn off power
- Shut down the system automatically and `reboot` to multi-user mode

The commands available to `root` for doing these types of system shutdown procedures include:

- `/sbin/init` (using run levels `S`, `1`, `0`, `5` or `6`)
- `/usr/sbin/shutdown` (using run levels `S`, `1`, `0`, `5` or `6`)
- `/usr/sbin/halt`
- `/usr/sbin/poweroff`
- `/usr/sbin/reboot`

The /sbin/init Command

You can use the `init` command to shutdown, `poweroff`, or `reboot` a system in a clean and orderly manner. It executes the `rc0` kill scripts, however, this command does not warn logged in users that the system is being shutdown, and there is no delay.

To shut down the system to single-user mode, use either run level S or 1, for example:

```
# init S
```

To shut down the system to stop the Solaris Operating Environment and display the ok prompt:

```
# init 0
```

To shut down the system and turn its power off:

```
# init 5
```

To shut down the system and then reboot to multi-user mode:

```
# init 6
```

The /usr/sbin/shutdown Command

The shutdown command is a script that invokes init to shutdown, poweroff, or reboot the system. It does execute the rc0 kill scripts to shutdown processes and applications gracefully. Unlike the init command, the shutdown command does the following:

- Notifies all logged in users that the system is being shutdown
- Delays the shutdown for 60 seconds by default
- Gives you the capability to include an optional descriptive message to inform your users

Command Format

```
shutdown [ -y ] [ -g grace-period ] [ -i init-state ] [ optional message ]
```

The -y option is used to pre-answer the final shutdown confirmation question so the command runs without user intervention.

The -g grace-period allows root to change the number of seconds from the 60-second default.

The -i init-state specifies the state init is to be in. By default, system state S is used.

To shut down the system to single-user mode, enter either run level S or 1, for example:

```
# shutdown -iS
```

To shut down the system to stop the Solaris Operating Environment and display the ok prompt:

```
# shutdown -i0
```

To shut down the system and turn off its power automatically:

```
# shutdown -i5
```

To shut down the system and then reboot to multi-user mode:

```
# shutdown -i6
```

The /usr/sbin/halt Command

The `halt` command performs an immediate shutdown. It does not execute the `rc0` kill scripts, it does not notify logged in users, and there is no delay.

To shut down the system to stop the Solaris Operating Environment and display the ok prompt:

```
# halt
```

The /usr/sbin/poweroff Command

The `poweroff` command performs an immediate shutdown. It does not execute the `rc0` kill scripts, no logged in users are notified, and there is no delay.

To shut down the system and turn off its power:

```
# poweroff
```

The /usr/sbin/reboot Command

The `reboot` command performs an immediate shutdown and brings the system to run level 3 by default. The `reboot` command differs from the `init 6` command because it does not execute the `rc0` kill scripts, and it does not notify logged in users.

To shutdown the system and then reboot to multi-user mode:

```
# reboot
```

Exercise: The Boot Process



Exercise objective – In this exercise you create a new startup script, make changes in `/etc/system`, and observe their effects.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

- In `/etc/init.d`, copy the `lp` script to a file called `banner`. Change the content of `banner` according to the instructions in step 3 of the tasks. Make `banner` executable and test that it works with the `start` and `stop` arguments. In `/etc/rc2.d`, create a hard link to `/etc/init.d/banner` called `S22banner`. In `/etc/rcS.d`, create a hard link to `/etc/init.d/banner` 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 `/etc/system`. Check if any instances of the `st` driver are loaded. Modify `/etc/system` to forceload the `st` driver. Reboot the system and verify that `st` driver instances are loaded.
- Edit `/etc/system` to exclude the main 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 to the `boot` command. Use your backup of `/etc/system` as required. Replace `/etc/system` with your backup when finished and reboot the system.

Tasks

1. Log in as `root` and open a terminal window. Change directory to `/etc/init.d`. Create a copy of the `lp` script and call it `banner`.

```
# cd /etc/init.d
# cp lp banner
```

2. In `/etc/init.d`, use `vi` to edit the `banner` script. Replace the line (line 16) that reads:

```
[ -f /usr/lib/lpsched ] && /usr/lib/lpsched
```

with a line that reads:

```
echo ""; /usr/bin/banner "SA-238 up"; echo""
```

Replace the line (line 20) that reads:

```
[ -f /usr/lib/lpshut ] && /usr/lib/lpshut
```

With a line that reads:

```
echo ""; /usr/bin/banner "SA-238 dwn"; echo""
```

3. Make the `banner` script executable and verify that it runs with both the `start` and `stop` arguments.

```
# chmod a+x banner
# ./banner start
# ./banner stop
```

4. Change directory to `/etc/rc2.d`. Create a hard link called `S22banner` that points to the same data as `/etc/init.d/banner`.

```
# cd /etc/rc2.d
# ln /etc/init.d/banner S22banner
```

5. Change directory to `/etc/rcS.d`. Create a hard link called `K99banner` that points to the same data as `/etc/init.d/banner`.

```
# cd /etc/rcS.d
# ln /etc/init.d/banner K99banner
```

6. Reboot the system and watch for the output of the script you just installed. Does the startup message from S22banner display?

```
# init 6
```

7. Login as `root` and open a terminal window. Use `init` to change to run level S. Does the shutdown message from K99banner display?

```
# init S
```

8. Enter the password for `root` in order to login at the command line. Change to run level 3.

```
# init 3
```

9. Login as `root` and open a terminal window. Change directory to `/etc`.

```
# cd /etc
```

10. Make a backup copy of `/etc/system` and name the back file `system.orig`.

```
# cp system system.orig
```

11. If your system uses a SCSI tape device, perform the following:

- a. Log in as `root` 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? _____

- b. Edit the `/etc/system` file so 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? _____

12. Edit `/etc/system` so 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
```

13. 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?

-
14. Use the `boot -a` command to boot the system and supply the name of your backup file called `etc/system.orig`. Enter carriage returns to accept the default values for all other boot parameters. 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>
```

15. Log in as `root` and open a terminal window. Copy `/etc/system.orig` to `/etc/system`. 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 exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

6. Reboot the system and watch for the output of the script you just installed. Does the startup message from S22banner display?

Yes.

7. Login as `root` and open a terminal window. Use `init` to change to run level S. Does the shutdown message from K99banner display?

Yes.

11. If your system uses a SCSI tape device, perform the following:

- a. Log in as `root` and open a terminal window. Use the `prtconf` command to list instances of the `st` driver currently loaded.

How many instances are reported? *None.*

- b. Log in as `root` and open a terminal window. Again list instances of the `st` driver currently loaded.

How many instances are reported? *The number will vary depending on how many SCSI controllers are present. You should see instances 0 through 6 for a system with one controller.*

12. Shut down the system to run level 0, and then attempt to boot it again.

What happened?

The system will be 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 alternate file for `/etc/system`.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Describe the four phases of the boot process
- ☐ Identify the directories that contain the kernel and its loadable modules
- ☐ Modify the kernel's configuration file
- ☐ Describe the eight Solaris Operating Environment run levels
- ☐ Define a system's current run level using the `who -r` command
- ☐ Explain the purpose of the `/etc/inittab` file
- ☐ Describe the steps in the `init` process to bring a system to multiuser mode
- ☐ List the directories that hold the run control scripts used to stop and start system processes and services.
- ☐ Describe the steps add a new run control script
- ☐ Use the following commands to shut down the system: `init`, `shutdown`, `halt`, `poweroff`, and `reboot`

Installing the Solaris 8 Operating Environment on a Standalone System

14 

Objectives

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

- State the different installation methods available for the Solaris 8 Operating Environment software
- Explain the hardware requirements for a Solaris 8 Operating Environment installation
- Identify the different Solaris 8 Operating Environment software CD-ROM editions
- List the five Solaris Software Groups
- Demonstrate how to install the Solaris 8 Operating Environment software on a networked, standalone system, using Solaris™ Web Start

The Solaris Operating Environment Software Installation Options

You can install the Solaris 8 software on a system using one of the following installation options:

- **Solaris Web Start 3.0 Installation** – Provides a graphical user interface-based, Java technology-powered software application that guides you through the installation of the Solaris Operating Environment and other software on a single system from a local or remote CD-ROM drive.
- **Solaris Interactive Installation Program** – Provides a graphical user interface that guides you step-by-step through installing the Solaris 8 Operating Environment software. This installation program does not enable you to install all the additional software, as with Solaris Web Start, it installs only the Solaris 8 Operating Environment software.
- **Solaris Installation Over the Network** – Provides the capability to install the Solaris Operating Environment software on a large number of systems that do not have a local CD-ROM drive. This eliminates the need to insert the Solaris 8 Operating Environment software CD-ROM on every system. You can install these systems from the remote Solaris 8 Operating Environment software CD images, which have been copied to an install server system's hard drive.
- **Solaris JumpStart Installation** – Provides the capability to automatically install the Solaris 8 Operating Environment software on a new system only, by inserting the CD labeled Solaris 8 Software 1 of 2 SPARC Platform Edition or Intel Platform Edition into the CD-ROM drive and turning on the system. The software components installed are specified by a default profile that is selected based on the model and disk size of the system.
- **Solaris Custom JumpStart Installation** – A type of installation in which the Solaris 8 Operating Environment software is automatically installed on a system based on a user-defined profile. You can customize profiles for different types of users and systems, and this is the most cost-effective option for installing the

Solaris Operating Environment software in a large enterprise. Provides a hands off installation across the network based on a central configured server.

Note – This module describes how to install the Solaris Operating Environment software on a single system with Solaris Web Start, Sun's graphical wizard, Java technology-powered software installation application.

Hardware Requirements of a Solaris 8 Operating Environment Installation

A desktop Solaris 8 Operating Environment installation requires:

- A SPARC-based or an Intel-based system
- 64 Mbytes of memory
- 2.3 Gbytes of disk space
- Access to a CD-ROM drive

The Solaris 8 Operating Environment Installation CD-ROM

The content of each CD-ROM in the Solaris 8 Operating Environment Media kit is as follows:

The Solaris 8 Operating Environment SPARC Platform Edition CD-ROM

- Solaris 8 Installation English SPARC Platform Edition
- Solaris 8 Software CD 1 of 2 SPARC Platform Edition
- Solaris 8 Software CD 2 of 2 SPARC Platform Edition
- Solaris 8 Documentation CD (English SPARC/Intel Platform Edition)

International Versions of the Solaris 8 Operating Environment

International versions of Solaris 8 contain:

- Solaris 8 Installation Multilingual CD SPARC Platform Edition
- Solaris 8 Software CD 1 of 2 SPARC Platform Edition
- Solaris 8 Software CD 2 of 2 SPARC Platform Edition
- Solaris 8 Languages CD - SPARC Platform Edition or Intel Platform Edition

International versions also include a two CD-ROM set labeled:

- Solaris 8 Documentation European SPARC/Intel Platform Edition, which contains English, French, German, Italian, Spanish, and Swedish documentation.
- Solaris 8 Documentation Asian SPARC/Intel Platform Edition, which contains Simplified and Traditional Chinese, Japanese, and Korean documentation.

Intel Versions of the Solaris 8 Operating Environment

An equivalent CD-ROM set is included with the Solaris 8 Intel Platform Edition, plus a diskette labeled Solaris 8 Device Configuration Assistant Intel Platform Edition.

Choosing the Correct CD for Your Installation Requirements

The following describes which CD-ROM is required when installing Solaris 8 using the different installation methods:

- Solaris Web Start uses the following CD-ROM set:
 - ▼ Solaris 8 Installation
 - ▼ Solaris 8 Software 1 of 2
 - ▼ Solaris 8 Software 2 of 2

All the other installation methods described earlier, use the Solaris 8 Software 1 of 2 and Solaris 8 Software 2 of 2 CD-ROM set.

The Solaris Operating Environment Software Arrangement

The Solaris Operating Environment software delivered on the Solaris 8 Software CD-ROM set

1 of 2 and 2 of 2 are organized into three types of components:

- Software Packages
- Software Clusters
- Cluster Configurations

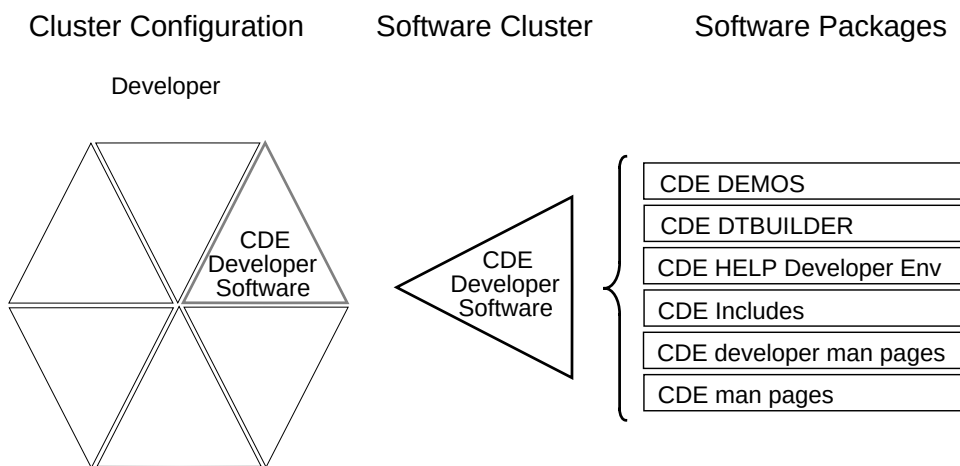


Figure 14-1 Solaris Operating Environment Software Components

Software Packages

A *software package* contains a group of files and directories in a category of related software (for example, system or application) and software installation scripts.

Software Clusters

During the software installation process, logical collections of software packages are grouped into *software clusters*. For example, the CDE software cluster includes the following packages:

SUNWdtab	SUNWdthed	SUNWdtmad	SUNWeudhr
SUNWdtbas	SUNWdthev	SUNWdtrme	SUNWeudhs
SUNWdtdem	SUNWdticn	SUNWdtwn	SUNWeudis
SUNWdtdmn	SUNWdtim	SUNWeudba	SUNWeudlg
SUNWdtdst	SUNWdtinc	SUNWeudbd	SUNWmfman
SUNWdtthe	SUNWdtma	SUNWeudda	

Some software clusters can contain only one software package.

Cluster Configurations

The *cluster configurations* are referred to during the installation process as the Solaris Software Groups. There are currently five software groups available, which include:

- Entire Solaris Software Group Plus OEM - SUNWCXall
- Entire Solaris Software Group - SUNWCa11
- Developer Solaris Software Group - SUNWCprog
- End User Solaris Software Group - SUNWCuser
- Core Solaris Software Group - SUNWCreq

The Solaris Operating Environment Software Groups

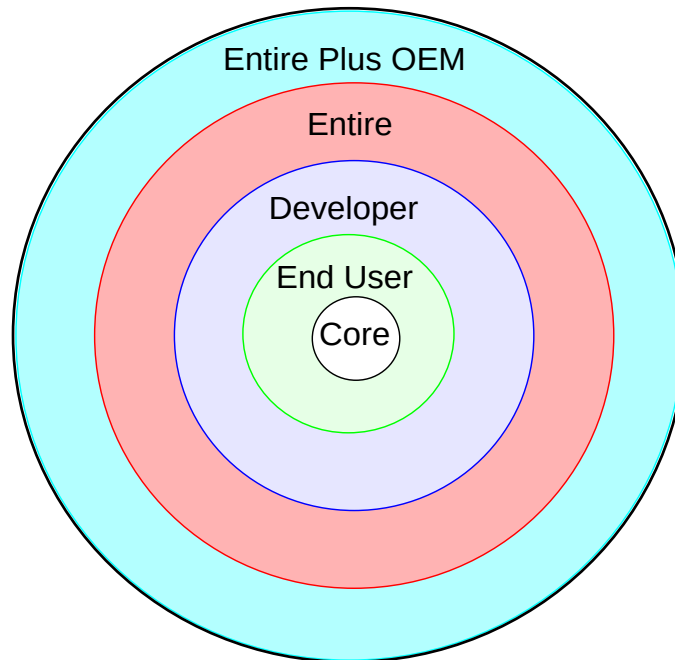


Figure 14-2 Solaris Operating Environment Software Groups

Core

Core is a software group that contains the minimum software required to boot and run the Solaris Operating Environment on a system. It includes some networking software and the drivers required to run the Common Desktop Environment (CDE) or OpenWindows desktop. It does not include the CDE or OpenWindows software.

Note – Approximate disk space requirement for Core is 718 Mbytes.

End User System Support

The End User System Support is a software group that contains the Core software group plus the recommended software for an end user, including OpenWindows or CDE and DeskSet software.

Note – Approximate disk space requirement for End User is 1.2 Gbytes.

Developer System Support

The Developer System Support is a software group that contains the End User System Support software group plus the libraries, include files, man pages, and programming tools for developing software.

Note – Approximate disk space requirement for Developer is 1.5 Gbytes.

Entire Distribution

The Entire Distribution is a software group that contains the entire Solaris 8 Operating Environment software release.

Note – Approximate disk space requirement for Entire Distribution is 1.9 Gbytes.

Entire Distribution Plus OEM Support

The Entire Distribution Plus OEM Support is a software group that contains the entire Solaris 8 Operating Environment software release, plus additional hardware support for Original Equipment Manufacturers (OEMs). This software group is recommended when installing the Solaris Operating Environment software on SPARC-based servers.

Note – Approximate disk space requirement for Entire Distribution Plus OEM is 2.1 Gbytes.

Planning an Installation on a Standalone System

The following installation procedures describe how to run Solaris Web Start to install the Solaris 8 Operating Environment software on a networked, standalone system from a local CD-ROM drive.

You can run Solaris Web Start in either of two ways:

- Graphical User Interface (GUI) – This requires a local or remote CD-ROM drive or network connection, frame buffer, keyboard, and monitor.
- Command Line Interface (CLI) – This requires a local or remote CD-ROM drive or network connection, keyboard, and monitor.

If Solaris Web Start detects a frame buffer for the system, it uses the GUI, if it does not it uses the CLI. The content and sequence of instructions in both are generally the same.

Note – You can select the Solaris Web Start's upgrade option during installation if the system is currently running Solaris 7 Operating Environment software. However, if the system is currently running the Solaris 2.5.1 or Solaris 2.6 Operating Environments, you must run an Interactive Installation to perform a Solaris 8 Operating Environment upgrade.

Pre-Installation Information

Before installing the Solaris Operating Environment software on a networked standalone system, you must provide the following information:

- Host name – A unique, commonly short name for the system. You can use the command `uname -n` command to determine the host name on an existing system.
- Host IP address – A software address representing the host address and network address.
- Name service type – Determine if the networked system is to be included in one of the following types of name service domains: NIS, NIS+, Other, or None.
- Subnet mask – Determine if the networked system is to be included in a particular subnet. The subnets 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 bandwidth to each host.

- Geographic location and time zone – A specific region where the system physically resides.
- Root password – A password assigned to `root` to gain access and `root` privileges on the system.
- Language – Determine the language to be used to install the Solaris Operating Environment. Use the CD labeled:
 - ▼ Solaris 8 Installation English SPARC Platform Edition – All prompts, messages, and other installation information is displayed in English only.

- ▼ Solaris 8 Installation Multilingual SPARC Platform Edition – Select a language in which to display prompts, messages, and other installation information:
 - Simplified Chinese
 - Traditional Chinese
 - English
 - French
 - German
 - Italian
 - Japanese
 - Korean
 - Spanish
 - Swedish

The last step in the pre-installation process is to make sure the following Solaris 8 CD-ROM set is available:

- Solaris 8 Installation English SPARC Platform Edition or Solaris 8 Installation Multilingual SPARC Platform Edition.
- Solaris 8 Software 1 of 2 SPARC Platform Edition and Solaris 8 Software 2 of 2 SPARC Platform Edition.
- Solaris 8 Languages SPARC Platform Edition (if using the Multilingual CD).

Note – Before a software installation, always back up any modifications or data that exist in the previous version of the Solaris Operating Environment, and restore them after the installation is complete.

Software Installation Using Solaris Web Start

A few seconds after the reboot is complete, the Welcome screen is displayed:



Figure 14-3 Localized Welcome Message

On the Welcome screen, click on Next to continue.

- Solaris Web Start generates a series of screens used to gather information about the system, as follows:

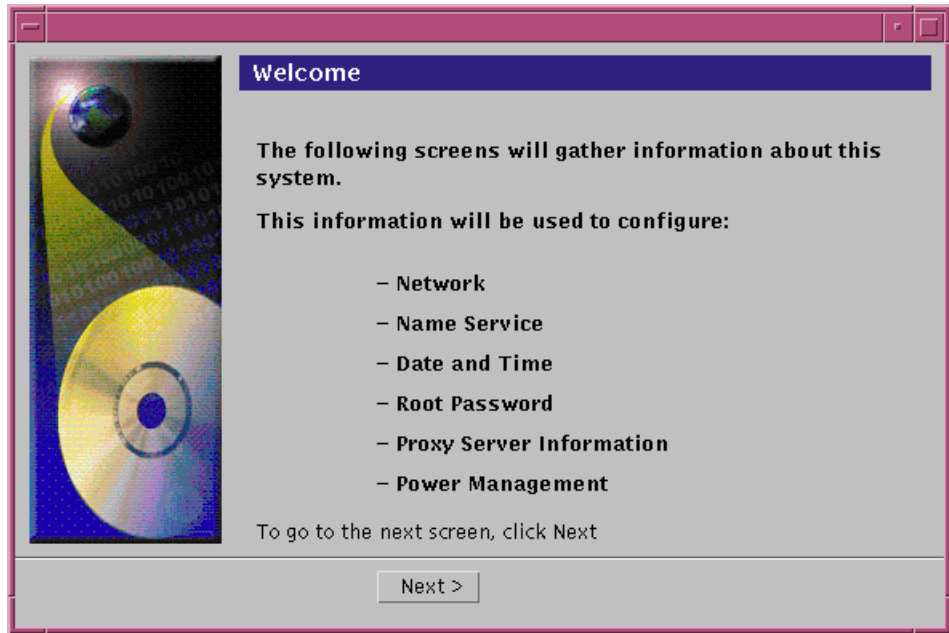
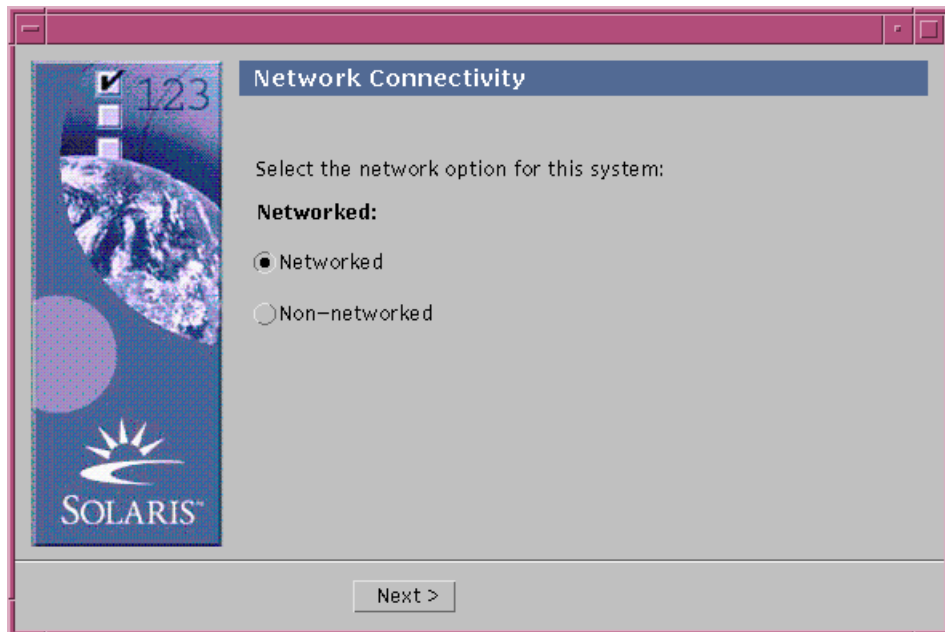


Figure 14-4 Preconfiguration Checklist

- The Network Connectivity dialog box:

**Figure 14-5** Network Connectivity Dialog Box

Make the appropriate selection depending on if the system will be networked or non-networked.

- The DHCP dialog box.

DHCP (Dynamic Host Configuration Protocol) is an application-layer protocol that enables individual computers, or clients, on a TCP/IP network to extract an IP address and other network configuration information from a designated and centrally maintained DHCP server or servers.

DHCP reduces the overhead of maintaining and administering a large IP network.

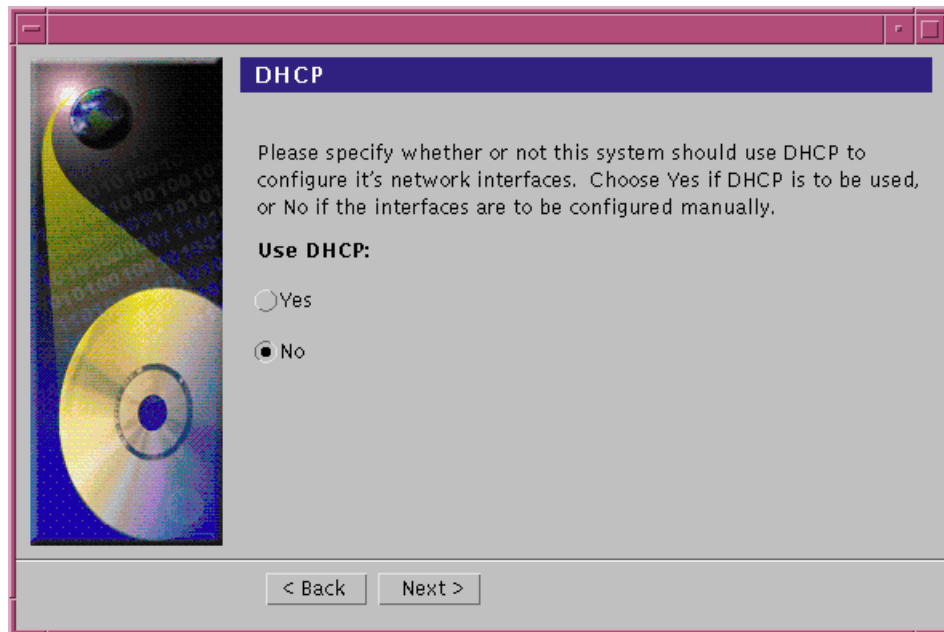


Figure 14-6 DHCP Dialog Box

Determine if the system will use DHCP to configure its network interfaces, or if the interfaces will be configured manually.

- The Host Name dialog box.

The host name consists of 2–255 characters, and it should be unique to each host on the network.

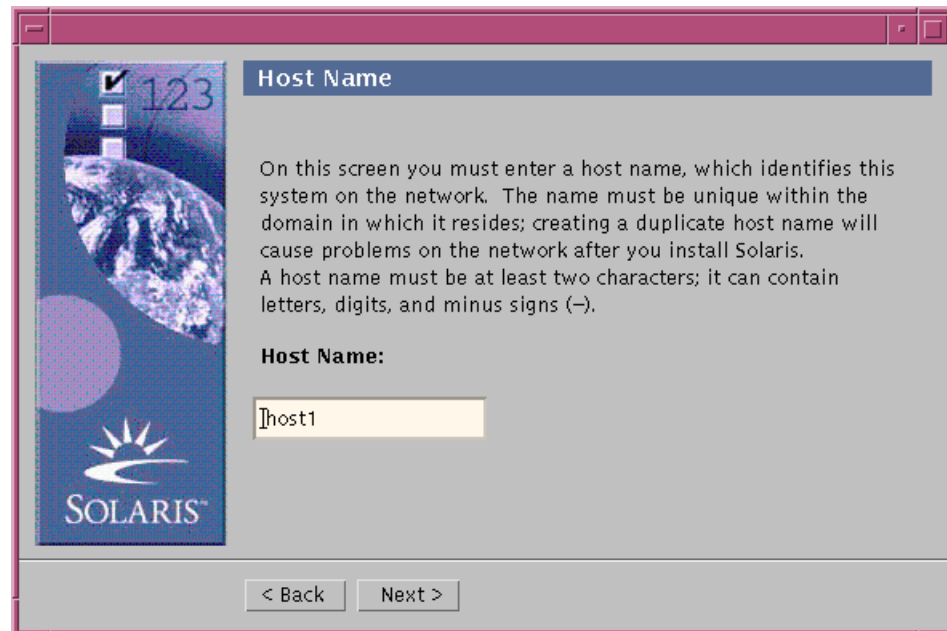


Figure 14-7 Host Name Dialog Box

- The IP Address dialog box.

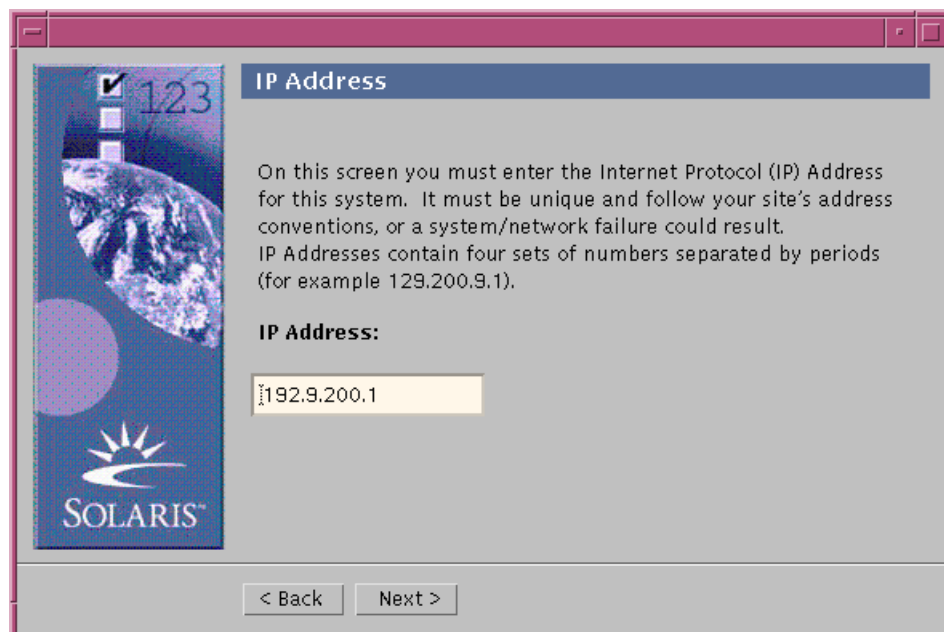


Figure 14-8 IP Address Dialog Box

- The Netmask dialog box.

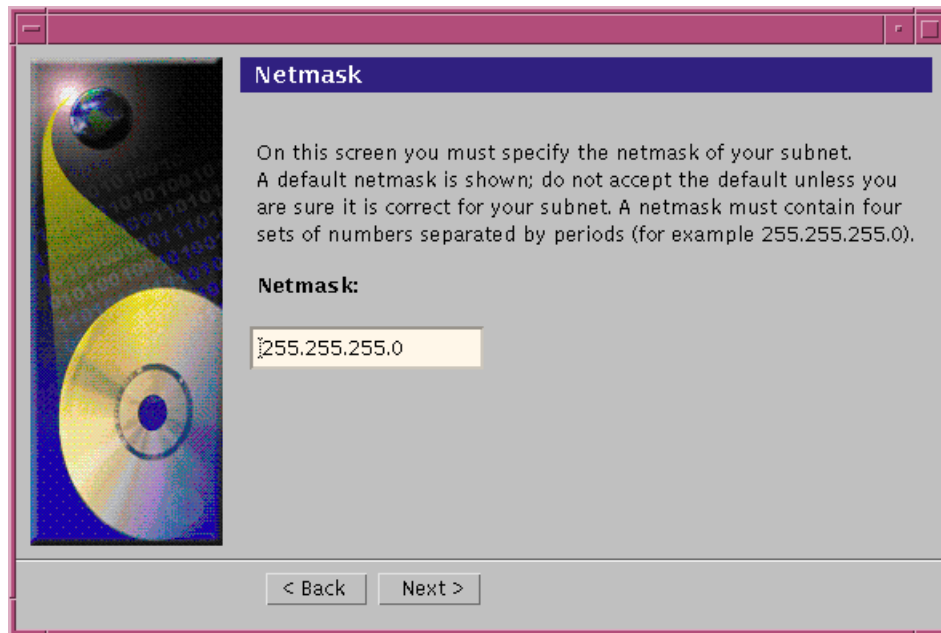


Figure 14-9 Netmask Dialog Box

- The IPv6 dialog box.

IPv6 is a new version (version 6) of Internet Protocol (IP) designed to be an evolutionary step from the current version, IPv4 (version 4).

It is an increment to IPv4. Deploying IPv6, using defined transition mechanisms, does not disrupt current operations. In addition, IPv6 provides a platform for new Internet functionality.

Determine if IPv6 will be enabled for this system.

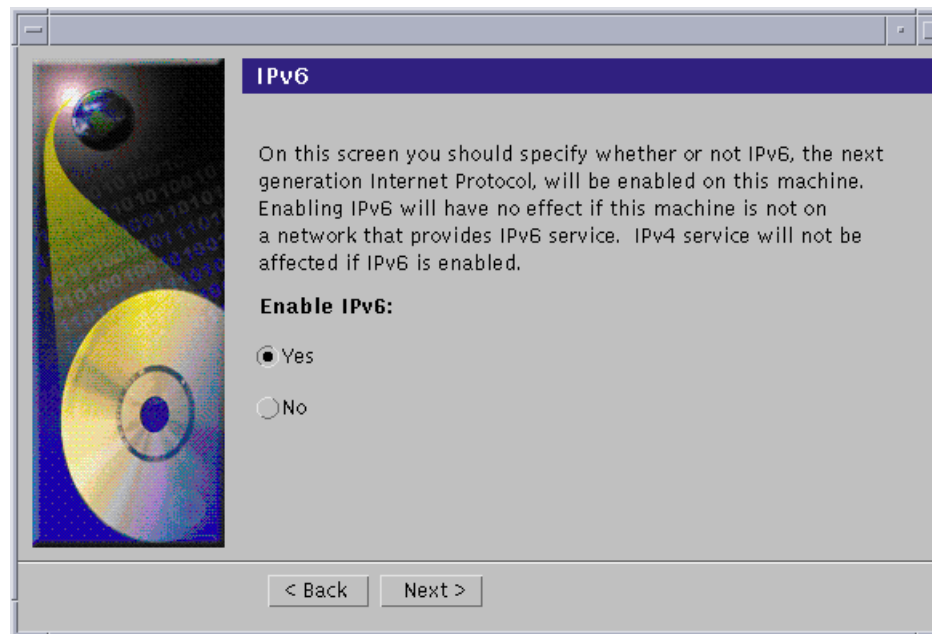


Figure 14-10 IPv6 Dialog Box

- The Name Service dialog box.

The name service concept centralizes the shared information in your network. A single machine, the name server, maintains the information previously maintained on each individual host.

The name servers provide such information as host names and IP addresses, user names, passwords, and automount maps.

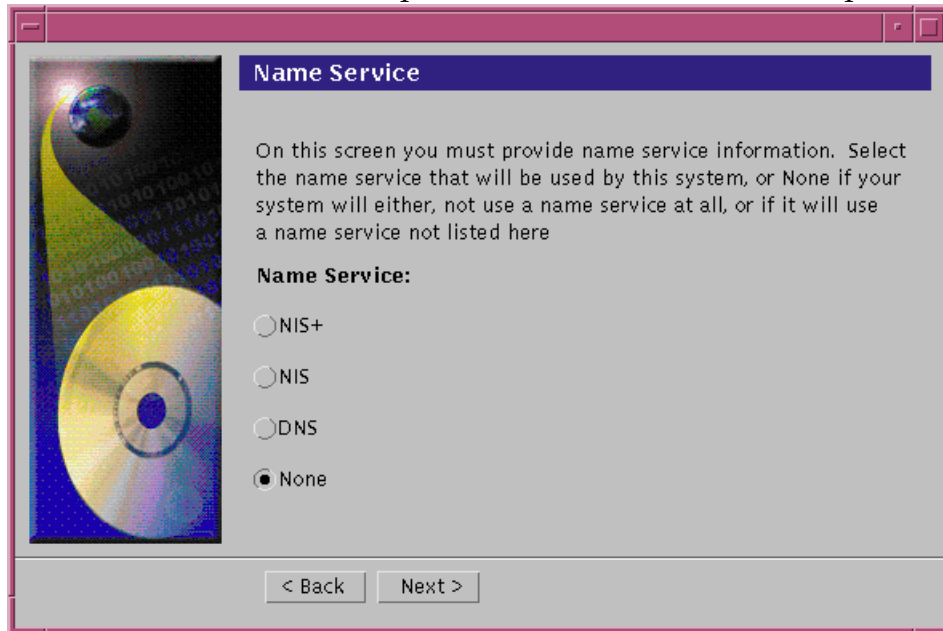


Figure 14-11 Name Service Dialog Box

Determine if the system will be using a name service.

- The Time Zone dialog box.

In the Time Zone dialog box, select the appropriate option to set the default time zone:

- ▼ Geographic Region
- ▼ Offset From GMT
- ▼ Time Zone File

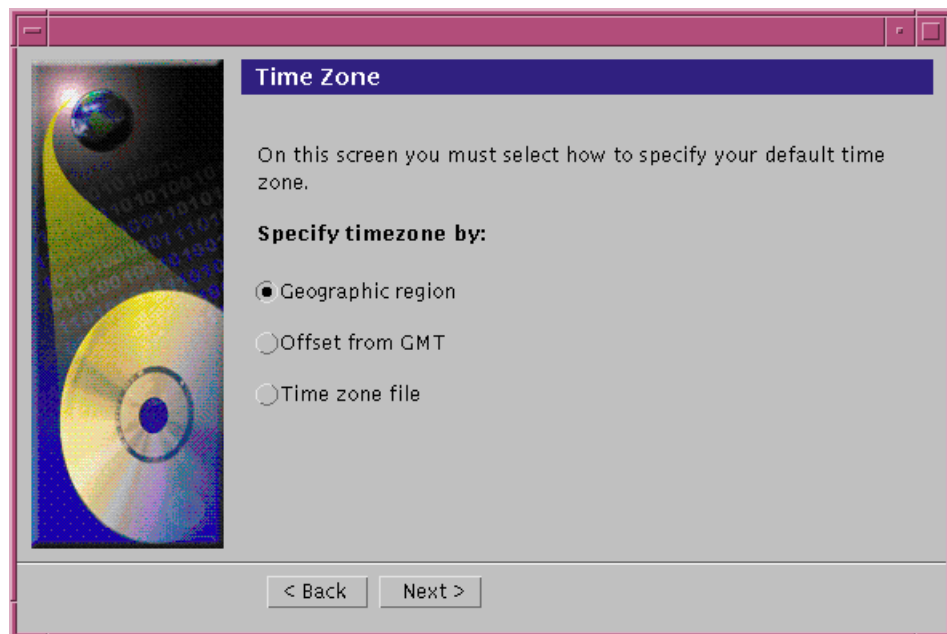


Figure 14-12 Time Zone Dialog Box

- The Date and Time dialog box.

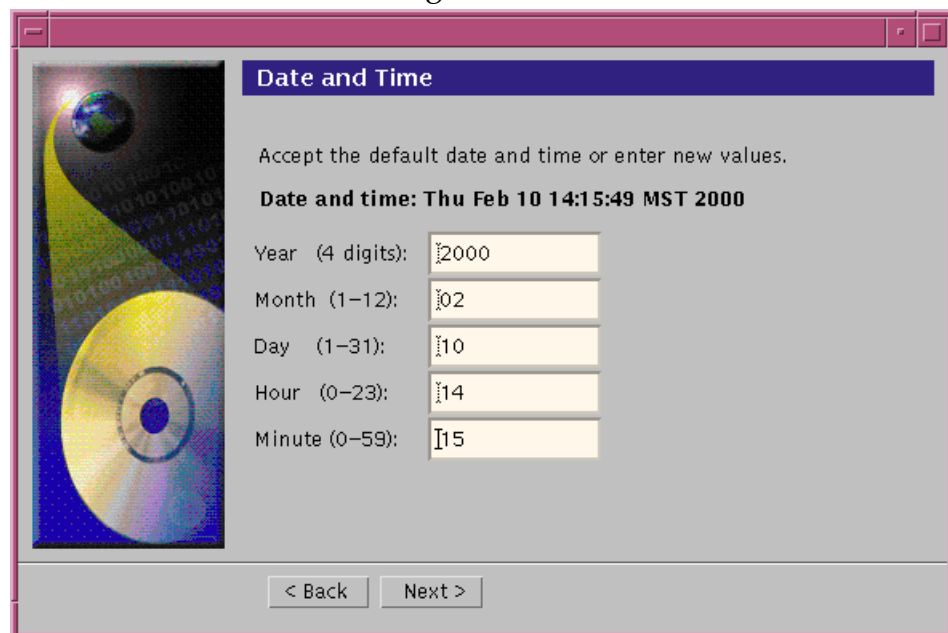


Figure 14-13 Date and Time Dialog Box

- The Root Password dialog box is displayed.

Note – You do not require a `root` password in this dialog box to continue.

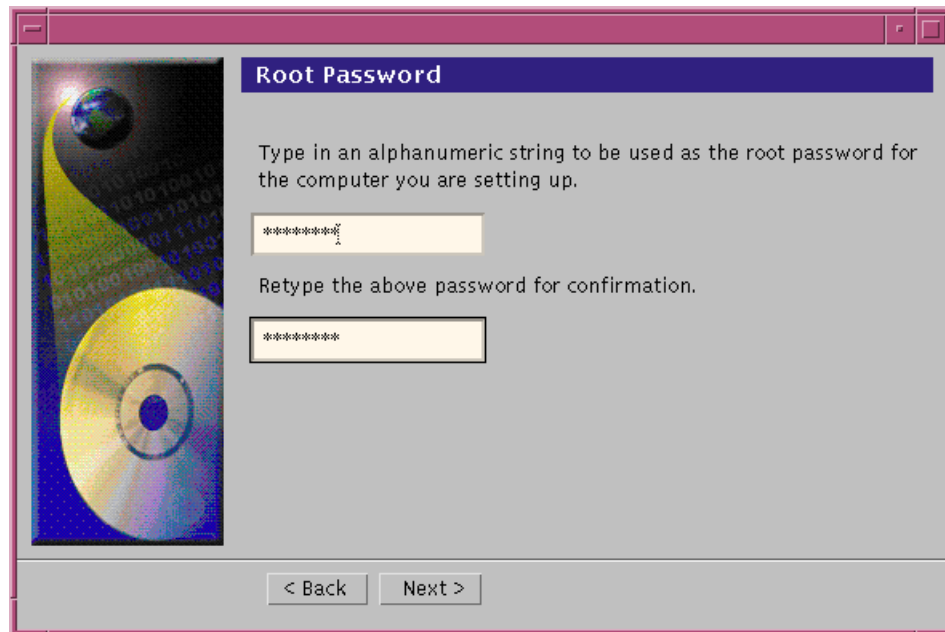


Figure 14-14 Root Password Dialog Box

- The Power Management™ dialog box.

Power Management software automatically saves the system state and turns the system off when it is idle for 30 minutes

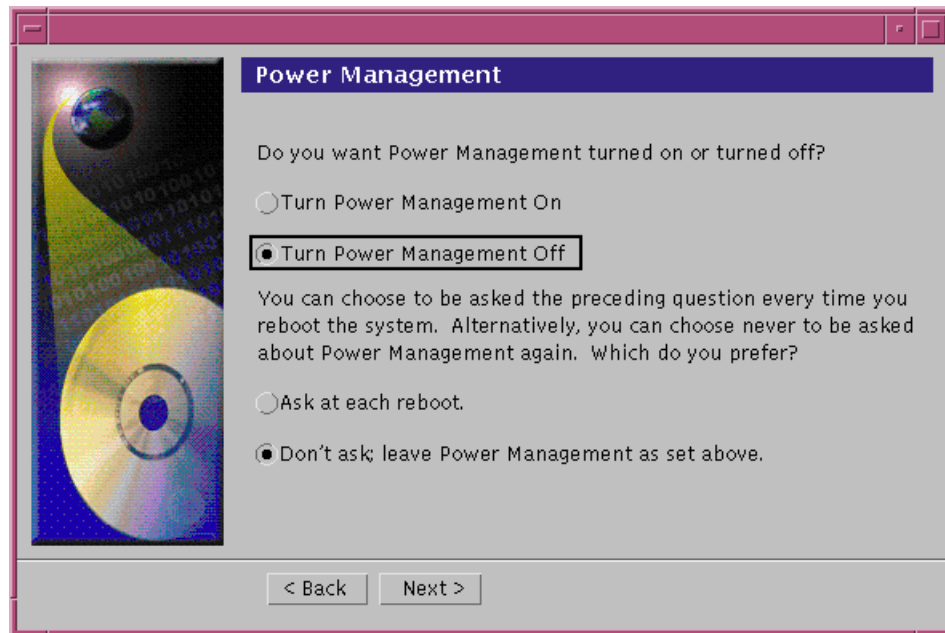


Figure 14-15 Power Management Dialog Box

- The Proxy Server Configuration dialog box.

A proxy server acts as an intermediary between a workstation user and the Internet to ensure security, administrative control, and caching service.

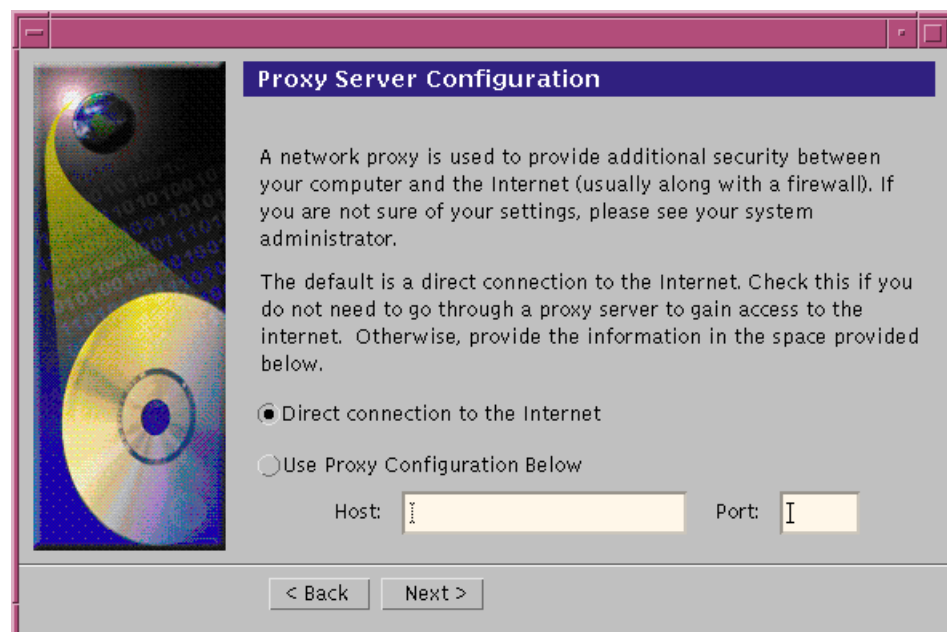


Figure 14-16 Proxy Server Configuration Dialog Box

- The Confirm Information dialog box. The Confirm Summary provides a display to check the list of responses that you have supplied to the Web Start utility.

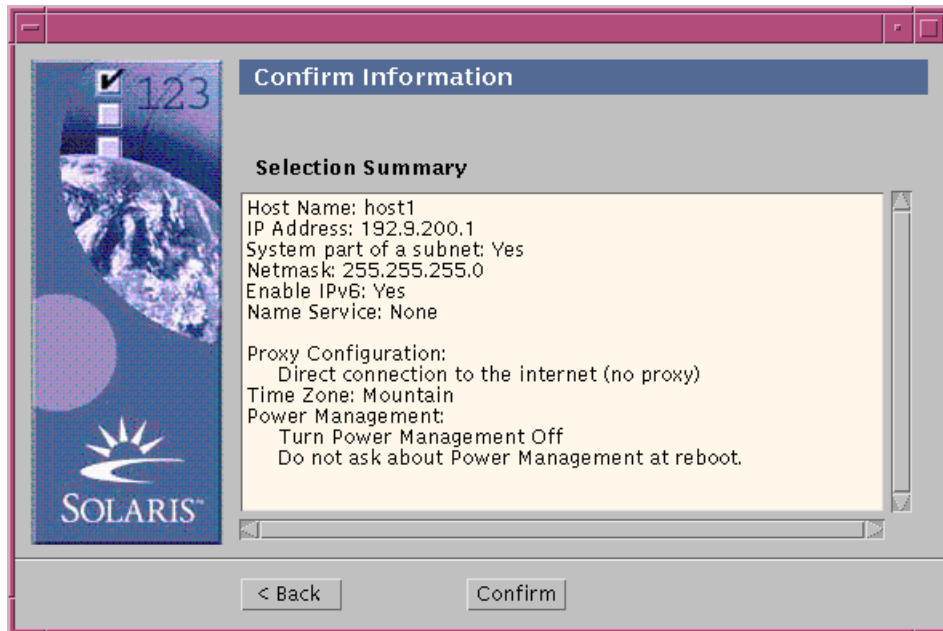


Figure 14-17 Confirm Information Dialog Box

The following message is displayed in the Confirm Information dialog box.

Please wait while the system is configured with your settings...

After a few minutes the Extracting dialog box will display, followed by the Solaris Web Start Installation Kiosk and Welcome to Solaris dialog box:



Figure 14-18 Solaris Web Start Installation Kiosk

The Kiosk is a browser-based environment in which information, such as documentation, web pages, and other content, is displayed as you install the Solaris Operating Environment software with Solaris Web Start.

Note – In some cases, the Kiosk might obscure a dialog box. To display an obscured dialog box, on the Kiosk menu, click on Send Kiosk to Background.

Installing the Solaris 8 Operating Environment

- The Solaris 8 Installation English SPARC Platform Edition or Solaris 8 Installation Multilingual SPARC Platform Edition CD is ejected and the Insert CD dialog box.

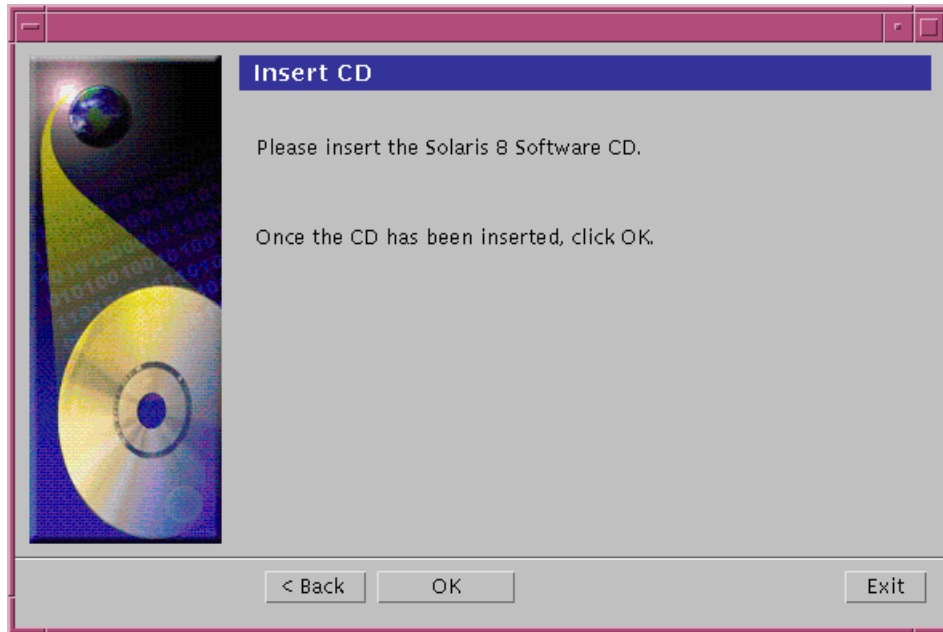


Figure 14-19 Insert CD Dialog Box

- The Reading CD and Initializing dialog boxes are displayed. After a few minutes the system is initialized and the Select Type of Install dialog box is displayed.

The default installation leads you through a generic installation process, providing default answers to all the configuration options.

However, if you have special disk or application requirements, you can select the custom installation.

The custom installation requires more in-depth knowledge of the installation process. You are prompted to make some decisions, such as: which disk(s) you want to install on, and how you want to slice up the disk(s)?

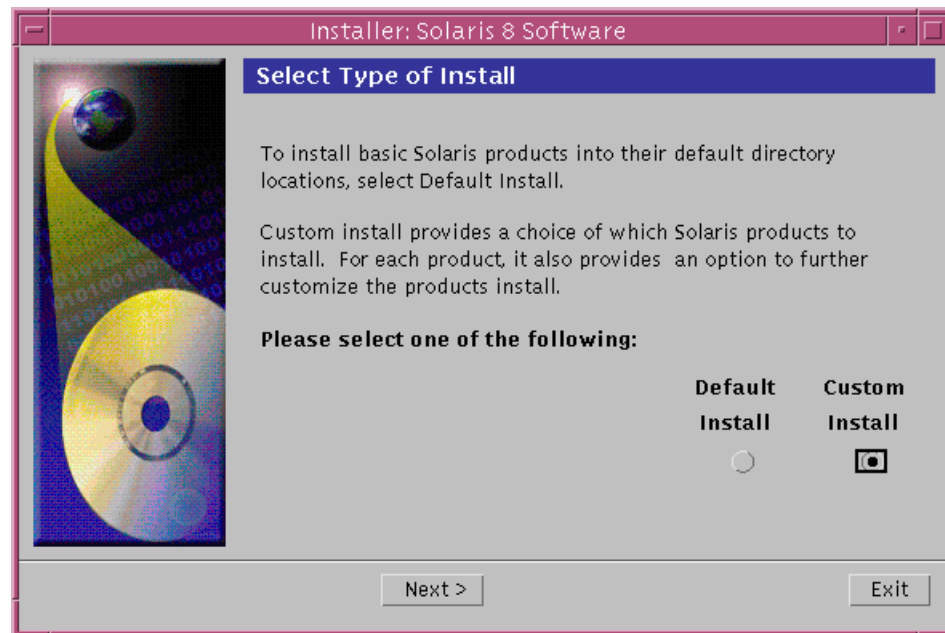


Figure 14-20 Select Type of Install Dialog Box

- The Select Software Localizations dialog box.

This screen allows you to select the geographic regions and localizations to be installed, if any, in addition to English.

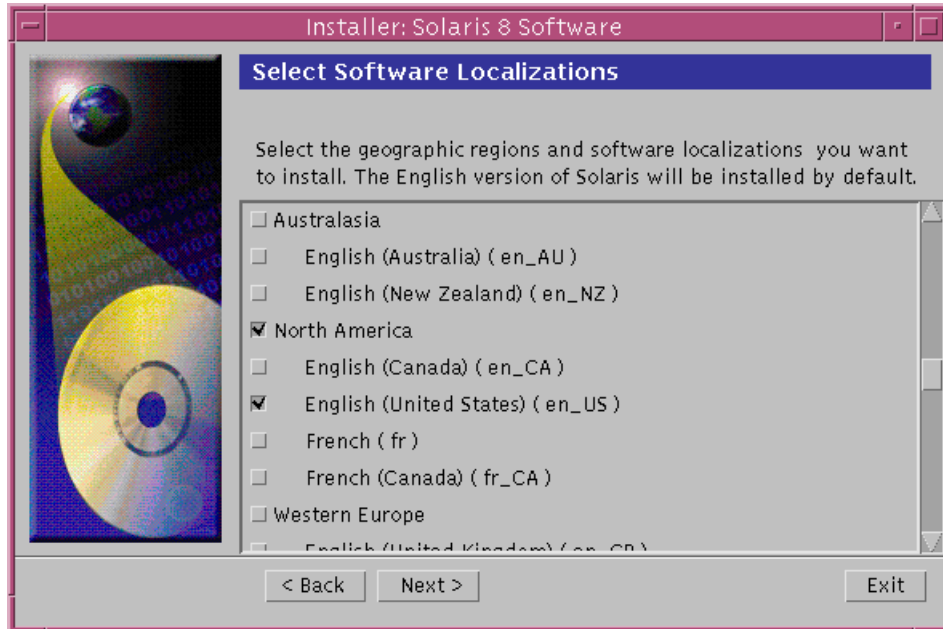


Figure 14-21 Select Software Localizations Dialog Box

- The Select System Locale dialog box.

This screen allows selection of a more definitive language localization relative to any languages that were previously selected.

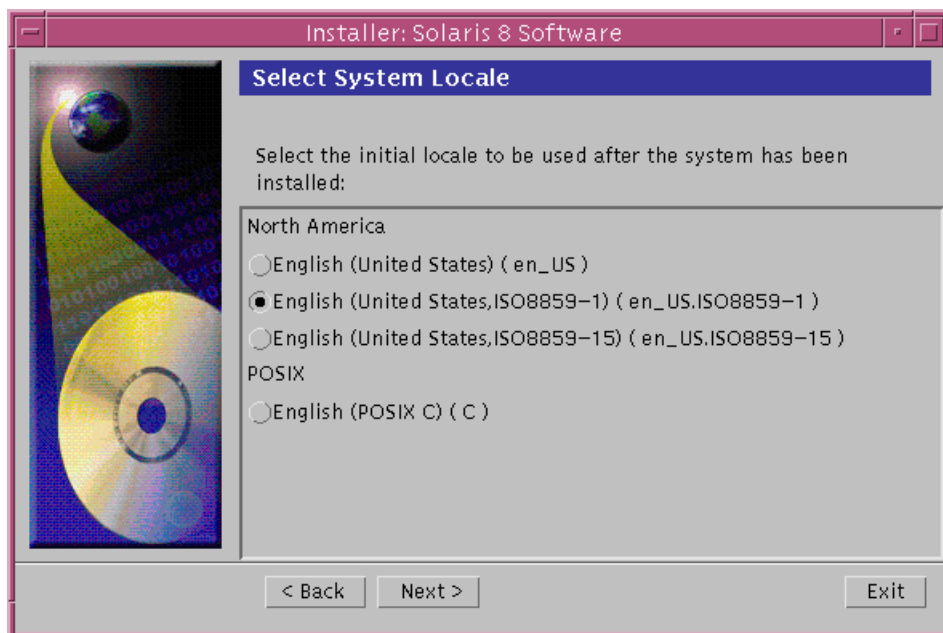


Figure 14-22 Select System Locale Dialog Box

Note – English (United States, en_US) is selected by default.

- The Select Products dialog box.

The additional software products listed here can be installed, in addition to the Solaris 8 software. For example:

- ▼ Solaris 8 Documentation CD – Contains the Solaris AnswerBook2™ server and the Solaris 8 Operating Environment collection.
- ▼ Solaris 8 Software 2 of 2 CD – Contains the Solaris 8 Early Access Software.
- ▼ Computer Systems Supplement CD– Contains value added software and documentation from Sun Microsystems Computer Systems.

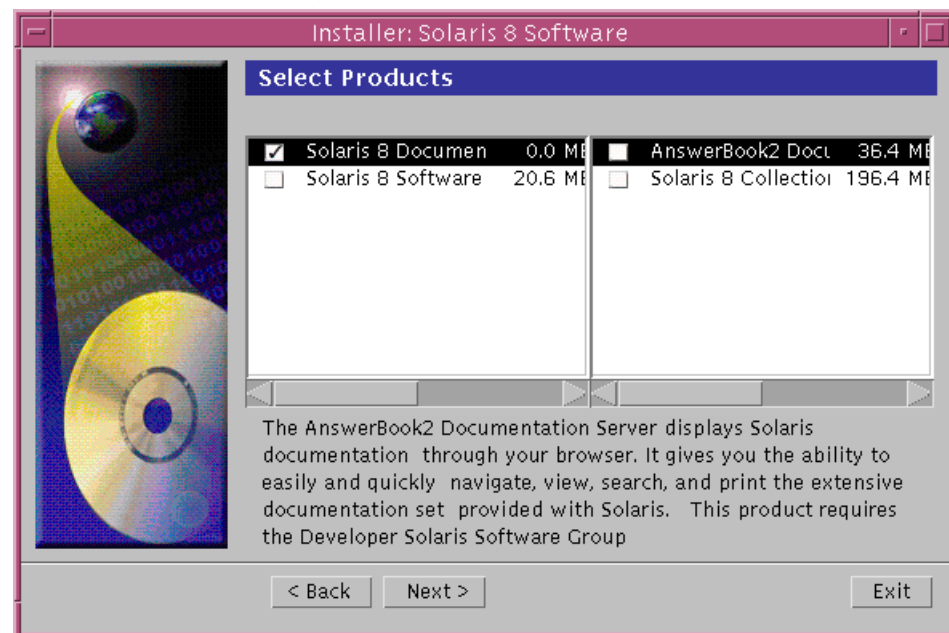


Figure 14-23 Select Products Dialog Box

Note – A description of each product is displayed when selected, by clicking on the appropriate box located in front of the product name.

- The Additional Products dialog box.

If you are installing additional third-party or other software products, select Product CD, Kiosk Download, or Local or Network File system, and click on Next.

- ▼ If you selected the Product CD, the Solaris 8 Software 1 of 2 SPARC Platform Edition CD is ejected and the Insert CD dialog box is displayed.
- ▼ If you selected the Kiosk Download, the Scanning dialog box is displayed. When Solaris Web Start is done scanning the Kiosk download area (`/webstart/kiosk/download`), the Select Products dialog box is displayed. The products that you can install with Solaris Web Start are listed in the windows.

- ▼ If you selected Local or Network File system, the Specify Network File system Path dialog box is displayed. You can enter the path to the file system, and the Scanning dialog box is displayed. When Solaris Web Start is done scanning the file system, the Select Products dialog box is displayed. The products that you can install with Solaris Web Start are listed in the window.

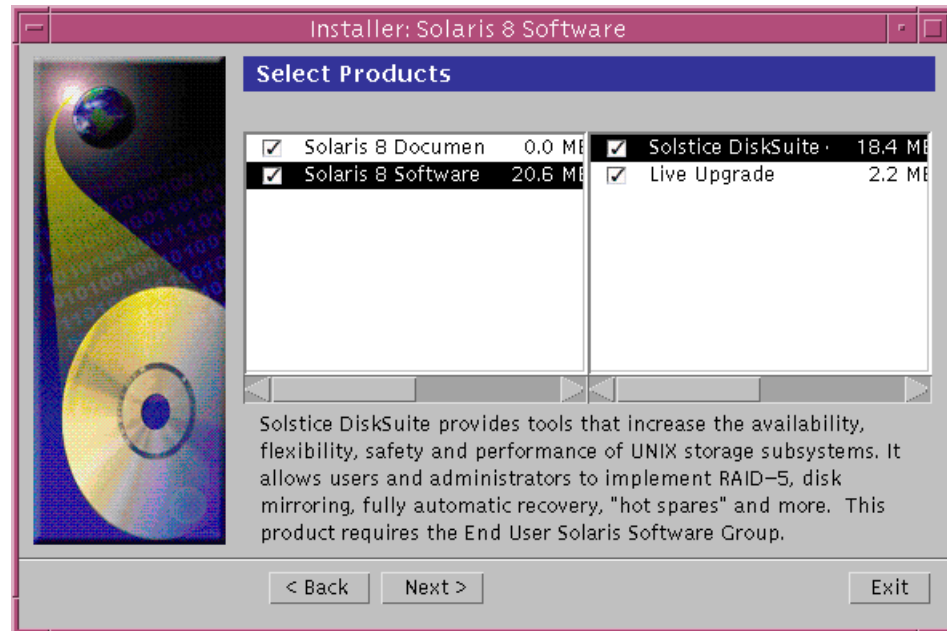


Figure 14-24 Additional Products dialog box

- The 64-Bit Selection dialog box.

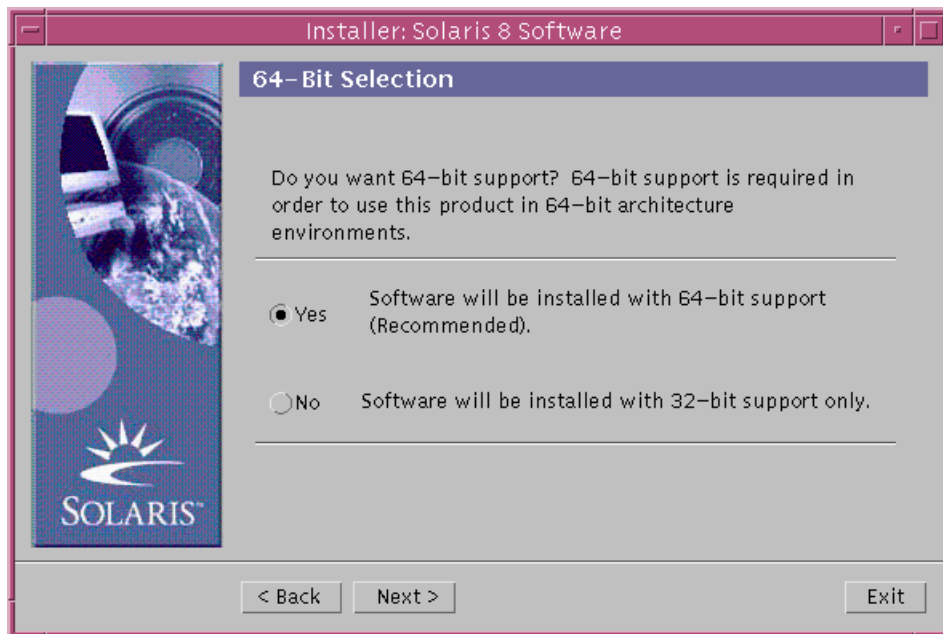


Figure 14-25 64-Bit Selection Dialog Box

- The Select Solaris Cluster Configuration dialog box.

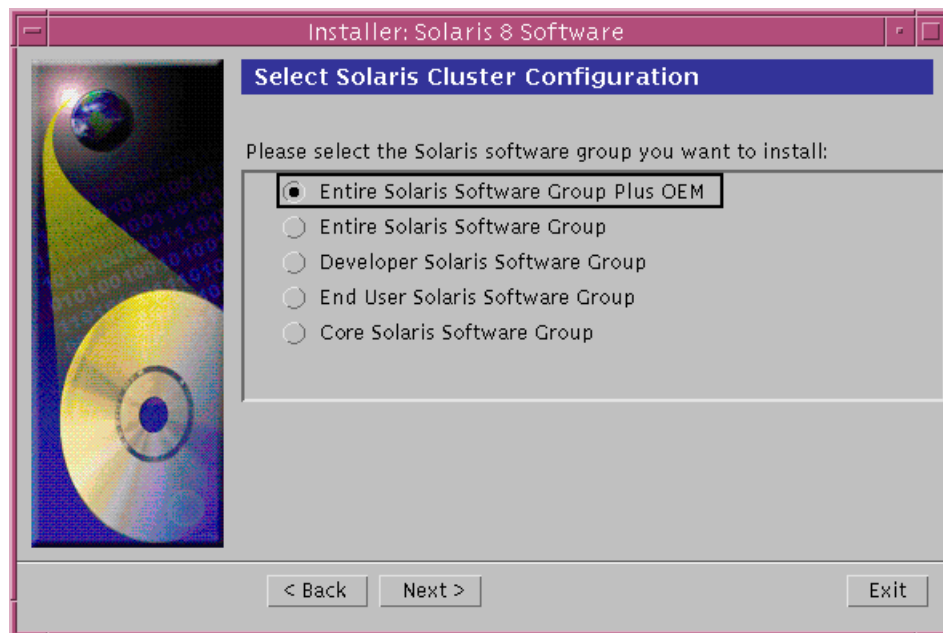


Figure 14-26 Select Solaris Cluster Configuration Dialog Box

Select the Solaris Operating Environment software group to be installed.

- The Disk Selection dialog box.

The window on the left indicates all the disk drives known by the system. The window on the right contains the default boot disk, by default.

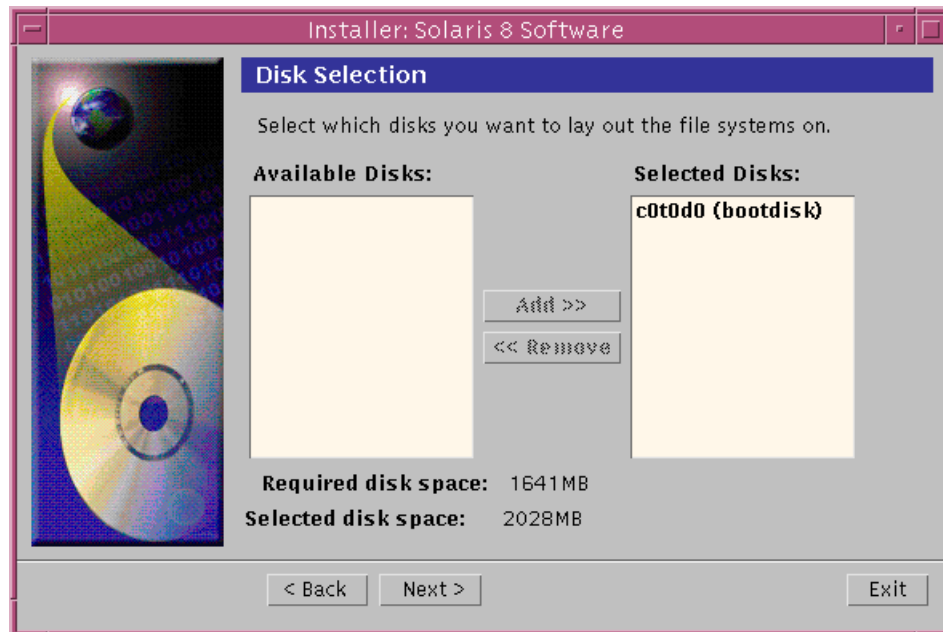


Figure 14-27 Disk Selection Dialog Box

Any disk that appears, or is moved into the window on the right will be affected by the Solaris Operating Environment software installation. Any disks remaining in the window on the left are ignored by the installation process.

- After a few seconds, the Gathering Disk Space Requirements dialog box is displayed, followed by the Lay Out File systems dialog box.

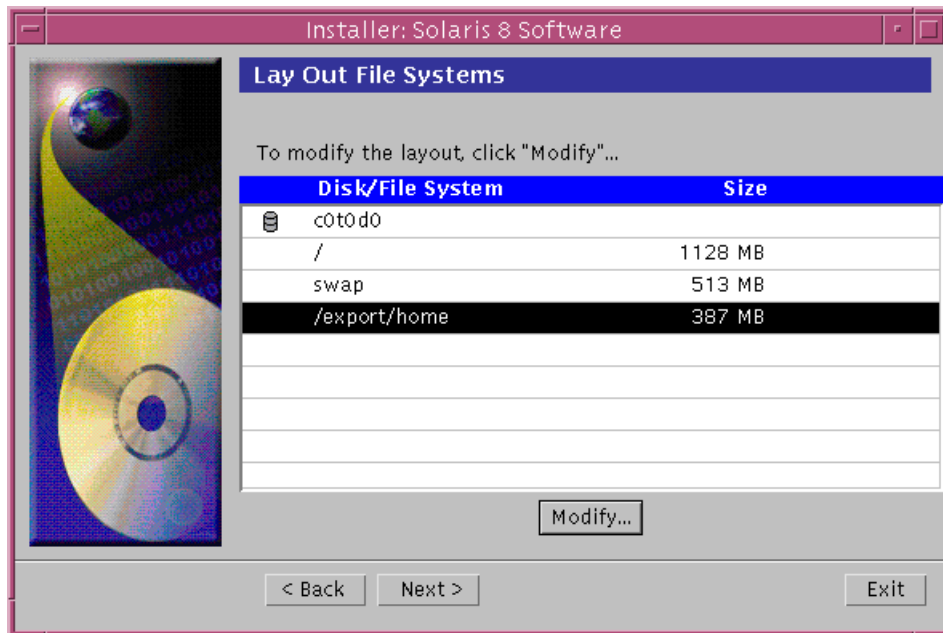


Figure 14-28 Lay Out File Systems Dialog Box

You can modify a default file system or add a new file system on a disk by selecting a disk, or a file system and clicking on the Modify option. The Disk dialog box is displayed.

The Disk dialog box provides an easy method for creating, renaming, moving, deleting, expanding, or shrinking disk slices or filesystems for each disk selected.

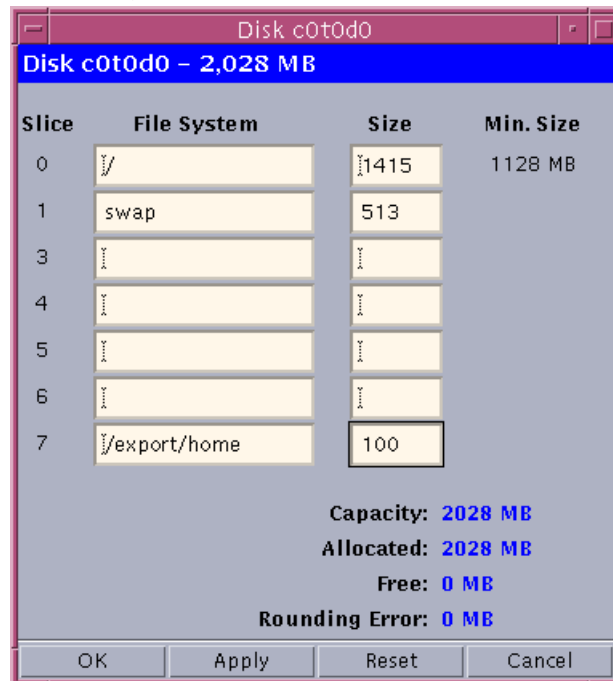


Figure 14-29 Disk Dialog Box

The default file systems that are created during the installation process include:

- ▼ Root on disk slice 0
- ▼ Swap on disk slice 1
- ▼ /export/home on disk slice 7

Note – You cannot change the size of the default swap slice in the Disk dialog box.

To add a new file system, or increase the size of a file system on a particular disk slice, you must first take away (or reduce) space from an existing slice.

By reducing the size of an existing slice, the excess space is displayed as free space at the bottom of the Disk dialog box.

This free space is then available for creating or increasing the size of other file systems.

When the entire disk space is allocated (or used), the free space value equals zero (0).

Note – Sometimes, a rounding error is displayed at the bottom of the Disk dialog box when allocating the last disk slice. This rounding error is less than 1 Mbyte, and it is necessary to ensure that the disk slice is made up of a whole number of cylinders.

If all free space is not used, an error message is displayed as a warning. If this warning occurred because of the rounding error, you can ignore it and continue the installation.

- The Ready to Install dialog box is displayed.

This screen displays a summary of your installation selections, including a list of additional products with their disk space allocation requirements.



Figure 14-30 Ready to Install Dialog Box

- The Installing dialog box.

Status messages and the name of each package that is added to the Solaris software are displayed in brackets above the progress bars.

The status of the entire installation is shown in the bottom progress bar.

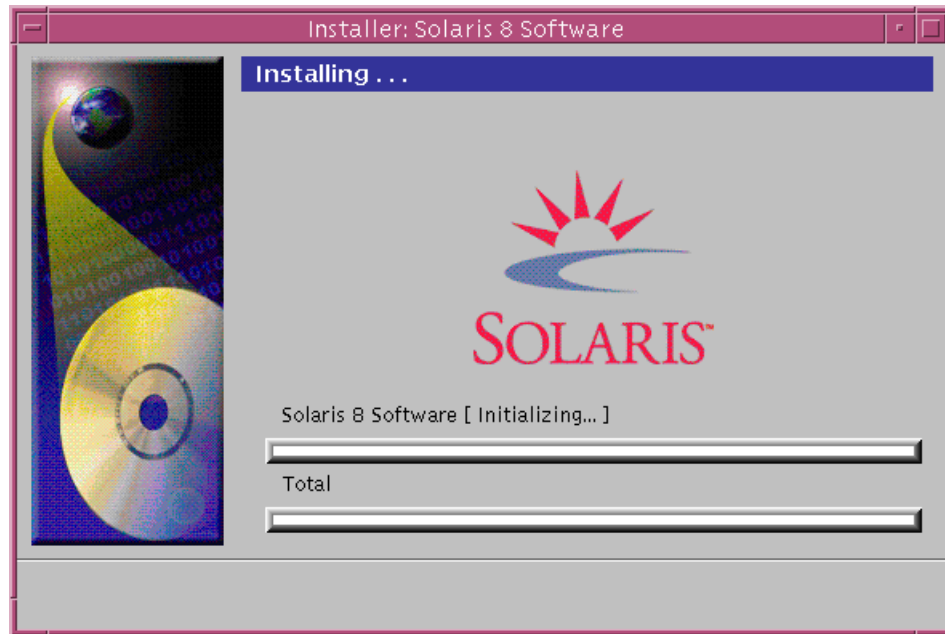


Figure 14-31 Installing Dialog Box

Note – This part of the installation can take a while, but the actual time depends on the software group that is being installed, the reallocation of any space if needed, and the speed of the network or local CD-ROM drive being used.

- When installation of the software on the Solaris 8 Software 1 of 2 SPARC Platform Edition CD is finished, the Installation Summary dialog box is displayed.

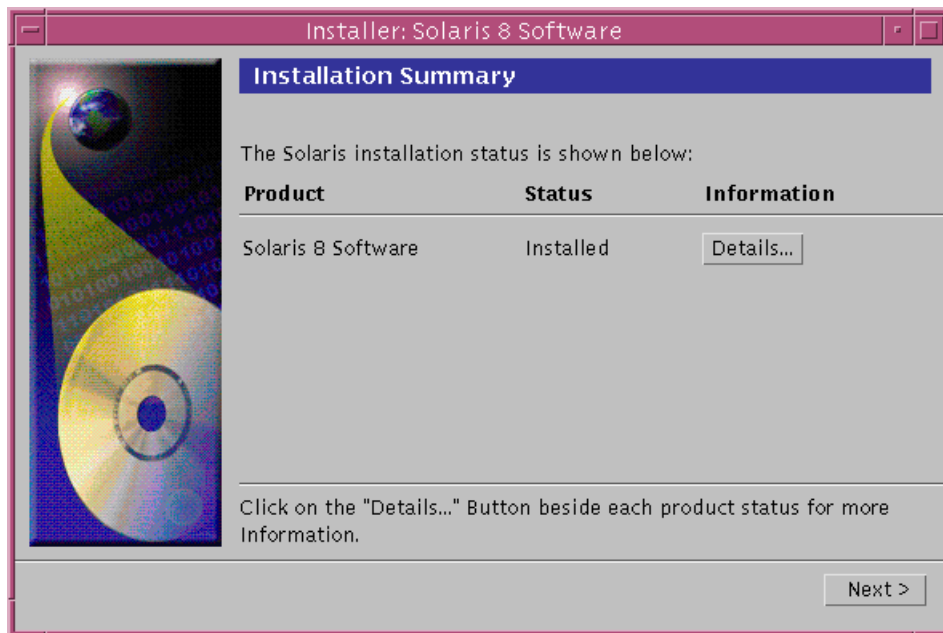


Figure 14-32 Installation Summary Dialog Box

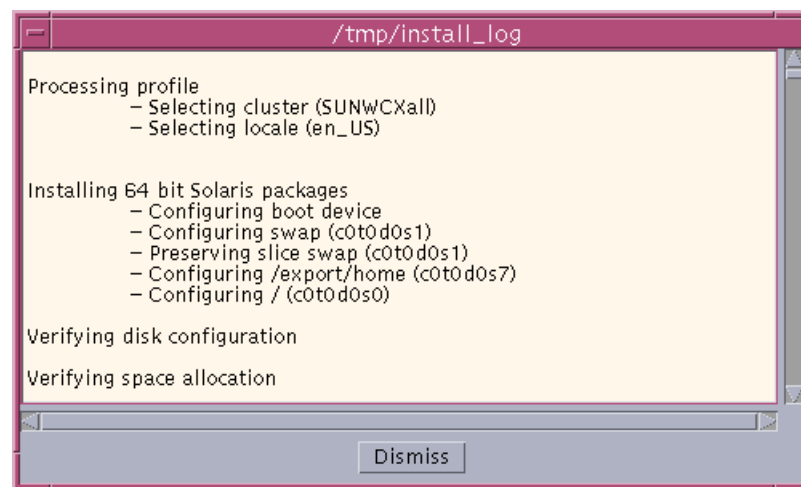


Figure 14-33 Details Dialog Box

- After viewing the log, click on Dismiss. You are returned to the Installation Summary dialog box. When you click on Next, the CD-ROM is ejected, and the Specify Media dialog box is displayed.

Additional Software

If you require additional software, determine if it will be installed from a CD-ROM or through a network file system.

- If it is installed from a CD-ROM, insert the CD-ROM specified on the Insert CD-ROM dialog box and click on OK.

The Reading CD-ROM, Launching Installer, and Extracting dialog boxes are displayed, followed by the Installing dialog box. When installation of the software is finished, the CD-ROM in the drive ejects, and the Installation Summary dialog box is displayed.

- If it is installed through a network file system, type the path to the network file system on the Specify Network Filesystem Path dialog box and click on Next.

The Launching Installer and Extracting dialog boxes are displayed, followed by the Installing dialog box. When installation of the software through the network is finished, the Installation Summary dialog box is displayed.

Note – Again, installation can take a while, but the actual time depends on the software group that is being installed, the reallocation of any space if needed, and the speed of the network or local CD-ROM drive being used.

In the next dialog box, select the Solaris Operating Environment desktop: either CDE or OpenWindows and click on OK.

After a few seconds, the Solaris Operating Environment desktop that you selected is displayed.

Exercise: The Solaris Operating Environment



Exercise objective – In this lab you will boot the system from the Solaris 8 Installation CD and install the Solaris Operating Environment software.

Preparation

This procedure requires a system configured with a 2 Gbyte boot disk or larger. Depending on the speed of CD-ROM devices in use, the complete installation process requires approximately the following amount of time:

- 8 minutes to boot from the Solaris 8 Installation CD.
- 7 to 15 minutes to load a mini-root file system into the swap slice and reboot.
- 30 minutes to load data from the Solaris 8 Software 1 of 2 CD.
- 13 to 20 minutes to load data from the Solaris 8 Software 2 of 2 CD.

Locate the Solaris 8 Software CD-ROM set. Your instructor will give you any last minute preparation details that might be required. Most panels presented during the installation process require that you select **Next** to continue.

Task Summary

In this exercise you will accomplish the following:

- Boot the system from the Solaris 8 Installation CD, and install the Solaris Operating Environment software. Create a configuration as follows:
 - ▼ Perform an initial install. Allow the swap slice to start at the beginning of the disk. Use the smallest swap size allowed.

- ▼ Assign host name, IP address, netmask, and naming service information compatible with the classroom configuration.
- ▼ Create a standalone system and use the Entire Distribution configuration cluster. Elect not to install additional software products.
- ▼ Create slices for `/`, `/usr`, and `/var` such that they use at least 200, 1375, and 100 megabytes respectively. Set the `root` password to “cangetin”.

Tasks

Complete the following steps:

1. Insert the Solaris 8 Installation CD into the CD-ROM drive.
2. If the Solaris Operating Environment is currently running, either log in as `root` and bring it to run level 0,
init 0
or abort the operating system by pressing **Stop-a**.
3. Set the system OBP parameters to their default values.
ok set-defaults
4. Boot the system from the CD-ROM. The boot process takes about 8 minutes to complete. Ignore error messages relating to network interfaces that are not attached (cable problem messages).
ok boot cdrom
5. The installation process begins automatically. Respond as indicated below to the initial questions asked by the installation process.

The system loads a mini-root file system into the swap slice and automatically reboots from it. This phase of the installation takes from 7 to 15 minutes to complete.

Initial or Upgrade?	Select Initial Install
Format root disk?	Enter y to format the disk.

Swap slice size	Enter the minimum value indicated. On sun4m systems this is 352, on sun4u systems, it is 384. Values indicated are expressed in Mbytes.

Can the swap slice start at the beginning of the disk?	Enter y to allow this.

Confirm selections	Enter y continue.

6. The WebStart installation begins once the system boots from the mini-root file system on the swap slice. Respond to prompts as indicated in the following section to create a system configuration compatible with the remaining lab assignments for SA-238. The WebStart install program presents various prompts and screens that require the information listed below

Welcome screen:	Select Next
Network Connectivity:	Select Networked
Use DHCP:	Select No
Host Name:	(provided by instructor)
IP Address:	(provided by instructor)
Netmask:	(provided by instructor)
Enable IPv6:	Select No
Name Service:	Select None
Specify Time Zone by:	Select by Geographic region
Geographic Region:	Select the region and time zone appropriate for your location.
Date and time:	Verify and set if required
Root Password:	Enter cangetin
Power Management:	Specified on sun4u systems. Select Turn Power Management Off, and Don't Ask
Proxy Server Configuration:	Select Direct connection to the inter-net
Confirm Information:	Select Confirm if the information displayed is correct, or click on Back to make changes before continuing

Welcome screen:	Select Next The Solaris 8 Installation CD automatically ejects Insert the Solaris 8 Software 1 of 2 CD. and select OK
Type of Install:	Select Custom Install
Software Localizations:	Select the geographic region(s) appropriate for your location
System Locale:	Select the locale appropriate for your location. For example, in the United States, select en_US . ISO8859 - 1
Select Products:	Deselect all categories listed
Additional Products:	Select None
64-Bit Selection:	Use the default - Yes for sun4u systems, and No for sun4m systems
Solaris Cluster Configuration:	Select Entire Solaris Software Group
Disk Selection:	Verify that the disk designated as the boot disk is listed in the Selected Disks column. Verify that the Selected disk space value is greater than the Required disk space value. If additional disk space is required, select a disk from the Available Disks column and add it to the Selected Disks column. Verify that sufficient selected disk space exists to install the software. Select Next when finished. The system gathers disk space information.
Lay Out File Systems:	Select the disk designated in the previous panel as your boot disk. Select Modify

If you're using a 2Gbyte boot disk, use the following partition information. Sizes are indicated in Mbytes.

The installation process automatically assigns slice 7 to `/export/home`. Remove this slice allocation before proceeding.

Slice	Use	Size
0	/	200
1	swap	(fixed)
3	/var	100
6	/usr	1375

If you are using a disk larger than 2 Gbytes, add any extra space to `/usr`.

If you are using more than one disk to store the file systems indicated, always place the root (/) file system on slice 0 on the boot disk. You might need to divide the `/usr` file system into two file systems, `/usr` and `/usr/openwin`.

Select **OK** when complete. Select **Next** to continue.

Ready to Install:

Select **Install Now**
This phase of the installation takes about 30 minutes to complete. The Solaris 8 Software 1 of 2 CD automatically ejects.

Installation Summary:

Select **Next**

Specify Media:

Select **CD**. Insert the Solaris 8 Software 2 of 2 CD into the CD-ROM drive. Select **Next**. If the CD-ROM tray opens again, close it and select **OK** to continue.

This phase of the installation takes from 13 to 20 minutes to complete.

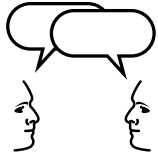
Installation Summary:

Select **Next**. The the Solaris 8 Software 2 of 2 CD automatically ejects.

Reboot:

Select **Reboot 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

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ State the different installation methods available for the Solaris 8 Operating Environment software
- ☐ Explain the hardware requirements for a Solaris 8 Operating Environment installation
- ☐ Identify the different Solaris 8 Operating Environment software CD-ROM editions
- ☐ List the five Solaris Software Groups
- ☐ Demonstrate how to install the Solaris 8 Operating Environment software on a networked, standalone system, using Solaris Web Start

Objectives

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

- Describe a software package
- View software package information using the `pkginfo` command
- Add a software package from the Solaris Software CD-ROM using the `pkgadd` command
- Verify the attributes and content of a software package using the `pkgchk` command
- Remove a software package installed on the disk using the `pkgrm` command
- View, add, and remove software packages using the `admintool`
- Add and remove a software package from a spool directory using the `pkgadd` and `pkgrm`

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Sun Part Number 805-7228-10

Software Packages

Software administration involves adding and removing software from systems. Sun and its third-party vendors deliver products in a form called a software package.

The term *package* refers to the method for distributing and installing software products to systems where the products will be used. In its simplest form, a package is a collection of files and directories

Note – All the required software packages are installed automatically during the Solaris Software installation process.

Software packages contain:

- Files that describe the package and the amount of disk space required.
- The actual software files to be installed on the system.
- Scripts that are run when the package is added and removed.

The tools for viewing, adding and removing software from a workstation after the Solaris software is installed on a system include:

- Package administration commands – `pkgadd`, `pkgrm`, `pkginfo`, and `pkgchk`
- The `admintool` utility – A graphical front-end to the `pkgadd` and `pkgrm` commands

The pkginfo Command

You use the `pkginfo` command to display information about the software packages that have been installed on the local system's disk.

Command Format

```
pkginfo [ -d [ device | pathname ] ] [ -l ] pkg_name
```

For example:

```
# pkginfo | more
<some output omitted>
application SUNWaxg      Solaris XGL 3.3 AnswerBook
application SUNwaadm     Solaris 7 System Administrator Collection
system      SUNwab2m      Solaris Documentation Server Lookup
system      SUNwab2r      Solaris Documentation Server
system      SUNwab2s      Solaris Documentation Server
system      SUNwab2u      Solaris Documentation Server
application SUNwabda      Sun Ultra 5/10 Hardware AnswerBook
application SUNwabe       Solaris 7 User Collection
application SUNwabsdk     Solaris 7 Software Developer Collection
```

The columns of information that are displayed are described below.

CATEGORY	Is the package category, such as application, system, ALE, or CTL.
PKGINST	Is the software package name; if it begins with SUNW, it is a Sun Microsystems product, otherwise it represents a third-party package.
NAME	Is a brief description of the software product.

Displaying Detailed Information for All Packages

To view all the available information about the software packages, use the `pkginfo` command with the `-l` option:

```
# pkginfo -l | more
```

Displaying Detailed Information for a Specific Package

To view information for a specific software package, specify its name on the command line, for example:

```
# pkginfo -l SUNWman
```

```
PKGINST:      SUNWman
NAME:         On-line Manual Pages
CATEGORY:     system
ARCH:         sparc
VERSION:      41.0, REV=31
BASEDIR:      /usr
VENDOR:       Sun Microsystems, Inc.
DESC:         System Reference Manual Pages
PSTAMP:       tinkertoym09133331
INSTRELEASE:  May 19 2000 16:50
HOTLINE:      Please contact your local service provider
STATUS:       Completely Installed
FILES:        6420 installed pathnames
              3 shared pathnames
              74 directories
              73925 blocks used (approx)
```

The last line (73925 blocks used (approx)), identifies the size of the package. A block is a 512-byte disk block. The `blocks used` number defines how much space is needed on the disk to install this package.

To determine how many packages are currently installed on disk, use the following command:

```
# pkginfo | wc -l
```

Displaying Information for Software Packages on CD-ROM

By default, the `pkginfo` command is used to access information about packages that have been installed on disk.

Displaying Detailed Information for All Packages on CD-ROM

To display information about software packages that resides on the Solaris Software CD-ROM (or other release media), use the `pkginfo` command with the `-d` option. This option defines the device on which the software packages reside.

```
# pkginfo -d /cdrom/cdrom0/s0/Solaris_8/Product | more
```

Displaying Detailed Information for Selected Package on CD-ROM

```
# pkginfo -d /cdrom/cdrom0/s0/Solaris_8/Product -l SUNWaudio
```

```
PKGINST:  SUNWaudio
NAME:      Audio applications
CATEGORY:  system
ARCH:      sparc
VERSION:   3.6.20, REV=1.1999.12.03
BASEDIR:   /
  VENDOR:  Sun Microsystems, Inc.
  DESC:    Audio binaries
  PSTAMP:  dtbuild38s19991204142646
INSTDATE:  May 19 2000 16:35
HOTLINE:   Please contact your local service provider
STATUS:    spooled
  FILES:   5 spooled pathnames
           2 directories
           3 executables
           4 package information files
           681 blocks used (approx)
```

The pkgrm Command

When a software package is removed from the system, the `pkgrm` command deletes all files associated with that package unless those files are also shared with other packages.

The command asks for confirmation to continue and might warn about possible package dependencies. If package dependencies do exist, it will again ask for confirmation to continue with the package removal process.

Command Format

`pkgrm pkg_name`

For example:

```
# pkgrm SUNWaudio
```

```
The following package is currently installed:
```

```
SUNWaudio      Audio applications
                (sparc) 3.6.4,REV=1.98.12.03
```

```
Do you want to remove this package? y
```

```
## Removing installed package instance <SUNWaudio>
```

```
## Verifying package dependencies.
```

```
WARNING:
```

```
    The <SUNWolrte> package depends on the package
    currently being removed.
```

```
WARNING:
```

```
    The <SUNWolaud> package depends on the package
    currently being removed.
```

```
WARNING:
```

```
    The <SUNWoldcv> package depends on the package
    currently being removed.
```

```
WARNING:
```

```
    The <SUNWxwkey> 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
```

Note – The message *filename* <shared pathname not removed> is displayed if a file is shared by two or more packages. It is removed only when the last package it is shared with is removed.

The pkgadd Command

When a software package is added, the `pkgadd` command uncompresses and copies files from the installation media to the local system's disk. This command will ask for confirmation to continue with package add process.

Command Format

```
pkgadd [ -d [ device | pathname ] ] pkg_name
```

For example:

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_8/Product SUNWaudio
```

```
Processing package instance <SUNWaudio> from  
</cdrom/sol_8_sparc/s0/Solaris_8/Product>
```

```
Audio applications  
(sparc) 3.6.4,REV=1.98.12.03  
Copyright 1999 Sun Microsystems, Inc. All rights reserved.  
Using </> as the package base directory.  
## Processing package information.  
## Processing system information.  
    2 package pathnames are already properly installed.  
## 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 <SUNWaudio> [y,n,?] y
```

```
Installing Audio applications as <SUNWaudio>
```

```
## Installing part 1 of 1.
```

```
Installation of <SUNWaudio> was successful.
```


The pkgchk Command

The `pkgchk` command checks installation completeness, pathname, file contents, and file attributes of a package.

Command Format

```
pkgchk [ options ] [-p path ...] [ pkg_name ]
```

The following example checks the contents and attributes of a software package currently installed on the system.

```
# pkgchk SUNWaudio
```

Note – If the `pkgchk` command does not display a message it indicates that the package was installed successfully.

To list the files contained in a software package, type:

```
# pkgchk -v SUNWaudio
```

To check any file to determine if its content and attributes have changed since it was installed with its software package, type:

```
# pkgchk -p /etc/passwd
ERROR: /etc/passwd
file size <414> expected <3391> actual
file cksum <34239> expected <17254> actual
```

The original `/etc/passwd` file has changed in size since the initial Solaris Operating Environment software installation.

This is indicated by the differences in file size and checksum. The checksum is used to validate transported data.

To list information on the selected files that make up a package, type:

```
# 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): 30116
Expected sum(1) of contents: 42039
Expected last modification: Dec 16 02:05:06 PM
Referenced by the following packages:
    SUNWadmc
Current status: installed
```

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 belonging to every software package, and the configuration of products installed can be viewed.

more /var/sadm/install/contents

The `pkgadd` command updates the `contents` file whenever new packages are installed.

The `pkgrm` command uses the `contents` file to determine where files for a software package are located on the system. Once a package is removed, `pkgrm` updates the `contents` file. This file can be queried to determine if a particular file has been installed on the system disk.

Identifying the Directory Location of a Command

Use the `grep` command to search the `/var/sadm/install/contents` file to determine if a particular file was installed, and the directory where it is located. For example, verify that the command `showrev` is installed on the system disk.

```
# grep showrev /var/sadm/install/contents
/usr/bin/showrev f none 0755 root sys 30116 42078 943868705 SUNWadmc
/usr/share/man/sman1m/showrev.1m f none 0444 bin bin 6398 62569 943312114
SUNWman
```

Search the Solaris Operating Environment CD-ROM for Command Information

Use the `grep` command to search for the `showrev` command on the distribution media. Instead of searching the `contents` file on the system disk, in this example the information for the `showrev` command is contained in the `pkgmap` file.

Every software package contained on the distribution media has its own `pkgmap`, which contains a content list of each package.

```
# grep showrev /cdrom/cdrom0/s0/Solaris_8/Product/*/pkgmap
/cdrom/sol_8_sparc/s0/Solaris_8/Product/SUNWadmc/pkgmap:1 f none
usr/bin/showrev 0755 root sys 31276 44676 938676470
```

Adding and Removing Packages With admintool

The package administration capabilities provided by the package commands `pkginfo`, `pkgadd`, and `pkgrm` are also available with `admintool`.

To Display Software Package Information

1. As root or as a member of the `sysadmin` group (GID 14), start `admintool`.

```
# admintool &
```

The Users window is displayed.

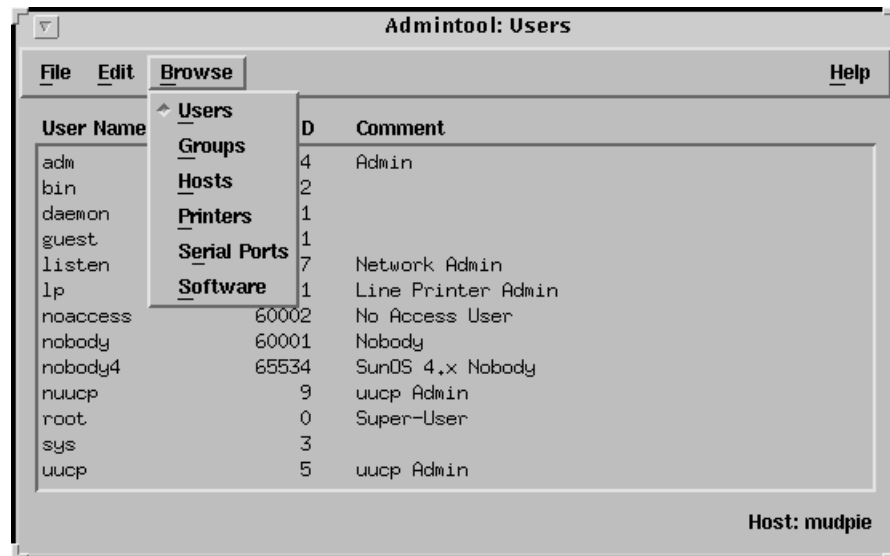


Figure 15-1 The `admintool` Users Window

2. From the Browse menu, select Software. The Software window is displayed.

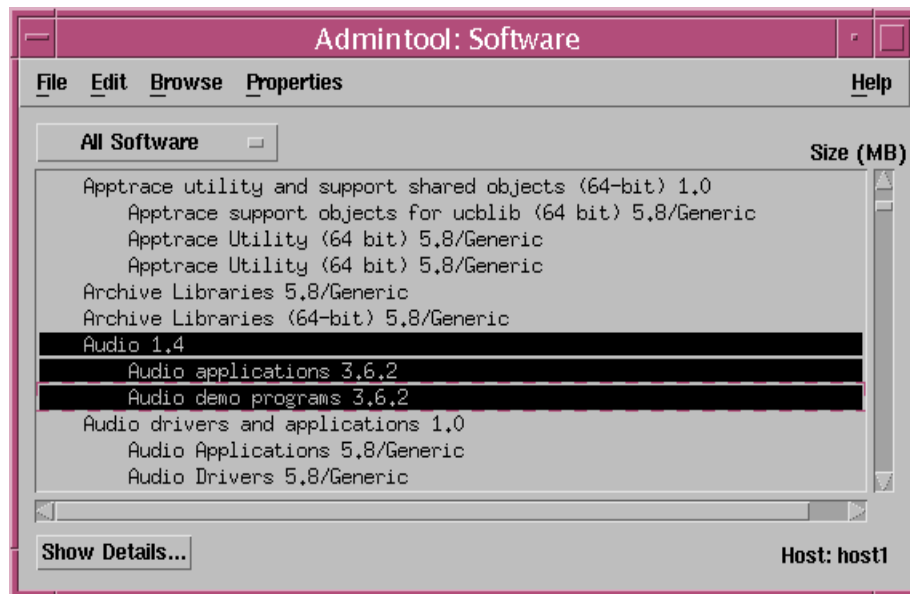


Figure 15-2 Admintool: Software Window

The Software option displays all software packages installed on the local system's disk. You can view only the system packages or the application packages.

3. Select the Audio 1.4. package.

The three lines relating to the Audio 1.4 software are highlighted.

4. Click on Show Details. The Software Details window is displayed.

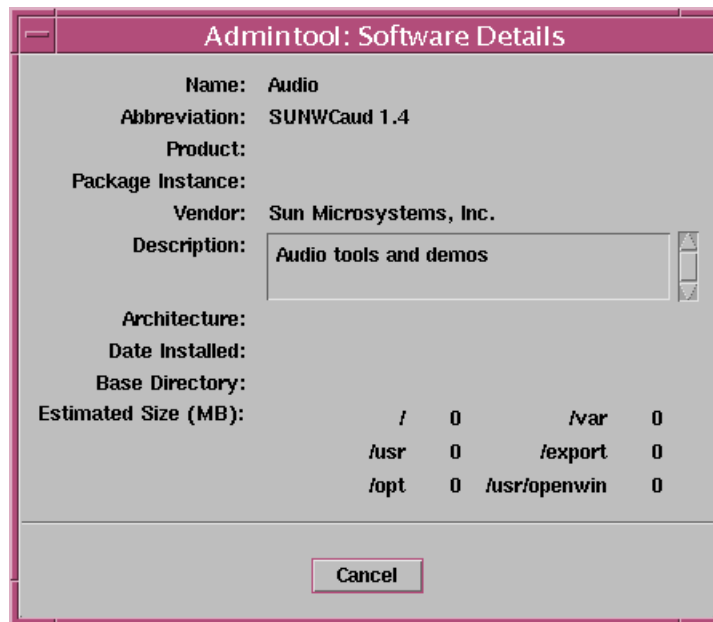


Figure 15-3 Software Details Window

This window shows limited information about the package, such as:

- ▼ Software name
- ▼ Abbreviated name
- ▼ Vendor
- ▼ Description

5. Click on Cancel. The Software window is displayed again.

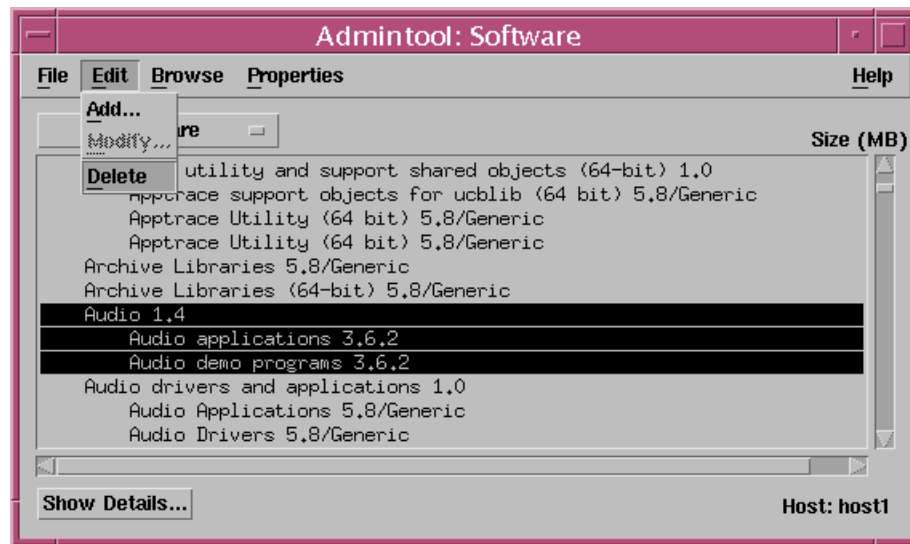


Figure 15-4 Deleting Software

6. From the Edit menu, select Delete. A confirmation window is displayed.

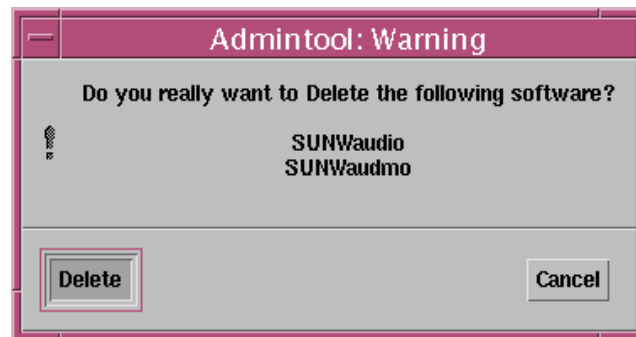


Figure 15-5 Delete Warning Window

7. Confirm this choice by clicking on Delete. The output of scripts used to remove the software is displayed.
8. Respond with y or yes to all questions regarding the removal of the package.

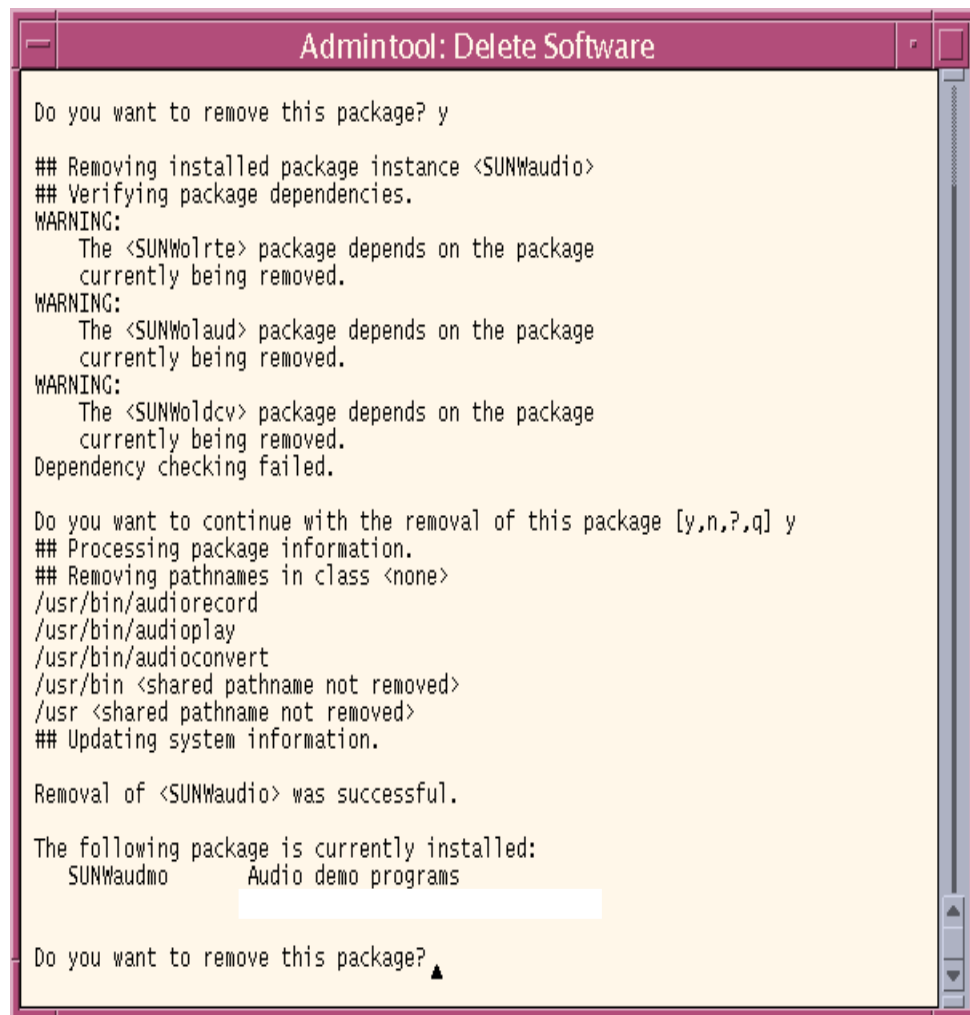


Figure 15-6 Dependency Checking

9. Press Return to continue.

The final message displayed in the window indicates that the removal of the package was successful.

Removal of <SUNWaudio> was successful.

Note – When a software package is removed by the `pkgrm` command, its package name remains in the window. To update the window, you must close and then reopen the window.

Managing Software With `admintool`

Adding a Software Package

1. Insert the Solaris Software CD-ROM 1 of 2 (if not already in the CD-ROM drive).
2. Start `admintool` (if it is not already displayed).
3. From the Edit menu select Add.

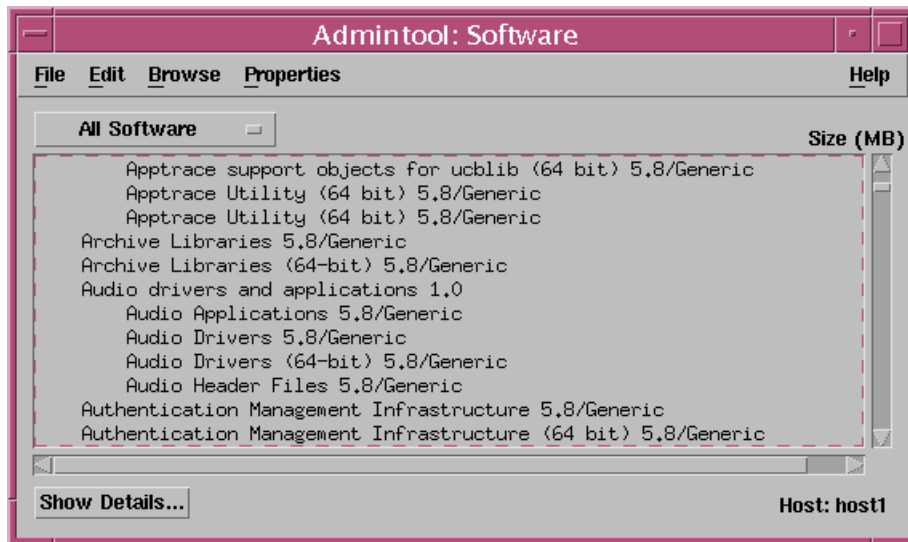


Figure 15-7 Adding Software Package

The Set Source Media window is displayed.

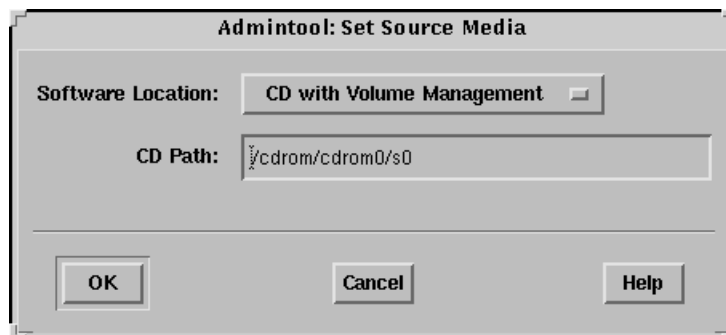


Figure 15-8 Set Source Media Window

The default volume management CD-ROM path is `/cdrom/cdrom0/s0`.

4. Click on OK.

The following are descriptions of set source media selection:

- Select CD with Volume Management if Volume Management (/usr/sbin/vold daemon) is running and the CD-ROM is a Solaris Software CD.
- Select CD without Volume Management if the system is *not* running Volume Management and the CD-ROM is a Solaris Software CD. The default path is usually displays as /export/install.
- Select Hard Disk if the software is on the local disk. Specify the absolute path of the directory where the software package(s) are located.

The Add Software window is displayed.

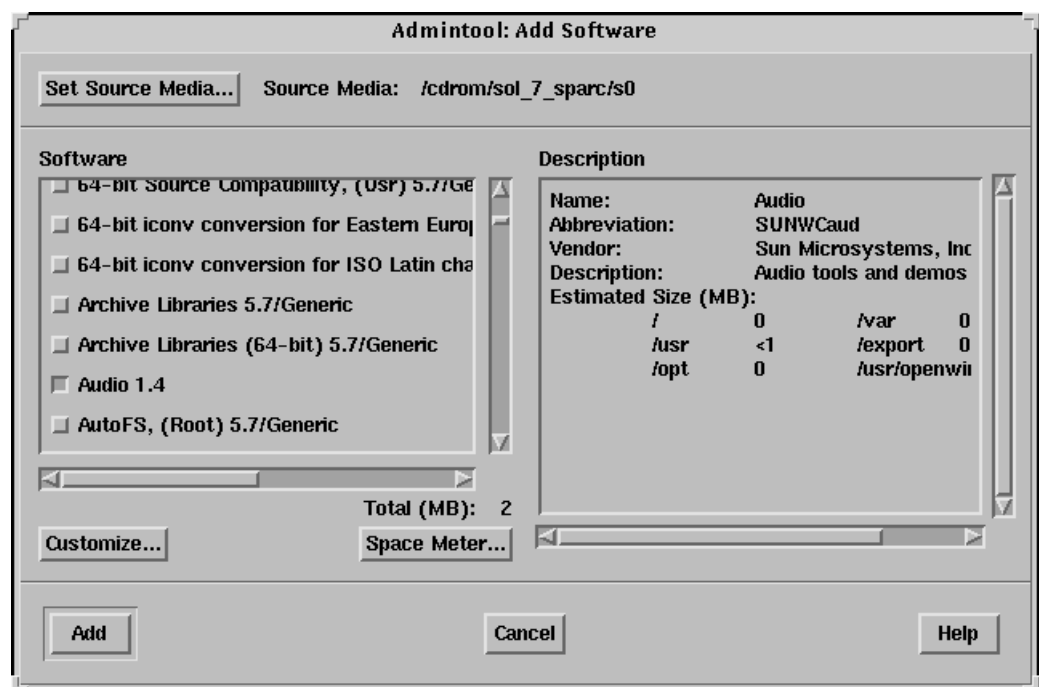


Figure 15-9 Add Software Window

The Add Software window lists the packages that you can install.

If no packages appear in the window, an incorrect location might have been specified for the source media. Reset by clicking on Set Source Media.

5. Select the Audio 1.4 package.
6. Click on Customize. The Customize Installation window allows you to select software package components.

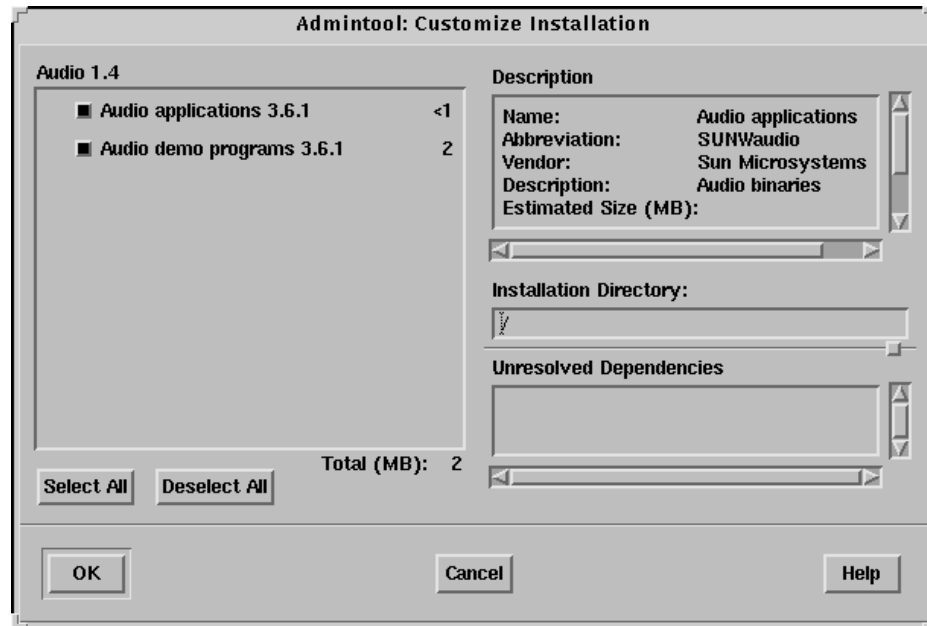


Figure 15-10 Customize Installation Window

- ▼ The Description field provides information about the specified package, including approximate disk space requirements.
 - ▼ The Installation Directory field specifies where the package will be installed. The default installation path is displayed. You can modify this path to install software in an alternative location.
 - ▼ The Unresolved Dependencies field displays other software packages that are required for this product to work properly.
7. Click on Cancel.

The Add Software window is displayed again.

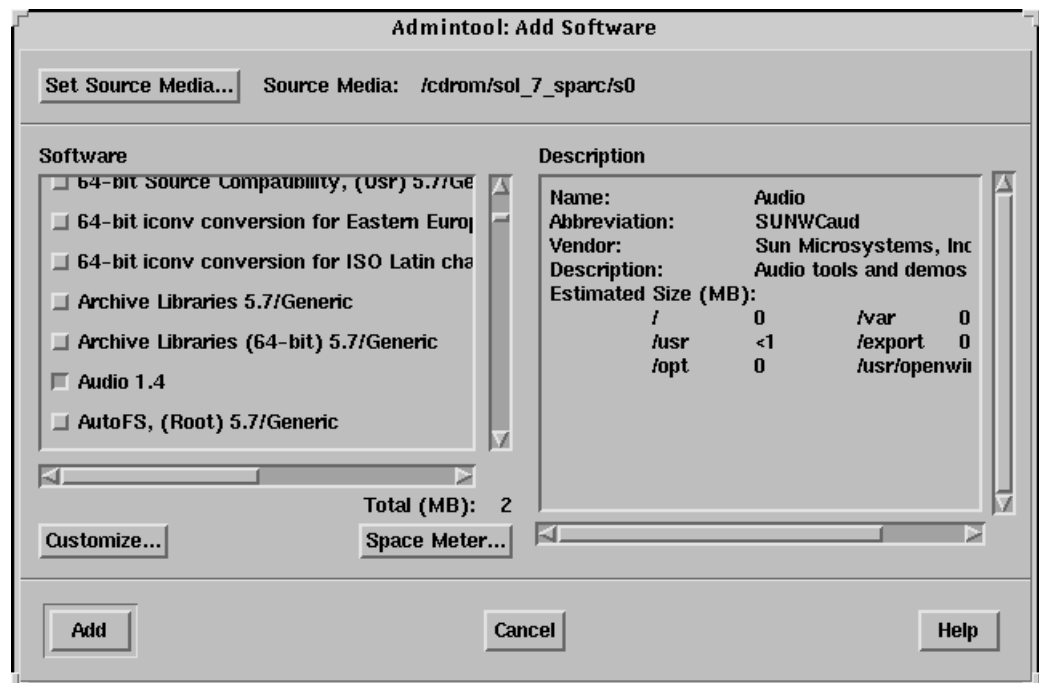


Figure 15-11 Add Software Window

8. Click on Add. The output of the scripts used to install the software package are shown. These scripts might be interactive and require input when requested.
9. Press Return to continue.

```

Admintool: Add Software

Processing package instance <SUNWaudio> from

Audio applications
(sparc) 3.6.4,REV=1.98.08.13

Using </> as the package base directory.
## Processing package information.
## Processing system information.
  2 package pathnames are already properly installed.
## 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 <SUNWaudio> [y,n,?] y

Installing Audio applications as <SUNWaudio>

## Installing part 1 of 1.

Installation of <SUNWaudio> was successful.

Processing package instance <SUNWaudmo> from
</cdrom/sol_7_sparc/s0/Solaris_2.7/Product>

Audio demo programs
(sparc) 3.6.1,REV=1.98.08.13

Using </> as the package base directory.
## Processing package information.
## Processing system information.
  2 package pathnames are already properly installed.
## 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 <SUNWaudmo> [y,n,?] y

Installing Audio demo programs as <SUNWaudmo>

## Installing part 1 of 1.

Installation of <SUNWaudmo> was successful.
press <Return> to continue
  
```

Figure 15-12 Add Software Process Output Window

The final message displayed in the window indicates the installation of the package was successful.

Installation of <SUNWaudio> was successful

Using a Spool Directory

For convenience, frequently installed software packages can be copied from the Solaris Software CD-ROM to a `spool` directory on the system.

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 into a `spool` directory is not the same as installing the packages on disk.

To copy a package into the `/var/spool/pkg` directory:

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_8/Product -s spool SUNWaudio
Transferring <SUNWaudio> package instance
```

The `-s` option with the keyword `spool` copies the package into the `/var/spool/pkg` directory by default.

Spooling Packages

You can specify a different directory location using the `-s` option. In this example, a new directory is created, then `pkgadd` is instructed to copy the package into the new `spool` directory.

```
# mkdir /export/pkg
# pkgadd -d /cdrom/cdrom0/s0/Solaris_8/Product -s /export/pkg SUNWaudio
Transferring <SUNWaudio> package instance
# ls /export/pkg
SUNWaudio
```

Removing Packages From the Spool Directory

You remove software packages from a spool directory using the `pkgrm` command with the `-s` option.

```
# pkgrm -s spool SUNWaudio
# pkgrm -s /export/pkg SUNWaudio
```

Package Administration Summary

The following sections summarize the tasks involved in package administration.

Package Command Summary

Table 15-1 summarizes the commands used for package administration.

Table 15-1 Package Administration

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 and contents of the path names belonging to packages.

Package Administration File and Directory Summary

Table 15-2 describes a list of the files and directories used with package administration.

Table 15-2 Files and Directories

File or Directory	Description
/var/sadm/install/contents	Software package map of the entire system.
/opt/pkgname	Preferred location for the installation of unbundled packages.
/opt/pkgname/bin or /opt/bin	Preferred location for the executable files of unbundled packages.
/var/opt/pkgname or /etc/opt/pkgname	Preferred location for log files of unbundled packages.

Exercise: Software Package Administration Commands



Exercise objective – In this lab you will use package-related commands and `admintool` to remove, install, and spool packages.

Preparation

Locate the Solaris 8 Software CD-ROMs. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise you will accomplish the following:

- Find the names of packages installed on your system that relate to manuals. List and record the status, install date, number of files, and number of blocks used by the `SUNWman` package. Obtain the same information from the spooled `SUNWman` package on the correct Solaris 8 Software CD-ROM. Remove and re-install the `SUNWman` package.
- Remove the `SUNWdoc` package from the system. Spool `SUNWdoc` from the correct Solaris 8 Software CD-ROM into the default spool area. Verify the presence of this package in the spool area. Add the `SUNWdoc` package. Remove `SUNWdoc` from the spool area.
- Use `admintool` to remove the `Audio1.4` software group from the system. Use `admintool` to add the `Audio Header Files 5.8/Generic` and `Audio demo programs 3.6.2` from the Solaris 8 2 of 2 Software CD-ROM. Eject the CD-ROM.

Tasks

Complete the following steps:

1. Insert the Solaris 8 Software 2 of 2 CD-ROM into the drive.
2. Use `pkginfo` to search for packages currently installed on your system that are related to manuals.

```
# pkginfo | grep anual
```

What packages were listed?

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, install date, number of files, and number of blocks used by this package?

4. Display a long-format listing of the information for the `SUNWman` package found on the Solaris 8 2 of 2 Software CD-ROM. Obtain the same information as in the previous step.

```
# pkginfo -d /cdrom/cdrom0/Solaris_8/Product -l SUNWman
```

5. Remove the `SUNWman` package from your system and verify it has been removed.

```
# pkgrm SUNWman  
# pkginfo SUNWman  
# man ls
```

6. Re-install the SUNWman package from the Solaris 8 Software 2 of 2 CD-ROM. Respond **y** to questions asked by **pkgadd**. Verify that the manual pages work.

```
# pkgadd -d /cdrom/cdrom0/Solaris_8/Product SUNWman
# man ls
```

7. Remove the SUNWdoc package from your system. Respond **y** to questions asked by **pkgrm**.

```
# pkgrm SUNWdoc
```

8. Eject the Solaris 8 Software 2 of 2 CD-ROM, and insert the Solaris 8 Software 1 of 2 CD-ROM. Use **pkgadd** to spool the SUNWdoc package into the default spool area.

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_8/Product -s spool SUNWdoc
```

9. Use the following commands to verify the presence of SUNWdoc in the default spool area.

```
# pkginfo -d spool SUNWdoc
# pkginfo -d /var/spool/pkg -l SUNWdoc
```

10. Install SUNWdoc. Observe the messages displayed and verify that the package is installed from **/var/spool/pkg**.

```
# pkgadd SUNWdoc
```

11. Remove SUNWdoc from the default spool area.

```
# pkgrm -s spool SUNWdoc
```

12. Eject the Solaris 8 Software 1 of 2 CD-ROM, and insert the Solaris 8 Software 2 of 2 CD-ROM.

13. Run **admintool**. Select the **Software** category from the **Browse** menu.

14. Scroll down the list and select **Audio1.4**. From the **Edit** menu, select **Delete**. Respond **y** to all questions asked by the delete process.

15. Select **Add** from the **Edit** menu. In the **Software Location** field, select **CD with Volume Management**. Specify the path: **/cdrom/cdrom0/Solaris_8/Product**. Click on **OK**.

16. Select the Audio Header Files 5.8/Generic and Audio demo programs 3.6.2 items from the Software list. Click on Add to add the selected software. Respond y to all questions asked by the installation process.
17. Once the installation is complete, select Exit from the File menu to quit admintool. Eject the CD-ROM.

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

2. Use `pkginfo` to search for packages currently installed on your system that are related to manuals. What packages were listed?

SUNWaman SUNWman SUNWmfman SUNWpl5m SUNWtltkm

These packages contain the Solaris 8 Reference Manual, the On-Line Manual Pages, CDE Motif Manuals, Perl5 On-Line Manual Pages, and ToolTalk manual pages respectively.

3. Display a long-format listing of the information for the `SUNWman` package installed on your system.

What is listed for the status, install date, number of files, and number of blocks used by this package?

Status: completely installed

Install date: This should match the date and time when you installed Solaris on your system.

Number of files: 6420

Number of blocks: 73925

4. Display a long-format listing of the information for the `SUNWman` package found on the Solaris 8 2 of 2 Software CD. Obtain the same information as in the previous step.

Status: spooled

Install date: There should be no install date indicated.

Number of files: 6424

Number of blocks: 73925

10. Install `SUNWdoc`. Observe the messages displayed and verify that the package is installed from `/var/spool/pkg`.

*pkgadd displays the following message on the first line of output:
Processing package instance <SUNWdoc> from </var/spool/pkg>*

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Describe a software package
- ☐ View software package information using the `pkginfo` command
- ☐ Add a software package from the Solaris Software CD-ROM using the `pkgadd` command
- ☐ Verify the attributes and content of a software package using the `pkgchk` command
- ☐ Remove a software package installed on the disk using the `pkgrm` command
- ☐ View, add, and remove software packages using the `admintool`
- ☐ Add and remove a software package from a spool directory using the `pkgadd` and `pkgrm`

Objectives

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

- List the locations to access patches
- Explain how to access patches from the World Wide Web and anonymous ftp
- Describe the different patch formats
- Prepare a patch for installation
- Install a patch using the `patchadd` command
- Demonstrate how to verify what patches are currently installed
- Remove a patch using the `patchrm` command

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Sun Part Number 805-7228-10

Patch Administration

The administration of patches involves installing or removing Solaris Operating Environment patches from a running Solaris Operating Environment.

A patch contains a collection of files and directories that replace existing files and directories that are preventing proper execution of the software. Some patches contain product enhancements.

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 **101945-02**, indicates that **101945** is the base code, and **02** is the revision number.

Patch Distribution

Sun customers have access to a general set of security patches and other recommended patches through the World Wide Web or anonymous ftp.

Sun customers who have a Sun ServiceSM contract, have access to the SunSolve database of patches and patch information, such as technical white papers, the Symptom and Resolution database, and more. These are available using the World Wide Web or anonymous ftp.

A SunService customer can request to receive the Patch Update CD-ROMs, which are released every six to eight weeks.

World Wide Web Patch Access

To access patches on the World Wide Web site, the workstation has to be:

- Able to access the Internet
- Capable of running Web browsing software, such as Netscape

To access patches using the World Wide Web, use the following 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

Or use the following URL, and navigate to the SunSolve patch database from the Support entry.

<http://www.sun.com>

From the Sun Microsystems home page, click on the Sales and Service button and navigate to the SunSolve patch database.

The patch database for publicly available patches are labeled "Public patch access."

The patch database for the comprehensive set of patches and patch information available to contract customers is labeled "Contract customer patch access." The customer's assigned Sun Service password is required to access this database.

SunSolve Site

Figure 16-1 shows a sample SunSolve Web page. This is the American-based site. Other sites are located at the bottom of this Home page.



Figure 16-1 Sample SunSolve Home Page

An Additional URL for Patch Access

The University of North Carolina maintains a public patch site, as a cooperative venture between Sun Microsystems, Inc. and the university.

Publicly available patches can be accessed by using the URL:

<http://metalab.unc.edu/pub/sun-info/sun-patches/>

Anonymous ftp Patch Access

To access patches using anonymous ftp, the workstation must be:

- Able to access the Internet
- Capable of running the ftp program

To access patches using ftp, use the ftp command to connect to:

`sunsolve.sun.com`

When ftp prompts for a login, enter `anonymous` as the login name.
When prompted for the password, enter your complete email address

After the connection is complete, the publicly available patches are located in the `/pub/patches` directory.

An Additional ftp Site for Patch Access

Publicly available patches can also be accessed by connecting to:

`http://metalab.unc.edu/pub/sun-info/sun-patches/`

This site is also maintained by the University of North Carolina.

The ftp Patch Access Procedure

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 `/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 (copying) both files to the local systems directory.

Note – To transfer patches, change the ftp transfer mode to binary, by typing `bin` at the ftp prompt.

For example:

```
# cd /tmp
# ftp sunsolve.sun.com
Name (sunsolve.sun.com:root): anonymous
331-
331-Welcome to the SunSolve Online FTP server.
331-
331-Public users may log in as anonymous.
331-
331-Contract customers should use the following 2-tier login procedure:
331-
331-At the 1st login prompt: sunsolve
331-          passwd: sunmicro
331-
331-At the 2nd login prompt: <sunsolve login name>/<sunsolve passwd>
331-example: myssID/mypasswd
331-
331-Public users may log in as anonymous; contract customers
331-should use the standard sunsolve login and password,
331-followed by their SunSolve account/password when prompted.
331-
331-sunsolve6 FTP server (Version wu-2.6.0(3) Wed Jan 5 15:02:27 MST
2000) ready.
331 Guest login ok, send your complete e-mail address as password.
Password:
<output omitted>
230 Guest login ok, access restrictions apply.
ftp> bin
200 Type set to I.
ftp> cd /pub/patches
ftp> ls 108277*
108277-01.zip
108277.readme
ftp> mget 108277*
mget 108277-01.zip? y
mget 108277.readme? y
ftp> cd
ftp> ls
ftp> bye
# cd /tmp ; ls
108277-01.zip
108277.readme
```

Downloading Patches

When patches are downloaded to the local system, the patches must be placed in a temporary directory to prepare them for installation. The directory most often used is the `/var/tmp` directory.

The most common reason for patch installation failure is directory permission/ownership problems. The `/var/tmp` directory is open to all and eliminates any of these types of problems.

Patch Informational Documents

There are important summary documents that list all recommended patches for every version of the operating system, including a detailed list of all patches for each operating system release.

Table 16-1 Patch Documents

Patch Document	Contents
Solaris8.PatchReport	A summary of all recommended patches for the Solaris 8 Operating Environment release.
8_Recommended.zip	A patch cluster containing all the recommended patches for the Solaris 8 Operating Environment release.
8_Recommended.README	Instructions for how to install the recommended patches for the Solaris 8 Operating Environment.

Start with the Patch Report document first. This report is divided into several different categories regarding information about all patches for a Solaris OS Release.

Listing Patch Documents Using ftp

The following example demonstrates how to use ftp to locate the Patch Report using a wildcard file search. Once found, the document is copied to a directory on the local system. For example:

```
# cd /var/tmp
# ftp sunsolve.sun.com
<output omitted>
ftp> cd /pub/patches
ftp> ls *8.PatchReport
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
Solaris8.PatchReport
Solaris8_x86.PatchReport
226 Transfer complete.
remote: *8*.PatchReport
48 bytes received in 0.00035 seconds (1.4e+02 Kbytes/s)
ftp> get Solaris8.PatchReport
ftp> bye
```

The Solaris8.PatchReport can then be read to determine what patch number(s) may need to be retrieved for installation on the system.

Title: Solaris 8 Patch Report Update as of 17/Apr/00

```
=====
                        Report Notes Section:
This report is generated to provide a summary list of patches released and
available from Sun Microsystems for the listed product. There are updates
of this report twice each month.<output truncated>.....
=====
```

```
=====
                        Quick Reference Section:
=====
```

New Patches Released Since Last Report:

This is the first Report.

Update Revs Released Since Last Report:

This is the first Report.

Solaris 8 Recommended Patches:

(No Official Recommended List At This Time)

Solaris 8 Patches Containing Security Fixes:

Solaris 8 Patches Containing Y2000:

Solaris 8 Obsoleted Patches:

```
=====
Solaris 8 Complete Listing of Released Patches:
Total Patches: 30
Total Bugfixes: 59
=====
```

SunOS Released Patch List:

Patch-ID# 108604-03

Synopsis: SunOS 5.8: Elite3D AFB Graphics Patch

BugId's fixed with this patch: 4234045 4294963 4300089 4303885 4308725

Changes incorporated in this version: 4303885 4308725

Date: Mar/16/00

Patch-ID# 108605-03

Synopsis: SunOS 5.8: Creator 8 FFB Graphics Patch

BugId's fixed with this patch: 4234045 4294963 4303885 4308725

Changes incorporated in this version: 4303885 4308725

Date: Mar/16/00

Patch-ID# 108609-01

Synopsis: SunOS 5.8: Buttons/Dials Patch

BugId's fixed with this patch: 4299526 <output truncated>.....

Figure 16-2 Sample Solaris 8 Patch Report

Note – Not all patches available from Sun Microsystems need to be installed. It is only necessary to install the Recommended Patches, Security Patches, and those required to fix problems specific to your site.

The /var/sadm/patch Directory

Historical information about all patches currently installed on a system is stored in /var/sadm/patch directory. For example:

```
# ls /var/sadm/patch
107558-05  107594-04  107630-01  107663-01  107683-01
107696-01  107817-01  107582-01  107612-06  107640-03
```

You should never modify or delete this directory. If you damage this directory, you can make it impossible to add or remove patches, add new software, or upgrade the Solaris Operating Environment without having to first reload the entire system software.

Patch Formats

Patches come in three different formats depending on the Solaris version and where the patch had been retrieved. For example:

- The Solaris 8 and Solaris 7 Operating Environment patches are in `zip` format, for example: `105050-01.zip`.

Note – Some patches that fix applications on the Solaris 7 Operating Environment can be in the `tar.Z` format.

- The Solaris 2.6 (and earlier) Operating Environment patches are compressed tar files in a `tar.Z` format, for example: `104040-01.tar.Z`.
- The Solaris 2.6 (and earlier) Patch Update CD-ROM contains patches that are `gzip` compressed tar files, for example: `112340-01.tar.gz`

Preparing Patches for Installation

For the Solaris 8 and Solaris 7 Operating Environments, use the `unzip` command to extract the patch files.

```
# /usr/bin/unzip 105050-01.zip
```

For Solaris 2.6 Operating Environment patches, use the `zcat` command to uncompress the patch files and the `tar` command to create the patch directories.

```
# /usr/bin/zcat 104040-01.tar.Z | tar xvf -
```

For the Solaris 2.6 Operating Environment patches retrieved from the Patch Update CD-ROM, use the `gzcat` command to uncompress and create patch directories.

```
# /usr/bin/gzcat 112340-01.tar.gz | tar xvf -
```

Patch Contents

Figure 16-3 shows the contents of a patch directory once it is extracted from the *.zip file.

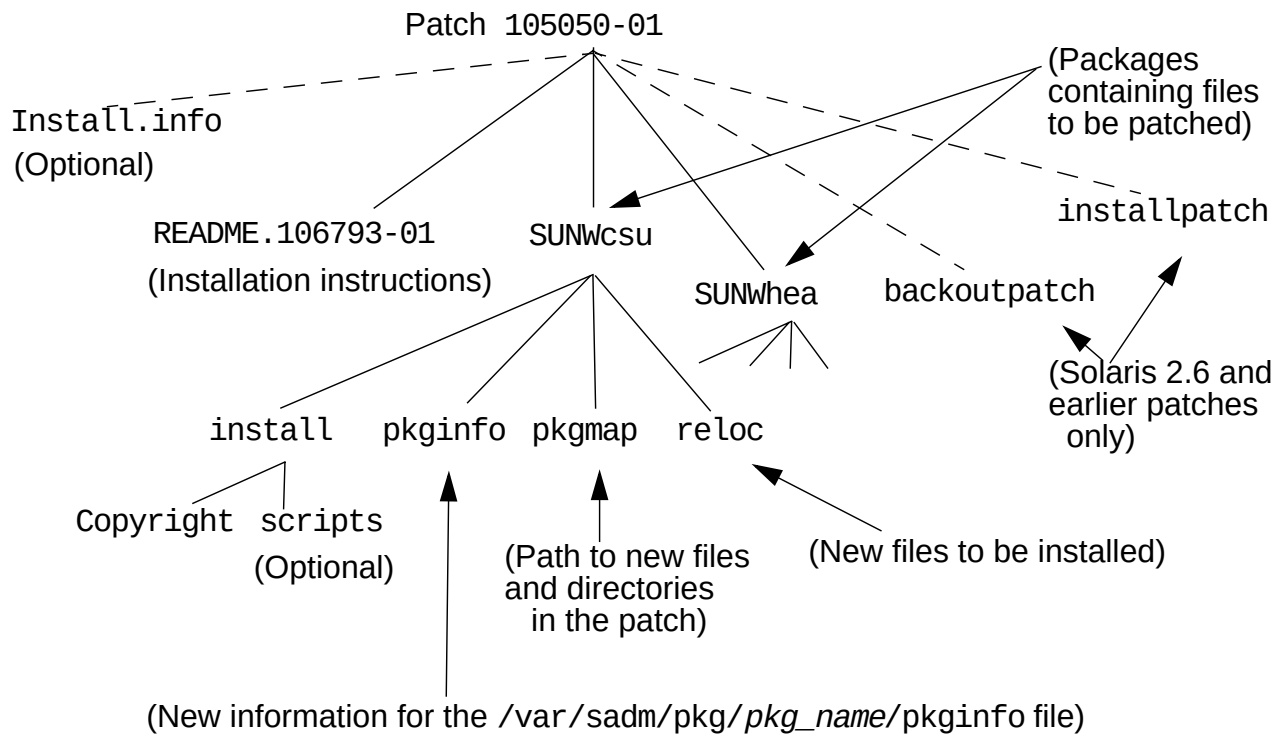


Figure 16-3 An Extracted Patch Directory

The patchadd and patchrm Commands

You have two commands available for managing patches:

- `patchadd` – Used to install unpacked patches to the Solaris Operating Environment.
- `patchrm` – Used to remove patches installed on the Solaris Operating Environment.

Installing a Patch

When a patch is installed, `patchadd` calls the `pkgadd` command to install the patch packages.

Patch installation procedure differs depending on the current version of the Solaris Operating Environment software installed on the system.

The examples below describe the procedure for patch installation on Pre-Solaris 2.6 Operating Environment, and those systems currently installed with Solaris 2.6 and above, (for example, the Solaris 7 or Solaris 8 Operating Environments).

Both examples assume the patch to be installed exists in the `/var/tmp` directory and has been prepared, or extracted for installation.

Installing a Patch in the Solaris 2.6 Operating Environment and Later Versions

For the Solaris 2.6 and above Operating Environments, use the `patchadd` command. The following shows how to install a patch using the `patchadd` command.

```
# cd /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 16-4 illustrates those components of the `/var/sadm` directory that are updated during the installation of patch 105050-01.

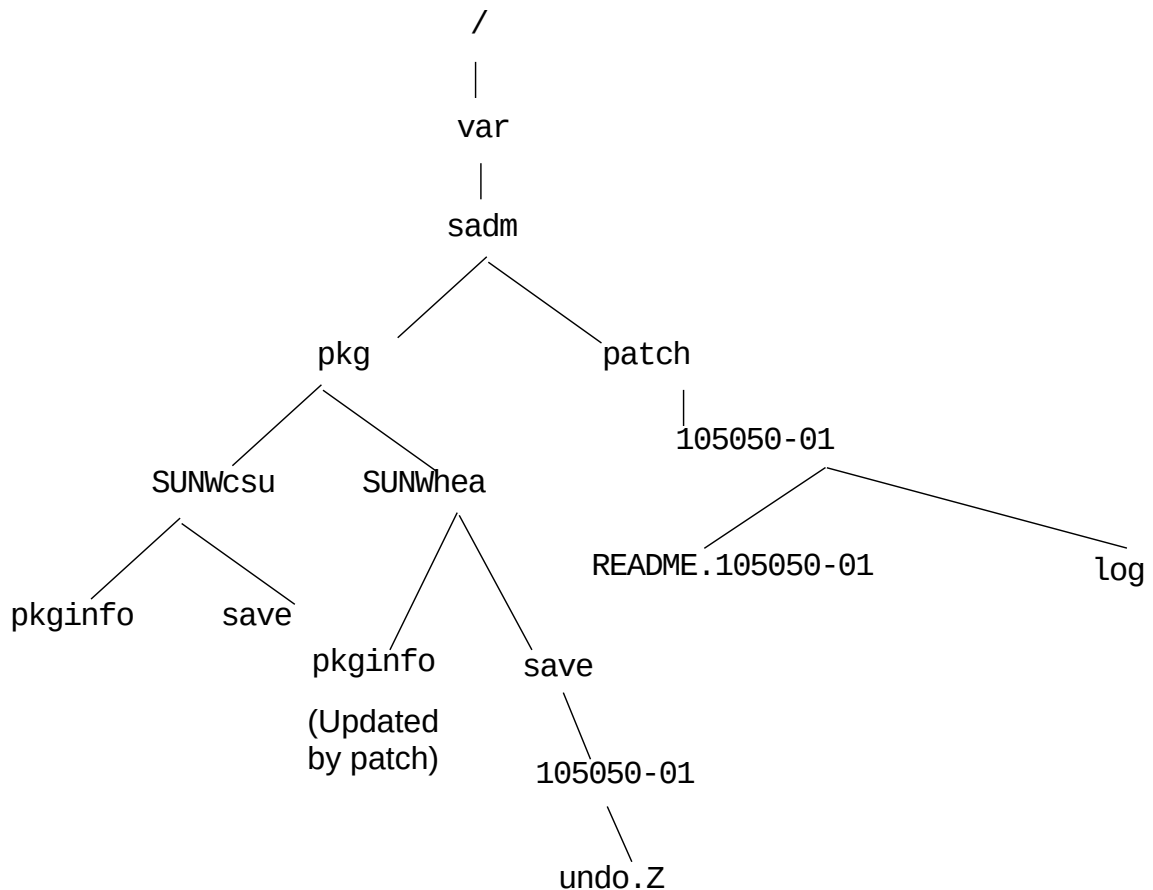


Figure 16-4 Updated `/var/sadm` directories

Installing a Patch in a Pre-Solaris 2.6 Operating Environment

Before the Solaris 2.6 Operating Environment, the `patchadd` command was not available in the Solaris Operating Environment. Instead, each patch contained an `installpatch` program.

The following shows the steps needed to install a patch on a system.

```

# cd /tmp/102301-01
# ls
Install.info      SUNWcsu          backoutpatch
README.102301-01 SUNWscpu         installpatch
  
```

```
# ./installpatch .
```

```
Checking installed packages and patches...
Generating list of files to be patched...
Verifying sufficient filesystem capacity (exhaustive
method)
Installing patch packages...
```

```
Patch number 102301-01 has been successfully installed.
See /var/sadm/patch/102301-01/log for details
```

```
Patch packages installed:
SUNWcsu
SUNWscpu
```



Caution – Both `patchadd` and `installpatch` have a `-d` option available that instructs the commands not to save copies of the files being updated or replaced in the `/var/sadm/patch` directory. This is often used to save disk space over time. However, it also prevents being able to back out or remove a patch from the system.

Checking Current Patch Status

Before installing patches, you should know about patches that have been previously installed on a system.

There are two commands available that 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
```

```
. . .
```

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 `patchadd -d` (which instructs `patchadd` 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 `pkgadd` to restore packages that were saved from the initial patch installation.

Removing a Patch from the Solaris 2.6 and Later Operating Environments

For the Solaris 2.6 and above Operating Environments, use the `patchrm` command. The following shows how to remove a patch using the `patchrm` command.

```
# patchrm 106793-01
```

```
Checking installed packages and patches...
```

```
Backing out patch 106793-01...
```

```
Patch 106793-01 has been backed out.
```

```
#
```

Removing a Patch from the Pre-Solaris 2.6 Operating Environments

Before the Solaris 2.6 Operating Environment, the `patchrm` command was not available. Instead, each patch contained an `backoutpatch` program.

```
# cd /var/sadm/patch/102301-01  
# ./backoutpatch 102301-01
```

Exercise: Patches Maintenance



Exercise objective – In this lab you will transfer a patch from a classroom server, apply the patch and then remove it.

Preparation

Your instructor will provide direction for accessing a patch located on a server that is available to systems in the classroom. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise you will accomplish the following:

- Create a directory to hold patches. Use `ftp` 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 `/var/sadm/patch` is empty.
- Read the README file associated with the patch to verify what Solaris release is appropriate for the patch. Add the patch and verify that it's installed. View the log for this patch found below `/var/sadm/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 `ftp` to transfer a patch from a classroom server into the directory you create. Use binary transfer mode. Your instructor will provide information about where to find a patch on the server. Close your `ftp` connection when finished. For example:

```
# cd /var/tmp
# ftp server1
(connection and login messages)
ftp> cd /export/patches
ftp> bin
200 Type set to I.
ftp> get 100000-01.zip
200 PORT command successful.
150 Binary data connection for 100000-01.zip
(192.9.200.1,32836) (49236 bytes).
226 Binary Transfer complete.
local: 100000-01.zip remote: 100000-01.zip
49236 bytes received in 0.045 seconds
(1072.57 Kbytes/s)
ftp> bye
221 Goodbye.
#
```

2. Use `unzip` to extract the patch from the zip archive. For example:

```
# unzip 100000-01.zip
```

3. Check to see if any patches are currently installed on your system.

```
# patchadd -p
```

`patchadd` makes the directory `/var/sadm/patch` the first time it runs.

4. Verify that the `/var/sadm/patch` directory is empty.

```
# ls /var/sadm/patch
```

5. Read the README file associated with the patch you unzipped. Verify the Solaris Release for which the patch is required.

```
# more 100000-01/README*
```

Solaris Release: _____

6. Add the patch.

```
# patchadd 100000-01
```

7. Verify that the patch is installed. What are the packages that the patch affects?

```
# patchadd -p
```

8. Examine the patch installation log.

```
# cd /var/sadm/patch/100000-01  
# more log
```

9. Remove the patch you just installed. Verify that the patch is no longer installed.

```
# cd  
# patchrm 100000-01  
# patchadd -p
```

Exercise Summary



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

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

3. Check to see if any patches are currently installed on your system.

patchadd should display the message:

No patches installed.

5. Read the README file associated with the patch you unzipped. Verify the Solaris Release for which the patch is required.

Solaris Release: 8

7. Verify that the patch is installed. What are the packages that the patch affects?

The packages affected by patches depend on the particular patches themselves.

9. Remove the patch you just installed. Verify that the patch is no longer installed.

patchadd should display the message:

No patches installed.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ List the locations to access patches
- ☐ Explain how to access patches from the World Wide Web and anonymous `ftp`
- ☐ Describe the different patch formats
- ☐ Prepare a patch for installation
- ☐ Install a patch using the `patchadd` command
- ☐ Demonstrate how to verify what patches are currently installed
- ☐ Remove a patch using the `patchrm` command

Objectives

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

- Identify the logical device names for tape drives
- Define the two different types of file system backups
- Backup a file system to tape using the `ufsdump` command
- Describe how to backup a file system to a remote tape drive
- Explain the purpose of the `/etc/dumpdates` file
- Restore a file system from tape using the `ufsrestore` command
- Describe the procedure for recovering file systems
- Use the `tar` command to manage multiple archives
- Use the `mt` command to control the actions of the tape drive

Additional Resources



Additional resources – The following reference can provide additional details on the topics discussed in this module:

- *Solaris 8 System Administration Guide, Volume I*, Sun Part Number 805-7228-10

Backing Up and Restoring File Systems

Backing up file systems is the task of copying file systems to removable media, such as tape, to safeguard against loss, damage, or corruption.

Restoring file systems means copying reasonably current backup files from removable media back to disk.

Importance of Regular File System Backups

Backing up file systems is one of the most crucial system administration functions. Backups should be performed on a regularly scheduled basis to prevent loss of data due to:

- Accidental deletion of files
- Hardware failures
- Problems when reinstalling or upgrading a system
- System crashes
- System break-in by an unauthorized user compromising data integrity
- Natural disasters

Tape Device Types

Figure 17-1 shows typical tape devices used for storing file systems during the backup process.

The media chosen depends on the availability of the equipment that supports it and the media selected to store the data.

Table 17-1 Tape Device Types

Media Type	Capacity
1/2-inch reel tape	140 Mbytes (6250 BPI)
1/4-inch (QIC) cartridge tape	8 Gbytes
8-mm cartridge tape	40 Gbytes
4-mm DAT cartridge tape	24 Gbytes
DLT 1/2-inch cartridge tape	70 Gbytes

The capacities shown are approximate and continue to increase. Check the documentation that comes with the tape device to determine its capacity.

Tape Device Naming

Logical Tape Device Names

All tape devices have logical device names that are used to reference the device on the command line. These logical device names use the following format:

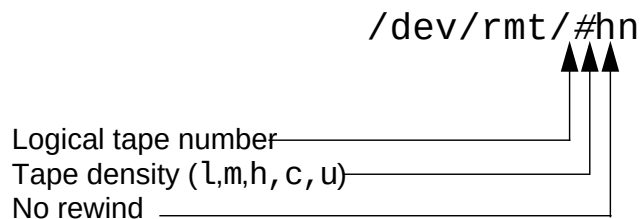


Figure 17-1 Logical Device Name Format

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

Tape device names are always numbered 0 and can include the following optional parameters:

- No Rewind: The letter “n” at the end of a tape device name indicates the tape is not to be rewound when the current operation completes.
- Tape Density: Five values can be given in the tape device name: l (low), m (medium), h (high), c (compressed), or u (ultra compressed).

Densities are tape-drive dependent. Check the manufacturer's documentation for the correct densities supported by a tape device.

The default can also be determined by device entries in the file `/kernel/drv/st.conf`.

Data Compression

Tape devices that support data compression contain internal hardware that performs the compression. Hardware-based compression is not as space efficient as using the Solaris `compress` command, though it is much faster.

Be aware that if a software compressed file is backed up using the tape device hardware compression option, the file will expand on tape to a size larger than its compressed version.

Types of File System Backups

As `root`, you can perform the following types of backups:

- Full – A complete file system backup
- Incremental – Only files in the file system that have been added or modified since a previous backup

The ufsdump Command

The `/usr/sbin/ufsdump` command is the recommended command for scheduled backups of complete file systems, as it is a resident command in the Solaris Operating Environment.

Note – Other backup programs are available from either Sun Microsystems, Inc., or third-party packages.

Command Format

```
ufsdump options [ arguments ] filesystem_name
```

You can use this command to back up a complete or a partial file system to backup media.

Common Options

The following are the common options for the `ufsdump` command:

- 0–9 – Backup Level. Level 0 is for a full backup of the whole file system. Levels 1 through 9 are for incremental backups of files that have changed since the last lower-level backup.
- v – Verify. After each tape is written, verify the contents of the media against the source file system. If any discrepancies occur, prompt the operator to insert new media, then repeat the process. Use this option only on an unmounted file system, any activity in the file system causes it to report discrepancies.

- S – Size estimate. Determines the amount of space needed on tape to perform the backup and display the estimated number of bytes required.
- l – Autoload. Use this option for an autoloading (stackloader) tape drive.
- o – Offline. When finished, take the drive offline, rewind (if tape), and if possible eject the media.
- u – Update the /etc/dumpdates file. An entry indicates the device name for the file system disk slice, the backup level (0-9), and the date. No record is written when the u option is not used. If an entry already exists for a backup at the same level, it is replaced.
- n – Notify all logged in users, who are members of the sys group, that ufsdump requires attention by sending messages to their terminals.
- f – Specify the tape device name where the file system will be copied. When the default tape device, /dev/rmt/0 is being used, it is not necessary to specify this device with the f option, it is assumed.
- *file system_to_backup* – Specify one of the following to be backed up. The file system's mount point name (e.g. /usr). The raw device name (/dev/rdisk/c#t#d#s#).

The /etc/dumpdates File

Each line in `/etc/dumpdates` file shows the file system backed up, the level of the last backup, and the day, date, and time of the backup.

The following is an example of a typical `/etc/dumpdates` file:

```
# cat /etc/dumpdates
/dev/rdisk/c0t2d0s6  0 Fri Jun 2   19:12:27  2000
/dev/rdisk/c0t2d0s0  0 Fri Jun 2   20:44:02  2000
/dev/rdisk/c0t2d0s4  5 Thu Jun 8   19:42:21  2000
```

When incremental backups are performed, the `ufsdump` command consults `/etc/dumpdates` to find the date of the most recent backup of the next lower level.

Then it copies all files that were modified or added since the date of that lower-level backup to the backup media.

After the backup is complete, a new entry, describing the backup just completed, replaces the entry for the previous backup at that level.

You can determine if backups are being done by viewing the `/etc/dumpdates` file. This is particularly important if a backup is not completed because of equipment failure, it will not be recorded in `/etc/dumpdates`.

Note – When restoring an entire file system, check `/etc/dumpdates` for a list of the most recent dates and levels of backups, to determine which tapes are needed to restore the entire file system.

Scheduling Backups

The dump level specified in the `ufsdump` command (0-9) determines which files are to be backed up. Specifying dump level 0 creates a full backup of the file system.

The numbers 1 through 9 are used to schedule incremental backups, but have no defined meanings. These are just a range of numbers used to schedule cumulative backups. The only meaning levels 1 through 9 have is in relationship to each other, as a higher or lower number.

Performing daily, cumulative incremental backups is the most commonly used backup scheme and is recommended for most situations. The following examples illustrate an incremental backup schedule for a particular file system.

A Sample Backup Strategy

The following is an example of using incremental levels to backup a file system.

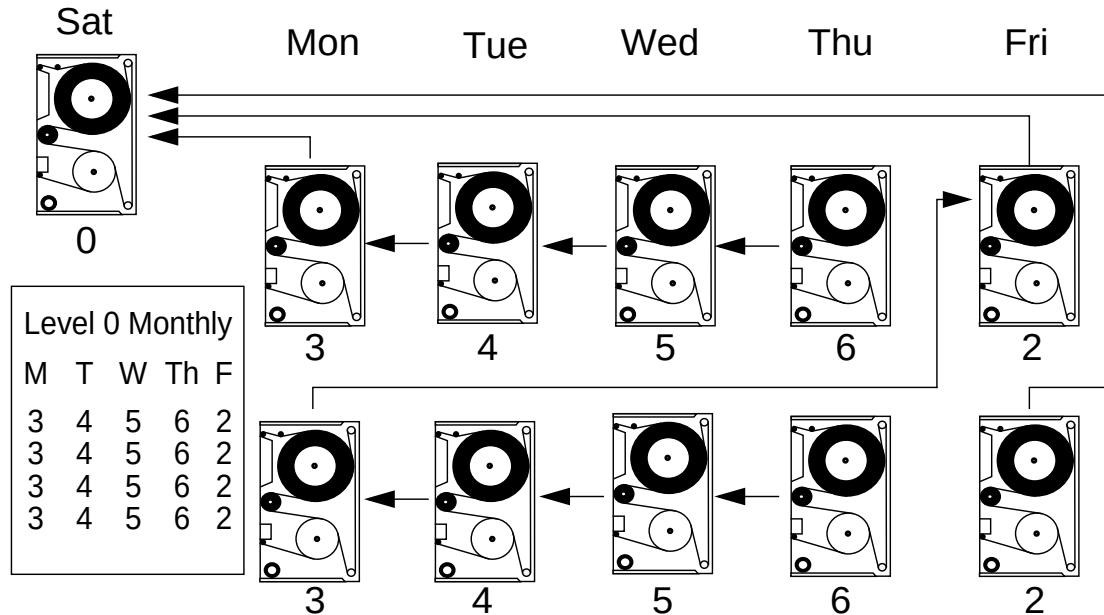


Figure 17-2 incremental Backup Strategy

- Full (level 0) backup is performed once each month.
- Level 3 backup is performed every Monday. Copies only new or modified files since the last lower level backup (for example, 0).
- Level 4 backup is performed every Tuesday. Copies only new or modified files since the last lower level backup (for example, 3).
- Level 5 backup is performed every Wednesday. Copies only new or modified files since the last lower level backup (for example, 4).
- Level 6 backup is performed every Thursday. Copies only new or modified files since the last lower level backup (for example, 5).
- Level 2 backup is performed every Friday. Copies only new or modified files since the last lower level backup, which is the level 0 backup at the beginning of the month.

Note – Most system administrators use the `crontab` utility to start a script that runs the `ufsdump` command.

Planning File System Backups

As part of determining a backup schedule, you need to choose:

- The file systems to backup
- The number of tapes for backup
- A backup device (for example, tape drive)
- The type of backup (for example, full or incremental)
- The procedures for marking and storing tapes

Finding File System Names

Display the contents of the `/etc/vfstab` file, and look at the `mount point` column for the name of the file system.

Determining the Number of Tapes

The size of the file system backup can be determined by using the following command.

For example:

```
# ufsdump 0S filesystem_name  
<number reported>
```

or

```
# ufsdump 3S filesystem_name  
<number reported>
```

The estimated number of bytes needed on tape to perform the backup is displayed.

Divide the reported size by the capacity of the tape to see how many tapes are needed to backup the file system.

Backing Up to Tape

You should bring the system to single-user mode and unmount the file system before doing a backup.

If you cannot unmount the file system, you need to be aware that backing up a file system, while operations, such as creating, removing, and renaming files are occurring, means some data will not be included in the backup.

1. Become `root` to bring 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 Jun 5 14:05:46 MDT 2000
```

```
Broadcast Message from root (pts/1) on host1 Mon Jun 5 14:05:46...
The system host1 will be shut down in 5 minutes
System is being shutdown for backup
```

2. Unmount all file systems (except `/` and `/usr`)

```
# umount /export/home
```
3. Check the integrity of the file system data with the `fsck` command, but only if the file system has been unmounted.

```
# fsck /export/home
```
4. Perform a full level 0 backup of the `/export/home` file system.

```
# ufsdump 0uf /dev/rmt/0 /export/home
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Mon Jun 5 2000 14:10:15 PM MDT
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t0d0s7 (host1:/export/home) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 125206 blocks (61.14MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 125182 blocks (61.12MB) on 1 volume at 747 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Mon Jun 5 2000 14:10:15 PM MDT
#
```

Performing Remote Backups

You can use the `ufsdump` command to perform a backup on a remote tape device.

When doing remote backups across the network, the system with the tape drive must have entries in its `/etc/hosts` file for every system that will be using the tape drive.

Command Format

`ufsdump options remotehost:tapedevice filesystem`

For example, to perform a full level 0 backup of the `/export/home` file system on `host1` to the remote tape device on `host2`, use the following command:

```
# ufsdump 0uf host2:/dev/rmt/0 /export/home
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Mon 5 Jun 2000 03:10:57 PM MST
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t0d0s7 (host1:/export/home) to
host2:/dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 125206 blocks (61.14MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 125182 blocks (61.12MB) on 1 volume at 704 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Mon 5 Jun 2000 03:10:57 PM MST
#
```

Restoring File Systems

Use the `ufsrestore` command to restore files and file systems that were backed up using the `ufsdump` command.

The reasons why a file system might need to be restored include:

- Rebuilding a damaged file system
- Reinstallation or upgrade of the Solaris Operating Environment software
- Reorganizing file systems on existing or new disks

The `ufsrestore` command copies files to disk, relative to the current working directory, from backups created using the `ufsdump` command.

Use `ufsrestore` to reload an entire file system hierarchy from a level 0 backup and incremental backups that follow it; or to restore one or more single files from any dump tape.

Command Format

```
ufsrestore options [ arguments ] [ filesystem ]  
ufsrestore options [ arguments ] [ filenames . . . ]
```

Common Options

The following describes the some options for the `ufsrestore` command:

- `t` – Lists the table of contents of the backup media.
- `r` – Restores the entire file system from the backup media.
- `x` – Restores only the files named on the command line.
- `i` – Invokes an interactive restore.
- `v` – Specifies Verbose mode. Displays pathnames to the terminal screen as each file is being restored.
- `f` – Specifies the tape device name.

The restoresymtable File

A `restoresymtable` file is created whenever restoring an entire file system from a backup tape. The `restoresymtable` file is used only by `ufsrestore` for *check-pointing*, which is information passed between incremental restores.

The `restoresymtable` file is not needed when the restore is complete and should be removed from the file system.

Preparing to Restore File Systems

The examples that follow demonstrate how to restore individual files; invoke restore in interactive mode to browse the contents of the backup tape; and restore an entire file system.

Before restoring files or file systems, you must determine the following:

- What file system backup tapes are needed
- The raw device name to restore the file system
- The temporary directory name to restore individual files
- The type of backup device to be used (local or remote)
- The backup device name (local or remote)

Restoring the *root* (/) File System

To restore the / (root) file system, boot from the Solaris CD-ROM and then run `ufsrestore`.

Note – If / (root), /usr, or the /var file system is unusable because of some type of corruption or damage, the system will not boot.

The following procedure demonstrates how to restore the / (root) file system on the boot disk `c0t0d0s0`.

1. Insert the Solaris 8 Software CD 1 of 2, and boot the CD-ROM with the single-user mode option.

```
ok boot cdrom -s
```

2. Create the new file system structure.

```
# newfs /dev/rdisk/c0t0d0s0
```

3. Mount the file system to an empty mount point directory, /a and change to that directory.

```
# mount /dev/dsk/c0t0d0s0 /a
# cd /a
```

4. Restore the / (root) file system from its backup tape.

```
# ufsrestore rf /dev/rmt/0
```

Note – Remember to always restore a file system starting with the level 0 backup tape and continuing with the next lowest level tape up through the highest level tape.

5. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

6. Install the `bootblk` in sectors 1–15 of the boot disk. Change to the directory containing the `bootblk`, and run the `installboot` command.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs
# installboot bootblk /dev/rdisk/c0t0d0s0
```

7. Unmount the new file system.

```
# cd /  
# umount /a
```
8. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdisk/c0t0d0s0
```
9. Reboot the system.

```
# init 6
```
10. Perform a full backup of the file system. For example:

```
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t0d0s0
```

Note – Always back up the newly created file system, as `ufsrestore` repositions the files and changes the inode allocation.

Restoring the /usr and /var File Systems

To restore the `/usr` and `/var` file systems repeat the steps described above, except step 6. This step is required only when restoring the (`/`) root file system.

Restoring Regular File Systems

To restore a regular file system, (for example, `/export/home`, or `/opt`) back to disk, repeat the steps described above, except steps 1, 6, and 9.

Example

```
# newfs /dev/rdisk/c#t#d#s#  
# mount /dev/dsk/c#t#d#s# /mnt  
# cd /mnt  
# ufsrestore rf /dev/rmt/#  
# rm restoresymtable  
# cd /  
# umount /mnt  
# fsck /dev/rdisk/c#t#d#s#  
# ufsdump 0uf /dev/rmt/# /dev/rdisk/c#t#d#s#
```

Invoking an Interactive Restore

The `ufsrestore i-i` command invokes an interactive interface for browsing through the backup tape's directory hierarchy and selects individual files to be extracted.

1. Become `root` and change to a temporary directory to place the extracted files.

```
# cd /var/tmp
```

2. Invoke the `ufsrestore` command with the interactive option.

```
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Mon June 01 15:17:09 2000
Dumped from: the epoch
Level 0 dump of / on host1:/dev/dsk/c0t3d0s0
Label: none
Extract directories from tape
Initialize symbol table.
```

3. Display the contents of the directory structure on the backup tape.

```
ufsrestore > ls
      2    *./          39      devices/    30847 net/
      2    *../         5122    etc/        15360 opt/
    161    .Xauthority  5120    export/    25611 proc/
    160    .Xdefaults  10240   home/      15381/sbin/
    159    .rhosts     40      kadb       35863 tmp/
   3085    .wastebasket/ 25608   kernel/    30848 tmp_mnt/
      3    bin         35      lib        20480 usr/
   3087    cdrom/      3       lost+found/ 25600 var/
  25610    dev/        20503   mnt/
```

To change directories on the backup tape:

```
ufsrestore > cd etc/inet
ufsrestore > ls
```

4. Add any file to be restored to the extraction list.

```
ufsrestore > add inetd.conf hosts
```

Files to be restored are marked with an asterisk (*) for extraction. If you are extracting a directory, all of its contents are marked for extraction.

In this example, two files are marked for extraction; and the `ls` command displays an asterisk in front of the selected file names: `*hosts` and `*inetd.conf`.

To delete a file from the extraction list, use the `delete` command:

```
ufsrestore > delete inetd.conf
```

The `ls` command displays `inetd.conf` without an asterisk.

5. To restore the selected file(s) from the backup tape:

```
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 ./etc/inet/hosts
Add links
Set directory mode, owner, and times.
set owner/mode for './'? [yn] n
```

6. Exit the interactive restore once the files are extracted.

```
ufsrestore> quit
```

7. Check the restored files, move them to their original or permanent directory location, and delete the files from the temporary directory.

```
# mv /var/tmp/etc/inet/hosts /etc/inet/hosts
# rm -r /var/tmp/etc
```

Note – Within an interactive restore, you can use the `help` command to display a list of available commands.

Controlling the Tape Drive

The `mt` command (magnetic tape control) is used to send instructions to the tape drive. Not all tape drives support all `mt` commands.

Command Format

```
mt [ -f tape-device-name ] command [ count ]
```

You use the `-f` option to specify the tape device name, typically a no-rewind device name.

- `status` – Displays status information about the tape drive.
- `rewind` – Rewinds the tape.
- `offline` – Rewinds the tape and, if appropriate, takes the drive unit off-line by unloading the tape.
- `fsf` – Forward skips *count* tape files.

Examples of Handling Multiple Archives

To create a tape archive of the current directory on the default tape drive, without the no rewind option, use the following command.

```
$ tar cvf /dev/rmt/0 .
```

The following example creates a tape archive of the current directory, on the default tape drive, using the no rewind option.

```
$ tar cvf /dev/rmt/0n /etc
```

This example positions the tape at the beginning of the third tar record.

```
$ mt -f /dev/rmt/0n fsf 1
```

To extract all files from tape and place them into the current directory, use the following command:

```
$ tar xvf /dev/rmt/0
```

Exercise: Backup and Recovery



Exercise objective – In this lab you will back up the / (root) file system, restore a single file from tape, and destroy and restore the root file system.

Preparation

Locate a tape appropriate for your system, from the Instructor. This exercise requires a system configured with a tape drive and a root (/) file system separate from /usr and /var. Identify the slice used to hold the root file system.

Task Summary

In this exercise you will accomplish the following:

Reboot the system to run state S. Use `ufsdump` 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 state 3.

- Use `ufsrestore` in interactive mode to restore the `/etc/inet/hosts` file from tape, and place it below `/var/tmp`.
- Recursively remove the `/kernel`, `/platform`, and `/devices` directories. Abort the operating system and attempt to boot the system from disk. Record what happens. Boot the system from the Solaris 8 Software CD 1 of 2 to run state S. Create a new file system on the root slice. Use `ufsrestore` to reload the root file system. Install a new boot block. Reboot the system and eject the CD-ROM.

Tasks

Creating a Backup of the / (root) File System

1. Log in as `root` and open a terminal window. Shut down the system to run state 0, then boot it to run state S. Supply the `root` password as required to enter run-state S.

```
# init 0  
(shutdown messages)  
ok boot -s
```

2. Insert a tape into your tape drive.
3. Use `ufsdump` to create a backup tape for the `root (/)` file system. For example:

```
# ufsdump 0uf /dev/rmt/0 /dev/rdisk/c0t0d0s0
```

4. Verify that the root file system has been recorded on tape:

```
# ufsrestore tvf /dev/rmt/0
```

5. Allow the system to continue to boot to run state 3.

```
# <Ctrl> d
```

Restoring /etc/inet/hosts from Tape

6. Log in as `root` and open a terminal window. Change directory to `/var/tmp`.

```
# cd /var/tmp
```

7. Run `ufsrestore` in interactive mode to retrieve the `/etc/inet/hosts` file from tape.

```
# ufsrestore if /dev/rmt/0  
ufsrestore > ls
```

8. Change directory to `/etc/inet` on tape, and list the files it contains.

```
ufsrestore > cd /etc/inet  
ufsrestore > ls
```


9. Add the `hosts` file to the list of files to extract. Display the list.

```
ufsrestore > add hosts
ufsrestore > marked
```
10. Extract the `hosts` file from tape. Specify volume number 1, do not set the owner and mode for ".", then quit `ufsrestore`.

```
ufsrestore > extract
Specify next volume #: 1
set owner/mode for '.'? [yn] n
ufsrestore > q
```
11. Verify that `etc/inet/hosts` exists below `/var/tmp`.

```
# ls etc/inet/hosts
```

Destroy and Restore the root Filesystem

12. Change directory to `/` and remove critical system files.

```
# cd /
# rm -r /kernel /platform /devices
```
13. Use `Stop-a` to abort the operating system. Attempt to boot the system from the boot disk. What happens?

```
<Stop> a
ok boot
```

14. Insert the Solaris 8 Software 1 of 2 CD. Boot the system from the CD to run state `S`.

```
ok boot cdrom -s
```
15. Use `newfs` to create a new file system on the root slice. Be certain to select the correct slice. The slice should match the one you used in step 3. Run `fsck` on the file system you create. For example:

```
# newfs /dev/rdisk/c0t0d0s0
# fsck /dev/rdisk/c0t0d0s0
```
16. Insert your root backup tape in the tape drive. Mount the new file system as `/a`. Change directory to `/a`. For example:

```
# mount /dev/dsk/c0t0d0s0 /a
# cd /a
```

17. Use `ufsrestore` to load the root data into the new file system.

```
# ufsrestore rf /dev/rmt/0
```

18. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

19. Install a new boot block in sectors 1-15 of the root slice. To do this, change to the directory containing the boot block, and run `installboot`.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs  
# installboot bootblk /dev/rdisk/c0t0d0s0
```

20. Change directory to `/`, and unmount the new file system.

```
# cd /  
# umount /a
```

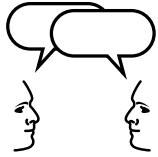
21. Reboot the system.

```
# reboot
```

22. Log in as `root` and open a terminal window. Eject the Solaris 8 Software CD 1 of 2.

```
# 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

Task Solutions

13. Use **Stop-a** to abort the operating system. Attempt to boot the system from the boot disk. What happens?

*The system fails to boot and displays the message: **bootblk: can't find the boot program***

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ☐ Identify the logical device names for tape drives
- ☐ Define the two different types of filesystem backups
- ☐ Backup a filesystem to tape using the `ufsdump` command
- ☐ Describe how to backup a filesystem to a remote tape drive
- ☐ Explain the purpose of the `/etc/dumpdates` file
- ☐ Restore a filesystem from tape using the `ufsrestore` command
- ☐ Describe the procedure for recovering filesystems
- ☐ Use the `tar` command to manage multiple archives
- ☐ Use the `mt` command to control the actions of the tape drive

New Features of the Solaris 8 Operating Environment



The following key features are the highlights of the Solaris 8 Operating Environment release:

- Internet Protocol version 6 (IPv6) adds increased address space and improves Internet functionality using a simplified header format, support for authentication and privacy, autoconfiguration of address assignments. IPv6 enables new quality-of-service capabilities.
- The Solaris 8 Operating Environment provides the Naming Service switch back-end support for Lightweight Directory Access Protocol (LDAP) based directory service.
- The Java™ 2 Software Development Kit (SDK) for Solaris significantly improves scalability and performance of Java applications.
- The Solaris 8 Operating Environment supports the Universal Disk Format (UDF) file system, enabling users to exchange data stored on CD-ROMs, disks, diskettes, DVDs, and other optical media.
- The Solaris Smart Card feature enables security administrators to protect a computer desktop or individual application by requiring users to authenticate themselves by means of a smart card.
- The PDA Synchronization (PDA sync) application synchronizes the data from applications such as Desktop Calendar, Desktop Mail, Memo, and Address, with data in similar applications on a user's Personal Digital Assistant (PDA).
- The Solaris 8 Software CDs and Languages CD include support for more than 90 locales, covering 37 languages.

- The Solaris Common Desktop Environment (CDE) contains new and enhanced features that incorporate easy to use desktop productivity tools, PC interoperability, and desktop management tools.
- The X Server is upgraded to the X11R6.4 industry standard which includes features that increase user productivity and mobility, including remote execution of X applications through Web browser on any Web-based desktop, Xinerama, Color Utilization Policy, EnergyStar support, and new APIs and documentation for the developer tool kits.
- The Solaris Installation CD provides a graphical, wizard-based, Java powered application to install the Solaris Operating Environment and other software.

Table 1-1 through Table 1-17 provide a more comprehensive description of the new features available in the Solaris 8 Operating Environment.

Table 1-1 Next Generation Internet Protocol

Feature	Description
IPv6	IPv6 adds increased address space and improves Internet functionality using a simplified header format, support for authentication and privacy, and autoconfiguration of address assignments. It enables new quality-of-service capabilities.

Table 1-2 Directory Services and Naming Enhancements

Feature	Description
Native LDAP	Native Lightweight Directory Access Protocol (LDAP) provides the Naming Service switch back-end support for LDAP based directory service. With the Solaris 8 Operating Environment, network administrators can now specify LDAP as the desired name service to directory entry access by copying the <code>nsswitch.ldap</code> template file to <code>/etc/nsswitch.conf</code> .

Table 1-3 Installation and Management

Feature	Description
Solaris Web Start enhanced installation CD	Solaris Web Start, a graphical, wizard-based, Java powered software application that installs the Solaris Operating Environment and other software, is now distributed on a separate installation CD.
Booting a system over the network with Dynamic Host Configuration Protocol (DHCP)	Configuration Protocol (DHCP) Network installs can now use DHCP to acquire boot parameters and network configuration information needed to boot a client over the network. DHCP booting is supported on certain SPARC and IA based systems.
DHCP Manager	DHCP Manager provides a Java technology-based graphical interface for configuring and managing the Solaris DHCP server and DHCP databases. It allows the system administrator to use a single tool to perform all DHCP management duties: set up and manage DHCP servers, manage client configuration options and macros, and manage networks and IP addresses that are under DHCP management.
Solaris Web-Based Enterprise Management (WBEM) Services	Solaris WBEM Services software is an implementation of Web-Based Enterprise Management (WBEM) standards and technologies in the Solaris Operating Environment. Intended for developers and administrators of WBEM-enabled environments, Solaris WBEM Services provides the Solaris Schema, extensions of the CIM Schema classes, and management, security, and logging services.
Support for IPv6 in system identification utilities	Systems can now be configured at install time to use IPv6 in addition to IPv4.

Table 1-3 Installation and Management (Continued)

Feature	Description
Support for domain name system (DNS) in system identification utilities	Solaris WBEM Services software is an implementation of Web-Based Enterprise Management (WBEM) standards and technologies in the Solaris Operating Environment. Intended for developers and administrators of WBEM-enabled environments, Solaris WBEM Services provides the Solaris Schema, extensions of the CIM Schema classes, and management, security, and logging services.
Support for domain name system (DNS) in system identification utilities	Identification utilities DNS has been added to the list of name services that can be configured through the system identification utilities.
Unlimited number of pseudo-terminals available	Solaris 8 software enables the opening on any number of pseudo-terminals (used by programs like <code>rlogin</code> and <code>telnet</code>).
Reading documentation from the Solaris 8 Documentation CD	The <code>ab2cd</code> script enables all users to read AnswerBook2™ documentation directly from the Solaris 8 documentation CD. It has been enhanced to provide better user feedback, to enable users to set the port number on which <code>ab2cd</code> runs, and to read documentation already installed on the user's system.
Product Registry	The Solaris Product Registry is a tool to manage software installed using Solaris Web Start 3.0 or the Solaris package management commands (<code>pkgadd</code>, for example). It enables you to: • View a list of installed and registered software and some software attributes • Install additional software products • Uninstall software • Browse for and launch an installer
IA: Boot partition in Solaris 8	Users running Solaris <i>Intel Platform Edition</i> must now designate a separate IA boot partition when using Solaris Web Start to upgrade to Solaris 8 <i>Intel Platform Edition</i> .

Table 1-3 Installation and Management (Continued)

Feature	Description
IA: CD-ROM boot	This new feature enables the user to boot a system from an installation CD (rather than the Device Configuration Assistant diskette, as was the case in the past) using the "El Torito" standard.
IA: Large disk support	By using improved BIOS interfaces to access the disk, Solaris 8 <i>Intel Platform Edition</i> now fully uses disks larger than 8 Gbytes.

Table 1-4 File System Enhancements

Feature	Description
Universal Disk Format (UDF) file system	The UDF file system, the industry-standard format for storing information on optical media technology, is supported in this Solaris release. The UDF file system can be used to exchange data on the following components when they contain a UDF file system: • CD-ROMs • Disks and diskettes • Digital versatile disc or digital video disc (DVD) DVD-ROM on supported platforms
NFS server logging	NFS server logging allows an NFS server to provide a record of file operations performed on its file systems. This feature is particularly useful for sites that make anonymous FTP archives available to NFS and WebNFS™ clients.
WebNFS JavaBeans component	The WebNFS JavaBeans™ component contains an XFileChooser class that extends the JFileChooser graphical component of the Java 2 API. This bean can be used by any Java 2 application that needs to display a file chooser to enable users to select a file for input (open) or output (save). Using XFileChooser an application can access a file on a local disk or on an NFS server through the use of NFS URL naming.

Table 1-4 File System Enhancements (Continued)

Feature	Description
Deferred access time updates on UFS file systems	Two new mount options, <code>dfratime</code> and <code>nodfratime</code> enable and disable deferred access time updates on UFS file systems. When enabled, writing access time updates for the file system may be deferred until the disk is accessed for a reason other than updating access times.
IA: Extended Memory (XMEM) support	XMEM support provides a mechanism that allows a single 32-bit process to efficiently allocate and manage more than 4 Gbytes of physical memory. The XMEM feature is implemented as a file system (<code>xmemfs</code>) that system administrators can mount and use to reserve memory for applications.

Table 1-5 Diagnostic and Availability Enhancements

Feature	Description
The <code>coreadm</code> command	The <code>coreadm</code> command provides flexible core file naming conventions and better core file retention.
Examining core files with <code>proc</code> tools	Some of the <code>proc</code> tools have been enhanced to examine process core files as well as live processes. The <code>proc</code> tools are utilities that can manipulate features of the <code>/proc</code> file system.
Improved device configuration (<code>devfsadm</code>)	The <code>devfsadm</code> command provides an improved mechanism for managing the special device files in the <code>/dev</code> and <code>/devices</code> directories, including support for dynamic reconfiguration events.
Improved system error messages	The system boot and error message format now provides a numeric identifier, module name, and time stamp to messages generated by the <code>syslog(1M)</code> logging facility. In addition, messages that were previously lost after a system panic and reboot are now saved.
Modular debugger (<code>mdb</code>)	The <code>mdb</code> command is a new extensible utility for low-level debugging and editing of the live operating system, operating system crash dumps, user processes, user process core dumps, and object files.

Table 1-5 Diagnostic and Availability Enhancements (Continued)

Feature	Description
Remote console messaging	This release includes the <code>consadm</code> command, which enables you to select a serial device as an auxiliary (or remote) console for troubleshooting remote system problems.
TCP/IP internal trace support	TCP/IP now provides internal trace support by logging TCP communication when a connection is terminated by a reset (RST) packet.

Table 1-6 Performance and Scalability Enhancements

Feature	Description
apptrace	A new application debugging tool, <code>apptrace</code> enables application developers and system support personnel to debug application or system problems by providing call traces to Solaris shared libraries, which may show the series of events leading up to a point of failure.
SPARC: busstat	A new system monitoring tool, <code>busstat</code> provides access to bus-related performance counters on supported SPARC platforms. Viewing these performance counters with <code>busstat</code> enables you to measure hardware clock cycles and bus statistics including DMA and cache coherency transactions on a multiprocessor system.
Faster boot for servers	Large servers now require significantly less time to boot.
New alternative to <code>poll()</code> interface	A second form of polling, <code>/dev/poll</code> is for the completion of I/O events and provides much higher performance when a large number of events must be polled for on-file descriptors that remain open for a long time. This feature supplements <code>poll(2)</code> ; it does not replace <code>poll(2)</code> .
prstat	The <code>prstat</code> utility iteratively examines all active processes on the system and reports various statistics based on the selected output mode and sort order.

Table 1-6 Performance and Scalability Enhancements (Continued)

Feature	Description
IA: Xeon enhancements	To maximize performance, Solaris 8 Intel Platform Edition now supports the Page Attribute Table (PAT) feature of IA32-bit processors (Pentium II and Pentium III).
IA: Added support for Physical Address Extension (PAE) mode	With the release of Pentium Pro, Intel introduced a mode called PAE on its advanced processors. By using PAE, Solaris Intel Platform Edition can address up to 32 Gbytes of physical memory.

Table 1-7 Security Enhancements

Feature	Description
Solaris Smart Cards	The Solaris Smart Card feature implements the Open Card Framework (OCF) 1.1 standard. Security administrators can use this technology to protect a computer desktop or individual application by requiring users to authenticate themselves by means of a smart card.
Default file system and directory permissions	Many system files and directories have different default ownership and more strict permissions than in previous releases.
Role-Based Access Control (RBAC)	Traditional superuser-based systems grant full superuser powers to anyone who can become superuser. With RBAC, administrators can assign limited administrative capabilities to normal users.
Centralized administration of user audit events	The file, <code>/etc/security/audit_user</code> , which stores audit preselection classes for users and roles, is now supported in the name switch. It is no longer necessary to set up the audit events for a user on each system to which the user has access.

Table 1-8 Realtime Systems Enhancement

Feature	Description
High resolution timers	High resolution timers The high resolution timers (HRTs) bypass the traditional 10 millisecond clock interface to expose the granularity of the physical clock interrupt from the hardware. Thus, the HRT interface allows a real time process to take control of one processor (of a multi-processor system) and operate to any required degree of precision in timing events
User-level priority inheritance	The real-time threads feature implements the POSIX interfaces (previously only dummied in) that let the high priority thread "lend" its priority to the low priority thread until it releases the lock.

Table 1-9 Common Desktop Environment (CDE) Desktop Enhancements

Feature	Description
Personal Digital Assistant (PDA) support	The PDA Synchronization (PDASync) is a Java-based application that enables users to easily synchronize their desktop calendar, mail, address book, and memos with their PDA.
Hot Key Editor	The Hot Key Editor enables users to predefine a series of commands to a given function key, resulting in increased productivity and efficiency.
Java Media Framework (JMF)	The JMF, a Java-based application, provides smooth streaming video file format support for MPEG1, MPEG2, Quicktime, and AVI, as well as audio support for MIDI. This feature enables users to take advantage of the real-time video creation and broadcast functionality.
SPARC: Audio Mixer	CDE now includes a new GUI tool, <code>sdtaudiocontrol</code> , that supersedes <code>audiocontrol</code> . <code>sdtaudiocontrol</code> , uses the features of the audio mixer, and provides more features.

Table 1-9 Common Desktop Environment (CDE) Desktop Enhancements

Feature	Description
SPARC: PC launcher 1.0	PC launcher 1.0 for SunPCi enables users to get seamless access and power to view, edit, and print many popular types of PC files or attachments instantly, by automatically launching the associated Windows application and file.
Netscape Application Launcher	The Netscape™ Application Launcher enables users to easily access and automatically launch Netscape files and associated Netscape applications such as Composer. This feature eliminates the need to run the entire Netscape environment, simplifying access to Netscape applications.
Print Client enhancements	Print Client now enables users to easily configure their own set of printers and default printer without any intervention from an administrator.
SDTImage enhancements	The SDTImage screen snapshot feature now enables users to easily and quickly capture a screenshot image from the command line.
Smart card support	CDE now supports smart card authentication security technology. Users can now use smart cards to authenticate their identity when logging in to CDE on a protected system, relogging in after a screen lock, or reauthenticating after the smart card is removed. CDE supports both external and internal smart card devices
ToolTips	ToolTips provides users with Balloon Help, a simple and short description of an icon function
X11R6.4 support	The X Server is upgraded to the X11R6.4 industry standard which includes key features that increase user productivity and mobility, including remote execution of X applications through Web browser on any Web-based desktop, Xinerama, Color Utilization Policy, EnergyStar support, and new APIs and documentation for the developer tool kits.
Extended control panel	This feature provides a unified, consistent, and extensible launchpad for desktop customization, such as desktop controls for color, font, backdrop, and Application Manager.

Table 1-10 Web Services

Feature	Description
Java Plug-in	Java Plug-in for the Solaris Operating Environment is an add-on product for Netscape Navigator™ that enables Java applets and JavaBeans components to run on Web pages using Java Runtime Environment (JRE) 1.2 instead of the default Java Virtual Machine (JVM) bundled with Navigator.
Netscape Communicator 4.7	Solaris 8 includes Netscape Communicator 4.7 and now installs it by default on your system.
Solaris Network Cache and Accelerator (NCA)	The Solaris NCA increases Web server performance by maintaining an in-kernel cache of Web pages accessed during HTTP requests.
Apache Web server	The open source Apache Web server is now bundled with Solaris. It includes all the standard Apache modules, including proxy server support, as well as the <code>mod_perl</code> module.

Table 1-11 Printing

Feature	Description
Print naming enhancement	This Solaris release supports the printers database in <code>/etc/nsswitch.conf</code> , the name service switch file. The printers database provides centralized printer configuration information to print clients on the network
Solaris Print Manager	Solaris Print Manager is a Java-based graphical user interface that enables you to manage local and remote printer access. This tool can be used in the following name service environments: NIS, NIS+, NIS+ with Federated Naming Service (FNS), and files.

Table 1-12 Language Support

Feature	Description
Universal language coverage	The Solaris 8 Operating Environment now includes support for more than 90 locales, covering 37 languages, on both the Solaris 8 Software CDs and the Solaris 8 Languages CD.
Improved language installation and setup	Changes to packaging on the language CD have reduced the storage requirements for a mixed language installation. A redesign of the install interface makes language selection and grouping extremely intuitive.
Expanded Unicode support	Solaris 8 continues to broaden support for Unicode with the addition of new Unicode (UTF-8) locales for Simplified Chinese and Traditional Chinese.
Customer-extensible codeset conversion (geniconvtbl)	With the Solaris 8 Operating Environment, developers can easily create and add to the Solaris system their own user-defined codeset conversions by using the geniconvtbl utility. Modification to existing Solaris codeset conversions is also supported.
Improved data interoperability	Data interoperability with non-Solaris environments has been improved in Solaris 8 with the addition of the following new iconv data conversion utilities: • iconv for Japanese mainframe data types • iconv for Microsoft data encodings (including user defined characters) • iconv for UTF-8 interoperability in China and Korea • iconv for various Unicode encoding formats and international and de facto industry standard codesets
New locales added	Two new locales have been added to Solaris 8 for Iceland (ISO8859-15) and Russia (ANSI1251). The new Russian locale is in addition to the existing Russian (8859-5) locale and provides native Microsoft data encoding support.

Table 1-13 Documentation

Feature	Description
AnswerBook2 Documentation Server updates	The AnswerBook2 Documentation Server has been updated for this release. Major changes since the Solaris 7 release include replacing the AnswerBook2 navigation icons with text, improved support for non-English locales, and minor changes to improve overall performance and stability.
Reference Manual reorganization	• The section of the SunOS Reference Manual that describes the C library functions (but does not include the system calls) now contains six books instead of one. These books are • Library Interfaces and Headers • Basic Library Functions • Networking Library Functions • Threads and Realtime Library Functions • Extended Library Functions • Curses Library Functions In addition, many of the man page suffixes have been changed to reflect the library that contains the function.

Table 1-14 Software Developer Environment

Feature	Description
SPARC: 64-bit Kodak Color Management System (KCMS) libraries	Kodak Color Management System™ (KCMS™) is now providing a 64-bit version of the libraries. Applications that currently use KCMS and are converted to the 64-bit operating environment can now retain color management.
Always ready Power Management™	With the Solaris 8 Operating Environment, a device driver using the new device Power Management interfaces will be power managed automatically.
cpustat and cputrack commands	The new cpustat and cputrack commands capture system-wide and per-process CPU statistics respectively, to monitor the performance of a system or a process.

Table 1-14 Software Developer Environment (Continued)

Feature	Description
Practical Extraction and Report Language (Perl) 5	The popular programming language, Perl 5.005_03, is included in the Solaris 8 release. Perl is commonly used for CGI scripting as well as automating complex system administration tasks.
Role-based access control (RBAC) for developers	The addition of RBAC to the Solaris Operating Environment gives developers the opportunity to deliver fine-grained security in new and modified applications. Developers can now create privileged functions that check for authorizations instead of checking for specific IDs such as superuser.
Secure path name change from /usr/lib to /usr/lib/secure	The secure directory from which files can be preloaded is now /usr/lib/secure for 32-bit objects and /usr/lib/secure/sparcv9 for 64-bit SPARCV9 objects.
Dynamic string token support	Greater flexibility in establishing instruction set specific, and system specific dependencies is provided with the new \$ISALIST, \$OSNAME, and \$OSREL dynamic string tokens.
strftime() function update	The %u conversion specification for the strftime() function represents a weekday as a decimal number [1,7], with 1 now representing Monday (rather than Sunday, as was the case in the Solaris 7 Operating Environment). This new behavior conforms to the X/Open CAE Specification, System Interfaces and Headers.
Alternate one-level libthread	An alternate threads implementation provides a one-level model in which user-level threads are associated one-to-one with lightweight processes (LWPs). This implementation is simpler than the standard implementation and may be beneficial to some multithreaded applications.
SPARC: audio mixer driver	The audio mixer driver now allows multiple applications to play and record audio simultaneously.
Updated DDI interfaces for cluster-aware device drivers	A documentation overview introduces the concept of device classes and the necessary interface modifications and additions for device driver writers.

Table 1-14 Software Developer Environment (Continued)

Feature	Description
8-bit visual support	The 8-bit visual shared library enables device drivers with only 24-bit hardware to display 8-bit visual applications.

Table 1-15 IA Hardware Enhancements

Feature	Description
Advanced Configuration and Power Interface (ACPI)	ACPI is a new, more flexible way to configure and control IA hardware. ACPI obsoletes Plug and Play BIOS and the Intel Multi-Processor Specification (MPSPEC). If ACPI is available on your IA based system, Solaris 8 automatically uses it to configure the hardware.
PCI hot-plug support	This feature enables standard PCI adapters to be hot-plugged into a machine with the hot-plug capability that is running Solaris Intel Platform Edition. You can now add (hot-add) or remove (hot-remove) adapters from a system while the system is still running.
Universal Serial Bus (USB) support	Solaris Intel Platform Edition now provides USB support for keyboards and mouse devices
X Server video driver enhancement	Solaris Intel Platform Edition now provides support for more video devices.

Table 1-16 IA SCSI Drivers

Feature	Description
IA: cadp driver enhancements	The Solaris cadp driver now supports Adaptec Ultra2 adapters.
IA: ncrs device driver enhancements	The Solaris ncrs device driver now supports the SCSI hot-plugging functionality and Ultra2 devices, in addition to general functionality and performance improvements.

Table 1-16 IA SCSI Drivers (Continued)

Feature	Description
IA: symhisl device driver	The symhisl device driver, which supports the adapters SYM22910 and SYM21002, is now included in Solaris <i>Intel Platform Edition</i> .

Table 1-17 Other Software

Feature	Description
Early Access Software	The Solaris 8 release includes an Early Access (EA) directory with EA software. For more information, refer to the README on the Solaris Software CD 2 of 2.
Freeware	Several freeware tools and libraries are included in the Solaris 8 release. These tools assist the development of tools for administration and development tasks. These tools include: • bash – Sh-compatible command language interpreter • bzip2 – Block-sorting file compressor • gpatch – Used to apply patch files to originals • gzip – GNU zip compression utility • less – A pager, like more • libz – Also known as 'zlib', this is a library that performs compression (specifically, RFCs 1950-1952) • mkisofs – Builds a CD image using an iso9660 file system • rpm2cpio – Transforms a package in RPM format (Red Hat Package Manager) to a cpio archive • tcsh – C shell with file name completion and command line editing • zip – Compression and file packaging utility • zsh – Command interpreter (shell) usable as an interactive login shell and as a shell script command processor

fsck – *Handling Error Messages* B

This appendix provides recommendations for using the fsck command to handle error messages.

The Phases of the fsck Command

The following sections describe the phases of the fsck command.

Initialization Phase

During the initialization phase of fsck, the invoking command-line syntax is parsed and checked. If there are no such errors, tables are set up and files are opened.

Errors during the phase include:

- bad inode number *inode-number* to gnode

A non-existent inode caused this internal error. fsck will exit.

- cannot alloc *size-of-block map* bytes for blockmap

Variations on this error include failure to allocate memory for the freemap and statemap.

- Can't open checklist file: *filename*

Typically, the /etc/vfstab file either does not exist or cannot be opened. fsck will exit. Check that the file exists and can be read.

- Can't open *filename*

This error is caused because the file system, *filename*, cannot be opened. If `fsck` is running in interactive mode, this error will be ignored and `fsck` will continue. Check that the reading and writing of the raw device for this file system are enabled.

- Can't stat root

This error is caused because `fsck` failed in its request for root directory statistics. `fsck` will exit. Restore the file system from the most recent backup.

- Can't stat *filename*

A variation of this error message might read:

Can't make sense out of name *filename*

This error indicates that a request for statistics about file system, *filename*, failed. Check the access modes.

- filename: (NO WRITE)

Either the `-n` option was specified or `fsck` could not open the file system, *filename*, for writing. When `fsck` is running in no-write mode, all diagnostic messages are displayed, but `fsck` does not attempt to fix anything.

- IMPOSSIBLE MINFREE=*percent* IN SUPERBLOCK (SET TO DEFAULT)

This error occurs when the superblock minimum space *percentage* is either greater than 99 percent or less than 0 percent. Entering `y` at the SET TO DEFAULT prompt will set the minfree parameter to the default of 10 percent; entering `n` at this prompt causes the system to ignore the error condition.

- INTERNAL INCONSISTENCY: *message*

message can be any of the following:

- ▼ MAGIC NUMBER WRONG
- ▼ SIZE PREPOSTEROUSLY LARGE
- ▼ TRASHED VALUES IN SUPER BLOCK



Caution – Use the `-b` option for `fsck` to indicate the use of an alternate superblock copy. The `-N` option for `newfs` will display alternate copies. Be sure to specify `-N`, otherwise, `newfs` will overwrite with a new file system.

Phase 1

During this phase, `fsck` examines the integrity of the inodes. It checks inode types, inode block numbers for bad or duplicate blocks, and inode sizes and formats.

Error messages displayed during this phase include:

- **`block-number BAD I=inode-number`**

This error is generated when the inode references a block number that is out of the possible range of block numbers for the file system.

This error condition can generate the `EXCESSIVE BAD BLKS` error message later in this phase if too many block numbers are out of range.

This error condition also generates the `BAD/DUP` error message in phases 2 and 4.

- **`BAD MODE: MAKE IT A FILE?`**

This error occurs because the inode contains a bad mode data element. Answering `y` to the prompt sets the mode to a reasonable value. Then, if this error continues to occur, it is an indication of possible disk damage.

- **`BAD STATE state-number TO BLKERR`**

This error occurs when the `fsck` state map is corrupted and shows an impossible state number. Restoring from a recent backup is appropriate when this happens.

- **`block-number DUP I=inode-number`**

This is an indication that more than one inode is claiming the same data *block-number*. This error condition can generate the `EXCESSIVE DUP BLKS` error message later in phase 1 if the inode has too many block numbers.

This error condition invokes phase 1B and generates the BAD/DUP error messages in phases 2 and 4.

- DUP TABLE OVERFLOW (CONTINUE)

This error occurs when fsck has no more room in its internal table that records duplicate block numbers. If fsck is running in preen mode, it will exit. Otherwise, typing y to the CONTINUE prompt will instruct fsck to continue checking. If the error reoccurs, kill processes and/or augment swap space to increase virtual memory and re-run fsck until it completes without error.

- EXCESSIVE BAD BLOCKS I=*inode-number* (CONTINUE)

This error occurs when too many (more than 10) data blocks for an inode are out of the range of possible block numbers for the file system.

Answering y to the CONTINUE prompt will instruct fsck to continue checking but the error is not cleared until an fsck session completes without an error.

- EXCESSIVE DUP BLKS I=*inode-number* (CONTINUE)

This error is similar to the EXCESSIVE BAD BLOCKS I=*inode-number* error except it applies to too many duplicate data blocks being claimed by an inode or the inode free list.

Answering y to the CONTINUE prompt will instruct fsck to continue checking but the error is not cleared until an fsck session completes without error.

- INCORRECT BLOCK COUNT I=*inode-number* (*number-of-BAD-DUP-or-missing-blocks* should be *number-of-blocks-in-filesystem*) (CORRECT)

This error indicates an inconsistency between the number of blocks claimed by an inode and the number of blocks in the file system. Answering y to the CONTINUE prompt will correct this error by replacing the former value with the latter.

- LINK COUNT TABLE OVERFLOW (CONTINUE)

This error occurs when fsck has no more room in its internal table that records allocated inodes with a link count of zero. If fsck is running in preen mode, it will exit. Otherwise, typing y to the CONTINUE prompt will instruct fsck to continue checking. If the

error reoccurs, you should kill processes and/or increase swap space to increase virtual memory and re-run `fsck` until it completes without an error.

- **PARTIALLY ALLOCATED INODE** *I=inode-number* (CLEAR)

This error occurs when an inode is neither allocated nor unallocated. This error is corrected when `fsck` is run in preen mode, otherwise answering `y` to the **CLEAR** prompt will clear the bad inode by zeroing out its contents.

If an inode is cleared, an **UNALLOCATED** error message will occur during phase 2 for all directories that contain this inode.

- **PARTIALLY TRUNCATED INODE** *I=inode-number* (SALVAGE)

This error occurs if the system crashes while truncating a file causing the inode size to be too small to contain the number of blocks allocated to it.

Answering `y` to the **SALVAGE** prompt completes the interrupted file truncation.

- **UNKNOWN FILE TYPE** *I=inode-number* (CLEAR)

This error occurs if the mode data element in the inode does not indicate a valid file type (pipe, special character, special block, regular, symbolic link, first-in, first-out [FIFO] file, or directory).

If `fsck` is running in preen mode, this condition is corrected. Otherwise, answering `y` to the **CLEAR** prompt will clear the inode which will cause an **UNALLOCATED** error during phase 2 for any directories that contain this inode.

Phase 2

During the previous phase, `fsck` may have discovered bad inodes that have been cleared, influencing the integrity of the directories that reference them. In phase 2 `fsck` continues with repairs by focusing on the integrity of the directories.

Error messages displayed during this phase include:

- **BAD INODE** *state-number* **TO DESCEND**

This error occurs when `fsck`, during internal processing, passes a bad *state-number* to the descend routine. (The descend routine traverses file system hierarchy many times during an `fsck` operation.)

Restoring from a recent backup is the best recourse.

The following error conditions are similar because they arise from bad *state-numbers* returned or found in inodes:

- ▼ BAD RETURN STATE *state-number* FROM DESCEND
- ▼ BAD STATE *state-number* FOR ROOT INODE
- ▼ BAD STATE *state-number* FOR INODE=*inode-number*
- BAD INODE NUMBER FOR '.' I=*inode-number* OWNER=UID
MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* (FIX)

The data of directories contains inodes of files that are contained in that directory. This includes entries for itself (.) and its parent directory (..). This error occurs when there is a mismatch between the inode number of the directory and the number which is in the directory's file listing.

Answering `y` to the FIX prompt alters the . inode number entry to match the inode number of the directory itself.

There is a similar error condition for a mismatch between the .. entry and the actual inode of the parent directory.

BAD INODE NUMBER FOR '..' I=*inode-number* OWNER=UID
MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* (FIX)

- DIRECTORY TOO SHORT I=*inode-number* OWNER=UID
MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* (FIX)

This error occurs when a directory size is smaller than the minimum directory size. Answering `y` to the FIX prompt adjusts the size to the required minimum.

The following error is similar in that the directory size is not a multiple of the block number size:

DIRECTORY *filename*: LENGTH *file-size* NOT MULTIPLE OF *block-number* (ADJUST)

- DIRECTORY CORRUPTED I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* (SALVAGE)

This error indicates a corrupted internal state for a directory. Answering y to the SALVAGE prompt is a drastic response that may cause the loss of some directory data. However, since directory data is nothing more than mappings of inode numbers to file names, the data of the referenced files themselves may survive and just become unreferenced. These unreferenced files may, during phase 3, be relinked to the lost+found directory and be recoverable.

- DUP/BAD I=*inode-number* OWNER=0 MODE=M SIZE=*file-size* MTIME=*modification-time* TYPE=*filename* (REMOVE)

This error occurs when duplicate or bad data blocks are found for a directory or file. When running in preen mode, these duplicate or bad block references are removed. Answering y to the REMOVE prompt will accomplish the same thing.

This error is similar to the following:

DUPS/BAD IN ROOT INODE (REALLOCATE)

In this case, the problem is specifically found in the root inode (inode number 2). Answering y to the REALLOCATE prompt will clear the bad inode and allocate another for root. The files, once there, will become unreferenced and placed in the lost+found directory.

- EXTRA '.' ENTRY I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* (FIX)

This error occurs when a directory contains more than one entry for itself (.).

Answering y to the FIX prompt removes the duplicate entry or entries.

This is similar to the following error, which is caused by more than one entry for the parent directory (..):

EXTRA '..' ENTRY I=*inode-number* OWNER=UID
MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename*(FIX)

- *hard-link-number* IS AN EXTRANEIOUS HARD LINK TO A DIRECTORY *filename* (REMOVE)

This error occurs when fsck discovers an extraneous hard link to a directory. This condition is fixed automatically by fsck while running in preen mode. Answering y to the REMOVE prompt will also remove the extraneous reference.

- *inode-number* OUT OF RANGE I=*inode-number*
NAME=*filename* (REMOVE)

A file name entry in a directory has been found to have an inode number greater than that of the last one in the inode list. Answering y to the REMOVE prompt deletes the offending entry.

- MISSING '.' I=*inode-number* OWNER=UID MODE=*file-mode*
SIZE=*file-size* MTIME=*modification-time*
DIR=*filename* (FIX)

This error occurs when the inode number associated with the . entry in the directory is unallocated.

Answering y to the FIX prompt allocates an inode for this entry.

The following error condition is similar. However, it applies to the .. entry:

MISSING '..' I=*inode-number* OWNER=UID MODE=*file-mode*
SIZE=*file-size* MTIME=*modification-time*
DIR=*filename* (FIX)

- MISSING '.' I=*inode-number* OWNER=UID MODE=*file-mode*
SIZE=*file-size* MTIME=*modification-time*
DIR=*filename* CANNOT FIX, FIRST ENTRY IN DIRECTORY
CONTAINS *filename*

This error occurs because the first entry in a directory is *filename* rather than the directory itself (.).

fsck cannot fix this error. An attempt to recover includes terminating fsck, mounting the file system, moving the offending file to another location, and then unmounting and running fsck again.

The following error condition refers to the same problem; the second directory entry should be the parent (..):

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename CANNOT FIX, SECOND ENTRY IN DIRECTORY
CONTAINS filename
```

- MISSING '.' I=*inode-number* OWNER=*UID* MODE=*file-mode*
SIZE=*file-size* MTIME=*modification-time*
DIR=*filename* CANNOT FIX, INSUFFICIENT SPACE TO ADD
'.'

This error occurs when the first directory entry is not the directory itself (..).

Fixing this error can be accomplished by restoring the directory from a backup.

The following error condition is similar, but it refers to a missing second directory entry which should be the parent (..):

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename CANNOT FIX, INSUFFICIENT SPACE TO ADD
'..'
```

- NAME TOO LONG *filename*

This error can be caused when a privileged user has created circular links to directories resulting in unusually long file names.

Removing the circular links should be attempted and fsck re-invoked.

- ROOT INODE UNALLOCATED (ALLOCATE)

This error occurs if the root inode (usually with an inode number of 2) is unallocated.

Answering y to the ALLOCATE prompt relinks the files that are normally in this directory. (During phase 3, fsck puts the newly unreferenced files in the lost+found directory.)

The following error is similar, but here the problem is that the root inode incorrectly shows the root as a non-directory file:

ROOT INODE NOT DIRECTORY (REALLOCATE)

- UNALLOCATED I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* type=*filename*(REMOVE)

This error occurs when a directory entry contains an inode that is unallocated. Answering y to the REMOVE prompt removes this entry.

- ZERO LENGTH DIRECTORY I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename*] (REMOVE)

This error occurs when a directory with a zero length is found. Answering y to the REMOVE prompt removes the directory. This will result in a BAD or DUPLICATE error message in phase 4.

This error can also occur during phase 4.

Phase 3

During this phase, repairs to unreferenced directories and missing or full lost+found directories are performed.

Error messages that are displayed during this phase include:

- BAD INODE *state-number* TO DESCEND

This error occurs when fsck, during internal processing, passes a bad *state-number* to the descend routine. (The descend routine traverses file system hierarchy many times during an fsck operation.)

Restoring from a recent backup is the best recourse.

- DIR I=*inode-number1* CONNECTED. PARENT WAS I=*inode-number2*

This is an advisory message, not an error condition. fsck is reporting that a directory was successfully placed in the lost+found directory and its parent entry (. .) changed to reflect this.

- DIRECTORY *filename* LENGTH *file-size* NOT MULTIPLE OF *block-number* (ADJUST)

This error occurs when the file size stored in a directory inode is not a multiple of the block size. Answering y to the ADJUST prompt will round the file size indication up to the appropriate block size value.

- lost+found IS NOT A DIRECTORY (REALLOCATE)

This error occurs when the lost+found directory entry indicates that it is not a directory. Answering y to the REALLOCATE prompt will properly allocate a directory inode for this entry.

This error can also be displayed in phase 4.

- NO lost+found DIRECTORY (CREATE)

This error occurs when no lost+found directory is found in the root of the file system. Answering y to the CREATE prompt will create one if possible. If a directory cannot be created, this will be indicated.

This error can also be displayed in phase 4.

- NO SPACE LEFT IN /lost+found (EXPAND)

When a file system is created, a lost+found directory of precalculated size is created. The size of this particular directory is different than the sizes of other directories in that it cannot be enlarged as often as other directories.

The EXPAND prompt is used to enlarge the lost+found directory. (This expansion is limited because the lost+found directory is populated during fsck repairs and allocating too many disk blocks for entries during repair can easily cause more damage to the file system.)

A failure to enlarge the lost+found directory will be indicated and the file needing to be placed in the directory will remain unreferenced, resulting in an UNREF error during phase 4.

To correct this condition, delete unnecessary entries in the lost+found directory.

This error can also be displayed in phase 4.

- UNREF DIR I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* (RECONNECT)

This error occurs when a directory inode is not found as another directory entry.

Answering y to the RECONNECT prompt will instruct fsck to place it in the lost+found directory. The success or failure of this action will be reported. If this reconnection fails, an UNREF error will occur during phase 4.

Phase 4

In this phase, fsck focuses on link count information in the inodes. It reveals problems with unreferenced files, the lost+found directory, bad link counts, bad or duplicate data blocks, and the free inode count.

Error messages that are displayed during this phase include:

- BAD/DUP type I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* (CLEAR)

This error occurs because bad or duplicate blocks were found in phase 1. Answering y to the CLEAR prompt will zero out the contents of the offending inode and deallocate it.

- LINK COUNT type I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* COUNT *link-count* SHOULD BE *corrected-link-count* (ADJUST)

This error occurs because the link count found in the inode of a directory or file was found to be bad. Answering y to the ADJUST prompt corrects the link count. If running in preen mode, fsck will correct this automatically unless the number of references are increasing. This error message might occur because of a hardware failure; if it does occur, an appropriate message will be displayed.

- UNREF FILE I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* (RECONNECT)

This error occurs when an inode which is not connected to a directory is found. Answering y to the RECONNECT prompt will instruct fsck to position it in the lost+found directory.

This message is similar to the following, except in this case, it is not possible to reconnect the inode:

```
UNREF type I=inode-number OWNER=UID MODE=file-mode  
SIZE=file-size MTIME=modification-time (CLEAR)
```

Answering y to the CLEAR prompt deallocates the inode.

Phase 5

This phase checks the free block map and the used inode map in the cylinder groups. It will reveal problems when the number of used inodes or free data blocks conflicts with the lists that manage them.

Error messages that appear during this phase include:

- BLK(S) MISSING IN BIT MAPS (SALVAGE)

Tracking which data blocks are free or used is implemented with bit maps in the cylinder groups. This error occurs when the map incorrectly represents the number of free data blocks. Answering y to the SALVAGE prompt will rebuild the map.

- CG *character-for-command-option*: BAD MAGIC NUMBER

This error indicates that the cylinder group maps have been destroyed. When running in preen mode, fsck attempts to build the bad map(s). If this fails, fsck exits and the best recourse is to restore the directory from a recent backup.

- FREE BLK COUNT(S) WRONG IN SUPERBLK (SALVAGE)

This error indicates that the free block count kept in the superblock does not match the actual number of free data blocks. Answering y to the SALVAGE prompt instructs fsck to write the correct value into the superblock.

This error is similar to the following, which indicates bad summary information in the superblock:

```
SUMMARY INFORMATION BAD (SALVAGE)
```

Cleanup Phase

At this point, the file system has been checked. Final cleanup and dismantling of fsck's internal processing structures now occurs. The following status messages are displayed:

- ***** FILE SYSTEM WAS MODIFIED *****

This summary message indicates that fsck modified the file system. If the file system was mounted or if it was the current root file system, reboot the system.

It is a good practice to then unmount a non-root file system and run fsck again.

- filename FILE SYSTEM STATE SET TO OKAY

This message indicates that the file system was marked as stable. Use fsck with the -m option to determine if you should run fsck again.

- filename FILE SYSTEM STATE NOT SET TO OKAY

This message indicates that the file system was not marked as stable. Use fsck with the -m option to determine if you should run fsck again.

Adding a Network Printer

The following sections describe how to add network printers.

Using Printer Vendor Supplied Tools

Adding a network printer involves the following main tasks:

1. Connect the printer to the network and turn on the power to the printer.
2. Consult the printer vendor's installation documentation for information about the hardware switches and cabling requirements.
3. Get an IP address and select a name for the printer node. This is equivalent to adding any node to the network.
4. Follow the printer vendor instructions to add the network printer to a Solaris 7 system.
5. Use the printer vendor instructions to configure the network printer. These will be specific to the vendor and printer.
6. Add client access to the new printer.

Now that the printer has been added, create access to the printer for the clients.

Setting Up the LexMark Optra Model Network Printer

The LexMark Optra Rn+ is one of many possible network printers available today. For the purpose of this training module, you will learn how to install and configure this printer. Configuration of other LexMark network printers in the Optra model family will be similar.

The following steps will enable you to set the appropriate network print parameters using the firmware program available on the front panel of the Optra Rn+ network printer:

1. Attach power and network cables to the printer.
2. Turn on the system and observe the liquid crystal display (LCD) panel on the front of the printer. There are several buttons situated on the LCD panel.
 - ▼ The MENU button displays a menu of selections.
 - ▼ The SELECT button enables you to select from a menu.
 - ▼ The RETURN button returns you to the previous display.
3. Push the MENU button.
4. Continue through the menus displayed (you will need to select MORE to see all choices) until you find the NETWORK MENU 2 selection; then press SELECT.
5. Go through the menu selections until you can select IP Protocol; then press SELECT.
6. Go through the menu selections until you can select IP Address; then press SELECT. (Again, you will need to select MORE to see all choices.)
7. Set the IP address and press SAVE. (This returns you to the previous menu.)
8. Go through the menu selections until you can select IP Netmask; then press SELECT.
9. Enter the IP Netmask and press SAVE. (Again, this returns you to the previous menu.)

10. Go through the menu selections until you can select the IP Gateway, if available, and press SAVE.
11. In a terminal window on a client system, test the printer's ability to respond to a `ping` command using the IP address you entered into the printer setup in step 7.

You should see a positive response. If you do not, revisit your printer configuration.

The network printer has been configured. The next step is to configure a Sun system as the network printer server.

Setting Up a Sun System as the Network Printer Server

Installing the Software Packages

The following steps enable you to establish a Sun system as the print server for the LexMark Optra printer using the MarkVision software:

1. Download the PostScript or portable document format (PDF) user guide and the 30-day demo license UNIX software package file from LexMark using the following URL:

<http://www.lexmark.com/networking/mkvreg.html>

Note – You will need to provide information about you and your company to acquire the demo software.

Although the file you download does not have a .Z extension, it is a compressed file. Rename the file to include the .Z extension so that the `uncompress` command will work.

2. Use the `uncompress` command on the newly renamed software package file.

```
# uncompress mv_demo.solaris2_sparc.pkg.Z
```

3. Prior to installing the packages, set the `NONABI_SCRIPTS` environment variable to `TRUE`. This will enable the MarkVision network printer installation program to run as a GUI.

For Bourne and Korn shells, run the following commands:

```
# NONABI_SCRIPTS=TRUE ; export NONABI_SCRIPTS
```

For C shell, run the following command:

```
print_server# setenv NONABI_SCRIPTS TRUE
```

4. Set the `OPENWINHOME` environment variable to the `/usr/openwin` value.

For Bourne and Korn shells, run the following commands:

```
# OPENWINHOME=/usr/openwin ; export OPENWINHOME
```


For C shell, run the following command:

```
print_server# setenv OPENWINHOME /usr/openwin
```

5. Create the home directory for Hypertext Transfer Protocol (HTTP) service if this server has access to the Internet and will provide web pages for other systems on the local network. The installation script will install the web pages in that location unless you specify another.

```
# mkdir -p /usr/ns-home/docs
```

6. Install all packages by using the following command:

```
# pkgadd -d ./mv_demo.solaris2_sparc.pkg
```

The script will install the following packages:

1	MVaction	MarkVision Action Client
2	MVbootp	Bootp
3	MVclient	MarkVision Client
4	MVfonts	MarkVision Fonts
5	MVserver	MarkVision Server
6	MVweb	MarkVision Intranet Client
7. Answer in the affirmative when prompted throughout the package installations. The default installation directory created is the /opt/lexmark directory, which must be used for all packages.
 8. During the installation process, the script will inform you that you need to run /opt/lexmark/setup.web to make information available from the Sun system through the local HTTP server (if you have one running).
 9. During the install of the MarkVision UNIX Client package, the script prompts you for the name of the MarkVision Server host name or IP address. Use the Sun system host name.
 10. During the installation of the MVfonts package, the script prompts you for the destination directory for fonts.

How should the fonts be installed?

A) For use with a font server.

B) As a standard system font.
C) As an additional font directory.
Or <return> to quit.

Enter B as your answer.

11. Accept the default /usr/openwin/lib/X11/fonts/misc for the destination directory.
12. Answer Y to the following question:

Do you accept the license agreement? (Y/N)
13. Continue to respond in the affirmative, accepting the defaults, for the rest of the package installation.

Configuring the Network Printer Software

To set up a Sun system as the network printer server, configure the network printer software using the following steps:

1. Start the markvision utility.

markvision &

The MarkVision window is displayed.

2. Select Add Printer from the Edit menu.

The Add Printer window is displayed.

3. Enter the printer name or IP address. If you use the printer name, there must be an entry in your local /etc/hosts file or NIS hosts map. For the purpose of this installation, the name OptraRn is used.

4. Click on OK.

5. The MarkVision window now shows the new printer.

6. Select Lexprt from the Utilities menu.

The Lexprt window is displayed.

7. Click on Create a Virtual Device.

The Device window is displayed.

8. Enter a device name.
9. Click in the field when finished to enable the Accept button.
10. Click on Accept.

The Transport options window is displayed.

11. Click on Network connection.

Note – If you select Network connection with end-of-job notification, users will receive email for each completed print job they submit.

The Adapter setup window is displayed.

12. Enter the name of the network printer.
13. Click in the message area in the lower part of the window to enable the Accept button.
14. Click on Accept.

The Network device options window is displayed.

The default answers will be appropriate.

You can modify the configuration when the printer sends warning messages by selecting Printer Intervention Required messages. You have the option of being notified by email or writing to the terminal, or receiving no notification.

Printer Response messages can be directed in the same manner; to the terminal, or email, or no notification. The default is no notification.

15. Click on Accept.

The Lexpert main menu window is displayed and the messages in the lower section of the window confirm the successful addition of the device.

16. Click on Create a queue.

The Create a queue window is displayed.

17. Enter the name of the printer, `OptraRn`, to create a queue.
18. Click in the scrolling messages window at the bottom of the window to enable the Accept button.
19. Click on Accept.

The Choose a Device window is displayed.

20. Click on the box to the left of the printer device name.

The Choose a Printer window is displayed.

21. Scroll through the list to match the printer with the same type being installed.

22. For the purpose of this configuration, select the Lexmark Optra plus laser printer family.

The Printer language window is displayed.

23. Since it is most likely that this printer will be printing in various formats (PostScript™, text, and so on), select the Auto button.

The Automatic language options window is displayed.

24. Observe the default settings for PostScript emulation by clicking on the box for that choice.

The PostScript emulation options window is displayed.

The default selections are adequate for this configuration.

25. Click on Cancel to return to the previous window.

The Automatic Language window is displayed once again.

26. Click on Accept.

The Main menu window is displayed.

The message area at the bottom of the window confirms that the queue for the OptraRn network printer was installed successfully.

27. From the File menu, select Exit.

28. Close the Markvision window.

Testing the Installation of the LexMark Network Printer

The following steps will provide confirmation that the network printer is available to the Solaris system:

1. Run the `lpstat` command to confirm installation of the printer.

```
# lpstat -t
scheduler is running
system default destination: none
device for OptraRn: /dev/null
OptraRn accepting requests since Sun Aug 8 21:05:11
MST 1999
printer OptraRn is idle. enabled since Sun Aug 8
21:05:12 MST 1999. available.
```

2. Run the `lpadmin` command to configure the OptraRn printer as the default printer for this system.

```
# lpadmin -d OptraRn
```

3. Confirm that OptraRn is the default printer by using the `lp` command to print a file.

```
# lp /etc/hosts
```

4. If you want to confirm PostScript capability, use the following command:

```
# lp /usr/openwin/share/images/PostScript/worldmap.ps
```

Setting Up an HP LaserJet 4000TN Network Printer

The Hewlett Packard LaserJet 4000TN is another one of many possible network printers available today. Configuration of other HP network printers will be similar to the procedure discussed here.

The following steps will enable you to set the appropriate network print parameters using the firmware program available on the front panel of the HP network printer:

1. Attach power and network cables to the printer per instructions included with the printer.
2. Turn on the system and observe the control panel on the front of the printer.

There are several buttons, LEDs, and an LCD readout located on the control panel (Figure C-1).

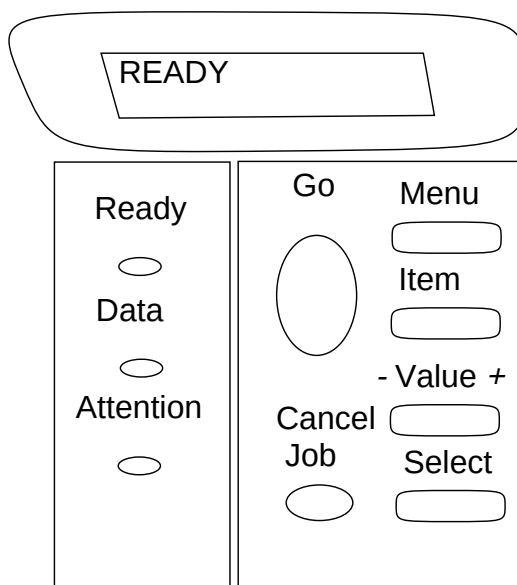


Figure C-1 HP LaserJet 4000TN Control Panel

- ▼ The Ready light indicates the printer is ready to print.
- ▼ The Data light means the printer is processing a print job.
- ▼ The Attention light indicates that some action is required. There is usually some additional information displayed on the LCD readout with regard to this condition.

- ▼ The Go button puts the network printer on- or offline.
 - ▼ The Cancel Job button terminates any job processing. It may take a few moments to terminate the job.
 - ▼ The Menu, Item, Value, and Select buttons are two-sided in function. The right side moves forward when pressed and the left side moves backward through the displayed parameters.
 - The Menu button displays a menu on the readout.
 - The Item button cycles through choices in the displayed menu.
 - The - Value + button enables changing the values of the menu item.
 - The Select button saves the selected value.
3. Continue through the menus displayed until you see the TCP/IP=ON. If the value is OFF, change it to ON.
 4. Continue through the menus until the CFG NETWORK MENU=NO selection is displayed and change the value to YES.
 5. Go through the menu selections until you can select CFG TCP/IP=NO. Change the value to YES and the TCP/IP menu will be displayed.
 6. Set BOOTP=NO.
 7. Set the IP address (IP), Subnet Mask (SM) and Default Gateway (GW) as appropriate for your network. The default IP address is 192.9.9.192.

Installing the HP JetAdmin Utility for UNIX

The following steps enable you configure the HP JetAdmin software to use a Sun Solaris system as the print server for an HP network printer:

1. Download the HP JetAdmin for Solaris software from the following URL:

<http://www.hp.com/cposupport/jsnav/netprn.html>
2. Copy the software file, SOLd515.PKG, to the /tmp directory and change to that directory.
3. Run the following command:


```
# pkgadd -d ./SOLd515.PKG
```

4. Accept the installation script default location of /opt/hpnp for the package.
5. At the configuration of sub-packages, enter **0** to install them all. No further interaction with the installation program is necessary.

Please configure the sub-packages you would like to install.

```
-----
0. Done altering installation configuration
1. [ N/A ] JetPrint
2. [ On  ] JobMonitor
3. [ On  ] HPNPF
4. [ On  ] HPNPD
5. [ On  ] CONVERT
?.      Help
-----
```

Select a number to toggle an installation option.
When done select 0. Select ? for help information: **0**

6. As specified at the end of the installation script:
 - ▼ Add the /opt/hpnp/man directory to the MANPATH variable.
 - ▼ Add the /opt/hpnp/bin directory to the PATH variable.
 - ▼ Add the /opt/hpnp/lib to LD_LIBRARY_PATH variable.
7. Run the JetAdmin software by typing `jetadmin` at the prompt.

```
# jetadmin
```

The Main menu for JetAdmin is displayed.

```
*****
*                               *
*             MAIN MENU        *
* HP JetAdmin Utility for UNIX (Rev. D.05.15) *
*****
```

- 1) Configuration (super-user only):
 - configure printer, add printer to spooler

- 2) Diagnostics:
 - diagnose printing problems
- 3) Administration (super-user only):
 - manage HP printer, JetDirect
- 4) Administration (super-user only):
 - manage JetAdmin
- 5) Printer Status:
 - show printer status, location, and contact
- ?) Help q) Quit

Please enter a selection (q - quit):

8. Select option 2, Diagnostics.

The Diagnostics menu for JetAdmin is displayed.

```
*****
*                DIAGNOSTICS                *
*   HP JetAdmin Utility for UNIX   *
*****
```

System:

- 1) Verify installation of software
- 2) Examine spooler operations

Printer/Network:

- 3) Test network printer's accessibility
- 4) Print a test page to printer
- 5) View HP JetDirect interface's current configuration
- 6) Check BOOTP and TFTP operations (super-user only)
- 7) Show HP JetDirect interface's protocol settings
- 8) Display operational and protocol statistics

?) Help q) Quit

Please enter a selection:

9. Select option 1 to verify successful installation of the JetAdmin software.

The program tests for JetAdmin software installation and returns the following messages:

The HP JetAdmin Utility for UNIX (D.05.15) is installed.

Software verification passed.

Press the return key to continue ...

10. Press the Return key and type **q** to return to the Main menu.

11. Select option 1 to configure the printer software.

The Configuration menu for JetAdmin is displayed.

```
*****
*                CONFIGURATION                *
*      HP JetAdmin Utility for Unix      *
*****
```

Printer Network Interface:

- 1) Create printer configuration in BOOTP/TFTP database
- 2) Remove printer configuration from BOOTP/TFTP

Spooler:

- 3) Add printer to local spooler
- 4) Delete printer from local spooler
- 5) Modify existing spooler queue(s)

?) Help

q) Quit

Please enter selection:

12. Select option 3 to add the printer to the local spooler.

The program responds with a request for the printer name or IP address.

Please enter selection: 3

Enter the network printer name or IP address (q - quit): 172.20.3.90

Following is a list of suggested parameter values for this queue. You may change any settings by selecting the corresponding non-zero numbers.

The values will be used to configure this queue when '0' is selected. To abort the operation, press 'q'.

Configurable Parameters:	Current Settings
-----	-----
1) Lp destination (queue) name	[172_1]
2) Status log	[(No log)]
3) Queue class	[(not assigned)]
4) JobMonitor	[OFF]
5) Default queue	[NO]
6) Additional printer configuration...	

Select an item for change, or '0' to configure (q-quit):

13. Enter the number for the network printer parameters you want to change followed by a 0 to return to the previous menu.

For the purpose of this install, you can change options 2, 4, and 5 by entering their respective numbers. These are *toggle* choices; either on or off. Log files are default names.

14. Create a Queue class, use an arbitrary name, by selecting option 3 and, for this exercise, entering HPprinters at the prompt.

The resultant menu now displays the changes in the previous steps.

Configurable Parameters:	Current Settings
-----	-----
1) Lp destination (queue) name	[172_1]
2) Status log	[/opt/hpnp/tmp/172_1.log]
3) Queue class	[HPprinters]
4) JobMonitor	[ON]
5) Default queue	[YES]
6) Additional printer configuration...	

Select an item for change, or '0' to configure (q-quit):

15. Observe the following output when entering option 6, Additional printer configuration:

Following is a list of suggested parameter values for this printer.
To abort this operation, press 'q'.

Configurable Parameters:	Current Settings

```
-----  
1) Model Script: [net_ljx000]  
2) Default Printing Language [AUTO]  
3) Job Recovery [ON]  
4) True End-of-Job [ON]  
5) Banner Page [ON]  
6) PostScript Level [Level 2]  
-----
```

Select an item for change, or q when done: q

16. For the purposes of this lab, there is no need to modify these parameters. Enter **q** at the prompt and you will be returned to the configuration parameters listing.

The modified configuration parameters are displayed.

Following is a list of suggested parameter values for this queue. You may change any settings by selecting the corresponding non-zero numbers.

The values will be used to configure this queue when '0' is selected. To abort the operation, press 'q'.

```
Configurable Parameters:      Current Settings  
-----  
1) Lp destination (queue) name [172_1]  
2) Status log [ /opt/hpnp/tmp/172_1.log]  
3) Queue class [HPprinters]  
4) JobMonitor [ON]  
5) Default queue [YES]  
6) Additional printer configuration...
```

Select an item for change, or '0' to configure (q-quit):

17. Enter **0**, confirm all settings for the network printer configuration, and confirm that configuration as the new default.

Ready to configure 172_1.
OK to continue? (y/n/q, default=y) y

.....
Finished adding "172_1" to the spooler.

The configuration of the HP network printer is complete.

Testing the Installation of the HP Network Printer

The following steps will provide confirmation that the HP network printer is available to the Solaris system:

1. Run the `lpstat` command to confirm installation of the printer.

```
# lpstat -t
scheduler is running
system default destination: none
device for 172_1: /dev/null
172_1 accepting requests since Sun Aug 8 21:05:11 MST
1999
printer 172_1 is idle. enabled since Sun Aug 8
21:05:12 MST 1999. available.
```

2. Run the `lpadmin` command to configure the 172_1 printer as the default printer for this system.

```
# lpadmin -d 172_1
```

3. Confirm that 172_1 is the default printer by using the `lp` command to print a file.

```
# lp /etc/hosts
```

If you want to confirm PostScript capability, use the following command:

```
# lp /usr/openwin/share/images/PostScript/worldmap.ps
```

Enabling Access to a Network Printer

The following steps will enable you to add printer access from a print client using the Admintool:

1. Start Admintool on the system you want to add access to a remote printer to.

admintool &

The Admintool:Users window is displayed.

2. Select Printers from the Browse menu.

The Admintool:Printers Window is displayed.

3. Select Add Access to Remote Printer from the Edit menu.

The Add Access to Remote Printer window is displayed.

4. Fill in the window as required.

If you need information to complete a field, click on the Help button to see field definitions for this window or check with your instructor.

5. Click on OK.

The printer is displayed in the Admintool Printers window.

6. Exit **admintool**.

Index

Symbols

\$HOME/.rhosts file 3-67
/ directory 4-14
/bin directory 4-14
/dev directory 4-15
/devices directory 4-15
/etc directory 4-16
/etc/default directory 3-15
/etc/default/login
 variable 3-18
/etc/dumpdates 17-8
/etc/format.dat file 6-9
/etc/group file 2-7, 2-21
/etc/hosts/equiv file 3-67
/etc/hosts/ftpusers file 3-67
/etc/passwd file 2-21
/etc/path_to_inst file 5-15
/etc/shadow file 2-21, 3-4
/etc/shells file 3-73
/etc/system 13-8
/export directory 4-16
/home directory 4-16
/kernel 13-8
/kernel directory 4-16
/kernel/drv 13-9
/mnt directory 4-16
/opt directory 4-17
/platform/'uname
 -i'/kernel 13-8
/platform/'uname
 -m'/kernel 13-8
/sbin directory 4-17
/sbin/init 13-8

/tmp directory 4-17
/usr directory 4-17
/usr/kernel/drv 13-9
/var directory 4-17
/var/adm/loginlog file 3-5
/var/adm/utmpx file 3-6
/var/spool/pkg 15-23

A

accept command 11-34
access control lists 3-51
ACL 7-14
ACLs 3-51
admintool 2-5, 15-13
application server 1-8
at command 10-7, 10-12

B

backup superblock 7-6
banner command 12-10
boot 13-7
boot -a command 13-14
boot block 7-6, 13-7
boot command 12-11

C

chgrp command 3-35
chown command 3-33
client 1-7
commands
 accept 11-34

at 10-7, 10-12	mt 17-20
banner 12-10	newfs 7-15
boot 12-11	nvalias 12-25
boot -a 13-14	nvedit 12-26
chgrp 3-35	nvunalias 12-25
chown 3-33	patchadd 16-16
compress 17-5	patchrm 16-20
crontab 10-7	pkgadd 15-8
devalias 12-24	pkgchk 15-9
devfsadm 5-20	pkginfo 15-3
df 9-11	pkgrm 15-6
disable 11-34	poweroff 13-28
drvconfig 5-22	printenv 12-13
du 9-12	probe-ide 12-23
EEPROM 12-28	probe-scsi 12-22
enable 11-34	probe-scsi-all 12-22
ff 9-14	prstat 10-2, 10-5
finger 3-7	prtconf 5-16
fmthard 6-24	prvtoc 6-23
format 5-18, 6-2	ps 10-2
fuser 8-17	pwconv 3-4
getfacl 3-55	quot 9-14
getfacl -m 3-57	reboot 13-29
grep 15-12	reject 11-34
groupadd 2-34	reset 12-9, 12-15
groupdel 2-36	rusers 3-9
groupmod 2-35	set-defaults 12-16
groups 3-31	setenv 12-15
gzcat 16-13	setfacl -d 3-56
halt 13-28	setfacl -m 3-54
help 12-12	show-devs 12-19
id 3-32	showrev 15-12
init 13-26	shutdown 13-27
last 3-8	su 3-10, 3-12
ln 4-8	tar 16-13
lp 11-18	tunefs 7-16
lpadmin 11-34, 11-41	ufsdump 17-6, 17-13
lpmove 11-34	ufsrestore 17-14
lpr 11-18	umount 8-15
lpsched 11-43	umount -f 8-17
lpshut 11-43	umountall 8-16
ls 4-3	unzip 16-13
ls -l 3-55	useradd 2-30
modify 6-18	userdel 2-33
mount 8-5, 8-8, 8-24	usermod 2-32
mountall 8-14	verify 6-22

- who 3-6
- whoami 3-11
- zcat 16-13
- compress command 17-5
- CONSOLE variable 3-16, 3-19
- controller number 5-8
- crontab 10-7
- cylinder 5-4
- cylinder group blocks 7-8

D

- daemons
 - in.lpd 11-27
 - inetd 11-27
 - listen 11-28
 - lpNet 11-28
 - lpsched 11-27, 11-29
- data block 7-8
- date mounted 8-5
- deleting, user account 2-19
- devalias command 12-24
- devfsadm command 5-20
- device name 8-5
- df command 9-11
- directories
 - / 4-14
 - /bin 4-14
 - /dev 4-15
 - /devices 4-15
 - /etc 4-16
 - /etc/default 3-15
 - /export 4-16
 - /home 4-16
 - /kernel 4-16
 - /mnt 4-16
 - /opt 4-17
 - /sbin 4-17
 - /tmp 4-17
 - /usr 4-17
 - /var 4-17
 - home 2-15
 - lost+found 9-2
- disable command 11-34
- disk label 7-6
- disk number 5-9

- disk platter
 - cylinder 5-4
 - sector 5-4
 - track 5-4
- disk slice
 - controller number 5-8
 - disk number 5-9
 - slice number 5-9
 - target number 5-8
- disk-based file systems 7-2
- distributed file system 7-3
- drvconfig command 5-22
- du command 9-12

E

- EDITOR variable 10-10
- eeprom command 12-28
- effective group ID 3-11
- effective user ID 3-11
- EGID 3-11
- enable command 11-34
- Ethernet address 1-7
- EUID 3-11

F

- fdfs file system 7-3
- ff command 9-14
- file server 1-8
- file system
 - fdfs 7-3
 - hsfs 7-2
 - nfs 7-3
 - pcfs 7-2
 - procfs 7-3
 - swapfs 7-3
 - tmpfs 7-3
 - udf 7-2
 - ufs 7-2
- file systems
 - disk-based 7-2
 - distributed 7-3
 - pseudo 7-3
- files
 - \$HOME/.rhosts 3-67
 - /etc/default/login 3-18

- /etc/format.dat 6-9
- /etc/ftpusers 3-67
- /etc/group 2-7, 2-21
- /etc/hosts.equiv 3-67
- /etc/passwd 2-21
- /etc/path_to_inst 5-15
- /etc/shadow 2-21, 3-4
- /etc/shells 3-73
- /var/adm/loginlog 3-5
- /var/adm/utmpx 3-6
- finger command 3-7
- fmthard command 6-24
- format command 5-18, 6-2
- fsck 8-14, 9-2
- fuser command 8-17

G

- genunix 13-7
- getfacl command 3-55
- getfacl -m command 3-57
- GID 2-4
- geographic location 14-12
- grep command 15-12
- group identification number 2-4
- groupadd command 2-34
- groupdel command 2-36
- groupmod command 2-35
- groups command 3-31
- gzcat command 16-13

H

- halt command 13-28
- help command 12-12
- home directory 2-15
- host 1-7
- host IP address 14-12
- host name 14-12
 - system administration terms
 - host name 1-7
- hsfs file system 7-2

I

- id command 3-32
- in.lpd daemon 11-27

- inetd daemon 11-27
- init command 13-26
- init state 13-3
- inode table 7-8
- installation, Web Start 14-14
- instance names 5-14
- Internet address 1-7
- IP address 1-7

L

- language 14-12
- last command 3-8
- listen daemon 11-28
- ln command 4-8
- locking, user account 2-17
- logical device names 5-11
- login device types
 - console 3-6
 - pts 3-6
 - term 3-6
- login name 2-4
- login shell 2-3
- logs
 - ufs 8-10
- lost+found command 9-2
- lp command 11-18
- lpadmin command 11-34, 11-41
- lpmove command 11-34
- lpNet daemon 11-28
- lpr command 11-18
- lpsched command 11-43
- lpsched daemon 11-27, 11-29
- lpshut 11-43
- ls command 4-3
- ls -l command 3-55

M

- MAXWEEKS variable 3-20
- MINWEEKS variable 3-20
- modify command 6-18
- mount command 8-5, 8-8, 8-24
- mount options 8-5
- mount point 8-3, 8-5
- mount point 17-11
- mountall command 8-14

mounting 8-3
mt command 17-20

N

name service type 14-12
netstandardDefault Para
Font> script 11-24
newfs command 7-15
nfs file system 7-3
nvalias command 12-25
nvedit command 12-26
nvunalias command 12-25

P

partition table 6-4
PASSLENGTH variable 3-21
password 2-3
patchadd command 16-16
patchrm command 16-20
pcfs file system 7-2
permissions
 setgid 3-38
 setuid 3-37
 Sticky Bit 3-40
physical device names 5-12
PID 10-2
pkgadd command 15-8
pkgchk command 15-9
pkginfo command 15-3
pkgrm command 15-6
poweroff command 13-28
print
 client 11-4
 fault notification 11-6
 initialization 11-5
 memory 11-8
 queuing 11-5
 spooling space 11-7
 tracking 11-6
print server 1-8, 11-4
printenv command 12-13
printer
 local 11-4
 network 11-4
 remote 11-4

probe-ide command 12-23
probe-scsi command 12-22
probe-scsi-all
 command 12-22
process identification
 number 10-2
procfs file system 7-3
prstat command 10-2, 10-5
prtconf command 5-16
prvtoc command 6-23
ps command 10-2
pseudo file system 7-3
pwconv command 3-4

Q

quot command 9-14

R

reboot command 13-29
reject command 11-34
reset command 12-9, 12-15
root password 14-12
rusers command 3-9

S

scripts
 netstandard 11-24
sector 5-4
server 1-7
servers
 application 1-8
 file 1-8
 print 1-8
set-defaults command 12-16
setenv command 12-15
setfacl -d command 3-56
setfacl -m command 3-54
setgid permission 3-38
setuid permission 3-37
show-devs command 12-19
showrev command 15-12
shutdown command 13-27
slice number 5-9
software administration 15-2

- software groups
 - Core 14-9
 - Developer System
 - Support 14-10
 - End User System
 - Support 14-9
 - Entire Distribution 14-10
 - Entire Distribution Plus OEM
 - Support 14-10
- Sticky Bit permission 3-40
- su command 3-10, 3-12
- subnet mask 14-12
- SULOG variable 3-17
- superblock 7-6
- swapfs file system 7-3
- system administration terms
 - client 1-7
 - Ethernet address 1-7
 - host 1-7
 - IP address 1-7
 - server 1-7

T

- tar command 16-13
- target number 5-8
- time mounted 8-5
- time zone 14-12
- tmpfs file system 7-3
- track 5-4
- tunefs command 7-16

U

- udf file system 7-2
- ufs file system 7-2
- ufs log 8-10
- ufsboot 13-7
- ufsdump command 17-6, 17-13
- ufsrestore command 17-14
- UID 2-4
- umount command 8-15
- umount -f command 8-17
- umountall command 8-16
- unix 13-7
- unmounting 8-3
- unzip command 16-13

- user account
 - deleting 2-19
 - home directory 2-3
 - initialization files 2-3
 - locking 2-17
 - login shell 2-3
 - password 2-3
 - user name 2-3
- user home directory 2-3
- user identification number 2-4
- user initialization files 2-3
- user name 2-3
- useradd command 2-30
- userdel command 2-33
- usermod command 2-32

V

- variables
 - CONSOLE 3-16, 3-19
 - EDITOR 10-10
 - MAXWEEKS 3-20
 - MINWEEKS 3-20
 - PASSLENGTH 3-21
 - SULOG 3-17
- verify command 6-22
- vold 8-20
- VTOC 7-6

W

- Web Start 14-14
- who command 3-6
- whoami command 3-11

Z

- zcat command 16-13