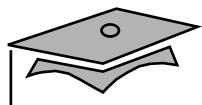


Sun Educational Services

Solaris™ 8 Operating Environment System Administration I

SA-238





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.



Preface

About This Course



Course Goal

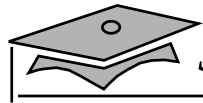
This *Solaris 8™ Operating Environment System Administration I* course provides you with knowledge and skills to administer the Solaris Operating Environment, including standalone installation, file system management, backups, process control, user administration, and device management.



Course Overview

The primary objective of this course is to teach new system administrators the basics of administering Sun workstations. You will perform basic administration tasks including

- Installing a standalone system
- Adding users
- Backing up and restoring file systems
- Adding printer support.
- Managing disks and creating file systems



Course Map

Introduction

Solaris 8
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

Solaris ufs
File System

Mounting
File Systems

Maintaining
File Systems

Processes and Printing

Scheduled
Process
Control

The Solaris
LP Print
Service

System Firmware, Boot Process, and Run Levels

The
Boot Prom

System Boot
Process

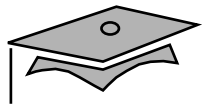
Software Installation and Administration

Installing
Solaris 8 on a
Standalone
System

Administration
of Software
Packages

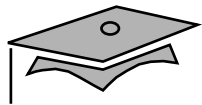
Administration
of Software
Patches

Backup and
Recovery

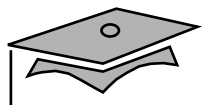


Module-by-Module Overview

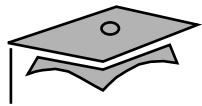
- Module 1 – "Introducing the Solaris 8 Operating Environment System Administration"
- Module 2 – "Adding Users"
- Module 3 – "System Security"
- Module 4 – "The Directory Hierarchy"
- Module 5 – "Device Configuration"
- Module 6 – "Disks, Slices, and Format"
- Module 7 – "The Solaris Operating Environment `ufs` File System"



- Module 8 – "Mounting File Systems"
- Module 9 – "Maintaining File Systems"
- Module 10 – "Scheduled Process Control"
- Module 11 – "The Solaris Operating Environment LP Print Service"
- Module 12 – "The Boot PROM"
- Module 13 – "System Boot Process"
- Module 14 – "Installing the Solaris 8 Operating Environment on a Standalone System"
- Module 15 – "Administering the Software Package"



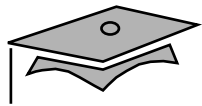
- Module 16 – "Administering the Software Patches"
- Module 17 – "Backup and Recovery"



Course Objectives

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

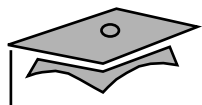
- Define basic system administration tasks and terms
- Add users and groups to the system
- Configure user initialization files
- Implement basic system security
- Create ACLs (access control lists) on files
- Identify disks configured on a system



- 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

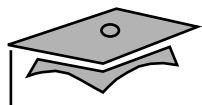


- Install the Solaris 8 Operating Environment software on a standalone workstation
- Add software packages
- Add a software patch
- Perform a root file system backup and restore



Skills Gained by Module

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												■	■				

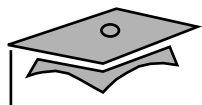


Skills Gained	Module																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Change system run levels																	
Install the Solaris 8 Operating Environment software on a standalone workstation																	
Add software packages																	
Add software patch																	
Perform a root file system backup and restore																	

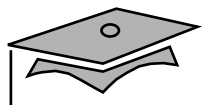


Guidelines for Module Pacing

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.				
"The Directory Hierarchy"	P.M.				
"Device Configuration"		A.M.			
"Disks, Slices, and Format"		A.M./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.		

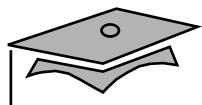


Module	Day 1	Day 2	Day 3	Day 4	Day 5
"The Boot PROM"				A.M.	
"System Boot Process"				A.M./ P.M.	
"Installing the Solaris 8 Operating Environment on a Standalone"				P.M.	
"Administering the Software Packages"					A.M.
"Administering the Software Patches"					A.M.
"Backup and Recovery"					P.M.

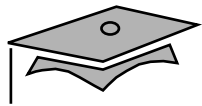


Topics Not Covered

- 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*



How Prepared Are You?

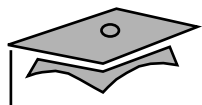
To be sure you are prepared to take this course, can you answer yes to the question below?

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



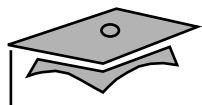
Introductions

- Name
- Company affiliation
- Title, function, and job responsibility
- Solaris Operating Environment experience
- Reasons for enrolling in this course
- Expectations for this course



How to Use Course Materials

- Objectives
- Lecture
- Exercise
- Check Your Progress



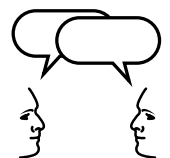
How to Use the Icons



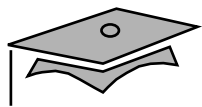
Additional resources



Demonstration



Discussion



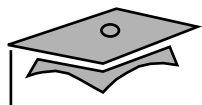
Exercise objective



Caution

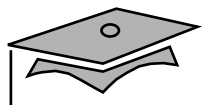


Warning



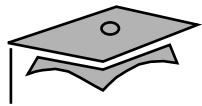
Typographical Conventions and Symbols

- `Courier` is used for the names of command, files, and directories, as well as on-screen computer output.
- **`Courier bold`** is used for characters and numbers that you type.
- *`Courier italic`* is used for variables and command-line placeholders that are replaced with a real name or value.
- *Palatino italics* is used for book titles, new words or terms, or words that are emphasized.



Module 1

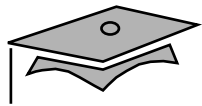
Introducing the Solaris 8 Operating Environment System Administration



Objectives

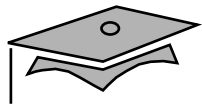
Upon completion of this module, you should be able to:

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



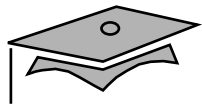
Roles of the System Administrator for a Standalone System

- Managing users' accounts
- Maintaining system security
- Configuring new devices
- Installing and partitioning disk drives
- Managing file systems
- Scheduling system-related jobs
- Maintaining print services



Roles of the System Administrator for a Standalone System

- Managing the boot PROM
- Configuring system initialization files
- Installing the Solaris Operating Environment software
- Administering software packages and patches
- Performing backup and recovery operations
- Managing disaster recovery
- Recovering critical file systems and rebooting successfully



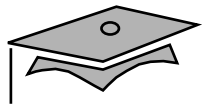
Roles of the System Administrator for Client-Server Systems

- Configuring a network environment
- Setting up the `syslog` utility
- Configuring and administering a network file system (NFS) environment
- Configuring `cacheFS` file systems
- Using `automount`
- Setting up name services



Roles of the System Administrator for Client-Server Systems

- Configuring boot protocols
- Installing and configuring Solstice AdminSuite™
- Installing the Solaris Operating Environment using the Jumpstart™ program



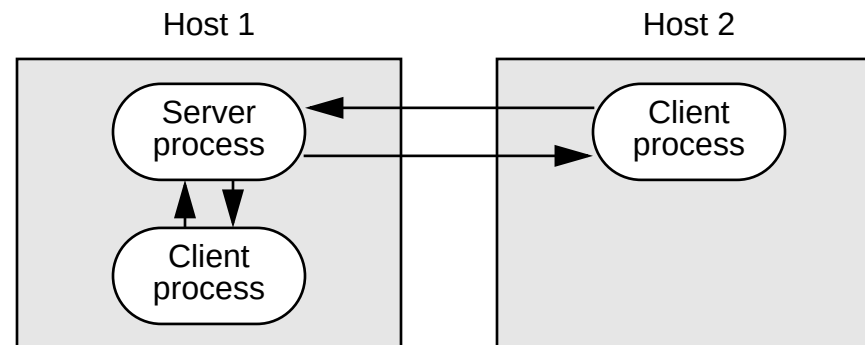
System Administration Terms

Some common system administration terms include:

- Host
- Host name
- Internet (IP) address
- Ethernet address
- Server
- Client



Distributed Computing Environment





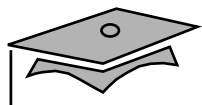
Check Your Progress

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



Module 2

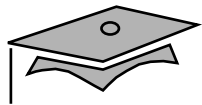
Adding Users



Objectives

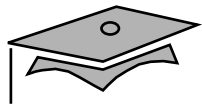
Upon completion of this module, you should be able to:

- 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`



Objectives

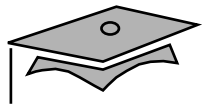
- Add, modify, and delete group accounts for the local system with the commands `groupadd`, `groupmod`, and `groupdel`
- Define the two different types of shell initialization files
- Describe the shell startup activities during login for the three main Solaris Operating Environment shells
- List the 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



Setting Up User Accounts

Each user account consists of five main components:

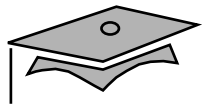
- User name
- Password
- User's home directory
- User's login shell
- User initialization files



Managing User Accounts

Before adding user accounts to the system, you must determine the following information for each new user:

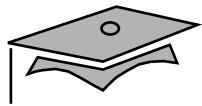
- Login name
- User identification (UID) number
- Group identification (GID) number
- Comment
- home directory
- Login shell
- Password aging



Managing User Accounts with `admintool`

`admintool` enables you to maintain and modify local system files from the following categories:

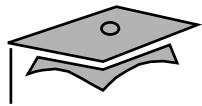
- Users
- Groups
- Hosts
- Printers
- Serial ports
- Software



Storing User and Group Account Information

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

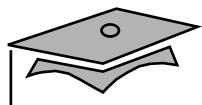
- `/etc/passwd` – Authorized system users have login account entries in the `/etc/passwd` file.
- `/etc/shadow` – All passwords are encrypted and maintained in a separate shadow file named `/etc/shadow`.
- `/etc/group` – The `/etc/group` file defines the default system group accounts.



Creating and Managing Accounts from the Command-line

The following command-line tools add, modify, and delete user accounts and group accounts on the local system:

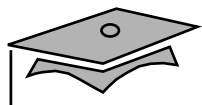
- `useradd` – Adds a new user account
- `usermod` – Modifies a user's account
- `userdel` – Deletes a user's account
- `groupadd` – Adds a new group account
- `groupmod` – Modifies a group account
- `groupdel` – Deletes a group account



Creating Accounts with useradd

- The useradd command adds an entry for the new user into the `/etc/passwd` and `/etc/shadow` files.
- The useradd command also copies all the initialization files in the `/etc/skel` directory to the user's new home directory automatically.
- Command format:

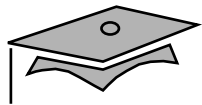
```
useradd [-u uid][-g gid][-G gid [,gid,.. ]][-d dir][-m][-s shell][-c comment] loginname
```



Modifying a User Account with usermod

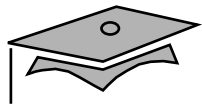
- The usermod command modifies the components existing in a user account.
- Command format:

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



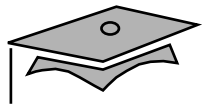
Deleting User Accounts with the `userdel` Command

- The system administrator can use the `userdel` command to delete a user 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`



Creating Accounts with groupadd

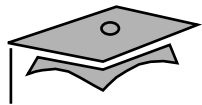
- As `root`, you can create new group accounts on the local system using the `groupadd` command.
- This command adds an entry for the new group into the `/etc/group` file.
- Command format:
`groupadd [-g gid [-o ]] groupname`



Modifying a Group Account with groupmod

- The groupmod command modifies 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
```



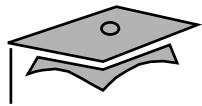
Deleting Group Accounts with the `groupdel` Command

- 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`



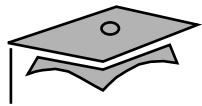
Exercise: Adding Users and Groups

- Objectives
- Tasks
- Discussion
- Solutions



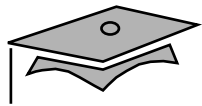
Using the Initialization Files

- 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

- The system initialization files are maintained by the system administrator to provide an environment for the entire community of users who login 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:
 - ▼ `/etc/profile`
 - ▼ `/etc/.login`.



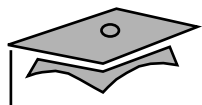
User Initialization Files

- The user's initialization files are set up by the system administrator and placed 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.



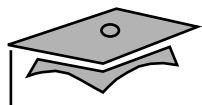
Initialization Files for the Six Shells

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		/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



Shell Variables

Variable Name	Set By	Description
LOGNAME	Set at login	Defines the user's login name.
HOME	Set at login	Sets the path to the user's home directory. Default argument for cd.
SHELL	Set at login	Sets the path to the default shell.
PATH	Set at login	Sets the default path the shell searches to find commands.
MAIL	Set at 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.
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.



Setting Environment Variables

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



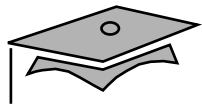
Default User Initialization Files

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



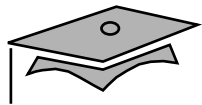
Exercise: Using the Initialization Files

- Objectives
- Tasks
- Discussion
- Solutions



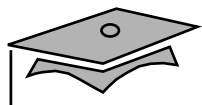
Check Your Progress

- 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`



Check Your Progress

- 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



Module 3

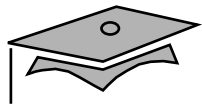
System Security



Objectives

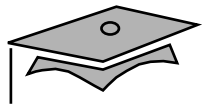
Upon completion of this module, you should be able to:

- 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



Objectives

- 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
- Create, modify, and delete access control lists (ACLs) on files
- Explain how the special permissions `setuid`, `setgid`, and the Sticky Bit can affect system security
- Control remote login access by maintaining three basic network files: `/etc/hosts.equiv`, `$HOME/.rhosts`, and `/etc/ftpusers`



Managing System Security Overview

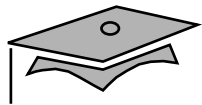
Some basic steps that you should take to manage security at the user, file, system, and network level include:

- 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



The pwconv Command

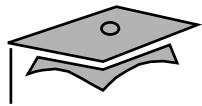
- The pwconv command creates and updates the /etc/shadow file with information from the /etc/passwd file.
- It is the pwconv command that relies on the special value of x in the password field of /etc/passwd.
- The x indicates the password for the user already exists in the /etc/shadow file.



Recording Failed Login Attempts

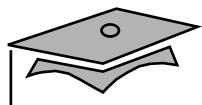
- You can save failed login attempts to a file, which is a useful tool for determining if attempts are being made to break into a system.
- You can record failed login attempts in the file `/var/adm/loginlog`.
- By default, the `loginlog` file does not exist. To enable logging, this file must be created with read and write permissions for `root` only. For example:

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

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.
- If a user is logged on remotely, the remote host name for that user is displayed.

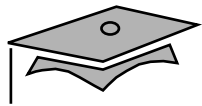


Displaying User Information

- To display detailed information about users either locally or remotely, use the `finger` command.
- `finger` 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, login shell, and the name of the host if logged in remotely.
- Command format:

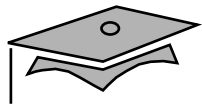
```
finger -m username
```

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



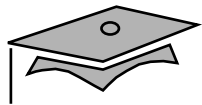
Displaying a Record of Login Activity

- Use the `last` command to display a record of all logins and logouts with the most recent activity at the top of the output.
- The `last` command looks in the `/var/adm/wtmpx` file, which records all logins and logouts.
- Each entry includes user name, the login device, the host logged in from, the date and time logged in, the time of log out, and the total login time in hours and minutes, including entries for system reboot times.



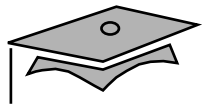
Displaying Users on Remote Systems

- The `rusers` command produces output similar to the `who` command, but displays users logged in on remote hosts.
- A remote host will only respond 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 ]`



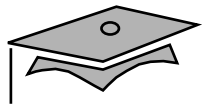
Accessing root Privileges

- You should only log in to the `root` account to perform administration tasks. Avoid performing routine work as `root`.
- This protects the system from unauthorized access, reducing the likelihood that the system will be left unattended with `root` logged in.
- You can become `root` on a system by either:
 - ▼ Logging in directly as `root` and supplying the `root` password.
 - ▼ Logging in as a regular user, then invoke the `su` command and supplying the `root` password.



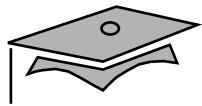
Effective User ID and Effective Group ID

- When you run the `su` command, the UID and GID are changed to the new user. These are referred to as the *effective* user ID (EUID) and *effective* group ID (EGID).
- 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.
- 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.



Using the su Command to Become root

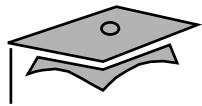
To use the su command to become root:

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

```
$ su
```

```
Password:
```

```
#
```

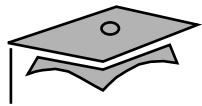
Using the su Command to Become root

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 you switched to, type `whoami` and press Return.

```
# whoami
root
```



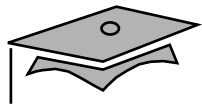
Using the su Command to Become root

5. To determine where you are 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  
$
```

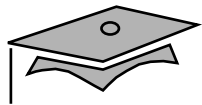


Using the su Command to Become Another Regular User

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, and press Return. Type the password for the user account and press Return.

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



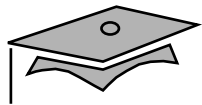
Using the su Command to Become Another Regular User

2. Determine the login name of the user you 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
```



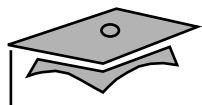
Using the su Command to Become Another Regular User

4. To display the login name of the user you originally logged in as, type `who am i` and press Return.

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

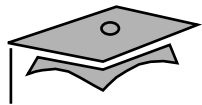
5. To return to the original user status and home directory, type the following command and press Return.

```
$ exit
```



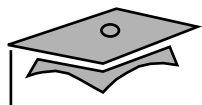
Exercise: User Access

- Objectives
- Tasks
- Discussion
- Solutions



Check Your Progress

- 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



Determining a User's Group Membership

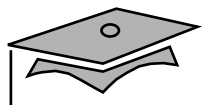
The `groups` command displays group memberships for the user.

- For example, to see what groups you belong to:

```
# groups  
staff class
```

- To list the groups that a specific user belongs to:

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

Identifying a User Account

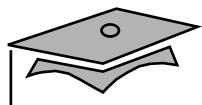
The `id` command is used to identify users by listing their UID, username, group ID, and group name.

- To view your account information:

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

- To view all account information for a specific user:

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



Changing a File's Ownership

The `chown` command is used to change the original owner of a file or directory to another user on the system.

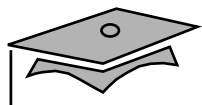
- Changing File Ownership

- ▼ In this example `user1` owns a file named `file7`

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

- ▼ Give this file to a new user named `user2`

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



Changing Directory Ownership

- In this example, user1 owns a directory called dir4

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

- 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
```



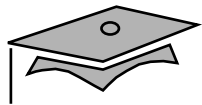
Changing User and Group Ownership Simultaneously

- You can use the `chown` command to change both the ownership and group membership of a file or directory at the same time:

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

- You can use the `-R` option to recursively descend a directory hierarchy, changing ownership and group membership on the directory and its contents:

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



Changing a File's Group Ownership

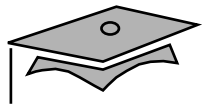
The `chgrp` command is used to change the group ownership of a file or a directory to another group on the system.

- 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`

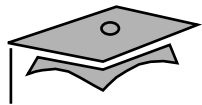
```
# chgrp class file4
# ls -l file4
-rw-r--r--  1 user1  class          874 Jun 1 15:09  file4
```



Special File Permissions

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

- Set user identification (`setuid`) permission
- Set group identification (`setgid`) permission
- Sticky Bit permission



The `setuid` Permission

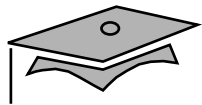
When set user identification (`setuid`) permission is set on an executable file, a user or process that runs this executable file is granted access based on the owner of the file (usually `root`) instead of the user who started the executable.

- The `setuid` permission displays as an “s” in the owner’s execute field, for example:

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

- To set the `setuid` permissions on an executable file use the `chmod` command and the octal value `4000`, for example:

```
# chmod 4555 executable_file
```



The setgid Permission

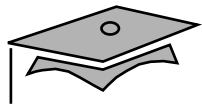
The set group identification (setgid) permission is similar to setuid, except the effective group ID of the user or the process is changed to the group owner of the file.

- The setgid permission displays as an “s” in the group’s execute field, for example:

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

- To set a setgid permission on an executable file use the chmod command and the octal value 2000 for example:

```
# chmod 2555 executable_file
```

The Sticky Bit Permission

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

- The Sticky Bit displays as the letter “t” in the other’s execute field, for example:

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

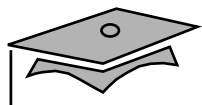
- To set the Sticky Bit permission on a directory, use the `chmod` command and the octal value `1000`, for example:

```
# chmod 1777 public_directory
```



Exercise: File Owners, Groups, and Special Permissions

- Objectives
- Tasks
- Discussion
- Solutions



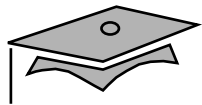
Check Your Progress

- 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



Access Control Lists

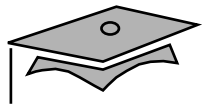
Access Control Lists (ACLs) provide greater control over file access permissions and provide better file security for the file owner, group, other, specific users and specific groups.



ACL Commands

ACLs can be set or viewed using these commands:

Command/Option	Description
<code>getfacl filename(s)</code>	Display 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).
<code>setfacl -f acl_file</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



ACL Entries

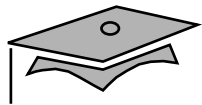
An ACL entry consists of the following fields, separated by colons

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



ACL Entry Examples

- `u[ser]::perm` – Sets permissions for file owner.
- `g[roup]::perm` – Sets permissions for owner's group.
- `o[ther]:perm` – Sets permissions for users other than the owner or owner's group.
- `u[ser]:UID:perm` or `u[ser]:username:perm` – Sets permissions for a specific user.
- `g[roup]:GID:perm` or `g[roup]:groupname:perm` – Sets permission for a specific group.
- `m[ask]:perm` – Sets ACL mask. Indicates the maximum permissions allowed for all users, (except the owner) and for all groups.

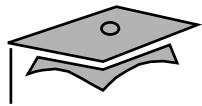


Adding ACL Permissions on a File

To add ACL permissions on a file, use the `setfacl -m` command.

- This example creates an ACL entry on `file.txt` for a specific user named `user8` with permissions to read and write this 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:---
```

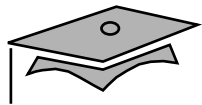



Modifying ACL Permissions on a File

This example modifies the permissions of a file's 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:---
```



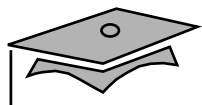
Determining if a File has an ACL

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

- By using the `getfacl` command
- By using the `ls -l` command

Using the `ls -l` command on any file with 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
```

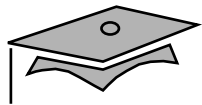


Deleting an ACL Entry on a File

To delete an ACL entry from a file, use the `setfacl -d` command and specify the entry type and the UID (username) or GID (groupname).

- This example deletes an ACL entry from `file.txt`:

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



Replacing an Entire ACL on a File

To replace the entire ACL on a file, you must specify the basic set of user, group, other, mask permissions and filename(s).

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

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



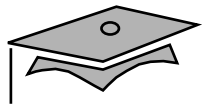
Exercise: Access Control Lists

- Objectives
- Tasks
- Discussion
- Solutions



Check Your Progress

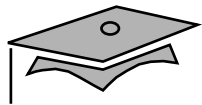
- Create, modify, and delete access control lists (ACLs) on files



Managing Remote Access Issues

The three network files 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



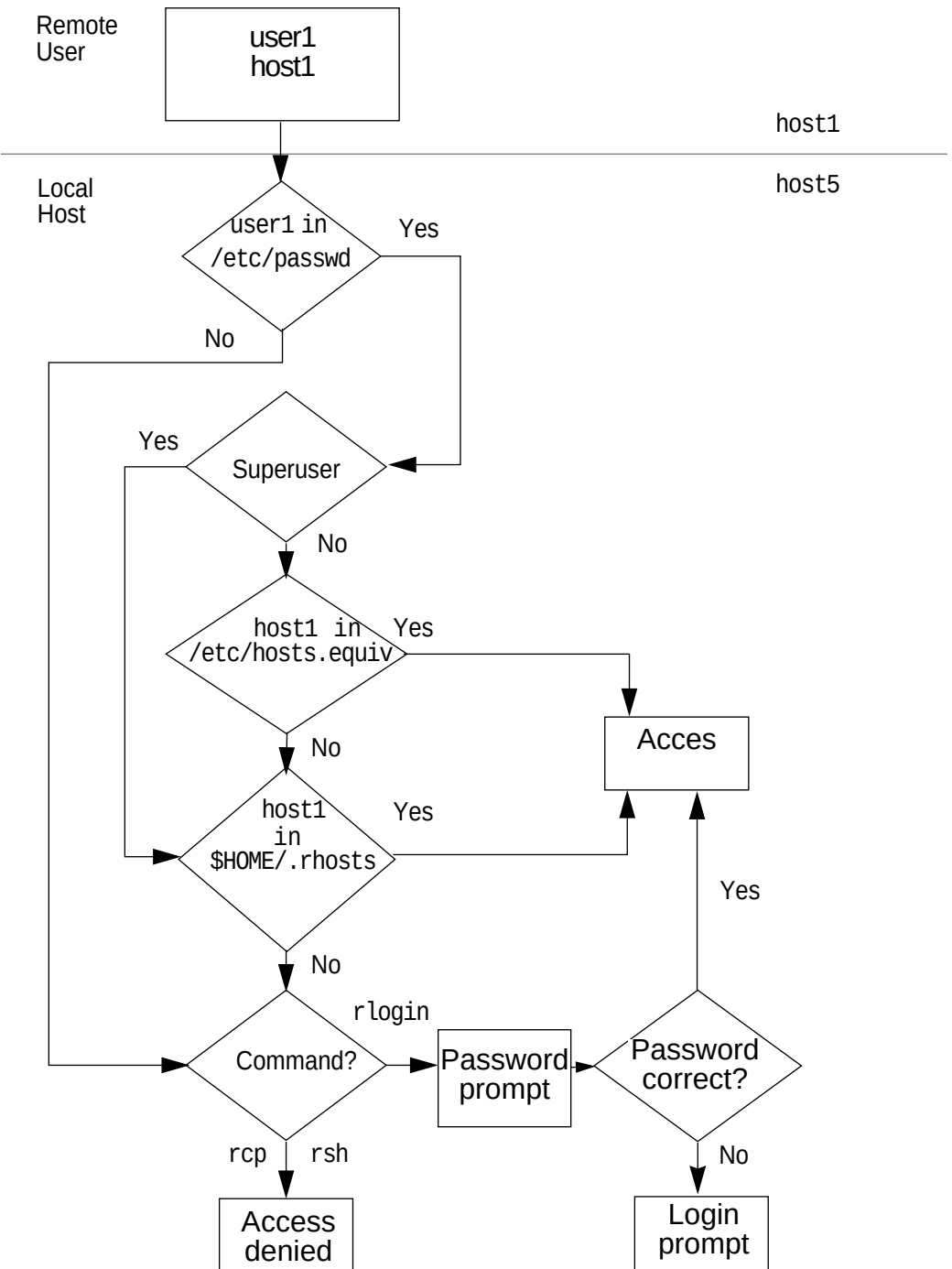
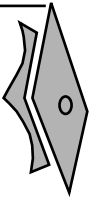
The `/etc/hosts.equiv` and `$HOME/.rhosts` Files

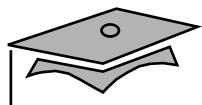
Typically, when login access to a local host is requested by a remote user, the first file read is the `/etc/passwd` file.

- If there is no entry in the local host's `/etc/passwd` file for the remote user, access is denied.

The `/etc/hosts.equiv` and `$HOME/.rhosts` files bypass this standard password-based authentication to determine if a remote user is allowed access to the local host.

- Based on information contained in these two files, (if they exist), determines if remote user access is granted or denied.





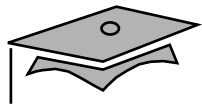
Entries in `/etc/hosts.equiv` and `$HOME/.rhosts` Files

These two files have the same format, although the same entries in each file have a different effect.

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

Both files contain a list of one-line entries, which can include:

- *hostname*
- *hostname username*
- *+*



The `/etc/hosts.equiv` File

For regular users, this file is used to identify remote hosts and remote users considered to be *trusted*.

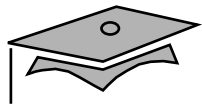
- If a local host's `/etc/hosts.equiv` file contains the a remote host name, all regular users of that remote host are trusted and need not supply a password to log in to the local host.
- Useful for sites where it is common for regular users to have accounts on many different systems, eliminating the security risk of sending ASCII passwords over the network.
- The `/etc/hosts.equiv` file does not exist by default.



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.
- If a remote host name is listed in this file, it is considered to be a trusted host and remote user access is granted on the local host.
- The `$HOME/.rhosts` file does not exist by default, it must be created in the user's home directory.

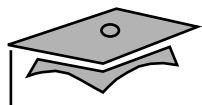


Restricting FTP Logins

The file `/etc/ftpusers` is used to list the names of users prohibited from running an `ftp` login on the system.

- Each line entry contains the login name for each restricted user.
- The default `ftpusers` file lists these account entries:

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



The `/etc/shells` File

The `/etc/shells` file contains a list of the shells on the system. This file does not exist by default.

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

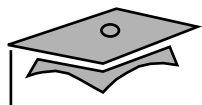
By creating this file, each shell to be recognized by the system must have a single line entry, consisting of the shell's path, relative to `/` (root), for example:

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



Exercise: Managing Remote Security Issues

- Objectives
- Tasks
- Discussion
- Solutions



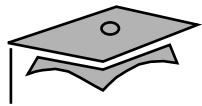
Check Your Progress

- Control remote login access by maintaining three basic network files: `/etc/hosts.equiv`, `$HOME/.rhosts`, and `/etc/ftpusers`



Module 4

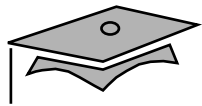
The Directory Hierarchy



Objectives

Upon completion of 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



Solaris File Types

The Solaris Operating Environment supports a standard set of files, which 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



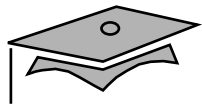
Identifying the File Types

Use the `ls -l` command to distinguish the different file types.

- The character in the first column of information indicates file type, for example:

```
# cd /etc ; ls -l
drwxr-xr-x  2 adm      adm          512 Apr  3 10:42 acct
lrwxrwxrwx  1 root     root          14 Apr  3 11:05 aliases -> ./mail/aliases
-rw-r--r--  1 root     bin           50 Apr  3 10:45 auto_home
(output truncated)
```

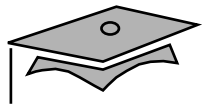
```
# cd /devices/pci@1f,0/pci@1,1/ide@3 ; ls -l
brw-----  1 root     sys          136,  0 Apr  3 11:11 dad@0,0:a
crw-----  1 root     sys          136,  0 Apr  3 11:11 dad@0,0:a,raw
(output truncated)
```



File Types

The characters that identify each file type includes:

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



File Names, Inodes, and Data Blocks

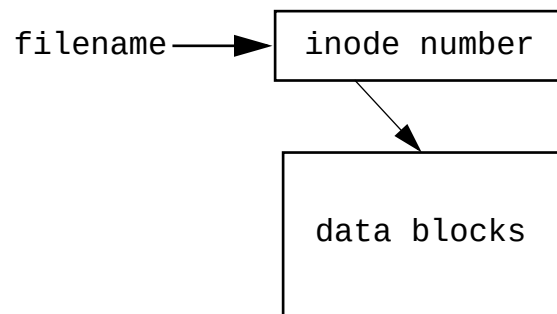
All files in the Solaris Operating Environment make use of a file name and a record called an inode. Most files also make use of data blocks.

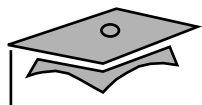
- File names are the objects most often used to access and manipulate files.
- Inodes are the objects the system uses to record information about a file.
- Data blocks are units of disk space used to store data.



File Names, Inodes, and Data Blocks

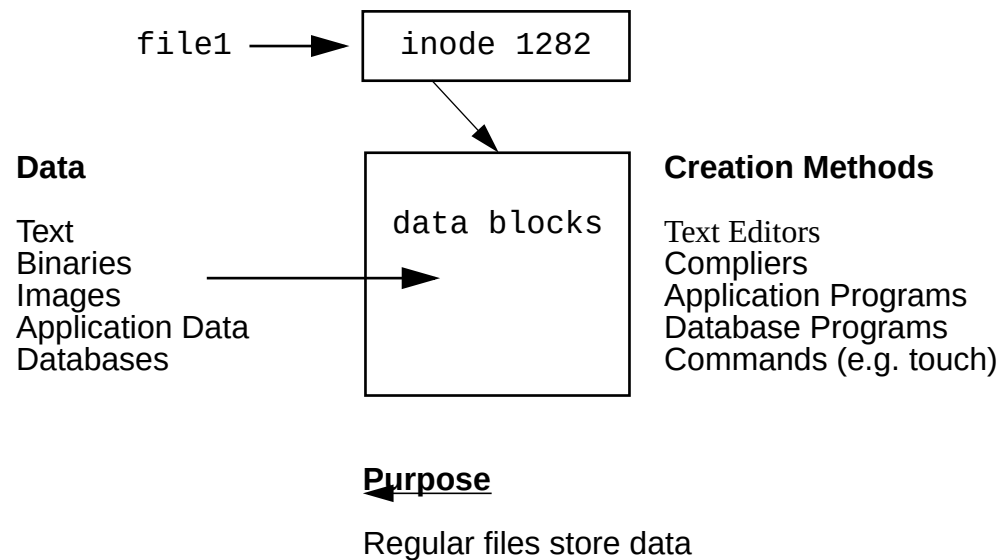
A filename is associated with an inode, and an inode provides access to data blocks.

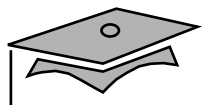




Regular Files

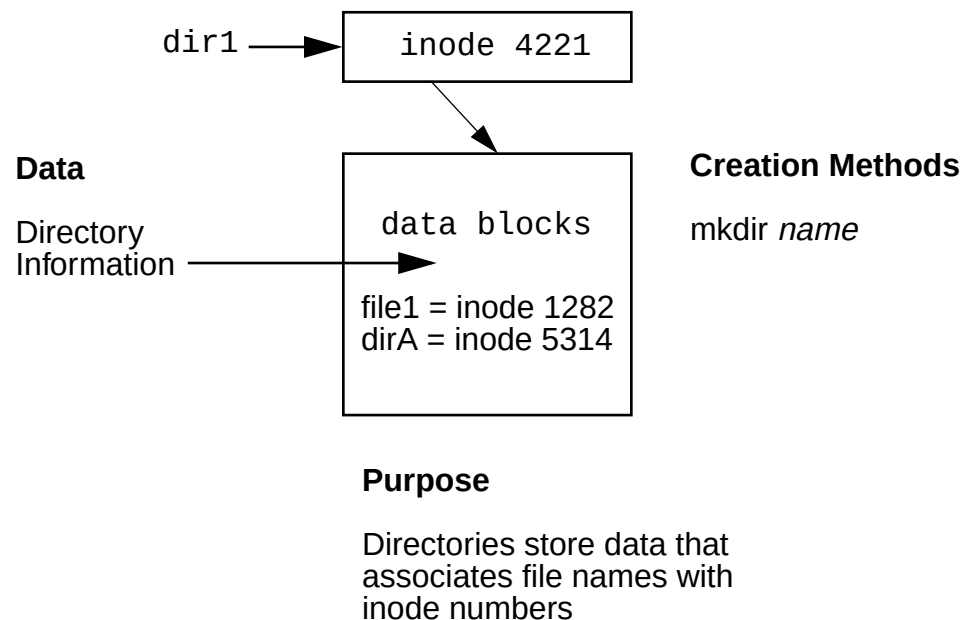
A regular file holds data.

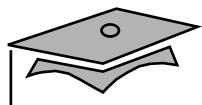




Directories

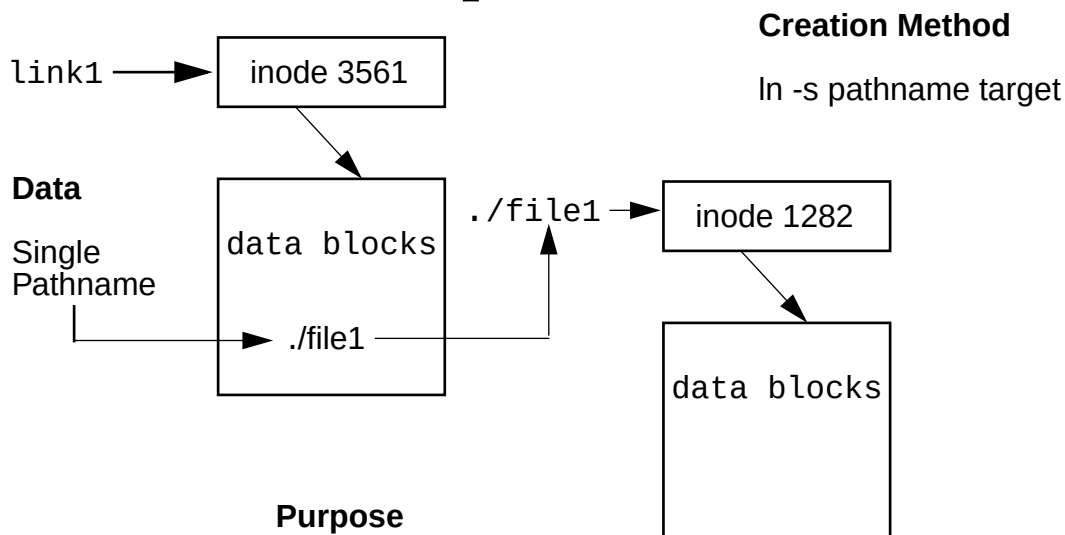
Directories store information that associates file names with inode numbers.

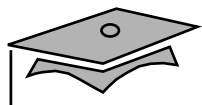




Symbolic Links

A symbolic link is a file that points to another file. A symbolic link contains the pathname of the file to which it points.





Device Files

A device file provides access to a device.

Data

Major and Minor
Device Numbers

dad@00,0:a



inode 90681

Creation Methods

devfsadm (Solaris 8)
drvconfig (<= Solaris 7)
mknod (Solaris 1)

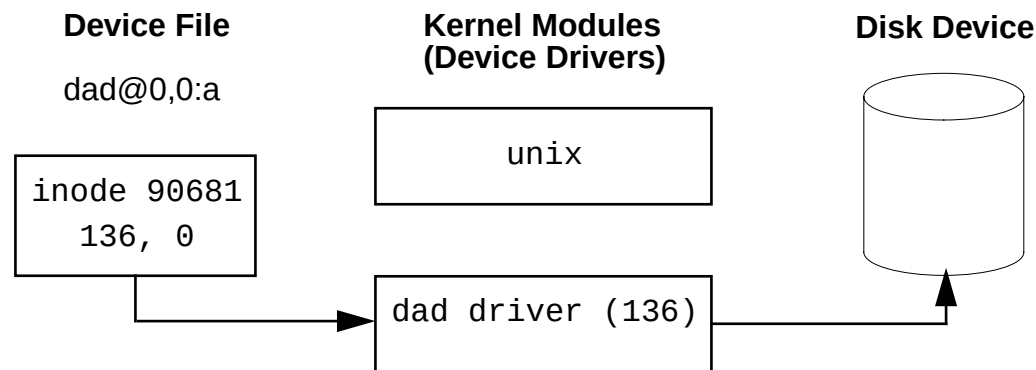
Purpose

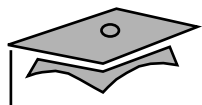
Device files activate devices.
Their major and minor device
numbers refer to specific device
drivers and individual devices



A Device File Example

This example shows the relationship between a 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.





Two Categories of Device Files

Device files fall into two categories: character-special devices and block-special devices.

- Character Device Files:

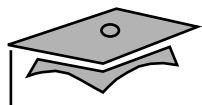
- ▼ The file type “c” identifies character device files.

```
crw----- 1 root    sys      136,  0 Apr  3 11:11 dad@0,0:a,raw
```

- Block Device Files:

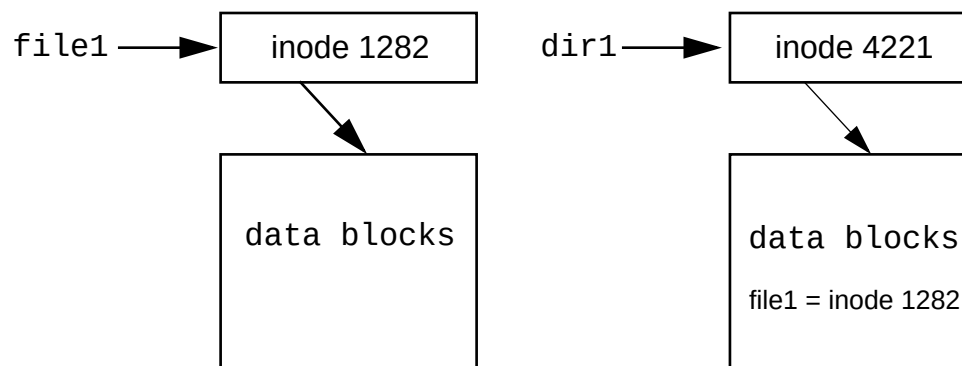
- ▼ The file type “b” identifies block device files.

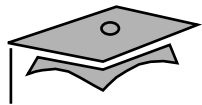
```
brw----- 1 root    sys      136,  0 Apr  3 11:11 dad@0,0:a
```



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.

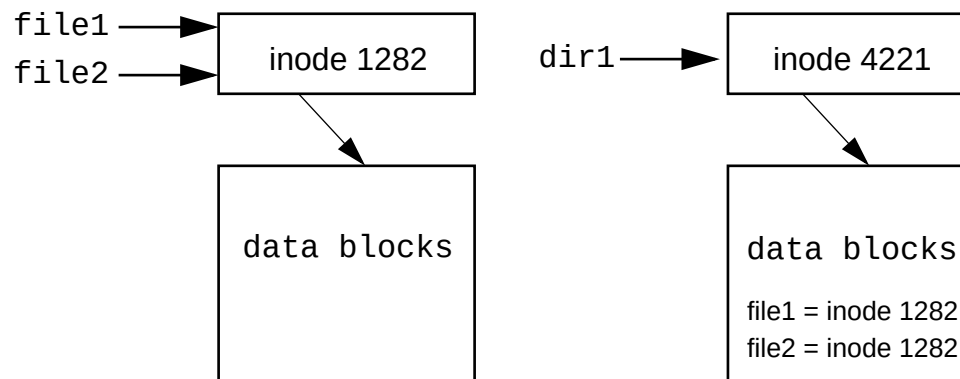


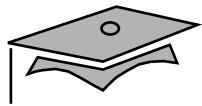


A Hard Link Example

Using the `ln` command, you can create new hard links to regular files.

- The command `ln file1 file2` would create a new directory entry called `file2`, associated with the same inode associated with `file1`.

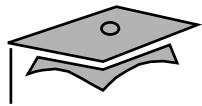




The root Subdirectories

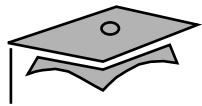
The Solaris Operating Environment is comprised of a hierarchy of critical system directories and files that are necessary for the operating system to function properly.

- `/` – Root of the overall filesystem name space.
- `/bin` – Location for standard system commands, and binary files.
- `/dev` – Primary location for logical device names.
- `/devices` – Primary location for physical device names.



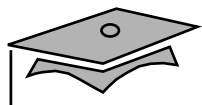
The root Subdirectories

- `/etc` – Host-specific system administrative configuration files and databases.
- `/export` – Default directory for commonly shared file systems.
- `/home` – Default directory or mount point for users home directories.
- `/kernel` – Directory of platform-independent loadable kernel modules required as part of the boot process.
- `/mnt` – Convenient, temporary mount point for file systems.



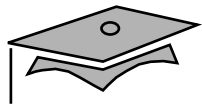
The root Subdirectories

- `/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.
- `/var` – Directory for varying files, which usually includes temporary, logging, or status files.



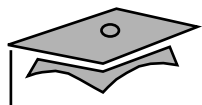
Exercise: Identifying File Types

- Objectives
- Tasks
- Discussion
- Solutions



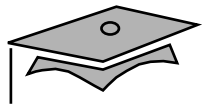
Check Your Progress

- 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



Module 5

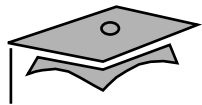
Device Configuration



Objectives

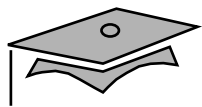
Upon completion of this module, you should be able to:

- 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

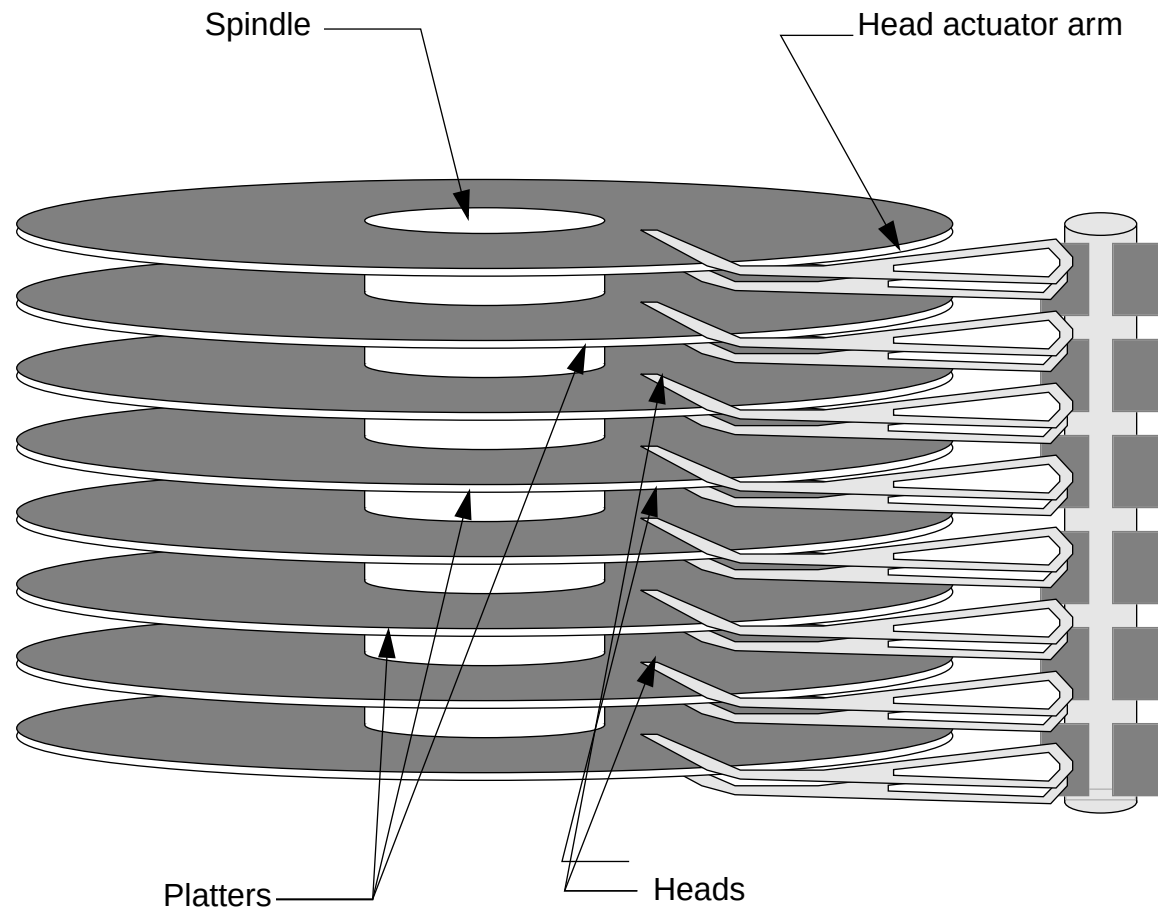


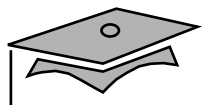
Objectives

- 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

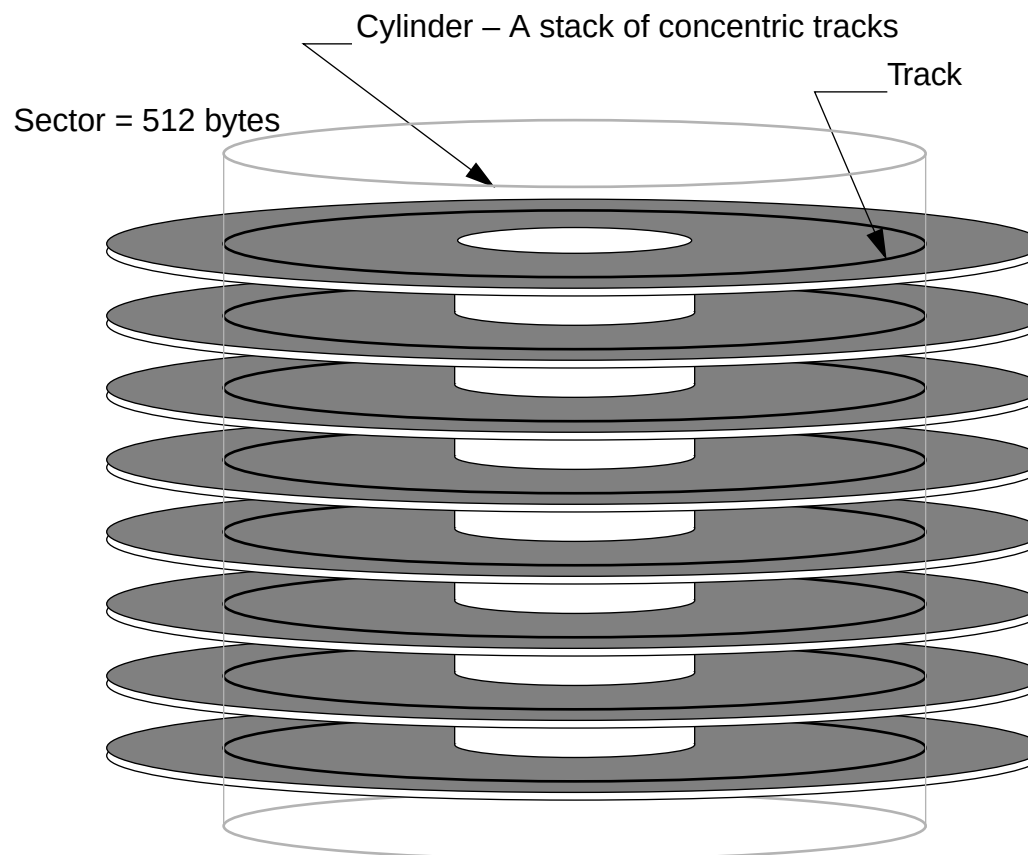


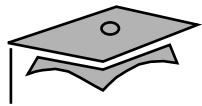
Basic Architecture of a Disk





Components of a Disk Platter

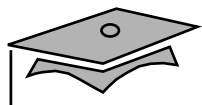




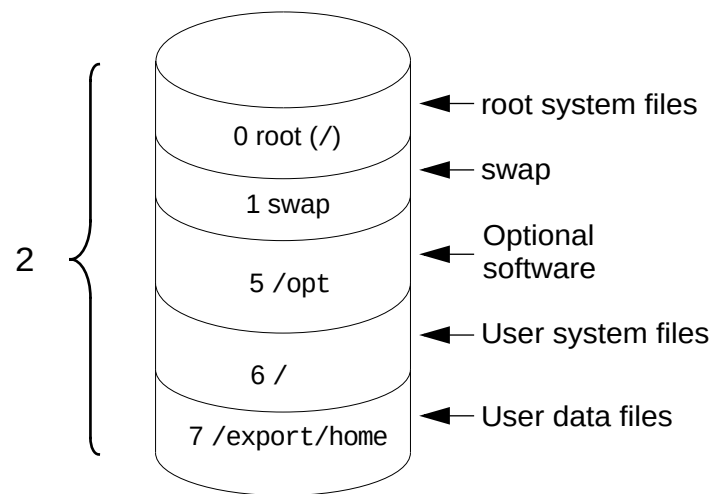
Disk Slices Defined

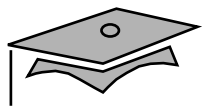
Disks can be divided into individual partitions, known as *slices*.

- Slices are groupings of cylinders commonly used to organize data by function.
- 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.



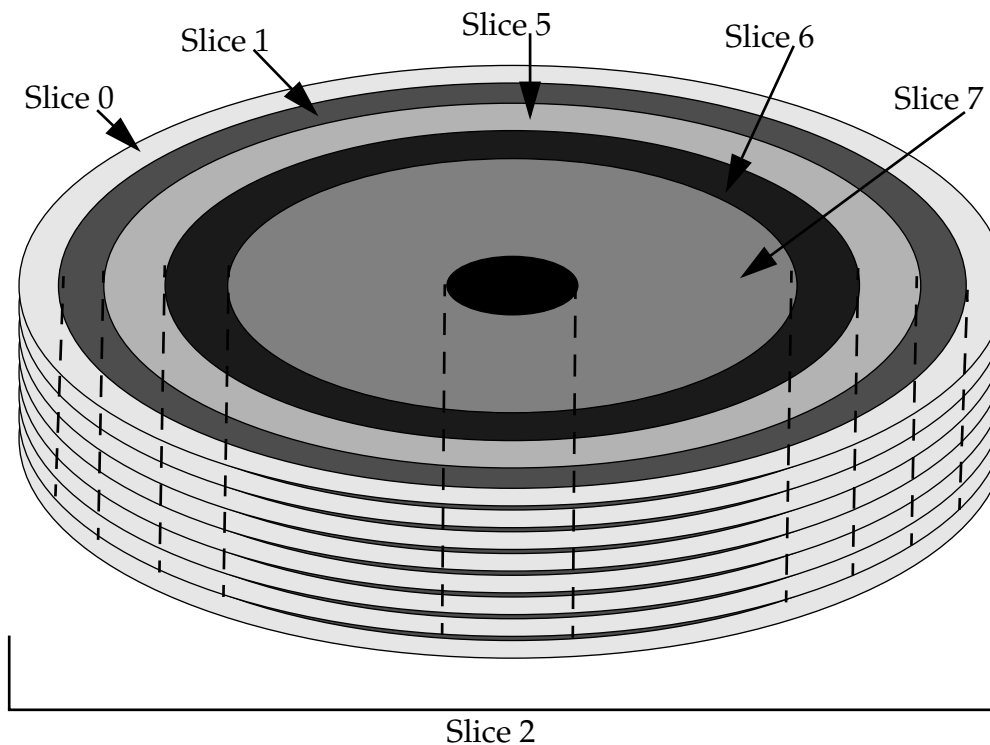
The Boot Disk

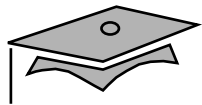




How Slices Reside on Disk

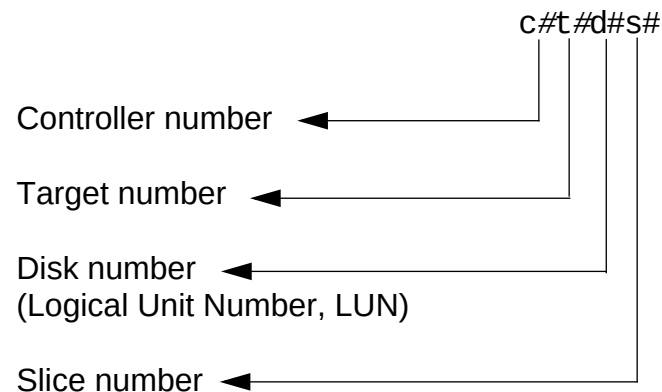
Each slice is defined by a starting cylinder and an ending cylinder, which determine the size of a slice.

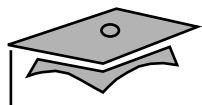




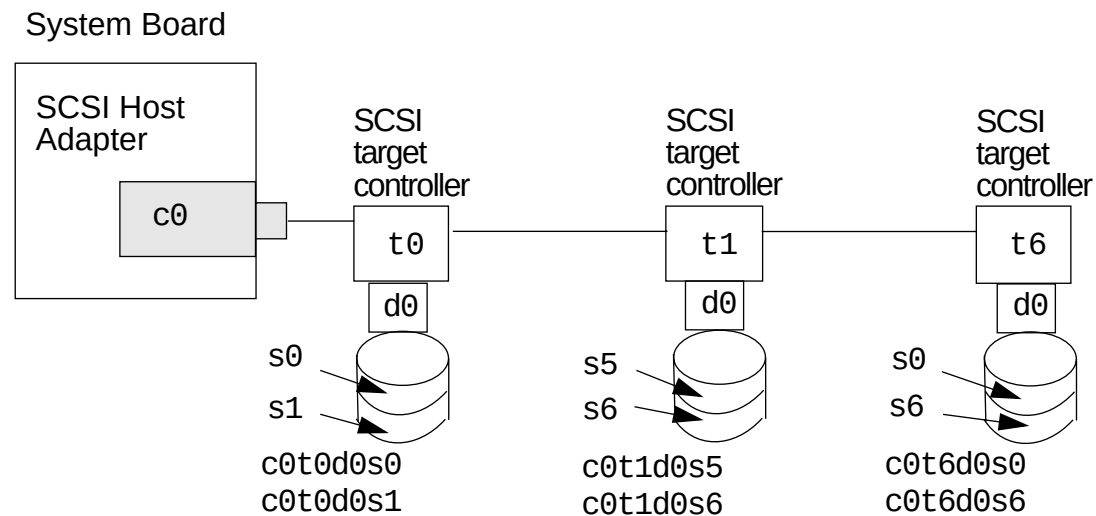
Disk Slice Naming Convention

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.



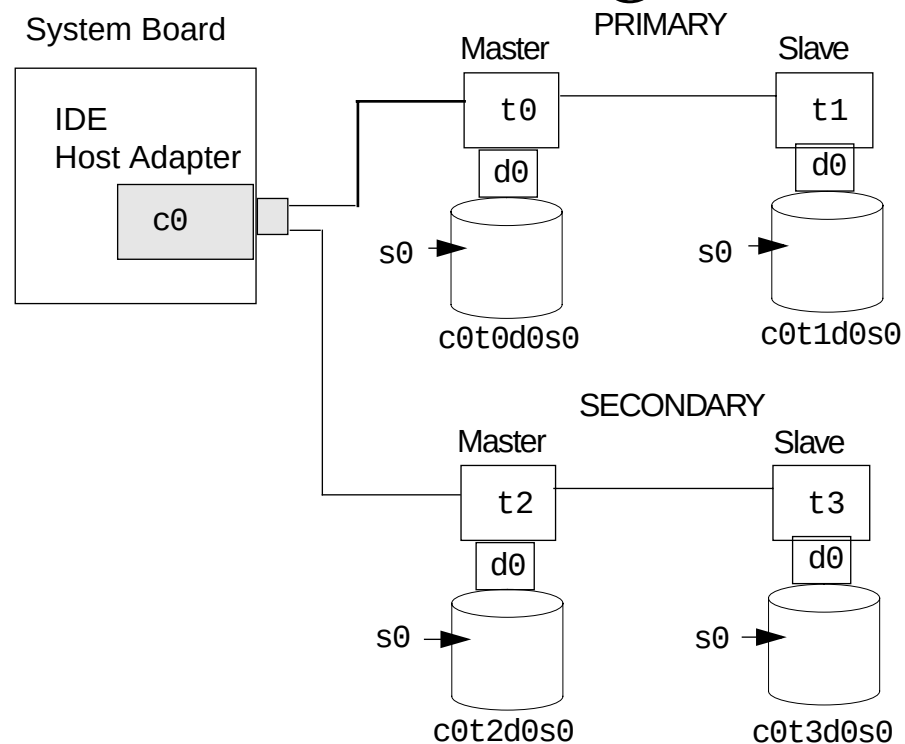


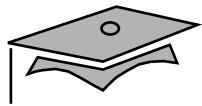
An Embedded SCSI Configuration





An IDE Configuration

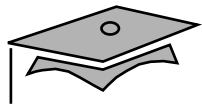




Device Naming Conventions

In the Solaris Operating Environment, all devices have three different types of name, depending on how the device is being referenced.

- Logical device names
- Physical device names
- Instance names



Logical Device Names

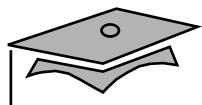
Logical device names are used to refer to a device on the command line.

- 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.

Logical device name for a disk contains the controller number, target number, disk number, and slice number:

`ls /dev/dsk`

<code>c0t0d0s0</code>	<code>c0t0d0s4</code>	<code>c0t3d0s0</code>	<code>c0t3d0s4</code>	<code>c0t6d0s0</code>	<code>c0t6d0s4</code>
<code>c0t0d0s1</code>	<code>c0t0d0s5</code>	<code>c0t3d0s1</code>	<code>c0t3d0s5</code>	<code>c0t6d0s1</code>	<code>c0t6d0s5</code>
<code>c0t0d0s2</code>	<code>c0t0d0s6</code>	<code>c0t3d0s2</code>	<code>c0t3d0s6</code>	<code>c0t6d0s2</code>	<code>c0t6d0s6</code>
<code>c0t0d0s3</code>	<code>c0t0d0s7</code>	<code>c0t3d0s3</code>	<code>c0t3d0s7</code>	<code>c0t6d0s3</code>	<code>c0t6d0s7</code>

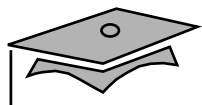


Physical Device Names

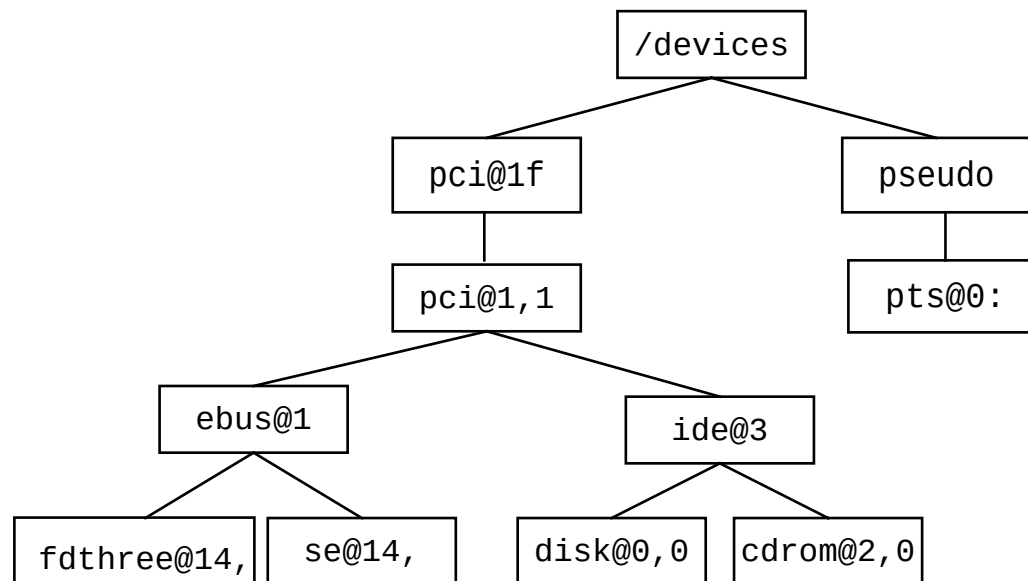
Physical device names uniquely identify the physical location of the hardware devices on the system

- Physical device names are maintained in the `/devices` directory.

```
# 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
```



The /devices directory Structure

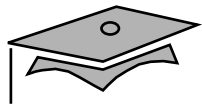




Instance Names

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

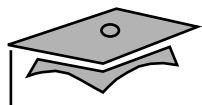
- An instance name is simply a shortened name for the physical device name, for example:
 - ▼ *sdn*
 - ▼ *dadn*



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.
- Maintained by the kernel



Sample /etc/path_to_inst File

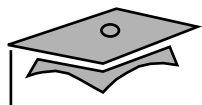
```
# more /etc/path_to_inst
#
#      Caution! This file contains critical kernel state
#
"/pci@1f,0" 0 "pci"
"/pci@1f,0/pci@1,1/ide@3/sd@2,0" 2 "sd"
"/pci@1f,0/pci@1,1/ide@3/dad@0,0" 0 "dad"
"/pci@1f,0/pci@1,1/ebus@1" 0 "ebus"
"/pci@1f,0/pci@1,1/ebus@1/fdthree@14,3023f0" 0 "fd"
"/pci@1f,0/pci@1,1/ebus@1/su@14,3062f8" 1 "su")
"/pci@1f,0/pci@1,1/ebus@1/se@14,400000" 0 "se")
"/pci@1f,0/pci@1,1/ebus@1/su@14,3083f8" 0 "su"
"/pci@1f,0/pci@1,1/ebus@1/ecpp@14,3043bc" 0 "ecpp"
"/pci@1f,0/pci@1,1/ebus@1/SUNW,CS4231@14,200000" 0 "audiocs"
"/pci@1f,0/pci@1,1/ebus@1/power@14,724000" 0 "power"
"/pci@1f,0/pci@1,1/network@1,1" 0 "hme"
"/pci@1f,0/pci@1,1/SUNW,m64B@2" 0 "m64"
"/pci@1f,0/pci@1" 1 "simba"
"/options" 0 "options"
"/pseudo" 0 "pseudo"
```



The prtconf Command

The `prtconf` command displays system configuration information; listing all instances of devices.

```
# 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
```



The format Command

The format command displays logical and physical device names for all currently available disks.

```
# format
```

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

```
AVAILABLE DISK SELECTIONS:
```

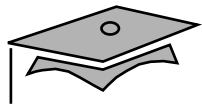
```
0. c0t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
```

```
   /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):
```

The Solaris 8 Operating Environment Device Reconfiguration

The Solaris 8 Operating Environment device reconfiguration is handled with the `devfsadm` command.

- Limit Configuration to a Specific Device Type

```
# devfsadm -c disk
```

- Specify Multiple Device Classes

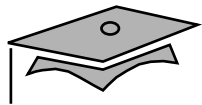
```
# devfsadm -c disk -c tape -c audio
```

- Limit Configuration to a Specific Device Driver

```
# devfsadm -i dad
```

```
# devfsadm -i sd
```

```
# devfsadm -i st
```



Pre-Solaris 8 Operating Environment Device Reconfiguration

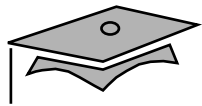
Pre-Solaris 8 Operating Environment device reconfiguration is handled with the `drvconfig` command

- To add a new disk device:

```
# drvconfig -i dad  
# disks
```

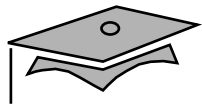
- To add a new tape drive:

```
# drvconfig -i st  
# tapes
```



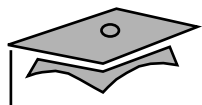
Exercise: Disk Configuration and Naming

- Objectives
- Tasks
- Discussion
- Solutions



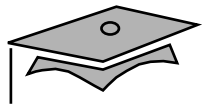
Check Your Progress

- 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



Module 6

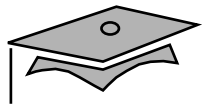
Disks, Slices, and Format



Objectives

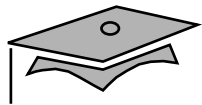
Upon completion of this module, you should be able to:

- 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



Objectives

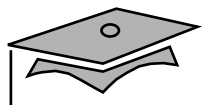
- Demonstrate how to view the disk's VTOC using two different commands: `verify` and `prtvto`
- Use the `fmthard` command to update the VTOC on a disk



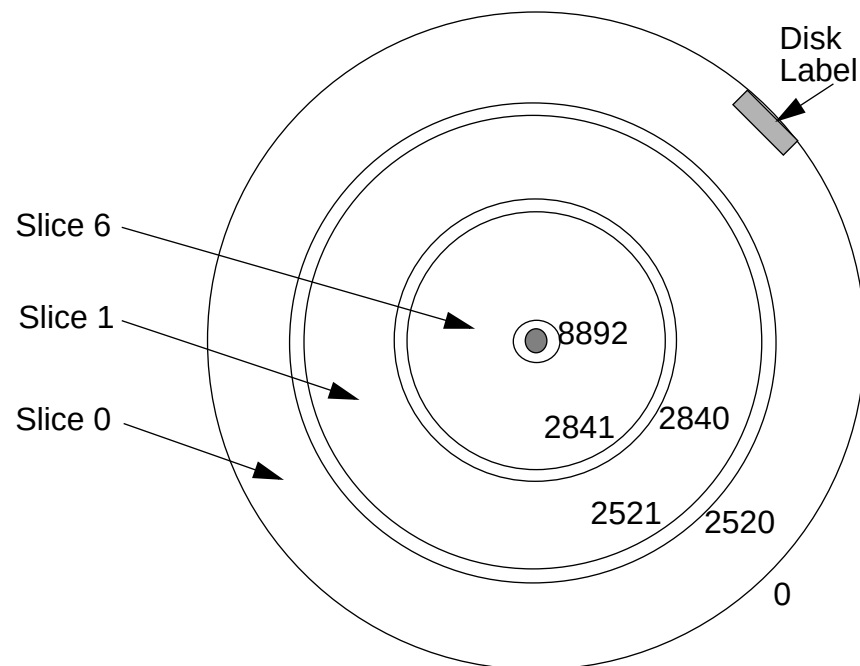
Disk Slices and the format Utility

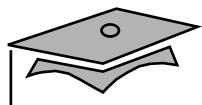
The format utility is a system administration tool used primarily to prepare hard disk drives for use in the Solaris Operating Environment.

- The main reason a system administrator uses the format utility is to divide a disk into disk slices.
- To divide a disk into slices, you 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



Disk Labels and Partition Tables





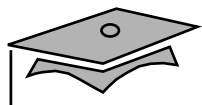
Disk Partition Table

The partition table primarily defines partition boundaries and the number of cylinders in a partition.

Current partition table (original):

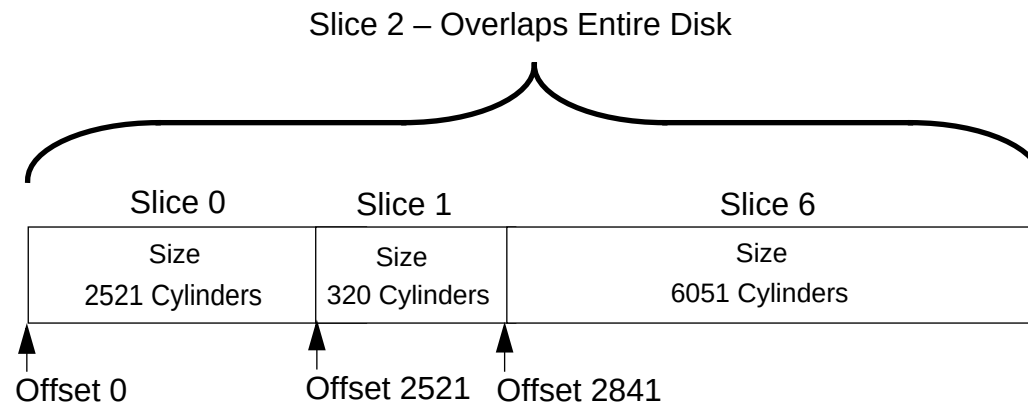
Total disk cylinders available 8892 + 2 (reserved cylinders)

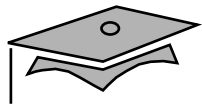
Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 2520	1.14GB	(2521/0/0) 2382345
1	swap	wu	2521 - 2840	147.66MB	(320/0/0) 302400
2	backup	wm	0 - 8892	4.01GB	(8892/0/0) 8402940
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	usr	wm	2841 - 8892	2.73GB	(6051/0/0) 5718195
7	unassigned	wm	0	0	(0/0/0) 0



Defining Disk Slices

Disk slices are defined by an offset and a size in cylinders. The offset is the distance from cylinder 0.

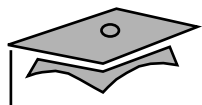




Disk Space and Undesirable Conditions

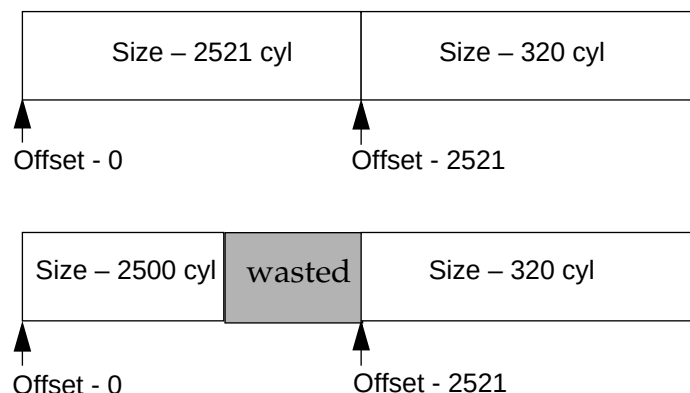
When creating or changing disk slices, two types of undesirable conditions can occur:

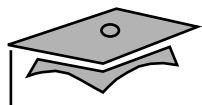
- Wasted disk space
- Overlapping disk space



Wasted Disk Space

Wasted disk space occurs when one or more cylinders are not allocated to a disk slice.

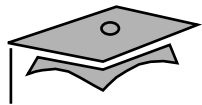




Overlapping Disk Slices

Overlapping disk slices occurs when one or more cylinders are allocated to more than one disk slice.

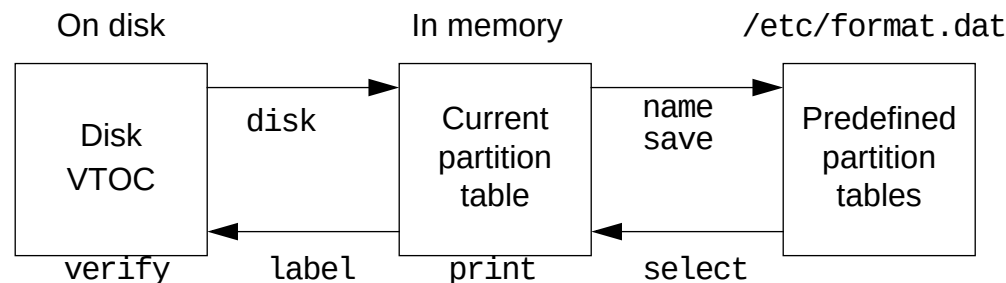


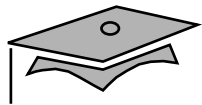


The `/etc/format.dat` File

The format utility reads the file `/etc/format.dat`

- This file is a table of available disk types and a set of predefined partition tables which can be used to partition a disk quickly.

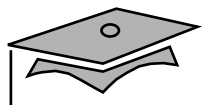




Viewing a Disk's VTOC

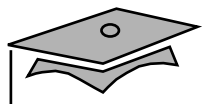
There are two methods available for locating and viewing a disk's label, or VTOC.

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



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

```
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
```



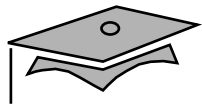
Reading a Disk's VTOC with the prtvtoc Command

```
# 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
```



Exercise: Disks, Slices, and Format

- Objectives
- Tasks
- Discussion
- Solutions



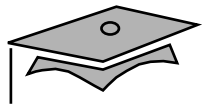
Check Your Progress

- 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 a disk's VTOC



Module 7

The Solaris Operating Environment ufs File System



Objectives

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

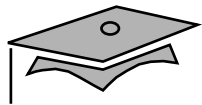
- Describe the three different types of file systems in the Solaris Operating Environment
- Define the term file system
- List the components that are contained in the structure of a file system
- Create a new ufs file system using the newfs command



File System Types Supported by the Solaris Operating Environment

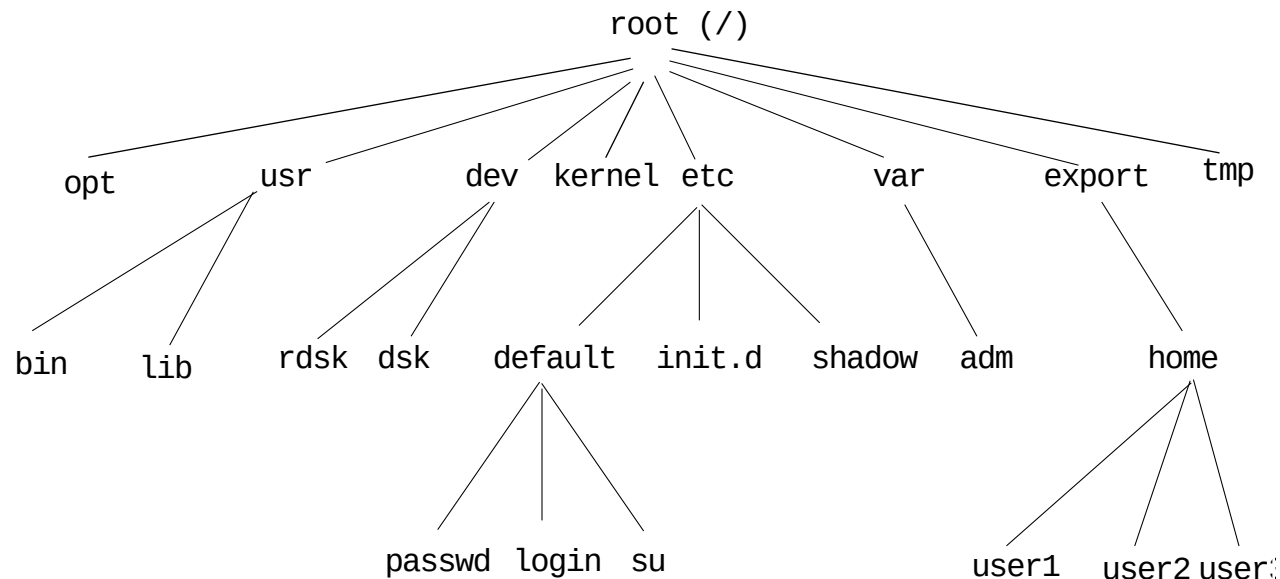
The Solaris Operating Environment supports three different types of file systems:

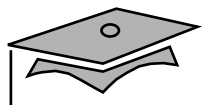
- Disk-based file systems
 - ▼ ufs, hdfs, pcfs, udf
- Distributed file systems
 - ▼ nfs
- Pseudo file systems
 - ▼ tmpfs, swapfs, fdcs, procfs



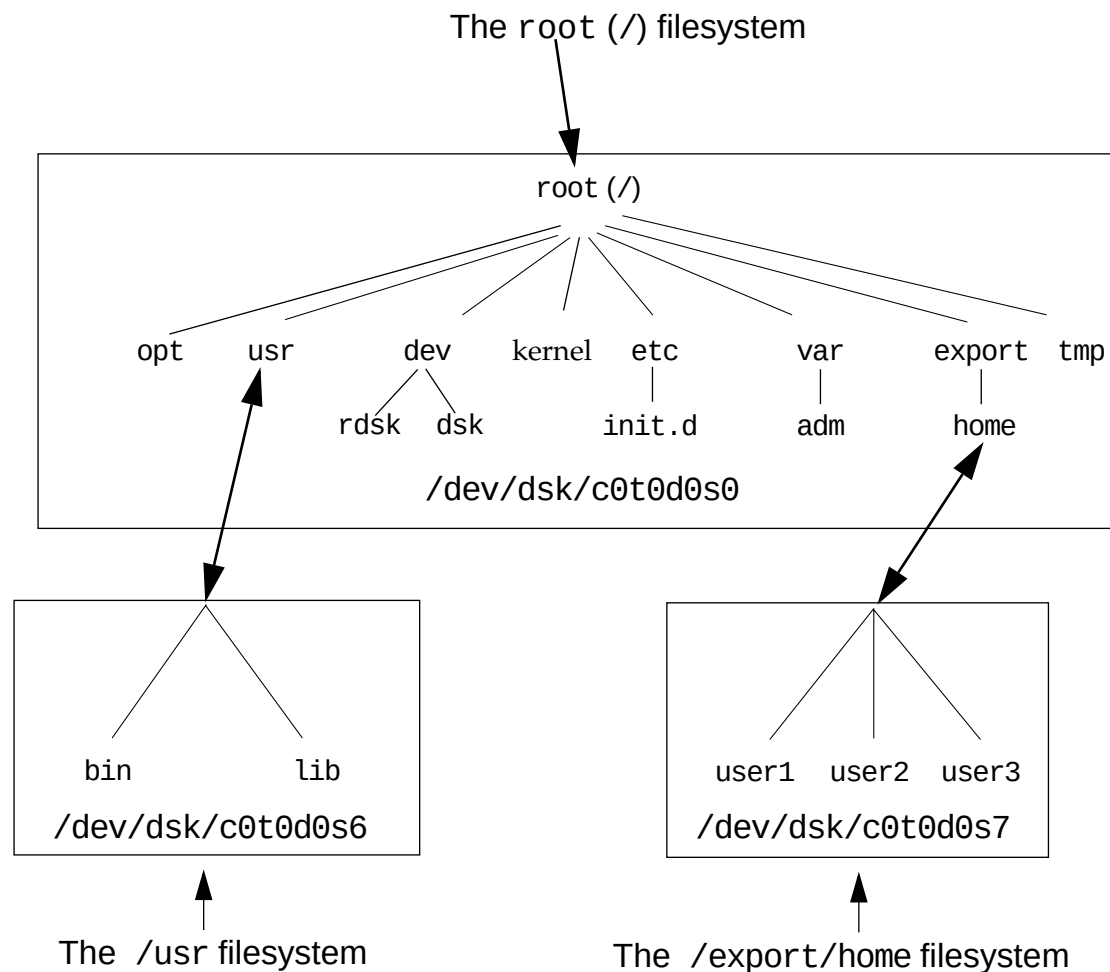
The Solaris ufs File System

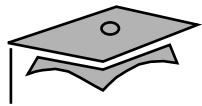
The Solaris Operating Environment stores data in a logical file hierarchy. This file hierarchy is referred to as the Solaris directory tree, which is formed by a number of file systems.





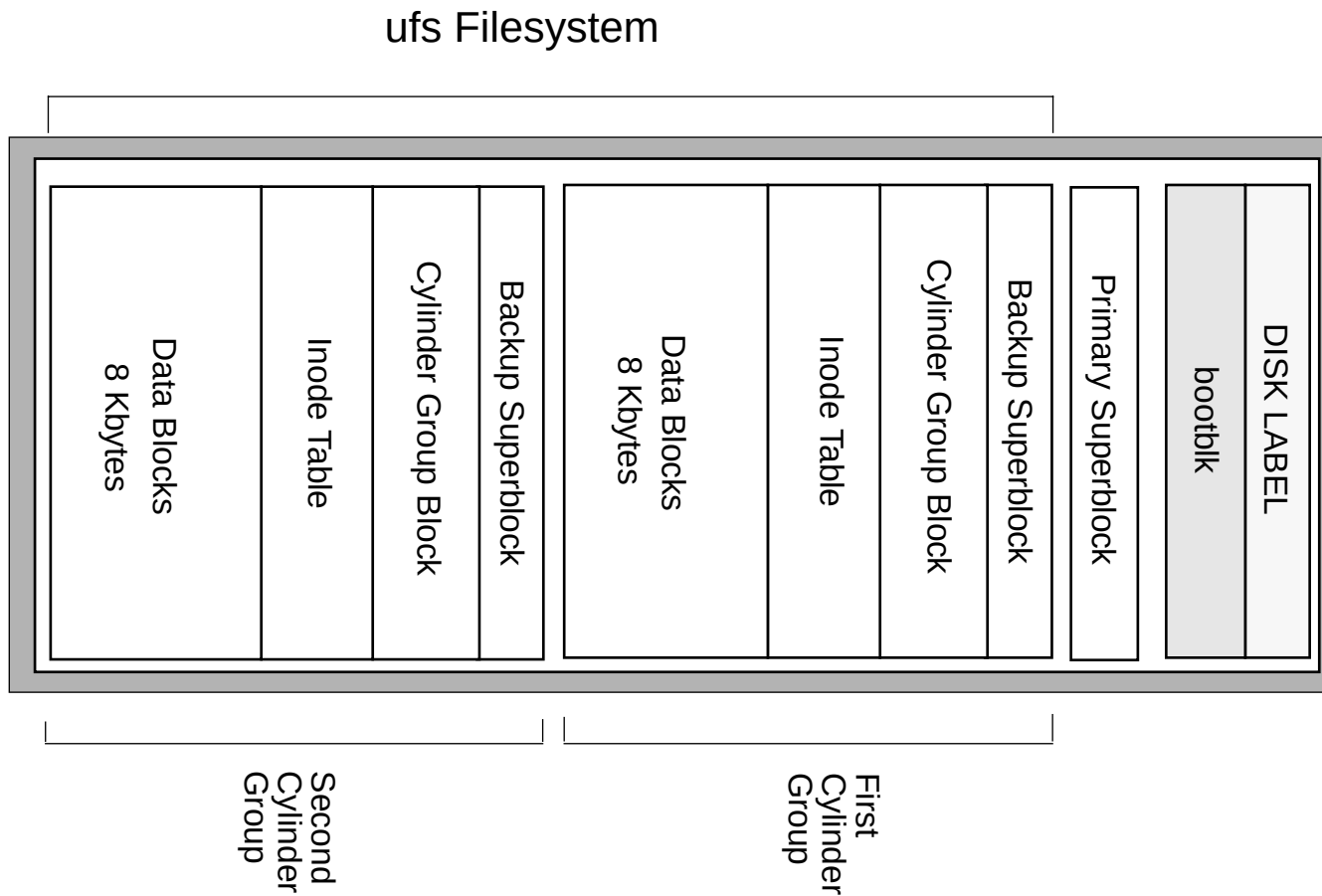
The Solaris ufs File System

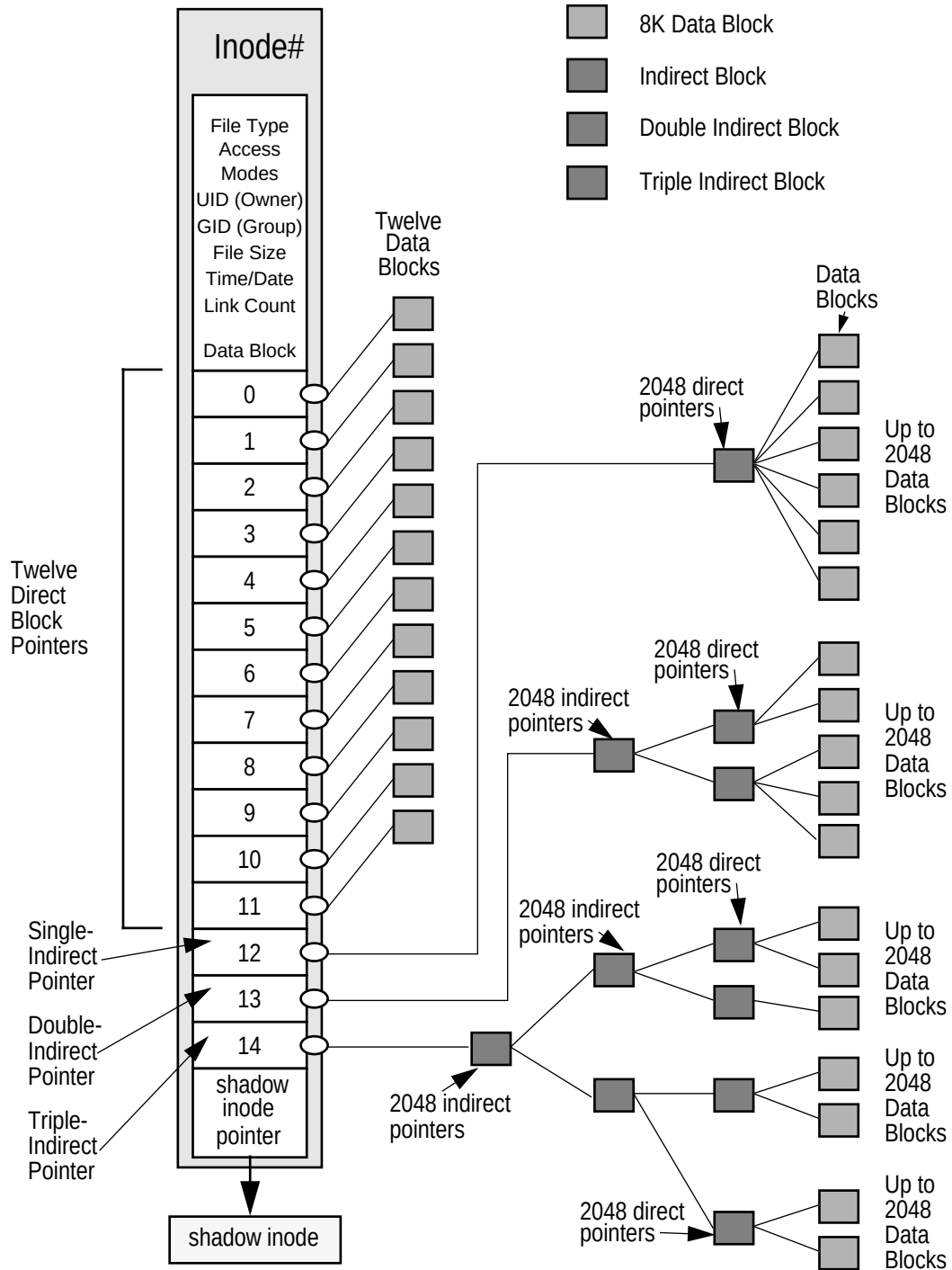
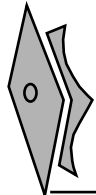


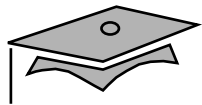


Basic Disk Structures

- The Disk Label (VTOC)
- The Boot Block
- The Superblock
- Cylinder Groups
 - ▼ Cylinder Group Blocks
 - ▼ Inode Table
 - ▼ Data Blocks



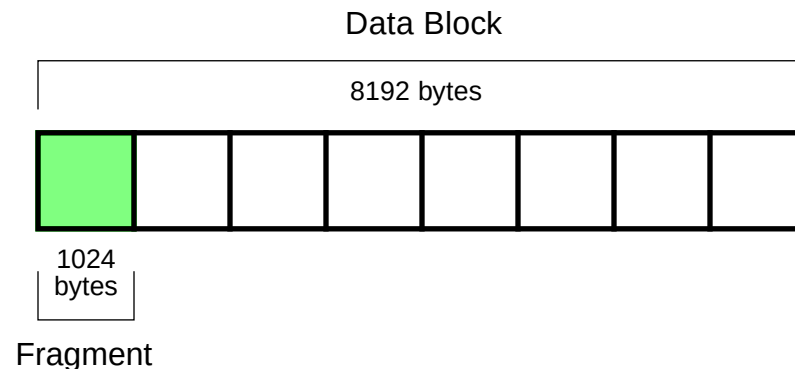


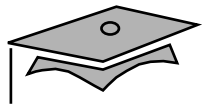


Data Blocks and Fragmentation

The method used by the ufs filesystem to store the contents of a file which is not large enough to fill one data block is called *fragmentation*.

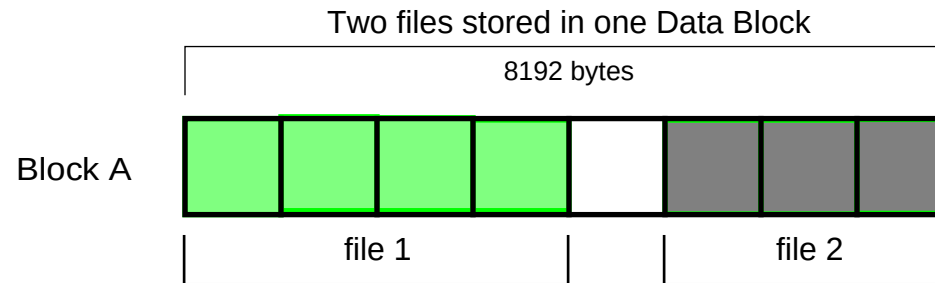
- Data blocks can be divided into eight fragments of 1024 bytes each, for the storage of small files.



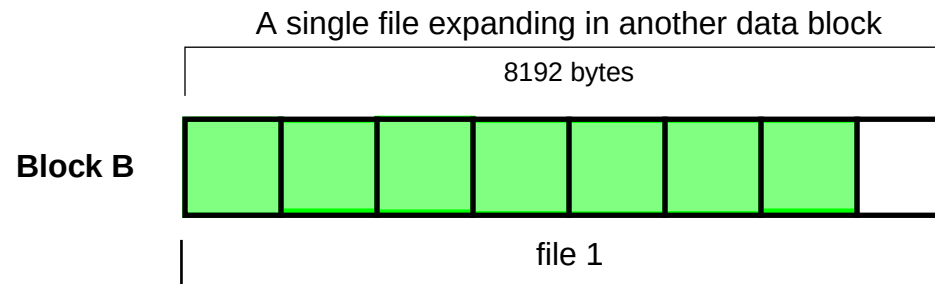


Data Blocks and Fragmentation

The content of two files can be stored in fragments of the same data block.

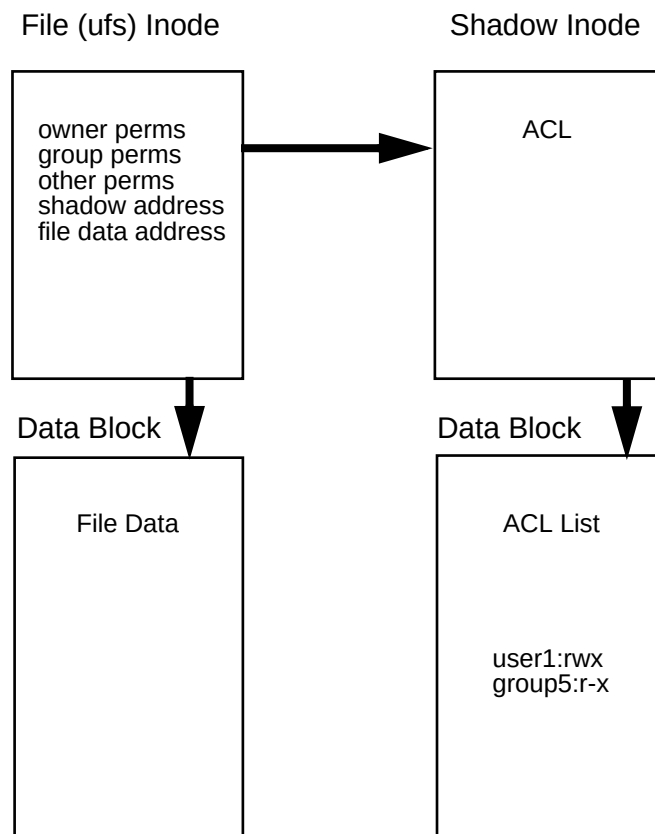


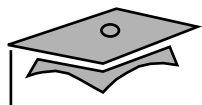
The ufs file system does not allow *fragments* of the same file to be stored in two different data blocks.





Shadow Inode





Creating ufs File Systems

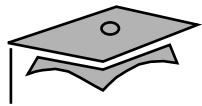
To construct a ufs file system on a disk slice, use the `newfs` command.

```
# newfs /dev/rdisk/c1t3d0s0
newfs: construct a new filesystem /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
```



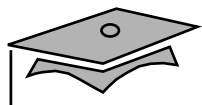

Exercise: Disks, Slices, and Format

- Objectives
- Tasks
- Discussion
- Solutions



Check Your Progress

- 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



Module 8

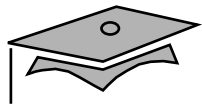
Mounting File Systems



Objectives

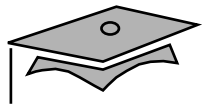
Upon completion of this module, you should be able to:

- 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



Objectives

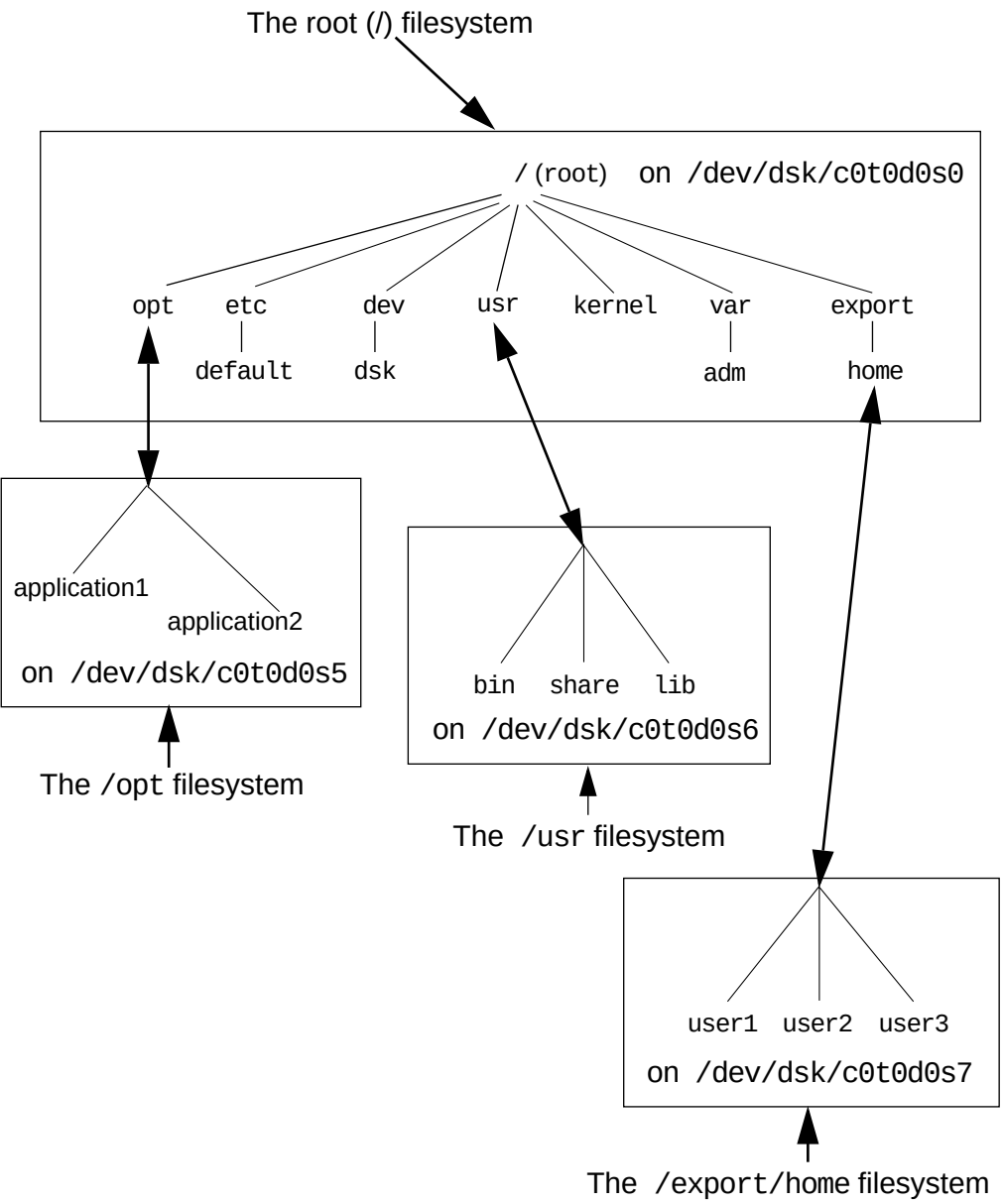
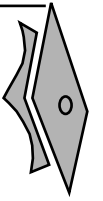
- Define the procedure for mounting different types of file systems
- List the system files used to determine a file system's type
- Unmount local and remote file systems using the commands `umount` and `umountall`
- Forcibly unmount a busy file system
- Describe how to mount and access file systems residing on removable media devices, such as diskettes and CD-ROMs

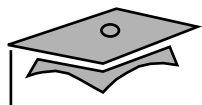


Working with File Systems

When a file system has been created, it must be attached to the Solaris 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.





Identifying Mounted File Systems

To determine which file systems are currently mounted, use the `mount` command.

- The `mount` command maintains the `/etc/mnttab` file.

mount

```
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic/dev=2200000 on Thu Apr 13 17:26:29 2000
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/onerror=panic/dev=2200006 on Thu Apr 13 17:26:30 2000
/proc on /proc read/write/setuid/dev=2d80000 on Thu Apr 13 17:26:28 2000
/dev/fd on fd read/write/setuid/dev=2e80000 on Thu Apr 13 17:26:31 2000
/etc/mnttab on mnttab read/write/setuid/dev=2f4000 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=2200007 on Mon Mar 6 17:58:21 2000
```

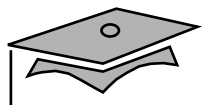
`/export/home on /dev/dsk/c0t0d0s7 /read/write/setuid/largefiles/onerror=panic/dev=2200007 on Mon Mar 6 17:58:21 2000`

↑
mount
point

↑
device
name

↑
mount
options

↑
date and time
mounted



Mounting File Systems

- To mount a local file system manually:

```
# mount /dev/dsk/c0t0d0s7 /export/home
```

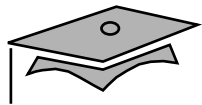
- ▼ The default action mounts a file system with the options: read/write, setuid, no logging, largefiles, and onerror

- To mount a filesystem as read-only:

```
# mount -o ro /dev/dsk/c0t0d0s7 /export/home
```

- To use multiple mount options on the command line:

```
# mount -o ro,nosuid /dev/dsk/c0t0d0s7 /export/home
```

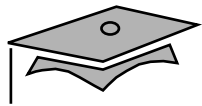


The Virtual File System Table

The `/etc/vfstab` file lists all the file systems that are to be automatically mounted at system boot time.

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	fd	-	no	-
/proc	-	/proc	proc	-	no	-
/dev/dsk/c0t0d0s1	-	-	swapfs	-	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	-



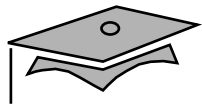
The `/usr/sbin/mountall` Command

- To manually mount all file systems listed in the `/etc/vfstab` file:

```
# mountall
```

- To mount only the local file systems listed in the `/etc/vfstab` file:

```
# mountall -l
```



Unmounting File Systems

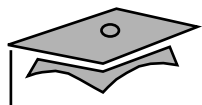
You can use the `/usr/sbin/umount` command to manually unmount a file system.

- To manually unmount a file system using the directory name:

```
# umount /export/home
```

- To manually unmount a file system using the mount point:

```
# umount /dev/dsk/c0t0d0s7
```



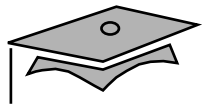
The `/usr/sbin/umountall` Command

- To manually unmount all file systems listed in the `/etc/mnttab` file:

```
# umountall
```

- To unmount only the local file systems listed in the `/etc/mnttab` file:

```
# umountall -l
```



Commands to Unmount a Busy File System

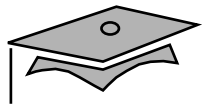
There are two methods available to make a file system available for unmounting if it is busy.

- **fuser** command – To list all the processes accessing the file system, and kill them if necessary:

```
# fuser -cu mount_point  
# fuser -ck mount_point
```

- **umount -f** command – To force the unmount of a file system:

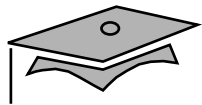
```
# umount -f mount_point
```



Removable Media Device Management

To access file systems on diskettes or CD-ROMS, the Solaris Operating Environment provides a standard interface referred to as Volume Management.

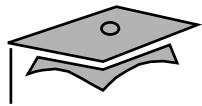
- Volume Management provides three major benefits:
 - ▼ It automatically mounts diskettes and CD-ROMs
 - ▼ It allows access to diskettes and CD-ROMs without having to become `root`
 - ▼ It can give other systems on the network automatic access to any diskettes and CD-ROMs currently inserted in the local system



Accessing Mounted Diskettes and CD-ROMs

To make working with diskettes and CD-ROMs simple for users, each device is mounted in an easy-to-remember location by `vol`.

- For diskettes, `vol` automatically mounts the device after you insert the diskette and run the `volcheck` command.
- For CD-ROMs, `vol` automatically mounts the device when you insert the CD into the drive.



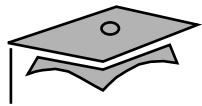
Accessing Mounted Diskettes and CD-ROMs

If `vol` detects that a mounted diskette or CD-ROM contains a file system, the device is mounted at the following locations:

Media Device	Access File Systems On
First Diskette Drive	<code>/floppy/floppy0</code>
First CDROM Drive	<code>/cdrom/cdrom0</code>

If `vol` detects that a mounted diskette or CD-ROM does not contain a file system, the raw device is accessible at the following paths:

Media Device	Access Raw Device On
First Diskette Drive	<code>/vol/dev/aliases/floppy0</code>
First CDROM Drive	<code>/vol/dev/aliases/cdrom0</code>



Administering Volume Management

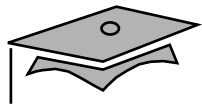
To restrict regular users from accessing diskettes or CD-ROMs on the system, `root` can terminate the volume management service.

- To stop volume management from running on a system temporarily:

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

- To restart the volume management service:

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



Mounting Different Types of File Systems

Mounting a `hsfs` file system type:

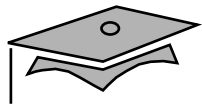
- As `root`, to mount a file system residing on a CD-ROM, when volume management services are stopped:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
```

Mounting a `pcfs` file system type:

- As `root`, to mount a file system residing on a diskette, when volume management services are stopped:

```
# mkdir /pcfs  
# mount -F pcfs /dev/diskette /pcfs
```



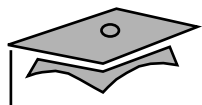
Determining a File System's Type

The `mount` command needs the file system type to be specified.

- It can be explicitly specified for `mount` on the command line, or `mount` can determine it by searching the following files.
 - ▼ `/etc/vfstab` for FS type field
 - ▼ `/etc/default/fs` for local file system type
 - ▼ `/etc/dfs/fstypes` for remote file system type

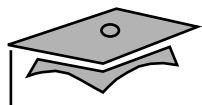
To determine a file system's type for use with `mount`:

```
# grep mount-point fs-table
```



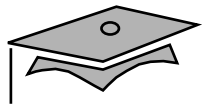
Exercise: Mounting File Systems

- Objectives
- Tasks
- Discussion
- Solutions



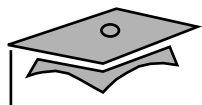
Check Your Progress

- 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



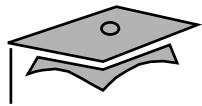
Check Your Progress

- 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



Module 9

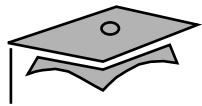
Maintaining File Systems



Objectives

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

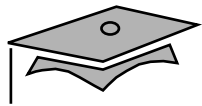
- Describe why `fsck` is necessary
- Describe how to check and repair a filesystem
- Display disk space usage by filesystems
- Display disk usage of a directory
- Display disk usage by user name
- Demonstrate how to repair the `/etc/vfstab` file when the system fails to boot up completely
-



The File System Check Program

File systems can become damaged from a variety of reasons:

- Corrupted from a power failure
- A software error in the kernel, a hardware failure
- Improper shutdown of the system



Data Inconsistencies Checked by fsck

The file system check program, `fsck`, checks for data consistency in file systems and corrects or repairs any inconsistencies or damage found.

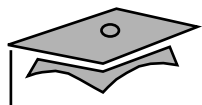
- The lost+found Directory
- Superblock Consistency
- Data Block Consistency
- Cylinder Group Block Consistency



Phases of fsck

fsck runs through five phases for each file system listed in the `/etc/vfstab` file that has a device to fsck and fsck pass entry.

```
# 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)
#
```



Using fsck

- To check a single unmounted file system

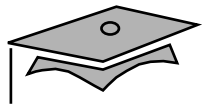
```
# fsck /dev/rdisk/c0t0d0s7
```

- To check a file system using its mount point directory name, as listed in the `/etc/vfstab` file

```
# fsck /opt
```

- To check and repair a 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)
```



Troubleshooting with fsck

- Reconnect an Allocated Unreferenced File

```
** Phase 3 - Check Connectivity
UNREF FILE I=788 OWNER=root MODE=100644
SIZE=19994 MTIME=Jan 18 10:49 1999
RECONNECT? y
```

- Adjust a Link Counter

```
** 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
```

- Salvage the Free List

```
** Phase 5 - Check Cyl groups
CG 0: BAD MAGIC NUMBER
FREE BLK COUNT(S) WRONG IN SUPERBLK
SALVAGE? y
```



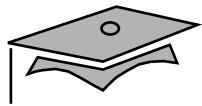
Using Backup Superblocks

If `fsck` fails because of a corrupted superblock it returns an error message indicating that it must be run using an alternative superblock backup to recover the file system.

```
# 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)
#
```

- To list the locations of all the alternative backup superblocks in the file system:

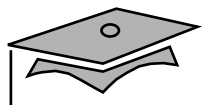
```
# newfs -N /dev/rdisk/c#t#d#s#
```



Monitoring File System Usages

There are four useful commands available for monitoring file system usage:

- `df` — Displays the number of free disk blocks and files
- `du` — Summarizes disk usage
- `ff` — Lists files names and statistics for a file system
- `quot` — Summarizes file system ownership

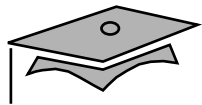


The df Command

The `df` command is used to display the amount of disk space occupied by mounted file systems.

```
# df -k
```

Filesystem	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 du Command

The du command displays the number of disk blocks (512 bytes) used by directories and files.

- To display disk usage in kilobytes:

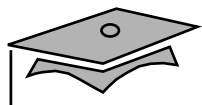
```
# cd /opt ; du -k
8      ./lost+found
11392  ./netscape
```

- To display disk usage including files:

```
# du -ak /usr
16      /usr/lost+found
6       /usr/kvm
723057  /usr
```

- To display a disk usage summary:

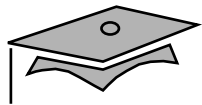
```
# du -sk /usr
723057  /usr
```



The ff Command

The **ff** command provides a list of pathnames and inode numbers of files in the file system.

```
# ff /dev/dsk/c1t3d0s5
/dev/dsk/c1t3d0s5:
inode#  pathname
inode#  pathname
inode#  pathname
inode#  pathname
inode#  pathname
inode#  pathname
inode#  pathname
inode#  pathname
inode#  pathname
inode#  pathname
```



The quot Command

The quot command displays how much disk space (in Kbytes) is being used by users.

- To display disk space being used by users on all mounted file systems:

```
# quot -af
```

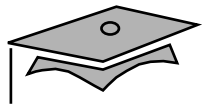
- To display a count of the number of files and space owned by each user for a specific file system:

```
# quot -f /dev/dsk/c1t0d0s5
```



Exercise: Maintaining File Systems

- Objectives
- Tasks
- Discussion
- Solutions



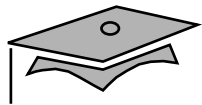
Check Your Progress

- Describe why fsck is necessary
- Describe how to check and repair a filesystem
- Display disk space usage by filesystems
- 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 up completely



Module 10

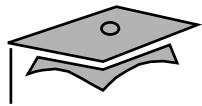
Scheduled Process Control



Objectives

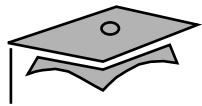
Upon completion of this module, you should be able to:

- 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`



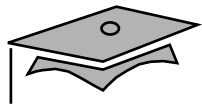
Objectives

- 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



Viewing Processes and PIDs

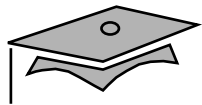
- The `ps` (process status) Command
- The CDE Process Manager
 - ▼ `/usr/dt/bin/sdtprocess &`
- The `prstat` Command



CDE Process Manager

The Solaris CDE environment provides a Process Manager window to monitor and control processes running on the local system.

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/dmi/dmispd
538	dsdm	root	0.0	518	2072	Dec_09	1	/usr/dt/bin/dsdm
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 dtNOIRj_
235	dtlogin	root	0.0	1618	6472	Dec_06	1	/usr/dt/bin/dtlogin -daemon
245	dtlogin	root	0.0	1659	6636	Dec_06	235	/usr/dt/bin/dtlogin -daemon
5848	dtmail	root	0.9	2382	9528	08:20:25	5847	/usr/dt/bin/dtmail
553	dtessio	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 dtOKVJ_z
563	dtterm	root	0.0	1650	6600	Dec_09	553	/usr/dt/bin/dtterm -session dtVenQC_
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	fbconsol	root	0.0	513	2052	Dec_06	1	/usr/openwin/bin/fbconsole -d :0
501	fbconsol	root	0.0	513	2052	Dec_09	491	/usr/openwin/bin/fbconsole

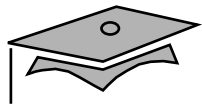


The `prstat` Command

The `prstat` command interactively examines and displays information about active processes on the system.

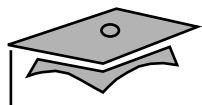
- By default, `prstat` displays information about all processes sorted by CPU usage.

```
# prstat
PID      USERNAME   SIZE    RSS    STATE   PRI    NICE    TIME    CPU    PROCESS/NLWP
191      root       2576K   1216K   cpu0     31      0    0:00.18  0.1%   prstat/1
164      daemon     3120K    552    sleep    59      0    0:00.00  0.0%   statd/4
<output truncated>
Total: 59 processes, 150 lwps, load averages: 0.00, 0.00, 0.01
```

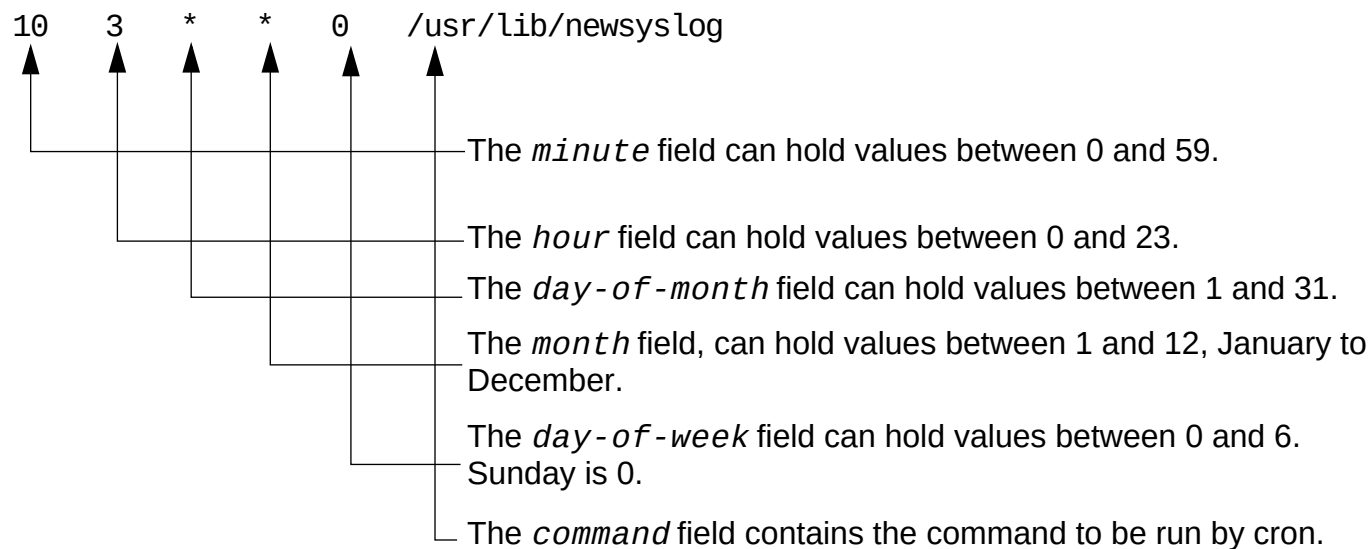


Scheduling the Automatic Execution of Commands

- The `crontab` Command:
 - ▼ Used to view, edit or remove a `crontab` file.
 - ▼ A `crontab` file is used to execute commands or scripts repetitively, at regularly scheduled intervals
- The `at` Command:
 - ▼ Used to execute a job at a specified time once.
- The `cron` daemon is responsible for scheduling and running `crontab` and `at` jobs.



The crontab File



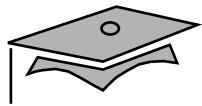


Crontab for the root User

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      */

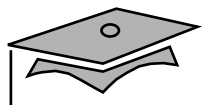
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
```



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`



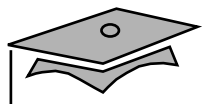
The at Command

- To execute the at command:

```
# 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
```



The at Command

- To display the jobs queued to run at specified times by ranking order:

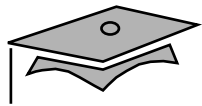
```
# atq
Rank      Execution Date      Owner      Job      Queue      JobName
1st       Apr  2, 2000 14:45    user2     891550468.a    a      stdin
```

- 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
```



Controlling at Access

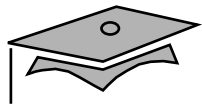
Control access to `at` with two files in the `/etc/cron.d` directory:

- `/etc/cron.d/at.deny`
- `/etc/cron.d/at.allow`



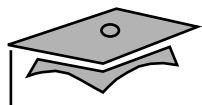
Exercise: Process Control

- Objectives
- Tasks
- Discussion
- Solutions



Check Your Progress

- 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



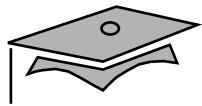
Check Your Progress

- Describe the location and format of a crontab file
- Demonstrate the steps to create, view, edit, and remove a crontab file



Module 11

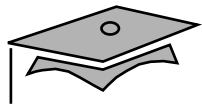
The Solaris Operating Environment LP Print Service



Objectives

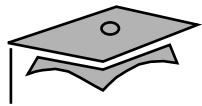
Upon completion of this module, you should be able to:

- 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



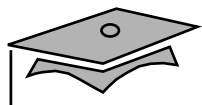
Objectives

- 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`



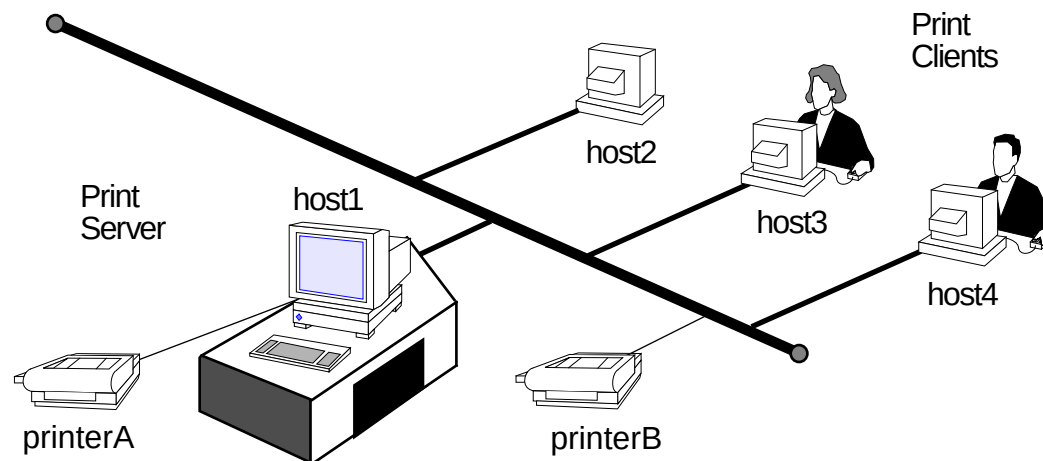
Print Management Tools

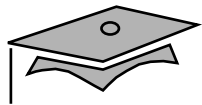
- Solaris Print Manager – A graphical user interface which provides the ability to configure and manage printers.
- Solaris Print Manager is new to Solaris 8 and is preferred over `admintool` as the method for installing and modifying printers and adding access to remote printers.
- `admintool` – A graphical user interface which is used to set up and manage printers on a local system.
- LP print service commands – A command-line interface which is used to configure and manage Mounting Filesystems



Types of Printer Configurations

- Local printer
- Network printer
- Remote printer

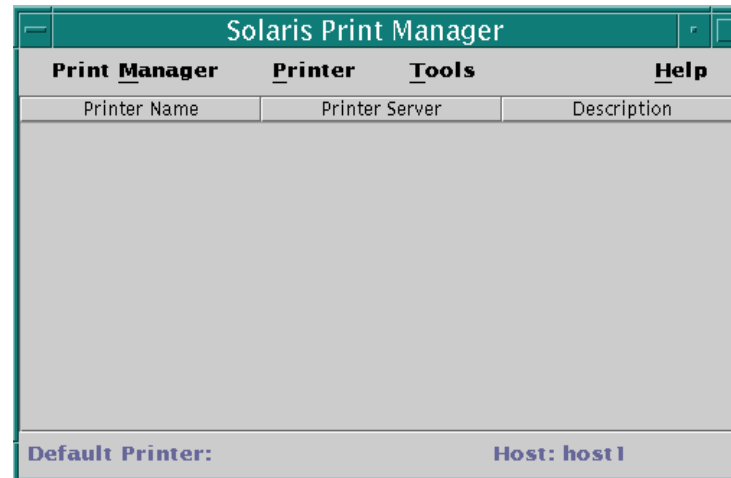




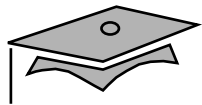
Solaris 8 Print Manager

- To start the Solaris Print Manager:

```
# /usr/sadm/admin/bin/printmgr &
```



- ▼ Add Access to Printer
- ▼ New Attached Printer
- ▼ New Network Printer



Printing the Solaris Operating Environment

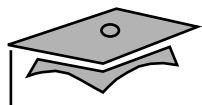
Users submit print requests from print clients using the `lp` or `lpr` command.

- Examples of Using the Print Command:

```
$ /bin/lp filename
```

or

```
$ /usr/ucb/lpr filename
```



Printing the Solaris Operating Environment

The following are examples of specifying the destination printer.

- Submitting a Print Request Atomic Style

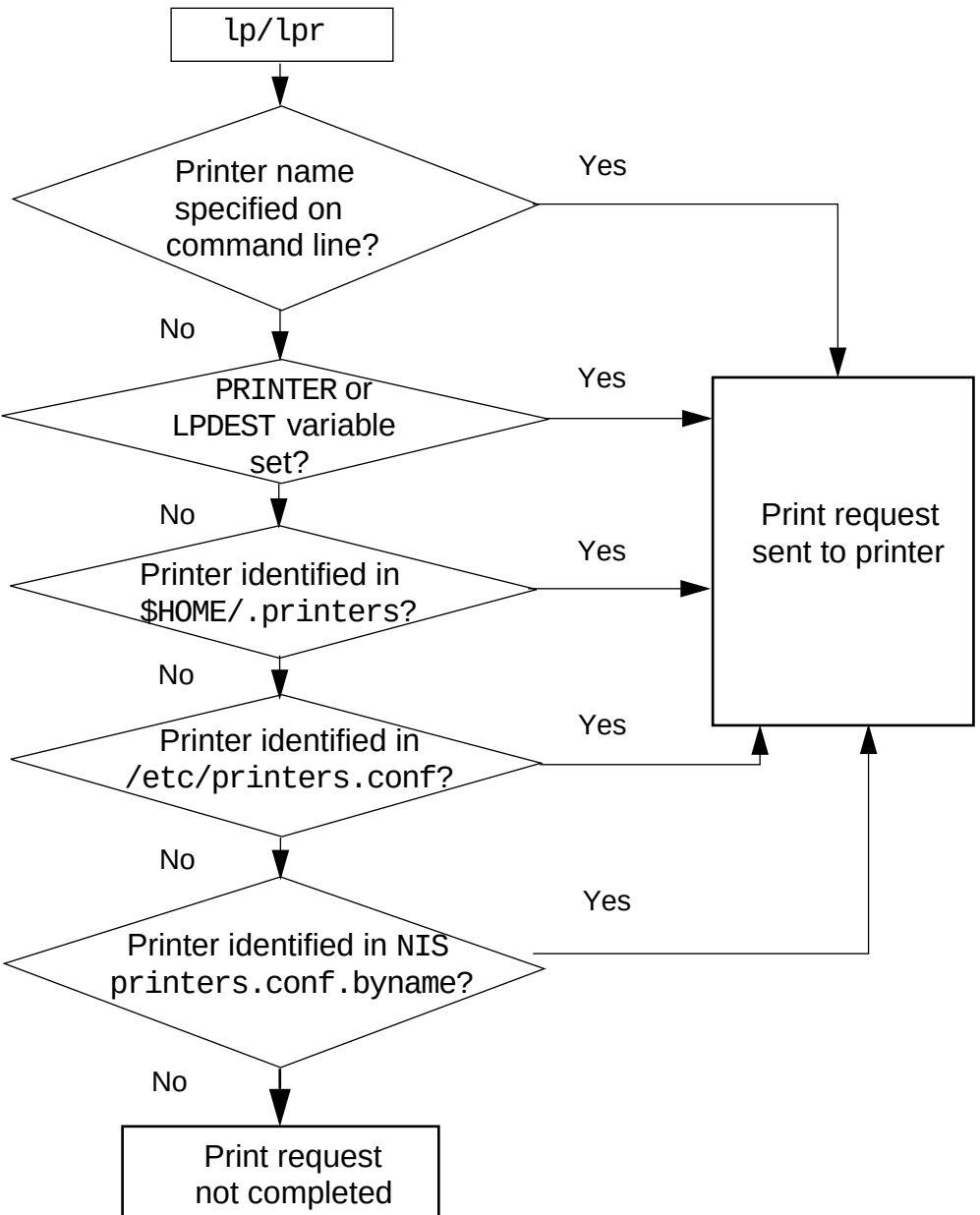
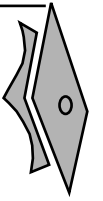
```
$ /bin/lp -d printerB filename
```

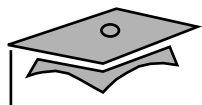
```
$ /usr/ucb/lpr -P printerB filename
```

- Submitting a Print Request POSIX Style

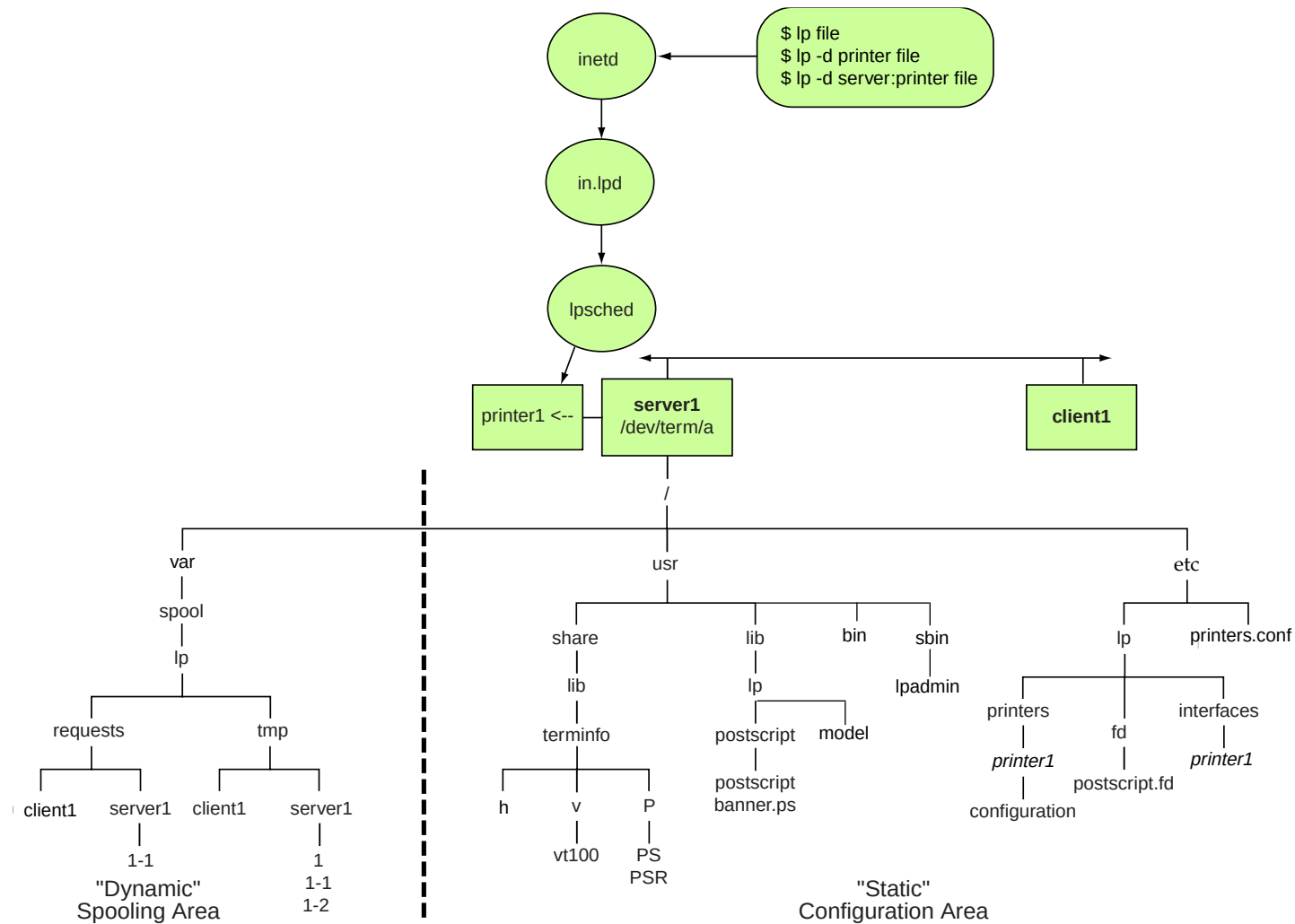
```
$ /bin/lp -d host1:printerA filename
```

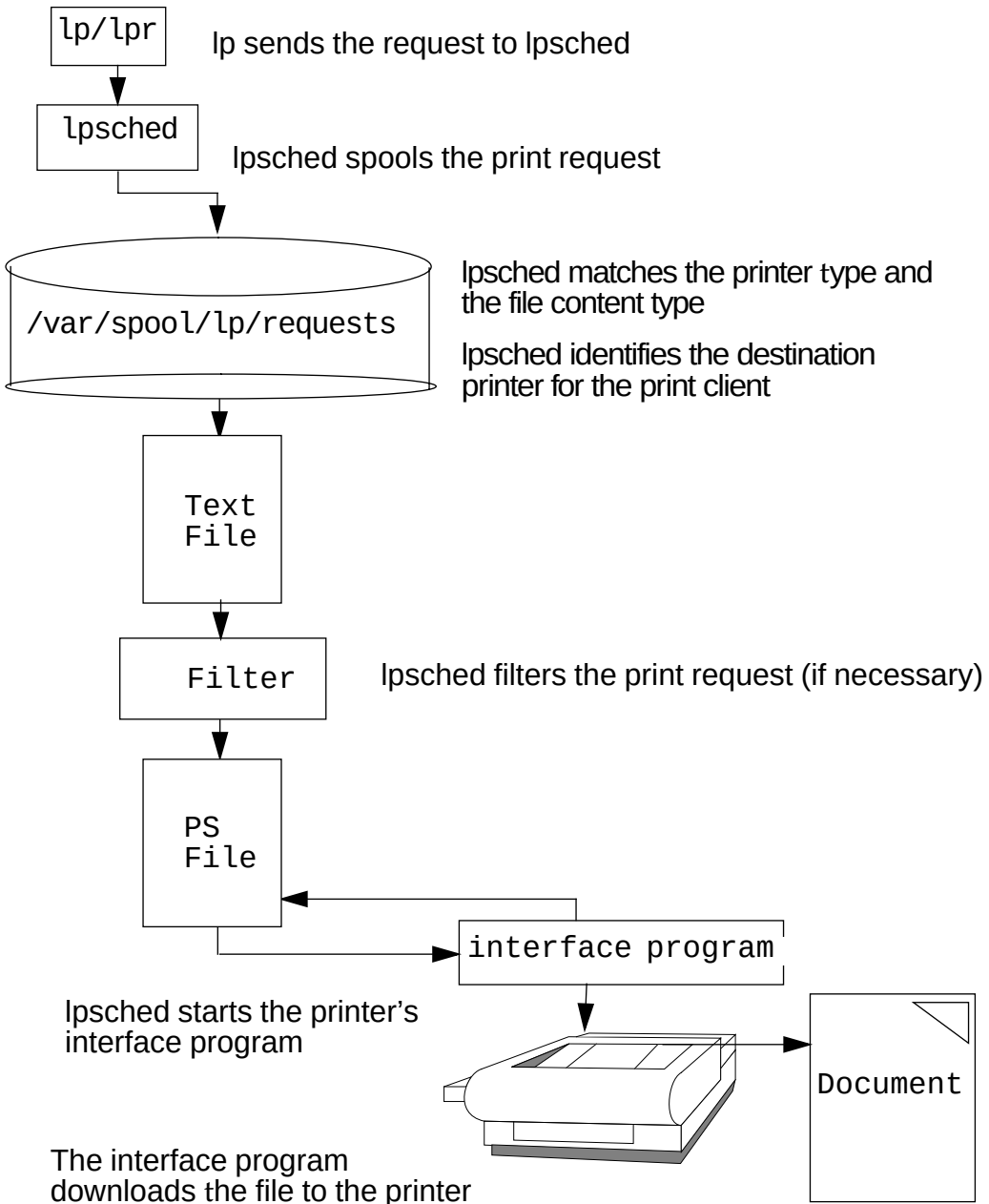
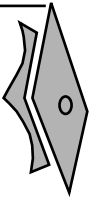
```
$ /usr/ucb/lpr -P host1:printerA filename
```

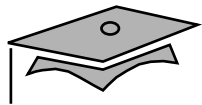




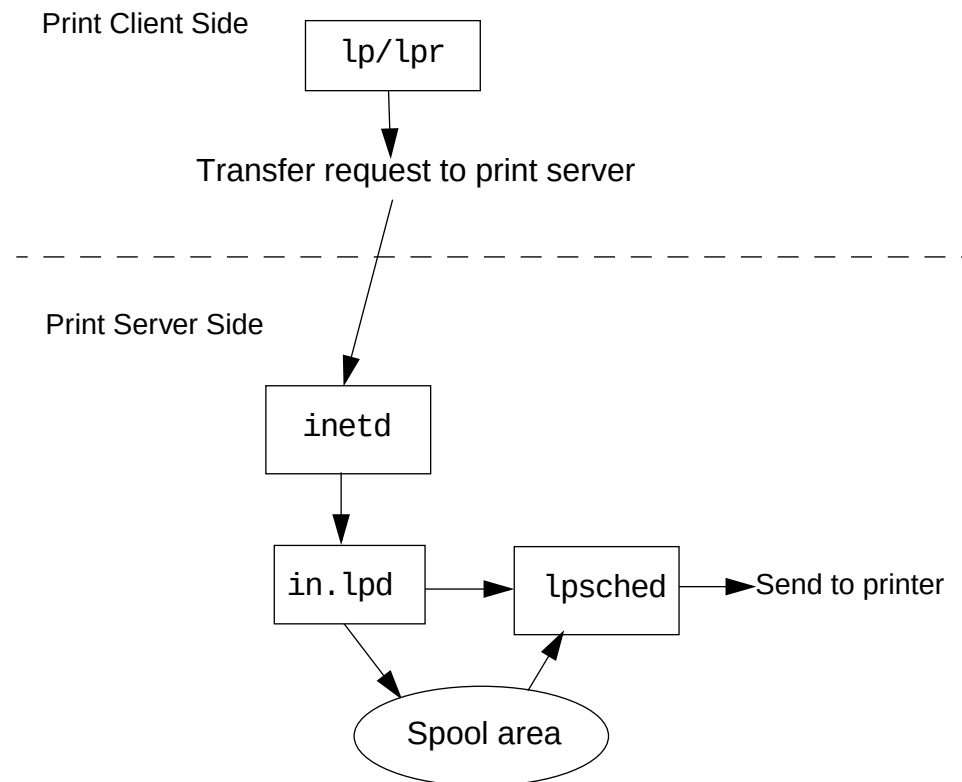
Locating the Destination Printer

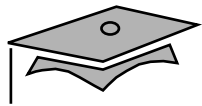






Remote Printing in a Solaris 2.6 to Solaris 8 Environment





The accept and reject Commands

These commands are used by `root` on the print server to permit or prevent print requests be queued on a specific printer.

- Using the `accept` Command to Allow Queuing

```
# accept printer-name
```

- Using the `reject` Command to Prevent Queuing

```
# reject [-r "reason" ] printer-name
```



The enable and disable Commands

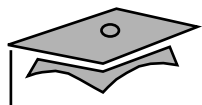
These commands are used by `root` on the print server to activate or to deactivate printers.

- Using the `enable` Command to Activate a Printer

```
# /usr/bin/enable printer-name
```

- Using the `disable` Command to Deactivate a Printer

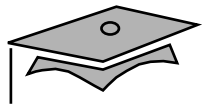
```
# /usr/bin/disable [-c | -w] [-r "reason"] printer-name
```



The `lpmove` Command

Use the `lpmove` command to move one or all print requests, from one printer to another printer.

```
# lpmove source_printer-name target_printer-name
```



Configuring the LP Print Service Using `lpadmin` Command

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 a Printer Class

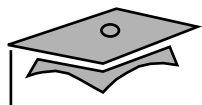
A printer class is created when the first printer is added to the printer class name. After a class is created, other printers can be added to it at any time.

- To create a printer class called `bldg2`:

```
# lpadmin -p printer-name -c class-name
```

- To add another printer to this class:

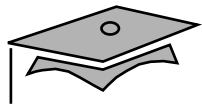
```
# lpadmin -p printer-name -c class-name
```



Setting or Changing a System's Default Printer

The `lpadmin` command can set or change an individual printer or a printer class to be the system's default destination for all print requests.

```
# lpadmin -d printername  
or  
# lpadmin -d printer-classname
```

Setting or Changing a System's Default Printer

- To set or change the default destination printer:

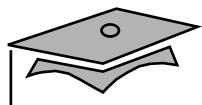
```
# lpadmin -d printerE
# lpstat -d
system default destination: printerE
```

- To set or change the default destination printer class:

```
# lpadmin -d bldg2
# lpstat -d
system default destination: bldg2
```

- To remove the default printer or printer class:

```
# lpadmin -d
```



Halt and Restart the LP Print Service

- To temporarily halt and then restart the LP print service using `lpshut` and `lpsched`:

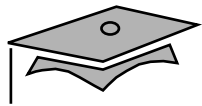
```
# /usr/lib/lpshut  
Print services stopped.
```

```
# /usr/lib/lpsched  
Print services started.
```

- To manually stop or start the LP print services using the print service script `/etc/init.d/lp`:

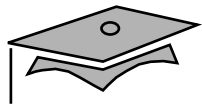
```
# /etc/init.d/lp stop  
Print services stopped.
```

```
# /etc/init.d/lp start  
Print services started.
```



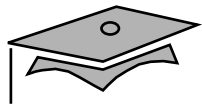
Exercise: LP Print Service

- Objectives
- Tasks
- Discussion
- Solutions



Check Your Progress

- 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



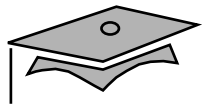
Check Your Progress

- 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`



Module 12

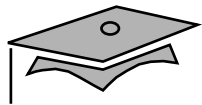
The Boot PROM



Objectives

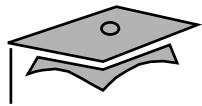
Upon completion of this module, you should be able to:

- 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



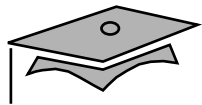
Objectives

- 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

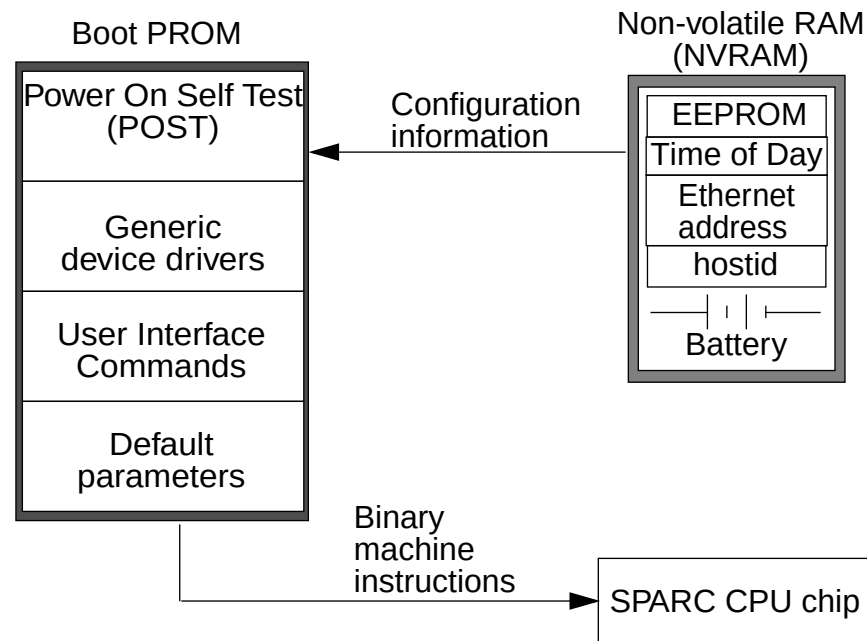


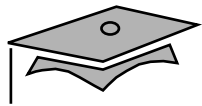
Objectives

- Delete a custom device alias name with the `nvunalias` command.
- Use the `eeeprom` command within Solaris to view or change the values of NVRAM parameters
- Demonstrate the steps to interrupt an unresponsive system



Basic Elements of the Boot PROM and NVRAM

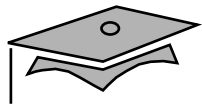




Power On Self Test (POST)

When a system's power is turned on, a low-level power on self test (POST) is initiated.

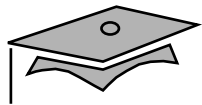
- At the successful completion of the POST phase, the boot PROM firmware takes control and performs the following initialization sequence:
 - ▼ Initialize the system
 - ▼ Probe the memory and then the CPU
 - ▼ Probe bus devices, interpret their drivers, and build a device tree
 - ▼ Install the console



The OpenBoot Goal

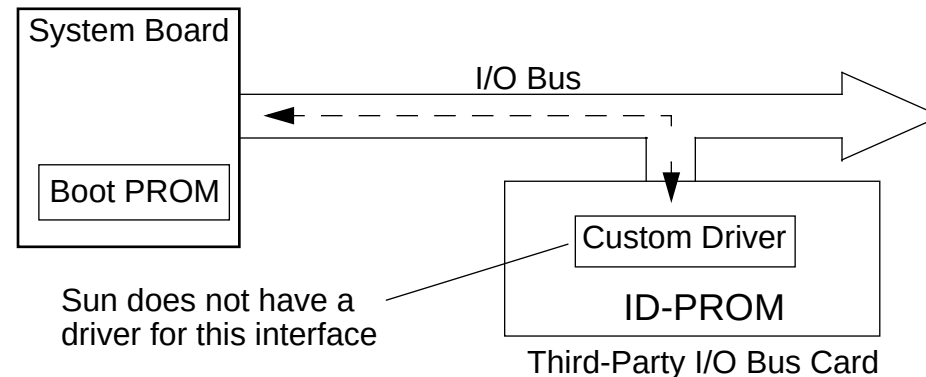
The overall goal of the OpenBoot Institute of Electrical and Electronics Engineers, (IEEE) standard is to provide the capabilities to:

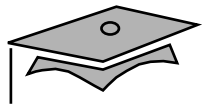
- Test and initialize system hardware
- Determine the systems hardware configuration
- Boot the operating system
- Provide interactive debugging facilities
- Enable the use of third party devices



Third Party Device Configuration

All versions of the OpenBoot architecture allows a third-party board to identify itself and load its own plug-in device driver.

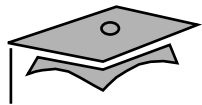




Basic BootPROM Configurations

Systems containing a single system board:

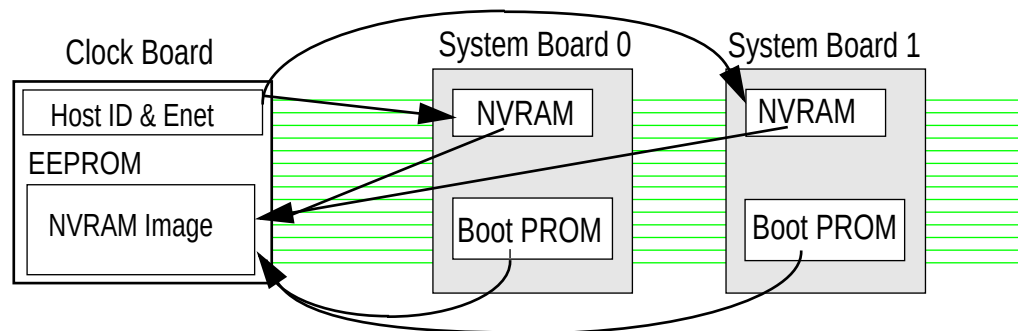
- The following Sun systems are configured with only one system board, which holds both the boot PROM and NVRAM chip.
 - ▼ SPARCstation™ 4, 5, 10, and 20
 - ▼ Ultra™ 1, 2, 5, 10, 30, 60, 80, 220, 250, 420, and 450

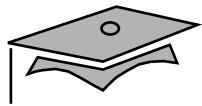


Basic BootPROM Configurations

Systems Containing Multiple System Boards

- The following Suns systems contain multiple System boards, which have a special boot PROM and NVRAM arrangement.
 - ▼ Enterprise 3X00, Enterprise 4X00, Enterprise 5X00, and Enterprise 6X00

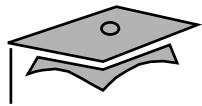




Controlling the POST Phase

The Stop key, located on the left-side of the keyboard, is used to effect the POST phase.

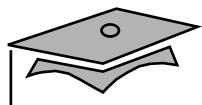
- To skip the POST phase at power up, power on the system while holding down the Stop key.
- To run extensive POST diagnostics during power up using STOP-d
- To reset the NVRAM parameter settings to the default values during power up using STOP-n
- To halt the Solaris Operating Environment to get to the PROM monitor prompt, use Stop-a



Basic Boot PROM Commands

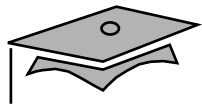
The boot PROM monitor provides a user interface for invoking OpenBoot commands.

- Some commonly used OBP commands:
 - ▼ **ok banner**
 - ▼ **ok boot**
 - ▼ **ok help**
 - ▼ **ok printenv**
 - ▼ **ok setenv**
 - ▼ **ok reset**



Basic Boot PROM Commands

- ▼ **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, (e.g. 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

The `boot` command is used to boot the Solaris Operating Environment from the `ok` prompt.

`ok boot`

- `s` — Boots the system to a single-user mode.

`ok boot -s`

- `a` — Boots the system interactively.

`ok boot -a`

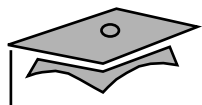
Enter filename of the kernel (kernel/unix):

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:



The boot Command

- **r** — Performs a reconfiguration boot.

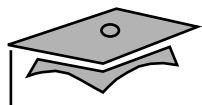
`ok boot -r`

- **V** — Boots the system while displaying more detailed device information to the console.

`ok boot -V`

`ok boot -rV`

`ok boot -sV`



The help Command

Use the `help` command to obtain help on the main categories contained in the OpenBoot firmware.

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

`nvrwrc` (making new commands permanent)

ok



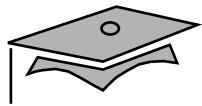
The printenv Command

The `printenv` command can be used to list all the NVRAM parameters.

ok **printenv**

Variable Name	Value	Default Value
tpe-link-test?	true	true
scsi-initiator-id	7	7
keyboard-click?	false	false
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
use-nvramrc?	false	false
diag-switch?	false	false

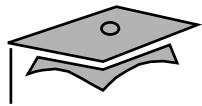
ok



The setenv Command

The `setenv` command is used to change the current values assigned to NVRAM parameters.

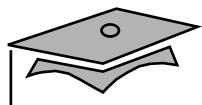
```
ok printenv auto-boot?
auto-boot? = true
ok
ok setenv auto-boot? false
auto-boot? = false
ok reset
Resetting ...
```

The reset Command

The `reset` command halts the system, clears all buffers and registers, and then either:

- 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

The `set-defaults` command is used to reset all parameters to their default values. It only affects 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.

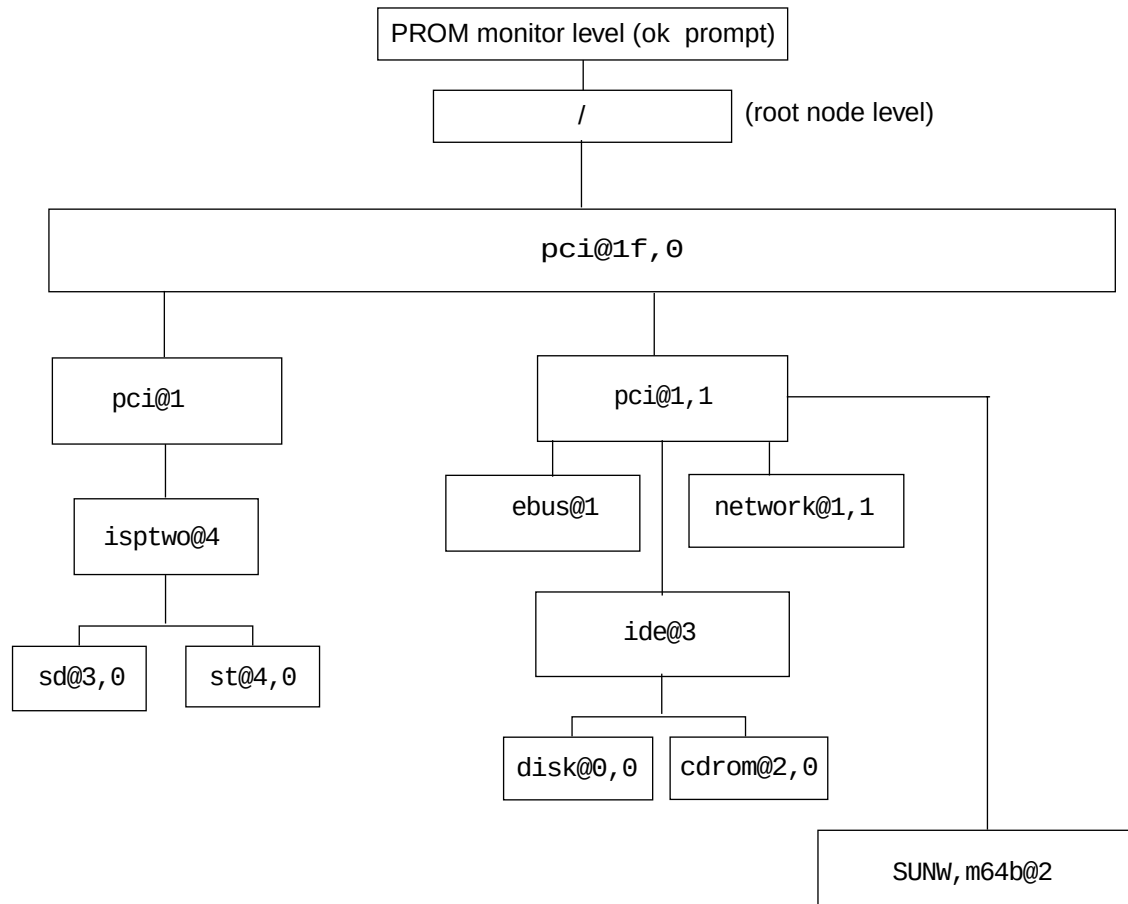
```
ok set-default parameter-name
```

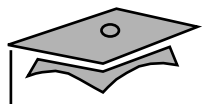


Device Tree



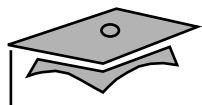
To View Device Path Names





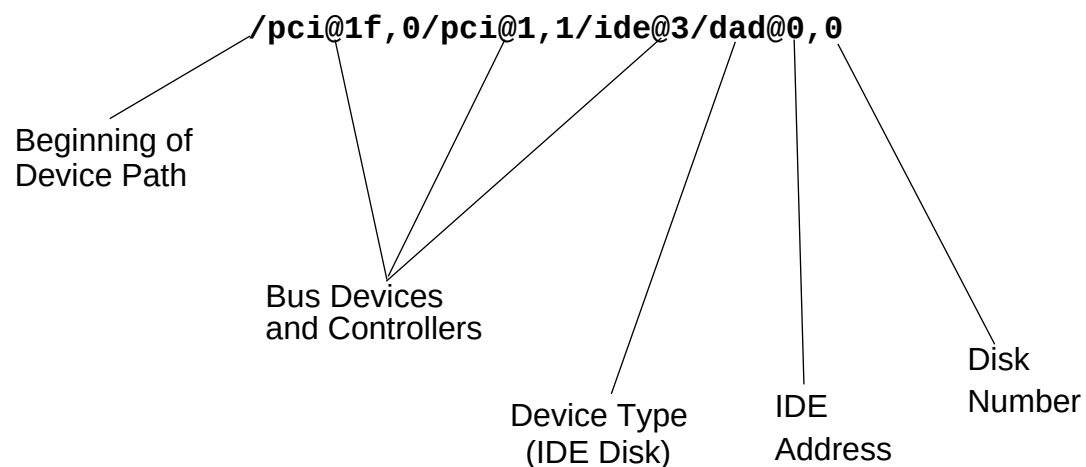
Use the `show-devs` command to view the entire device tree.

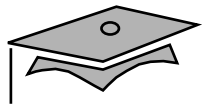
```
ok show-devs
/SUNw,UltraSPARC-IIi@0,0
/pci@1f,0
/virtual-memory
/memory@0,100000000
/pci@1f,0/pci@1
/pci@1f,0/pci@1,1
/pci@1f,0/pci@1/pci@1
/pci@1f,0/pci@1/pci@1/SUNw,isptwo@4
/pci@1f,0/pci@1/pci@1/SUNw,hme@0,1
/pci@1f,0/pci@1/pci@1/SUNw,isptwo@4/st
/pci@1f,0/pci@1/pci@1/SUNw,isptwo@4/sd
/pci@1f,0/pci@1,1/ide@3
/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/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
<output truncated>
```



Boot Disk Device Path Example

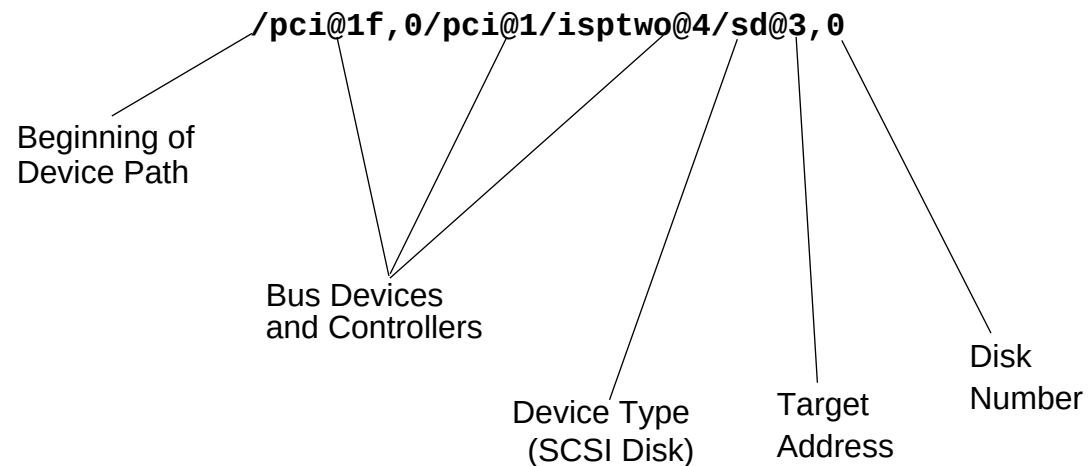
- Disk Device Path on an Ultra System With PCI Bus

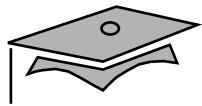




Boot Disk Device Path Example

- Disk Device Path on an Ultra System With PCI-SCSI Bus

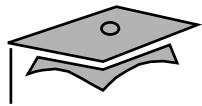




Using probe- Commands to Identify Devices

To identify the peripheral devices, such as disks, tape drives or CD-ROMs currently connected to the system, use the OBP commands:

- `probe-ide`
- `probe-scsi`
- `probe-scsi-all`



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:

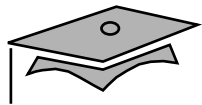
```
ok probe-scsi
```

```
Target 3
```

```
Unit 0    Disk SEAGATE ST1480 SUN0424626600190016
```

```
Target 6
```

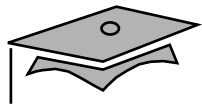
```
Unit 0    Removable Read Only device SONY CDR0M
```



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,isptwo@4  
Target 3  
Unit 0    Disk FUJITSU MAB3045S SUN4.2G1907  
Target 4  
Unit 0    Removable Tape EXABYTE EXB-8505SMBANSH20090
```



The probe-ide Command

The probe-ide command identifies the peripheral devices, currently only disks and CDRoms, attached to the on-board ide controller.

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

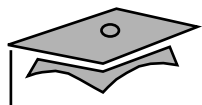
Device 3      ( Secondary Slave )
              Not Present
```



Identifying the Boot Device

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
```



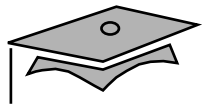
Creating Custom Device Aliases

To create a new device alias name to access the newly attached external device, use the command `nvalias`.

```
ok nvalias alias-name device-path
```

- Using `nvalias` to create custom device aliases

```
ok show-disks  
(select a disk from the list)  
ok nvalias mydisk /pci@1f,0/pci@1/pci@1/  
SUNW,isptwo@4/sd  
ok setenv boot-device mydisk  
boot-device = mydisk  
ok boot
```



Removing a Custom Device Aliases

To remove a custom device alias name, use the command `nvunalias`.

`ok nvunalias alias-name`

- Removing custom device aliases:

```
ok nvunalias mydisk
ok setenv boot-device disk
boot-device =          disk
ok reset
Resetting ...
```

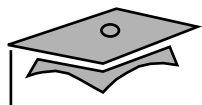
The `nvedit` Command

On Sun systems with PROM versions 1.x and 2.x, the `nvalias` command may not be available to create custom device alias names.



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
```



The eeprom Command

- To list all parameters with default and current values:

```
# eeprom
```

- To list a single parameter and its value:

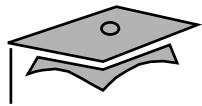
```
# eeprom boot-device  
boot-device=disk
```

- To change the value of the default boot device:

```
# eeprom boot-device=disk2
```

- To change the value of the auto-boot? parameter:

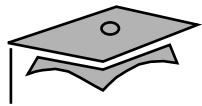
```
# eeprom auto-boot?=true  
auto-boot?=true
```

Interrupting an Unresponsive System

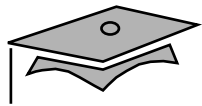
When a system freezes, or stops responding to the keyboard, it will need to be interrupted.

- To interrupt an unresponsive system:
 - ▼ Attempt a remote login on the unresponsive system to locate and kill the offending process.
 - ▼ Attempt to reboot the unresponsive system gracefully.
 - ▼ Hold down the Stop-a key sequence on the keyboard of the unresponsive system. The system is placed at the ok prompt. Synchronize the file systems using the sync command.



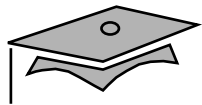
Exercise: OpenBoot Prom

- Objectives
- Tasks
- Discussion
- Solutions



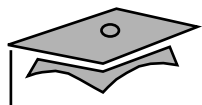
Check Your Progress

- 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 will boot
- 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



Check Your Progress

- 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 System to view or change the values of NVRAM parameters
- Demonstrate the steps to interrupt an unresponsive system



Module 13

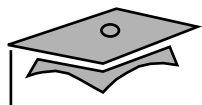
System Boot Process



Objectives

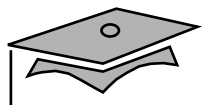
Upon completion of this module, you should be able to:

- 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



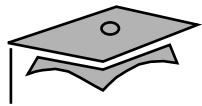
Objectives

- 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`



Solaris Run Levels

Run Level	Function
0	Shuts down the Solaris Operating Environment and displays the boot PROM ok prompt so that 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	To run in a single-user administrative state with access to all available file systems.
2	For multi-user operations. Multiple users can access the system. All system daemons are running except for the NFS server daemons.
3	For 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).

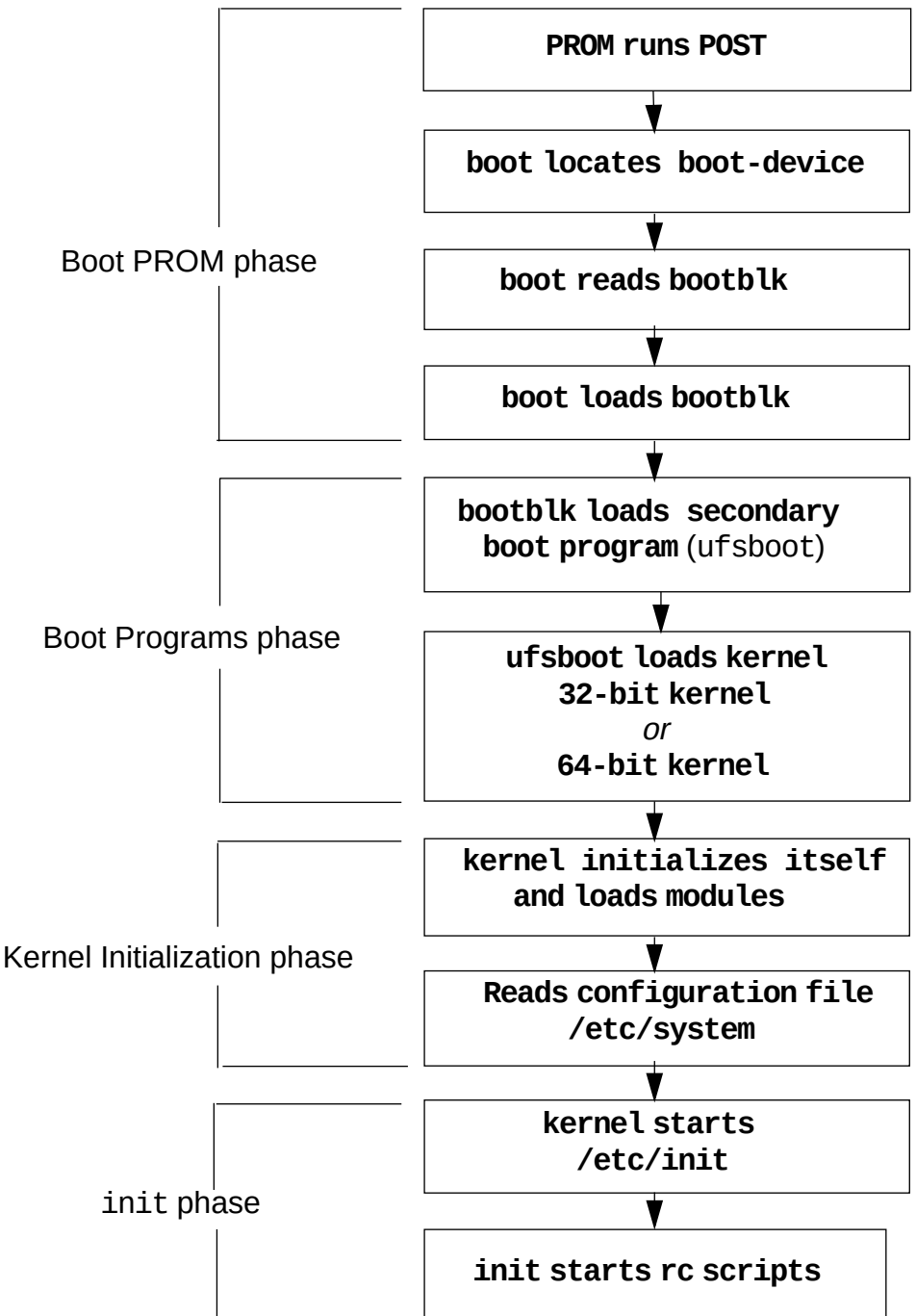
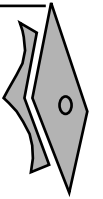


Determining the Current Run Level

To determine the run level a system has been brought to:

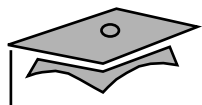
```
# who -r
```

```
      .  run level 3    Jun 9  08:30    3    0    S
      ^               ^               ^      ^      ^
      |               |               |      |      |
Current run level   Date and Time   Current  Number of times  Previous
                   of last run level run level  at this run level  run level
                   change                                     since last reboot
```





The Virtual Filesystem Table



Kernel and Modules Loaded in Memory

32-bit Kernel

```
/platform/'uname -m'/kernel/genunix  
/platform/'uname -m'/kernel/unix
```

64-bit Kernel

```
/platform/'uname -m'/kernel/sparcv9/genunix  
/platform/'uname -m'/kernel/sparcv9/unix
```

Module directories

```
/kernel  
/usr/kernel  
/platform/'uname -m'/kernel  
/platform/'uname -i'/kernel
```

MEMORY

Static Core

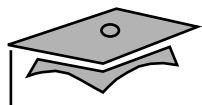
genunix
unix

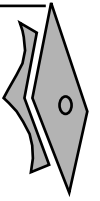
Device Driver Modules

Streams Modules

FS Modules

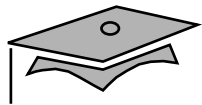
Sched Modules





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@0,800000/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).
```



The `/etc/inittab` File

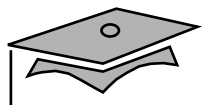
The `inittab` file defines three items for the `init` process:

- 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 four fields:

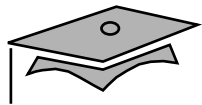
id	A 1 to 4 character identifier for the entry.
rstate	One or more run levels to which this entry applies.
action	How the process (in the next field) is to be treated.
process	The command or script to execute.

s3:3:wait:/sbin/rc3	> /dev/msglog 2<> /dev/msglog </dev/console



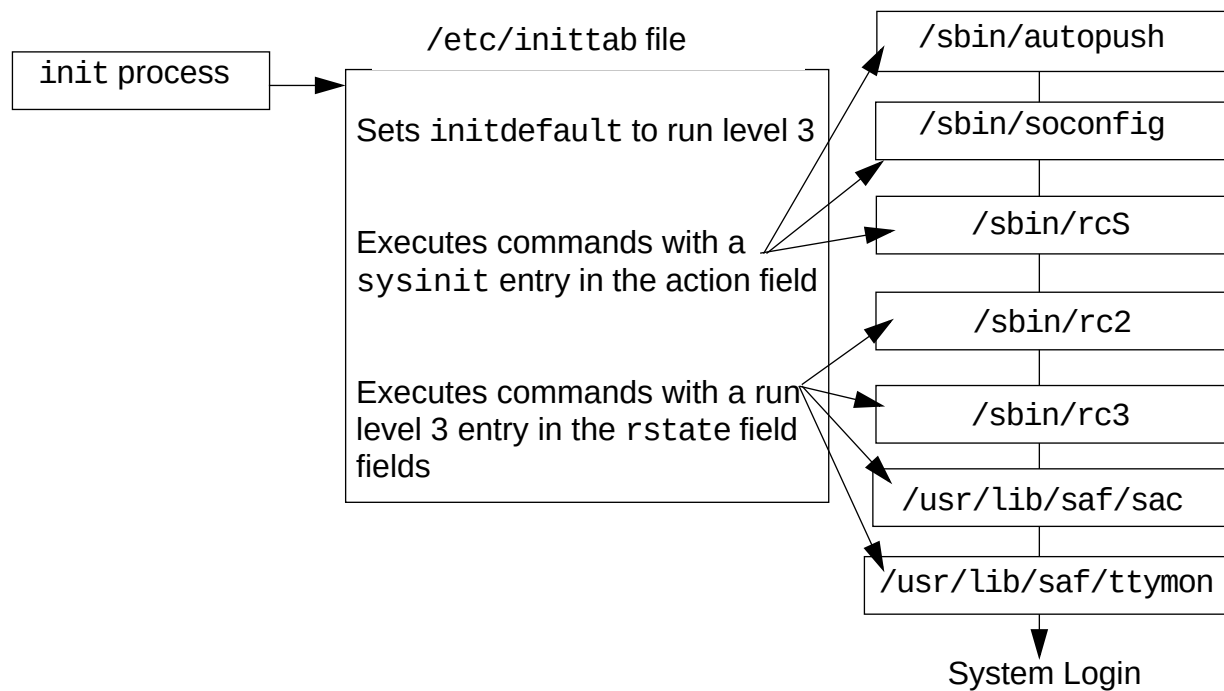
The /etc/inittab File

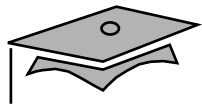
```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
ap::sysinit:/sbin/soconfig -f /etc/sock2path
fs::sysinit:/sbin/rcS sysinit >/dev/msglog 2<>/dev/msglog </dev/console
is:3:initdefault:
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/msglog
sS:s:wait:/sbin/rcS >/dev/msglog 2<>/dev/msglog </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 `init` Process

The following illustrates the process of bringing a system to the default run level 3.

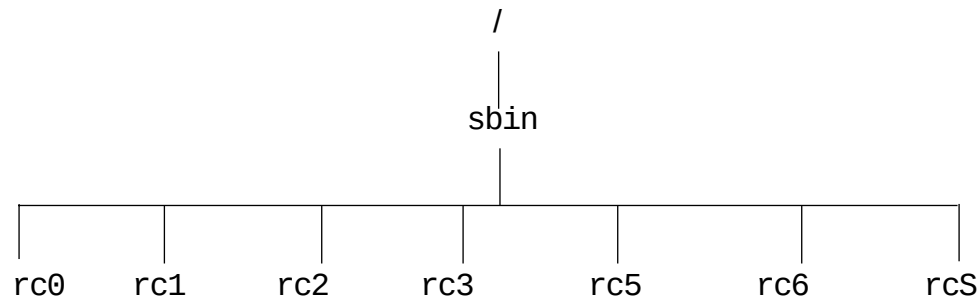




Run Control Scripts

Solaris provides a series of run control (rc) scripts to stop and start processes normally associated with run levels.

- Each run level has an associated rc script located in the /sbin directory.



```
# cd /sbin  
# ls -i rc*
```

```
47154 rc0  
47155 rc1
```

```
47156 rc2  
47157 rc3
```

```
47154 rc5  
47154 rc6
```

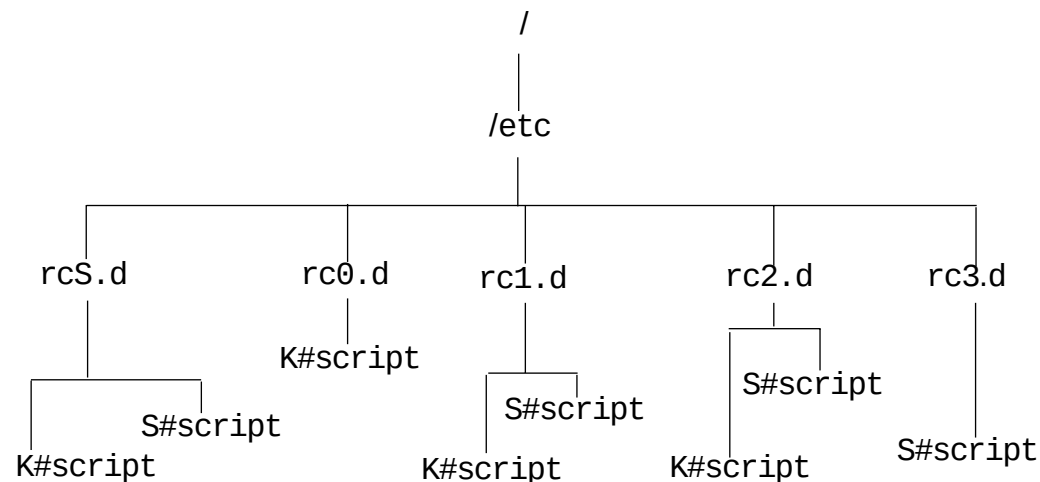
```
47158 rcS
```

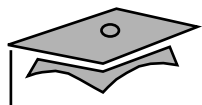


The /etc/rc#.d Directories

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.

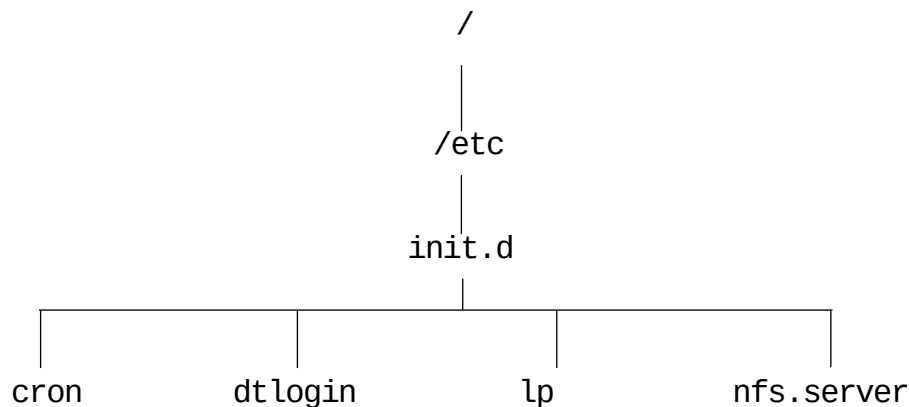




The `/etc/init.d` Directory

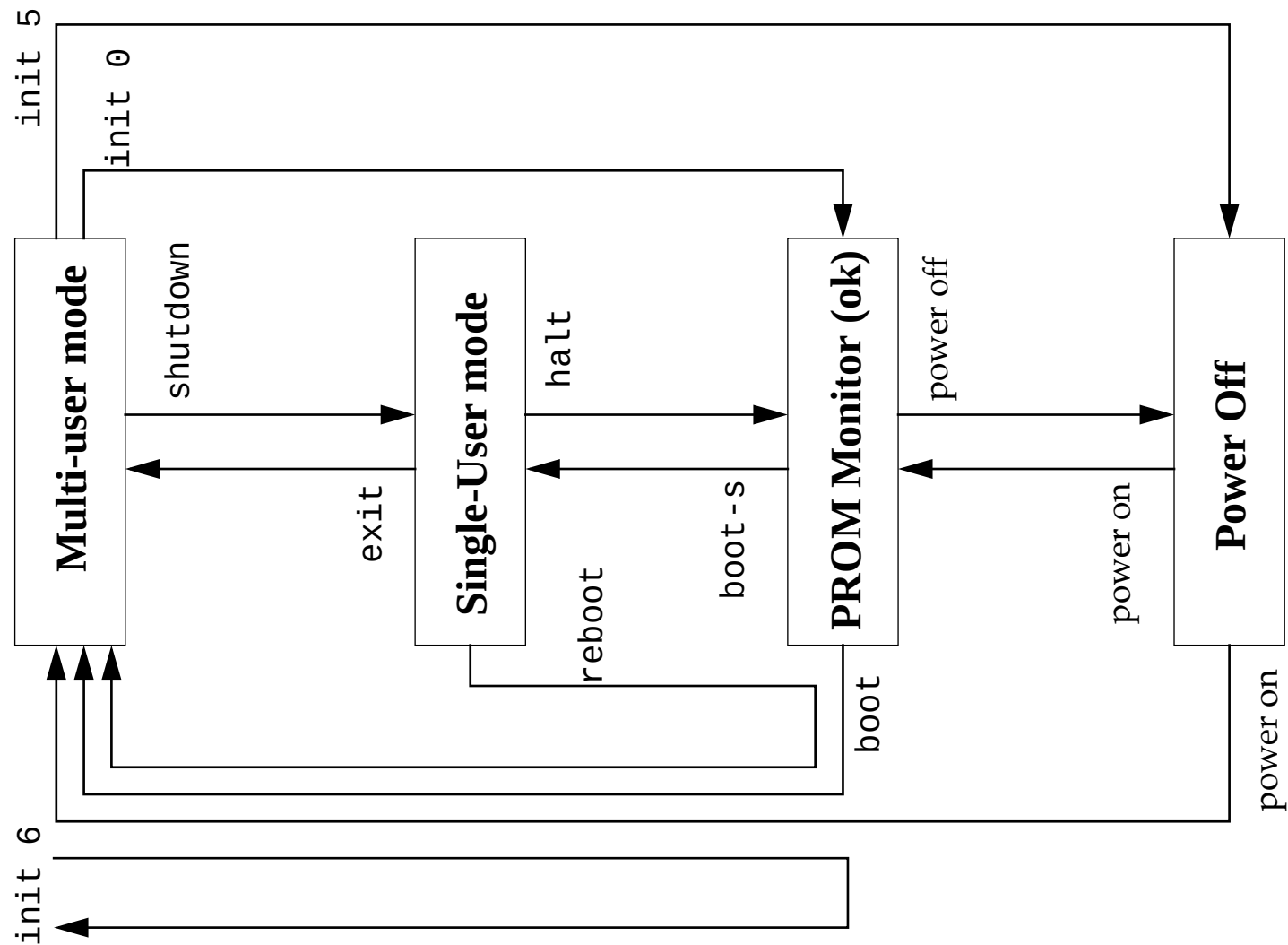
Run control scripts are located in the `/etc/init.d` directory.

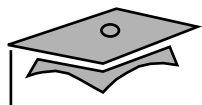
- These files are hard linked to corresponding run control scripts in the `/etc/rc#.d` directories.





Accessing Mounted Diskettes and

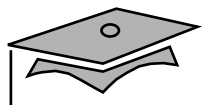




System Shutdown Procedures

The commands for system shutdown procedures include:

- `/sbin/init` (run levels S, 1, 0, 5 or 6)
- `/usr/sbin/shutdown` (run levels S, 1, 0, 5 or 6)
- `/usr/sbin/halt`
- `/usr/sbin/poweroff`
- `/usr/sbin/reboot`



The `/sbin/init` Command

The `init` command can be used 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 shutdown the system to single-user mode, use either run level `S` or `1`, for example:

```
# init S
```

To shutdown the system to stop Solaris and display the `ok` prompt:

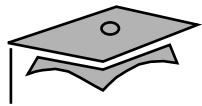
```
# init 0
```

To shutdown the system and turn its power off:

```
# init 5
```

To shutdown the system and then reboot to multi-user mode:

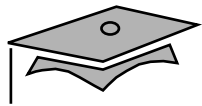
```
# init 6
```



System Shutdown Commands

The /usr/sbin/shutdown Command

- To shutdown the system to single-user mode:
shutdown -iS
- To shutdown the system and display the ok prompt:
shutdown -i0
- To shutdown the system and power off:
shutdown -i5
- To shutdown and reboot to multi-user mode:
shutdown -i6



The `/usr/sbin/halt` Command

- The `halt` command performs an immediate shutdown:

```
# halt
```

The `/usr/sbin/poweroff` Command

- To shutdown the system and turn off its power:

```
# poweroff
```

The `/usr/sbin/reboot` Command

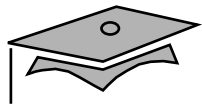
- To shutdown the system and then reboot to multi-user mode:

```
# reboot
```



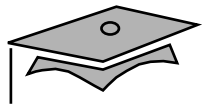
Exercise: The Boot Process

- Objectives
- Tasks
- Discussion
- Solutions



Check Your Progress

- 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 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



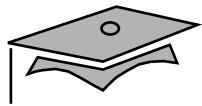
Check Your Progress

- 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 shutdown the system: `init`, `shutdown`, `halt`, `poweroff`, and `reboot`



Module 14

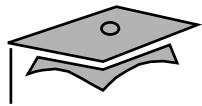
Installing the Solaris 8 Operating Environment on a Standalone System



Objectives

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

- State the different installation methods available for the Solaris 8 software
- Explain the hardware requirements for a Solaris 8 installation
- Identify the different Solaris 8 software CDROM editions
- List the five Solaris Software Groups
- Demonstrate how to install the Solaris 8 Software on a networked, standalone system, using Solaris Web Start



Solaris Software Installation Options

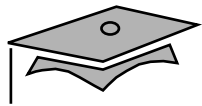
- Solaris Web Start 3.0 Installation
- Solaris Interactive Installation Program
- Solaris Installation Over the Network
- Solaris JumpStart Installation
- Solaris Custom JumpStart Installation



Hardware Requirements of a Solaris 8 Installation

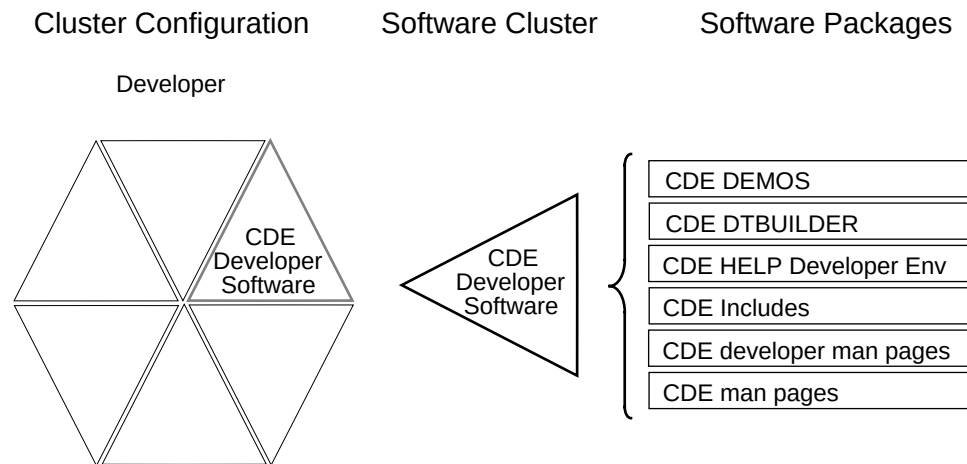
A desktop Solaris 8 installation requires:

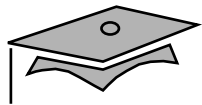
- A SPARC-based or an Intel-based system
- 64 Mbytes of memory
- 2.3 Gbytes of disk space
- Access to a CDROM drive



Solaris Software Arrangement

- Software Packages
- Software Clusters
- Cluster Configurations

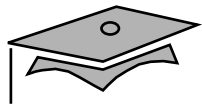




Cluster Configurations

The *cluster configurations* are referred to during the installation process as the Solaris Software Groups. There are currently five software groups available, which include:

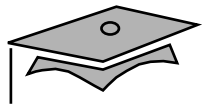
- Entire Solaris Software Group Plus OEM - SUNWCXall
- Entire Solaris Software Group - SUNWCa11
- Developer Solaris Software Group - SUNWCprog
- End User Solaris Software Group - SUNWCuser
- Core Solaris Software Group - SUNWCreq



Pre-Installation Information

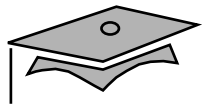
Prior to the Solaris software installation on a networked standalone system, the following information is required.

- Host name
- Host IP address
- Name service type
- Subnet mask
- Geographic location and Time zone
- Root password
- Language

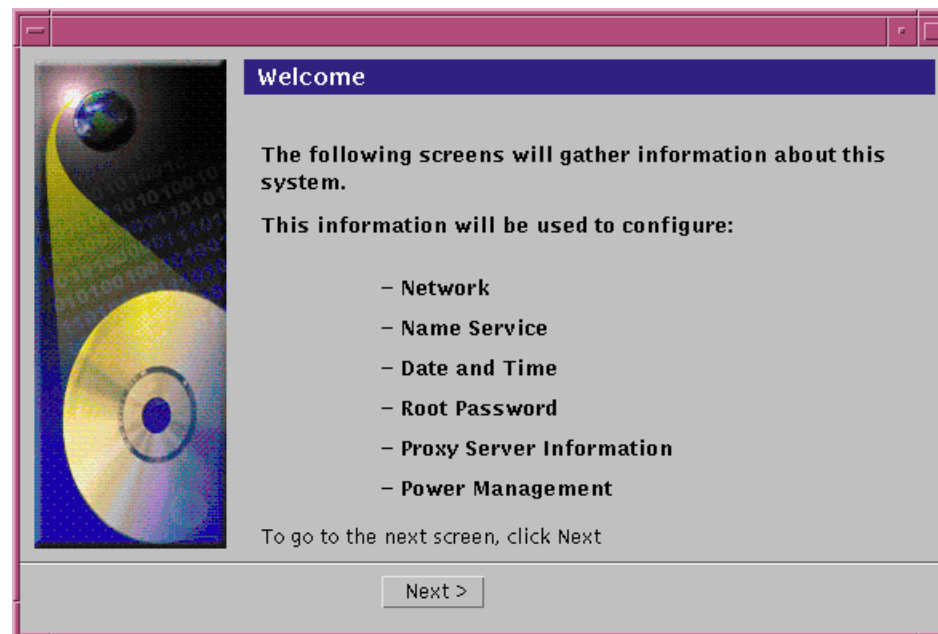


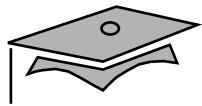
Software Installation Using Solaris Web Start



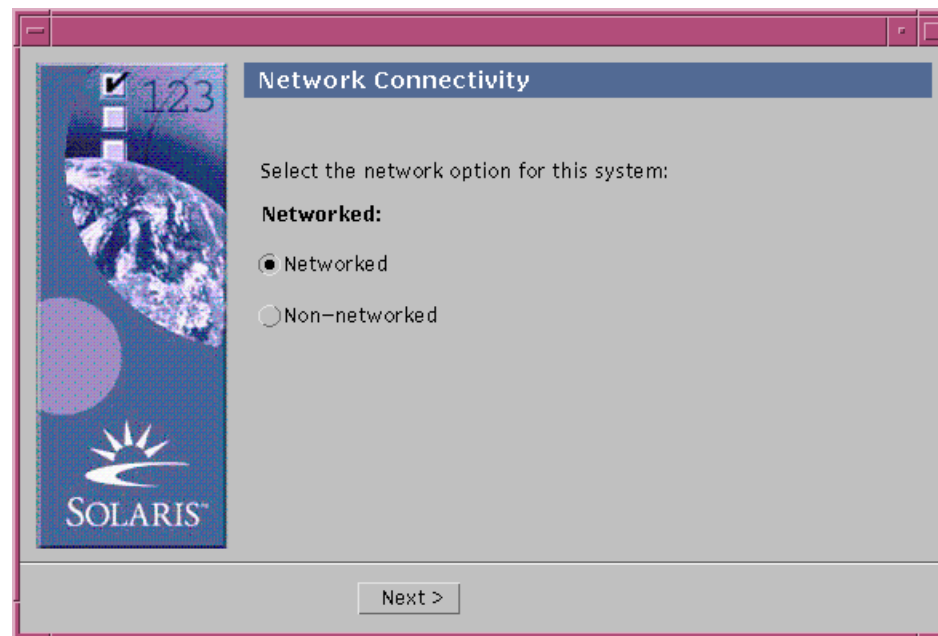


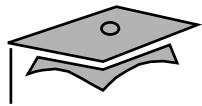
Preconfiguration Checklist





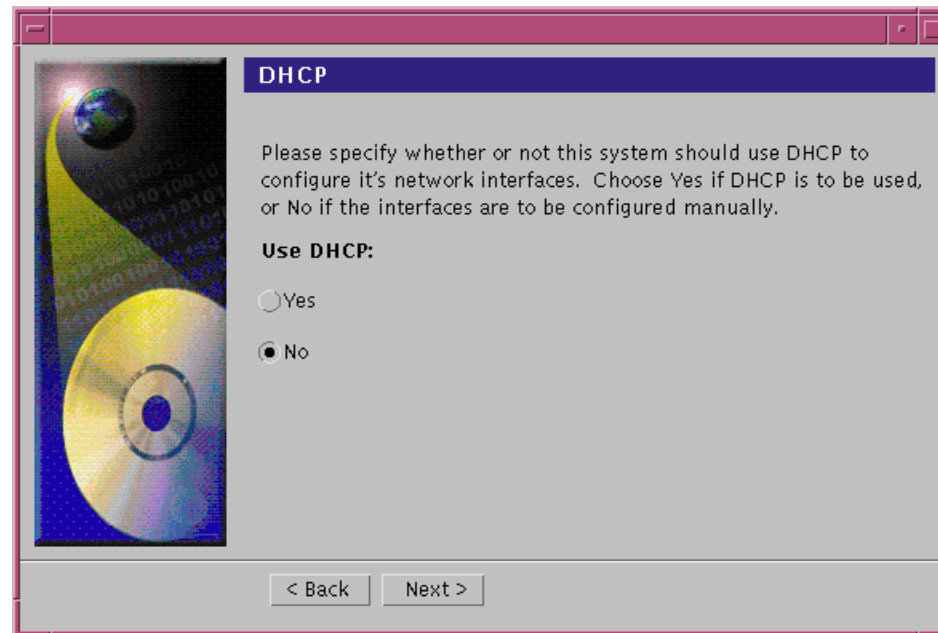
Network Connectivity

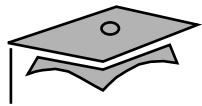




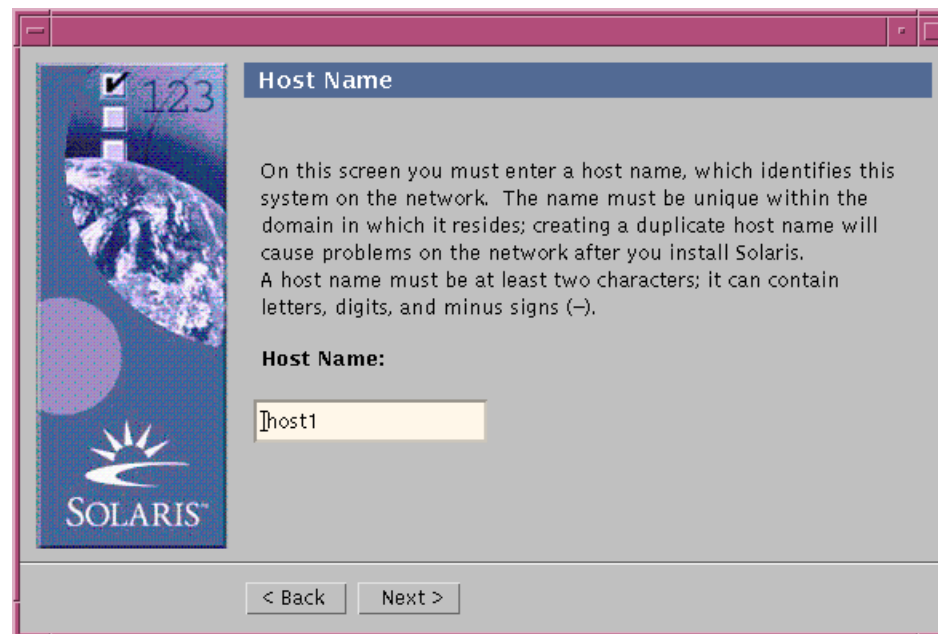
DHCP

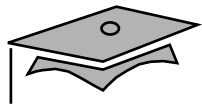
Dynamic Host Configuration Protocol



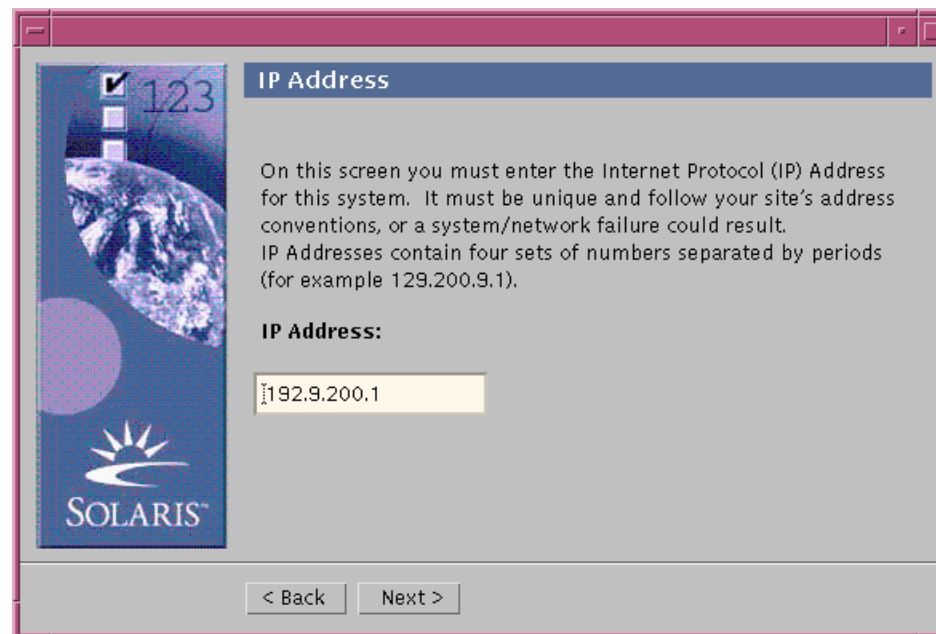


Host Name



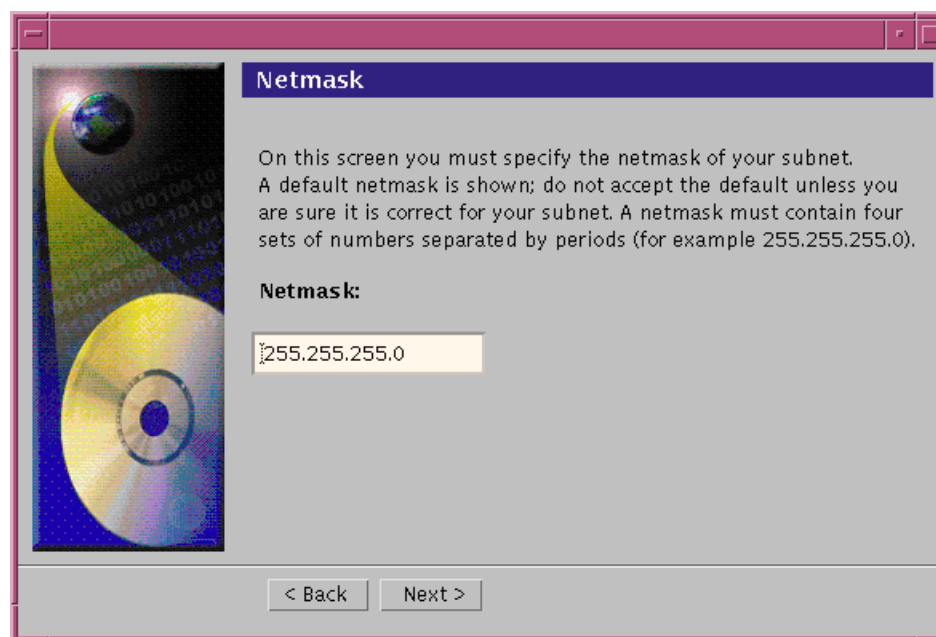


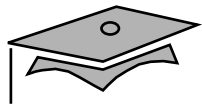
IP Address



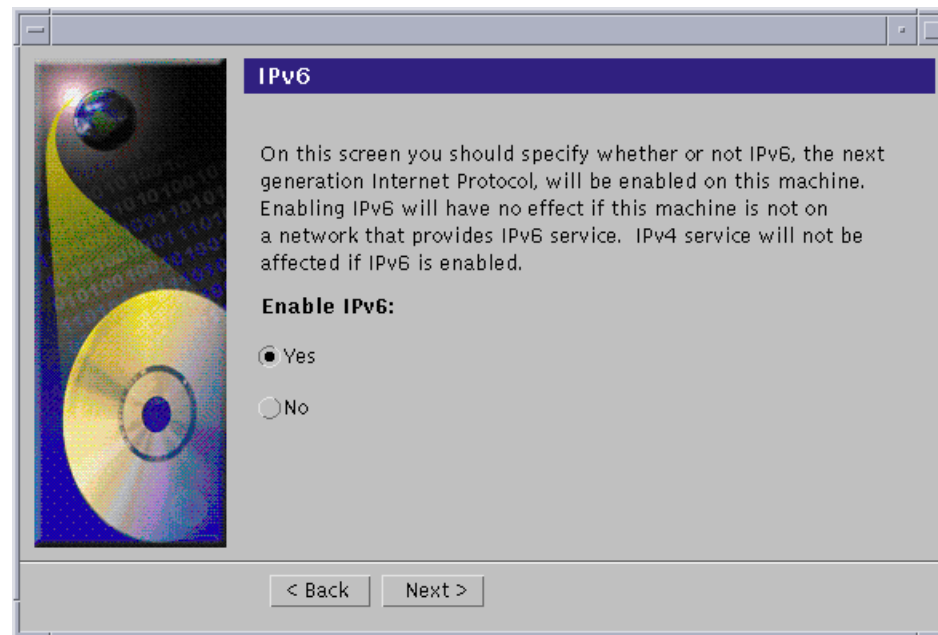


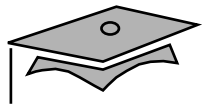
Netmask



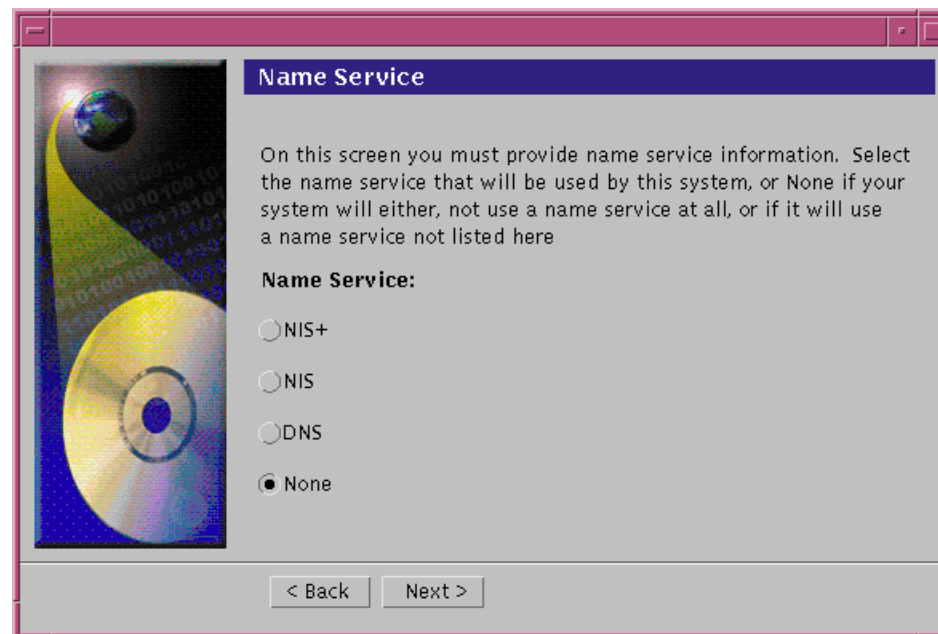


IPv6



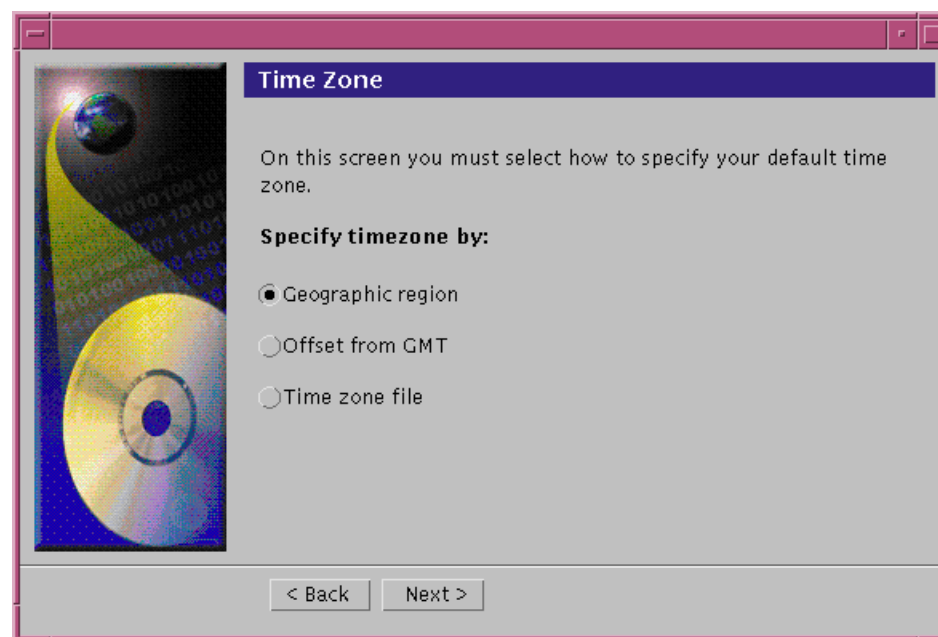


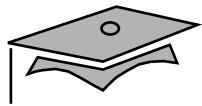
Name Service



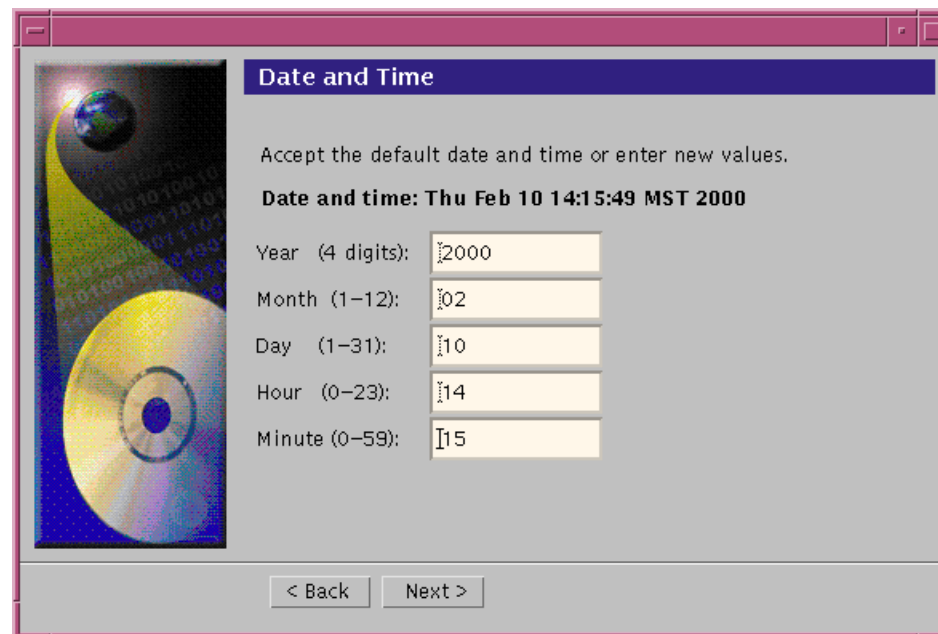


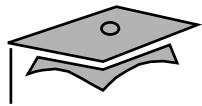
Time Zone



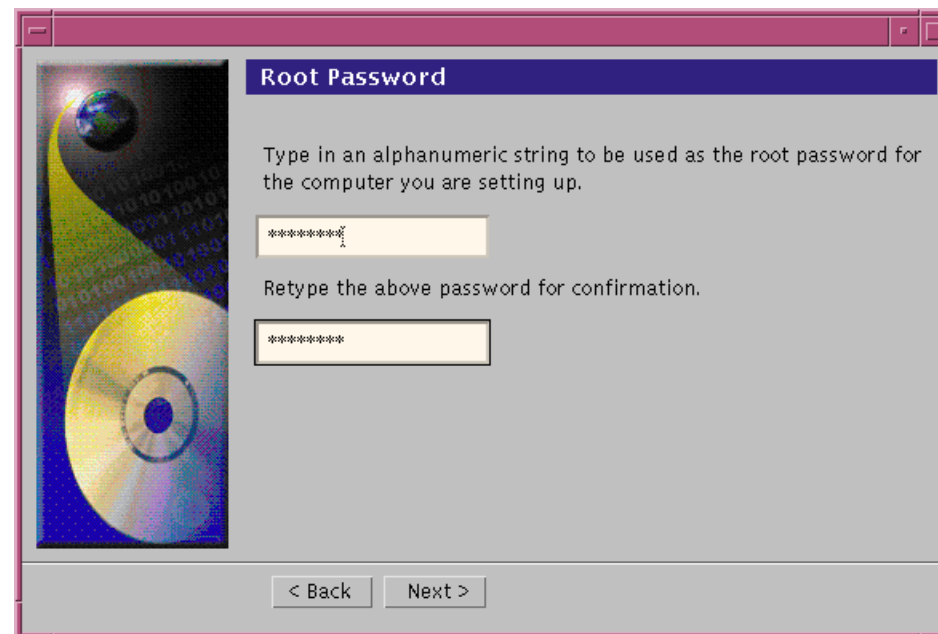


Date and Time





Root Password



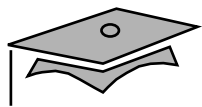
The image shows a window titled "Root Password" from the Solaris 8 installation process. On the left is a vertical graphic with a globe and a CD. The main area contains instructions to enter an alphanumeric root password. There are two input fields, both masked with asterisks. The first field is for the password, and the second is for confirmation. At the bottom are "< Back" and "Next >" buttons.

Root Password

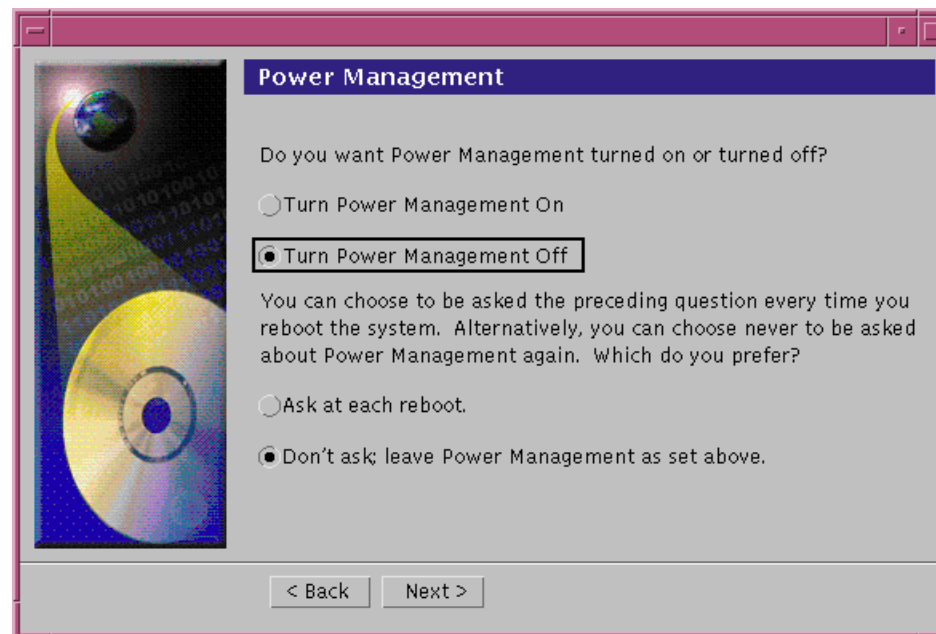
Type in an alphanumeric string to be used as the root password for the computer you are setting up.

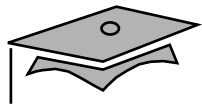
Retype the above password for confirmation.

< Back Next >

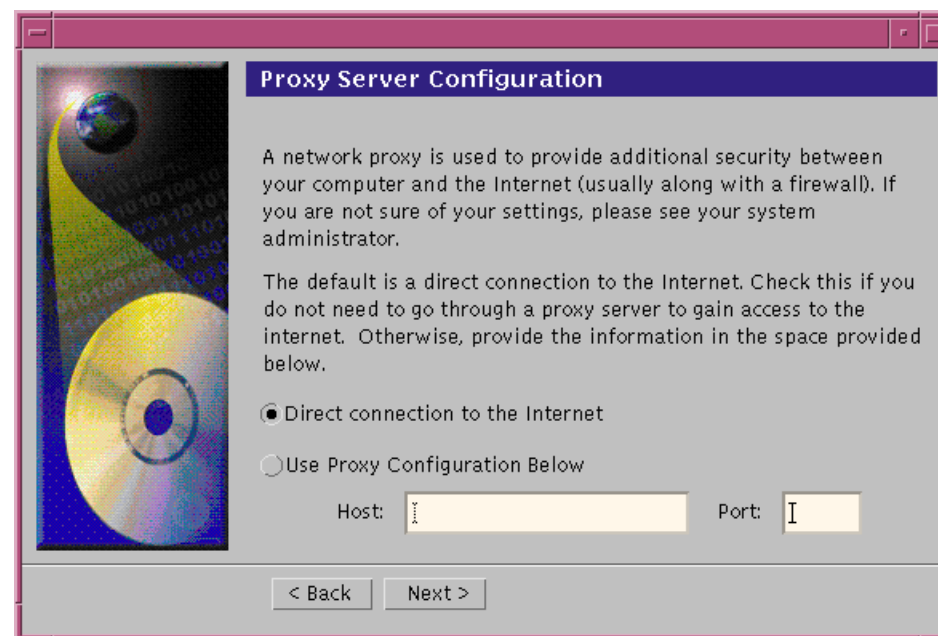


Power Management



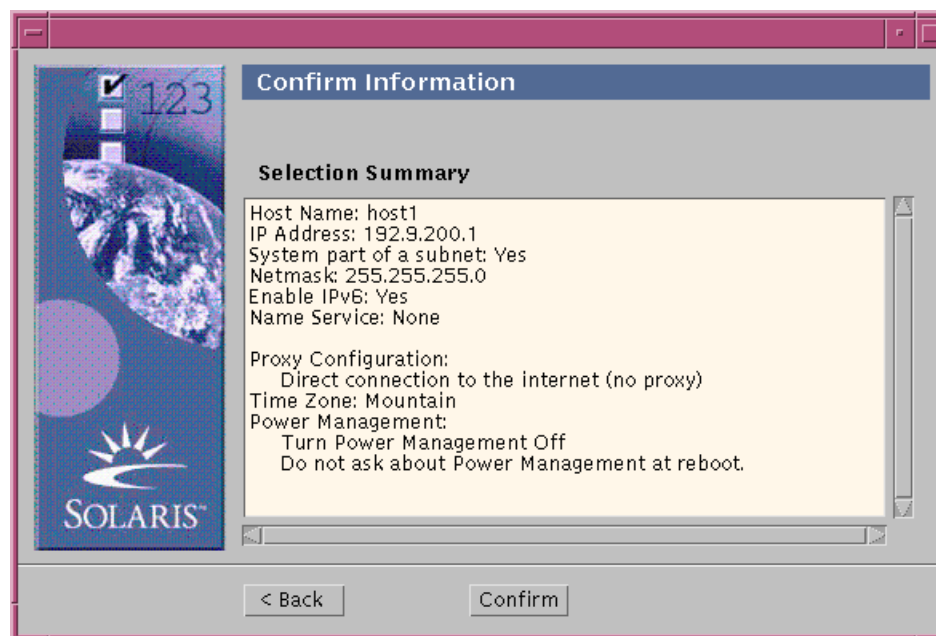


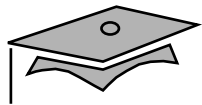
Proxy Server Configuration



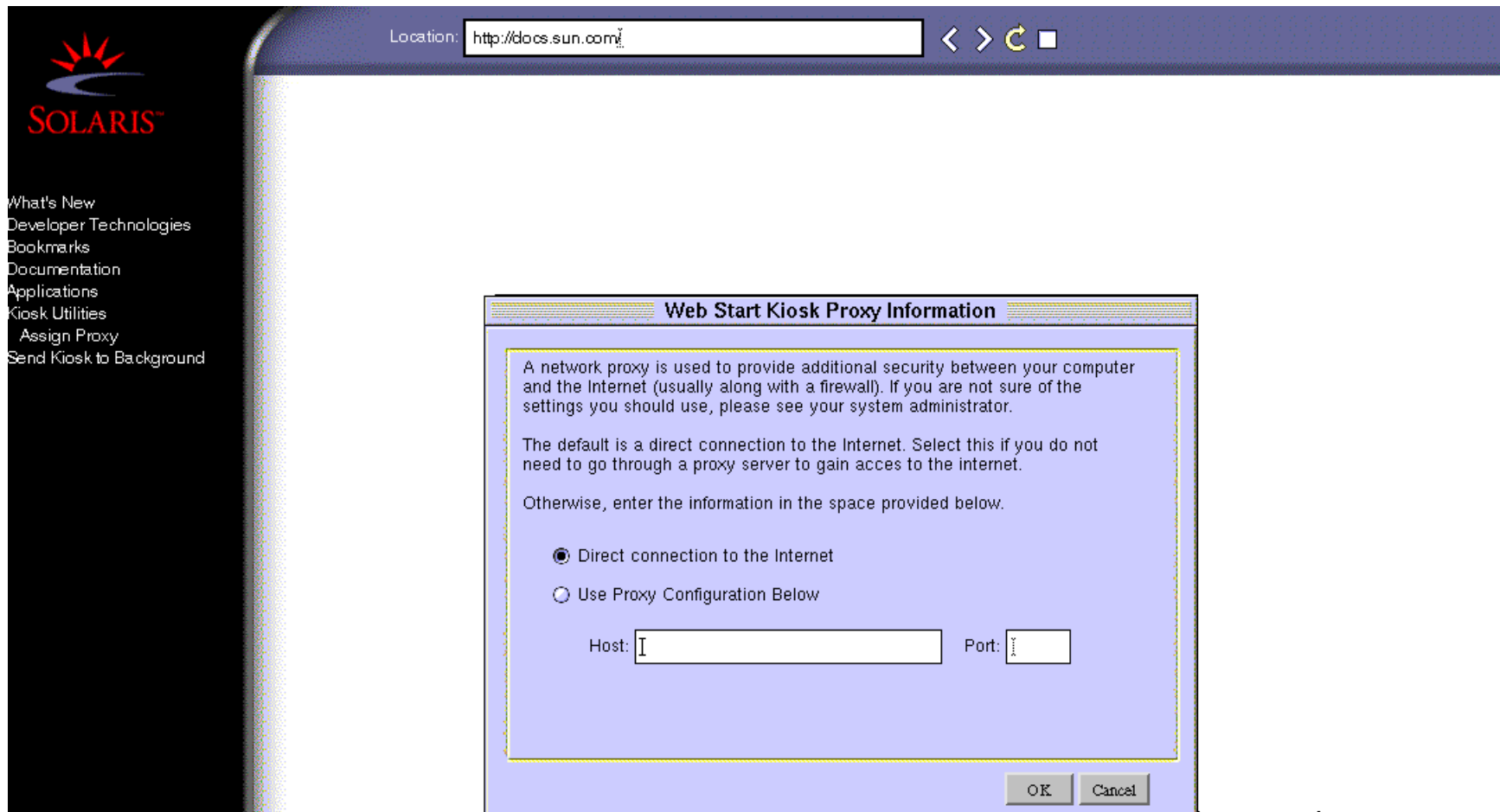


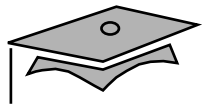
Confirmation Information



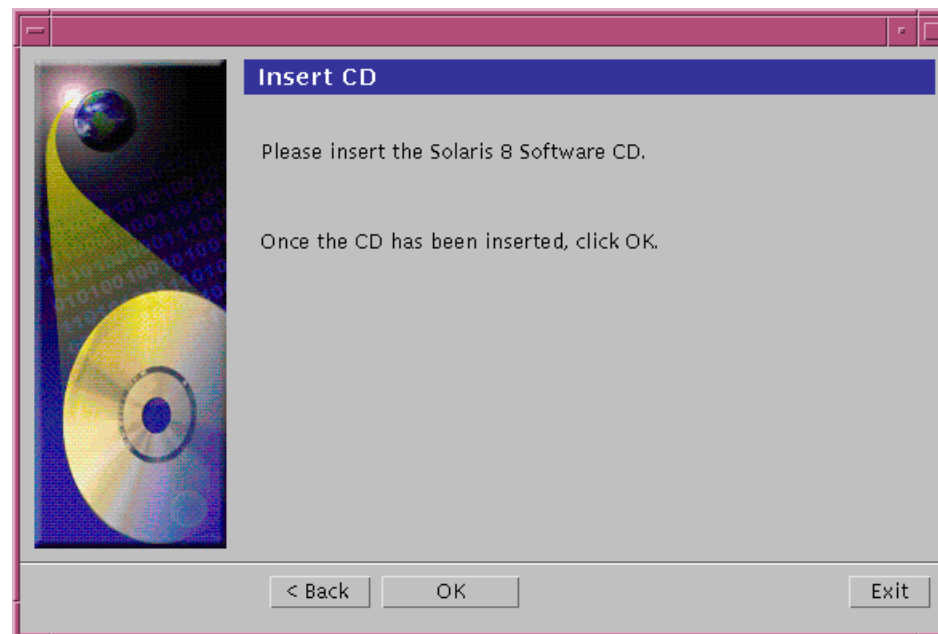


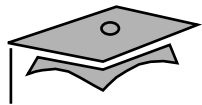
Solaris Web Start Installation Kiosk



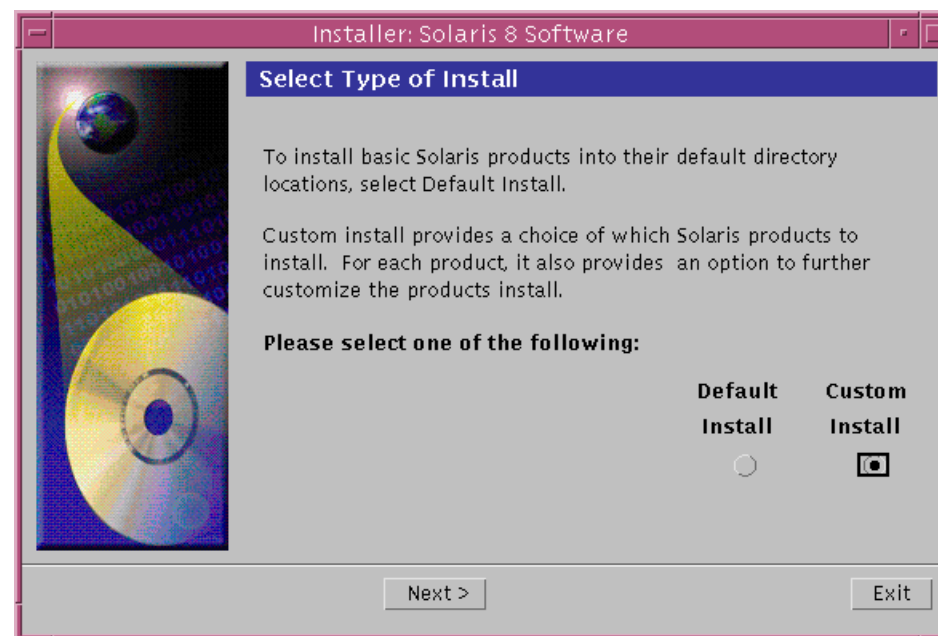


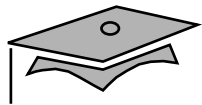
Insert CD





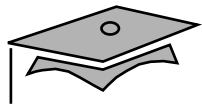
Select Type of Install





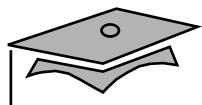
Select Software Localizations



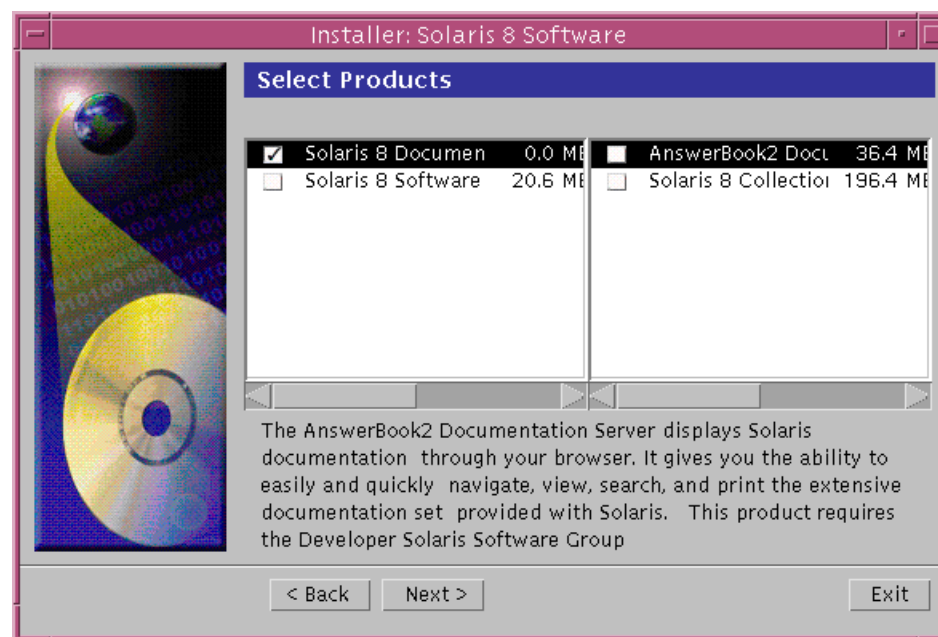


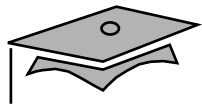
Select System Locale



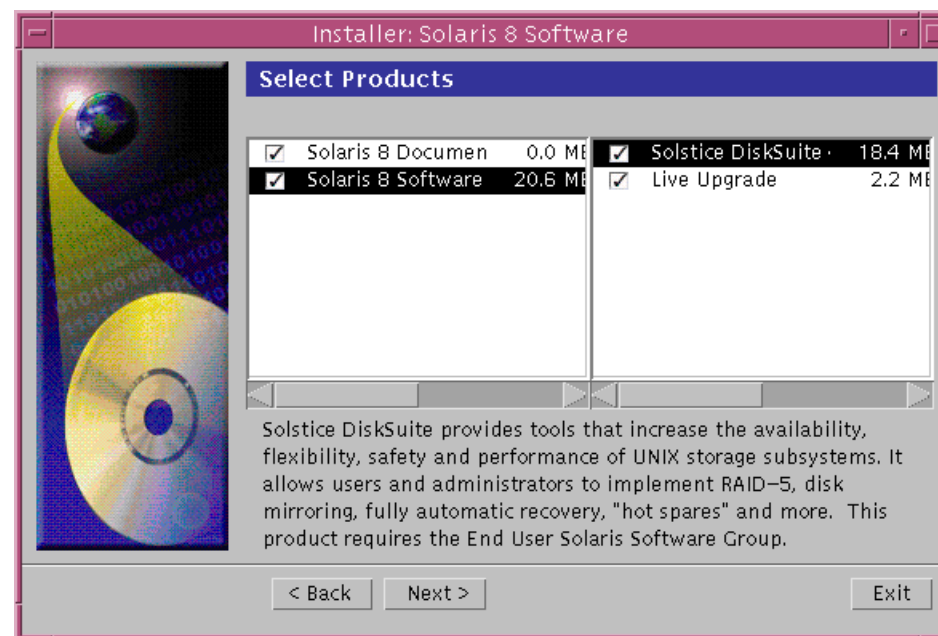


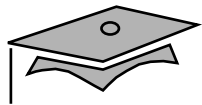
Select Products



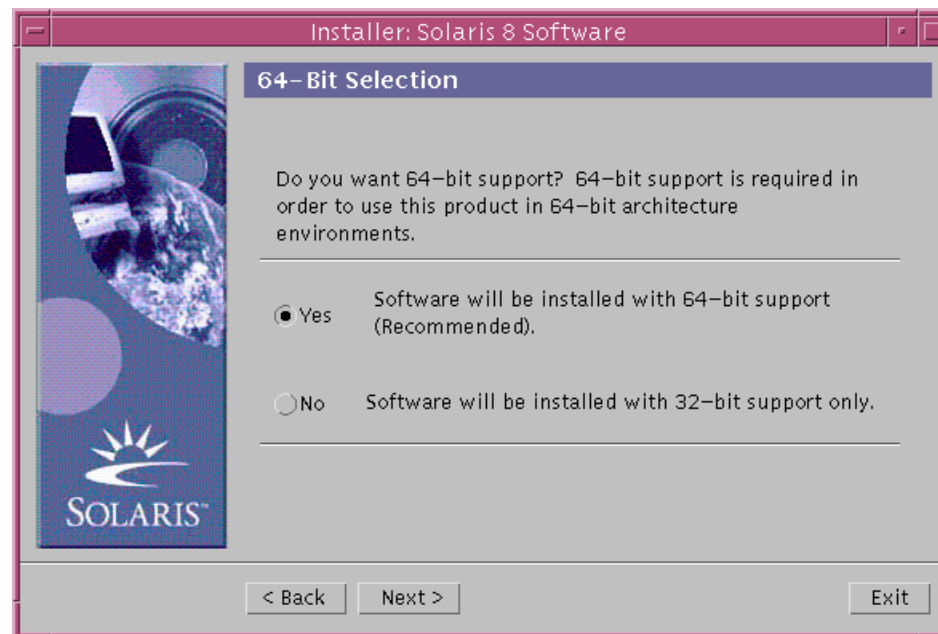


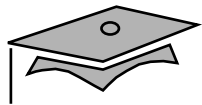
Additional Products



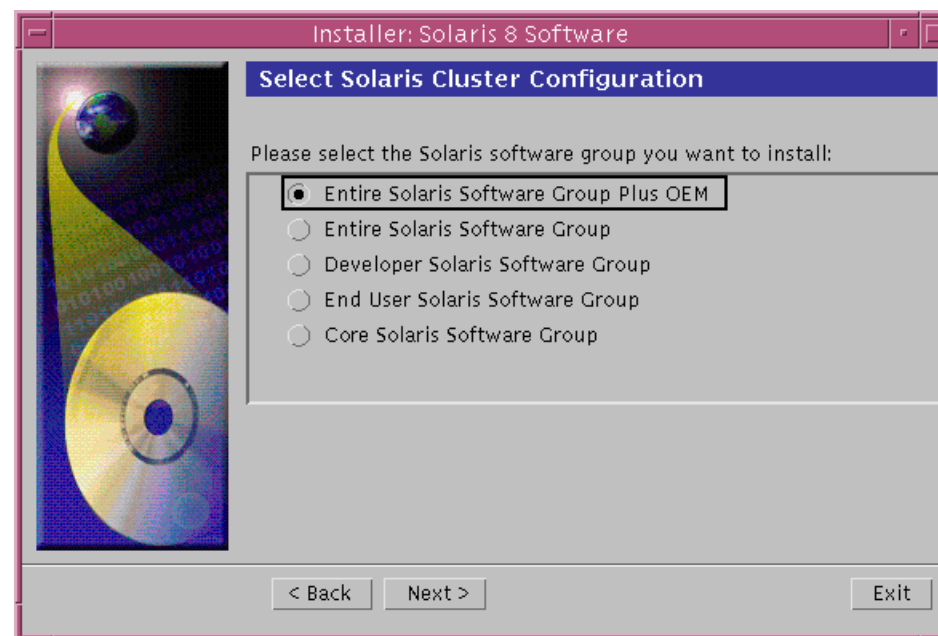


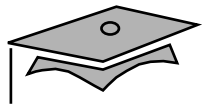
64 Bit Selection



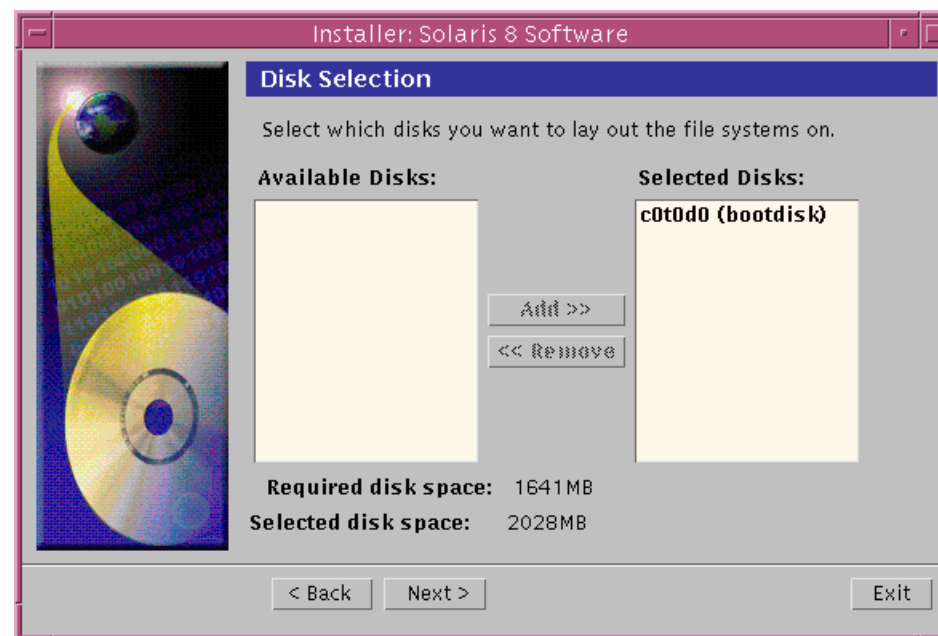


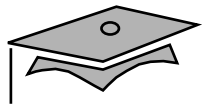
Select Solaris Cluster Configuration



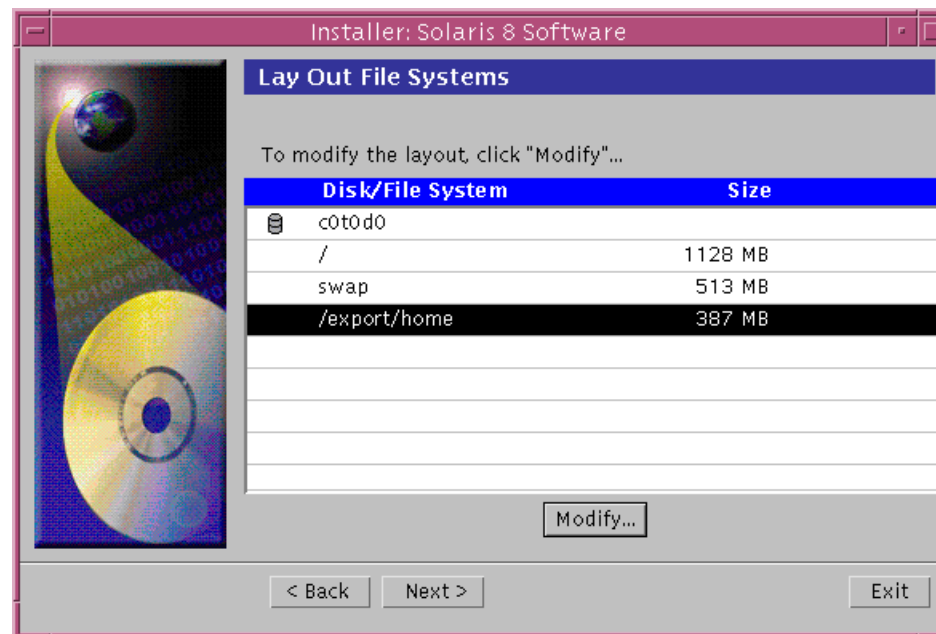


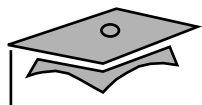
Disk Selection





Layout File Systems





Disk

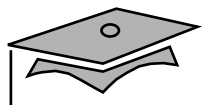
Disk c0t0d0

Disk c0t0d0 – 2,028 MB

Slice	File System	Size	Min. Size
0	/	1415	1128 MB
1	swap	513	
3			
4			
5			
6			
7	/export/home	100	

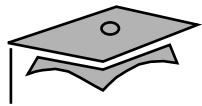
Capacity: 2028 MB
Allocated: 2028 MB
Free: 0 MB
Rounding Error: 0 MB

OK Apply Reset Cancel

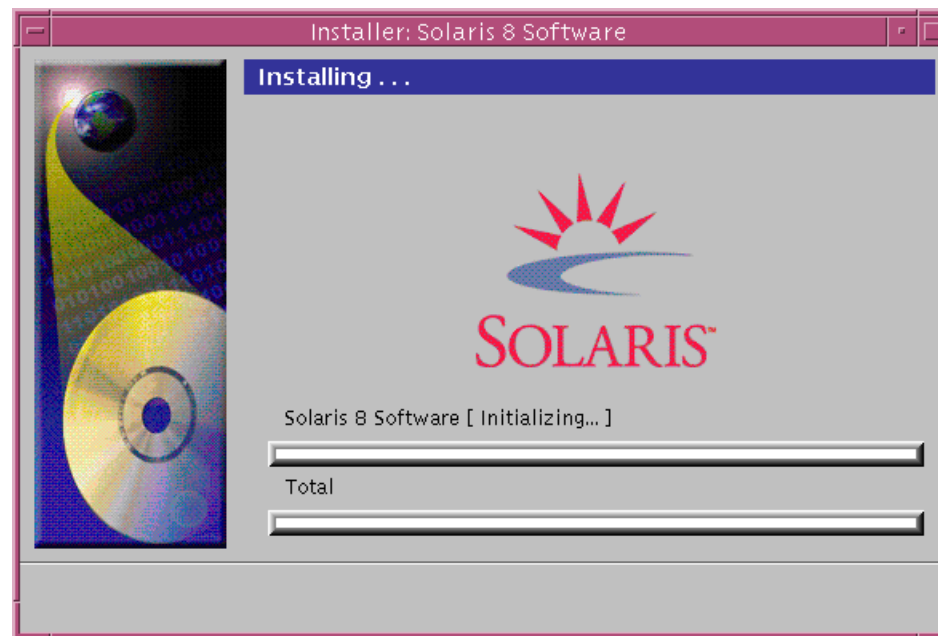


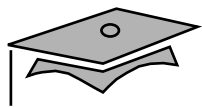
Ready to Install



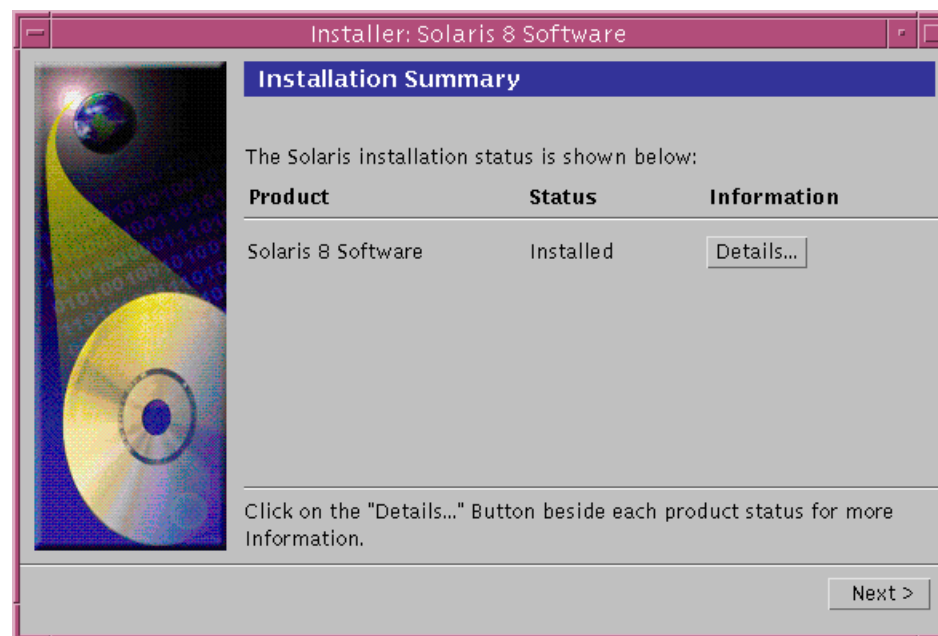


Installing



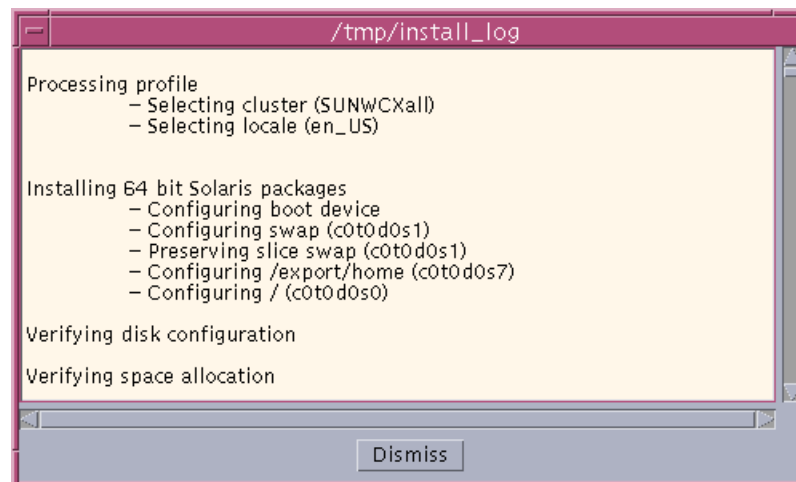


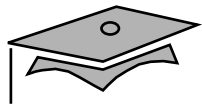
Installation Summary





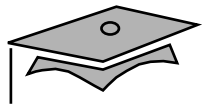
Details





Exercise: Solaris Operating Environment

- Objectives
- Tasks
- Discussion
- Solutions



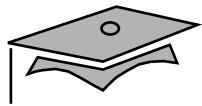
Check Your Progress

- State the different installation methods available for the Solaris 8 software
- Explain the hardware requirements for a Solaris 8 installation
- Identify the different Solaris 8 software CDROM editions
- List the five Solaris Software Groups
- Demonstrate how to install Solaris 8 software on a networked, standalone system, using Solaris Web Start



Module 15

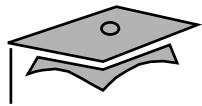
Administration of Software Packages



Objectives

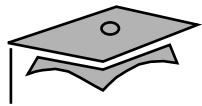
Upon completion of this module, you should be able to:

- 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



Objectives

- View, add, and remove software packages using the `admintool`
- Add and remove a software package from a spool directory using the `pkgadd` and `pkgrm`



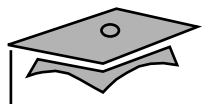
Software Packages

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.

Tools for viewing, adding and removing software include:

- Package administration commands: `pkgadd`, `pkgrm`, `pkginfo`, and `pkgchk`
- `admintool`



The pkginfo Command

The `pkginfo` command is used to display information about the software packages that have been installed on the local system.

```
# pkginfo | more
```

```
<some output omitted>
```

application	SUNWaxg	Solaris XGL 3.3 AnswerBook
application	SUNWadm	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

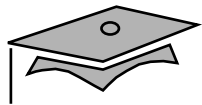


Displaying Detailed Information for a Specific Package

To view information for a specific software package, specify its name on the command line, for example:

```
# pkginfo -l SUNWman
```

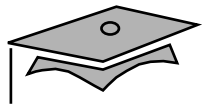
```
PKGINST:      SUNWman
NAME:         On-line Manual Pages
CATEGORY:    system
ARCH:        sparc
VERSION:     41.0,REV=31
BASEDIR:     /usr
VENDOR:      Sun Microsystems, Inc.
DESC:        System Reference Manual Pages
PSTAMP:      tinkertoym09133331
INSTRELEASE: May 19 2000 16:50
HOTLINE:      Please contact your local service provider
STATUS:      Completely Installed
FILES:        6420 installed pathnames
              3 shared pathnames
              74 directories
              73925 blocks used (approx)
```



Displaying Information for Software Packages on CDROM

- Displaying Detailed Information for All Packages on CDROM

```
# pkginfo -d /cdrom/cdrom0/s0/Solaris_8/Product | more
```



Displaying Information for Software Packages on CDROM

- Displaying Detailed Information for Selected Package on CDROM

```
# 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

```
# 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:
```

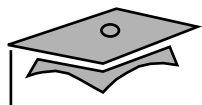
```
    The <SUNWoldcv> package depends on the package  
    currently being removed.
```

```
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
```



The pkgadd Command

```
# 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.
```

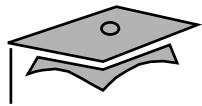
```
This package contains scripts which will be executed with super-user  
permission during the process of installing this package.
```

```
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

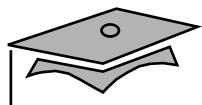
The pkgchk command checks installation completeness, pathname, file contents, and file attributes of a package.

- To check the contents and attributes of a software package currently installed on the system.

```
# pkgchk SUNWaudio
```

- To list the files contained in a software package.

```
# pkgchk -v SUNWaudio
```



The pkgchk Command

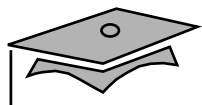
- To check any file to determine if its content and attributes have changed since it was installed with its software package.

```
# pkgchk -p /etc/passwd
```

```
ERROR: /etc/passwd
```

```
file size <414> expected <3391> actual
```

```
file cksum <34239> expected <17254> actual
```



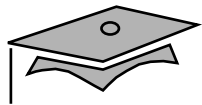
Identifying the Directory Location of a Command

- Search the `/var/sadm/install/contents` file to determine if a particular file was installed, and the directory where it is located.

```
# 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
```

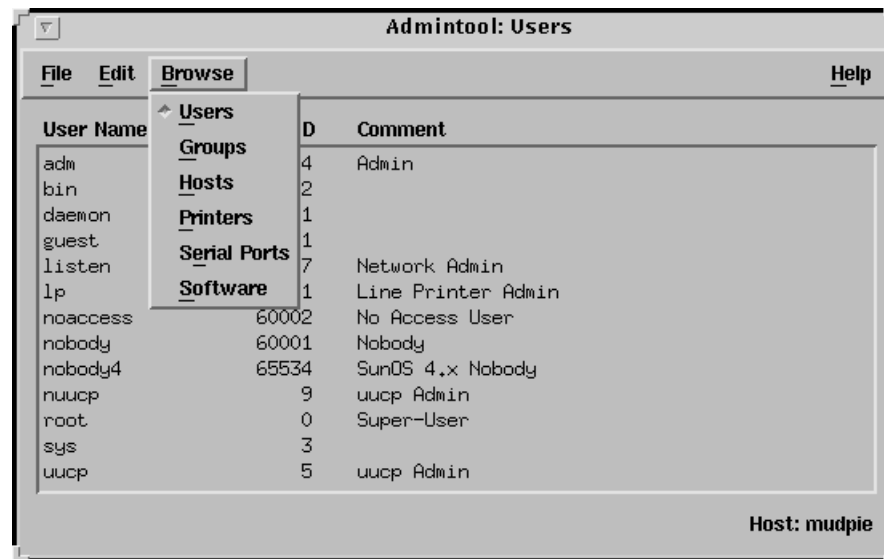
- Search on the distribution media to determine if a particular file exists.

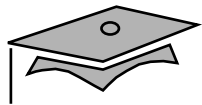
```
# grep showrev /cdrom/cdrom0/s0/Solaris_8/Product/*/pkgmap
/cdrom/sol_8_sparc/s0/Solaris_8/Product/SUNWadmc/pkgmap:1 f none usr/bin/showrev 0755 root sys
31276 44676 938676470
```



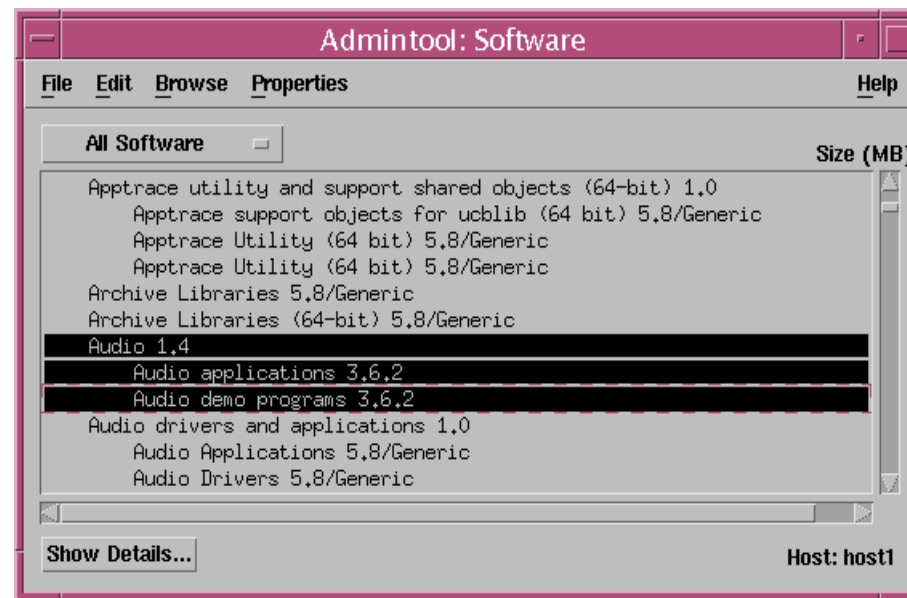
Adding and Removing Packages With admintool

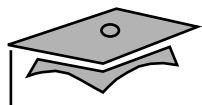
admintool &





Software Window





Software Details

Admin tool: Software Details

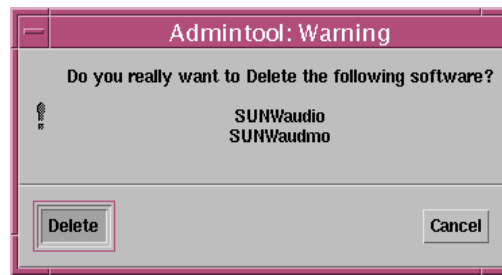
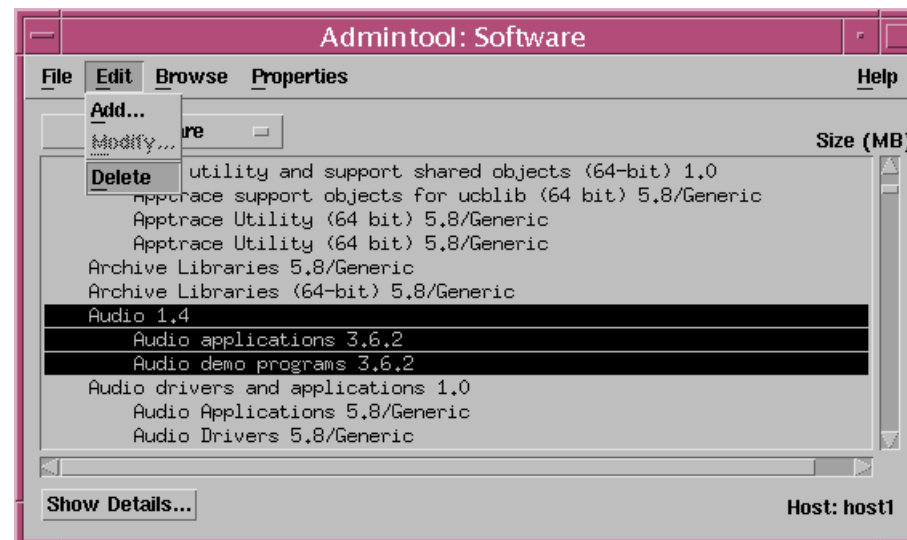
Name: Audio
Abbreviation: SUNWCaud 1.4
Product:
Package Instance:
Vendor: Sun Microsystems, Inc.
Description: Audio tools and demos
Architecture:
Date Installed:
Base Directory:
Estimated Size (MB):

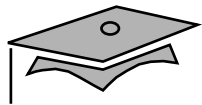
/	0	/var	0
/usr	0	/export	0
/opt	0	/usr/openwin	0

Cancel

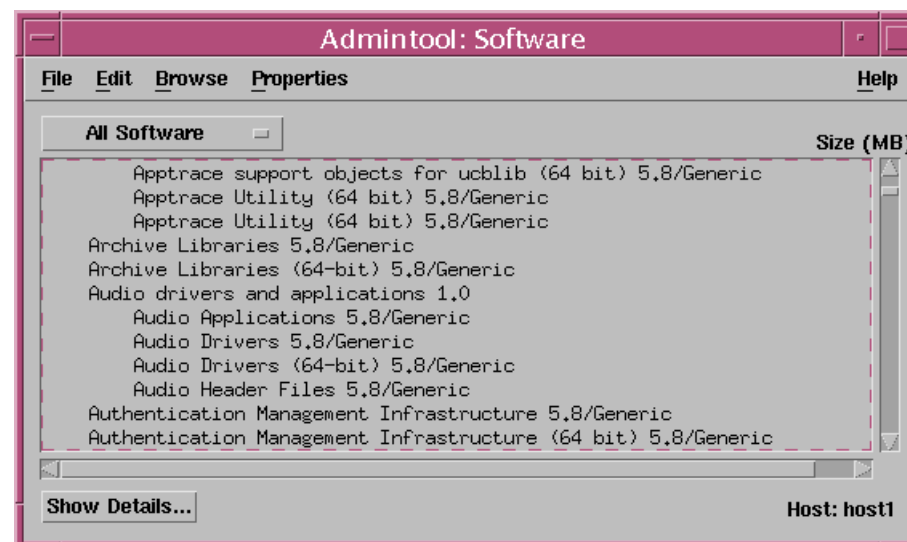


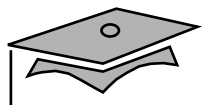
Delete Software



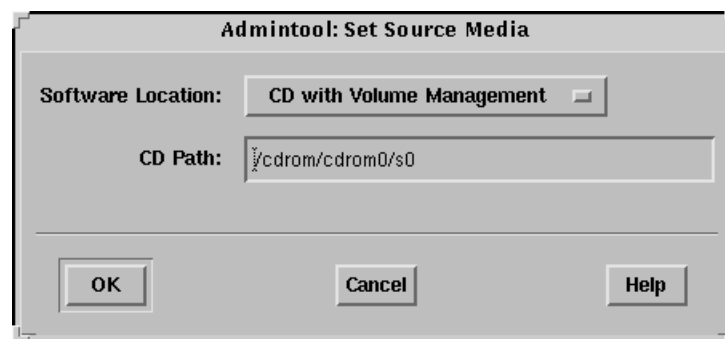


Adding Software Package

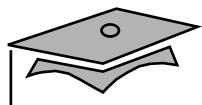




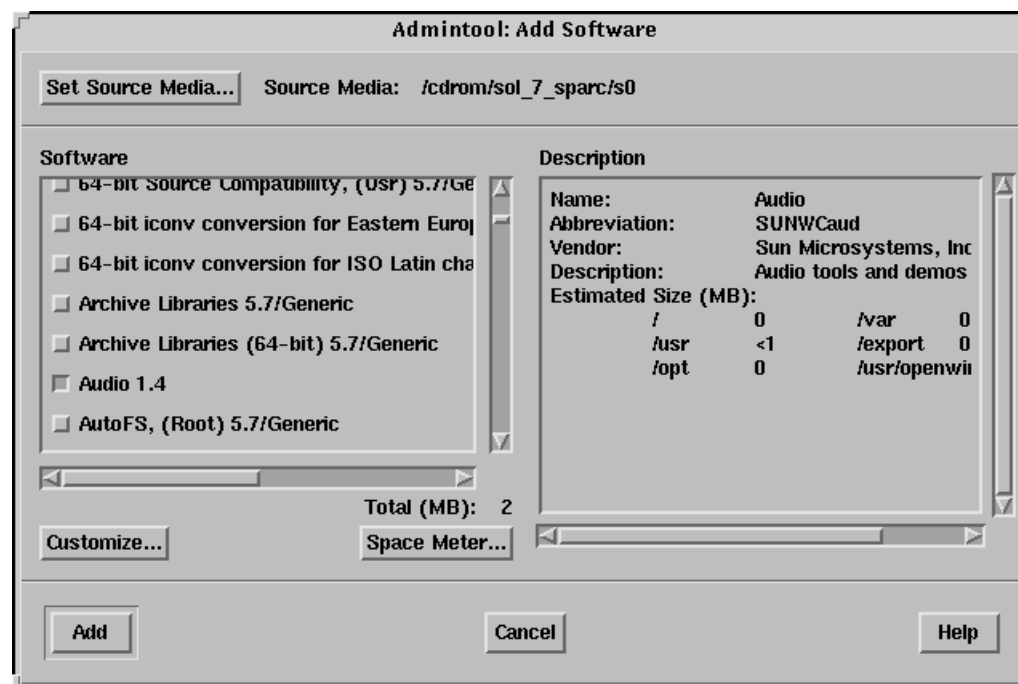
Set Source Media

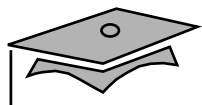


- CD with Volume Management
- CD without Volume Management
- Hard Disk

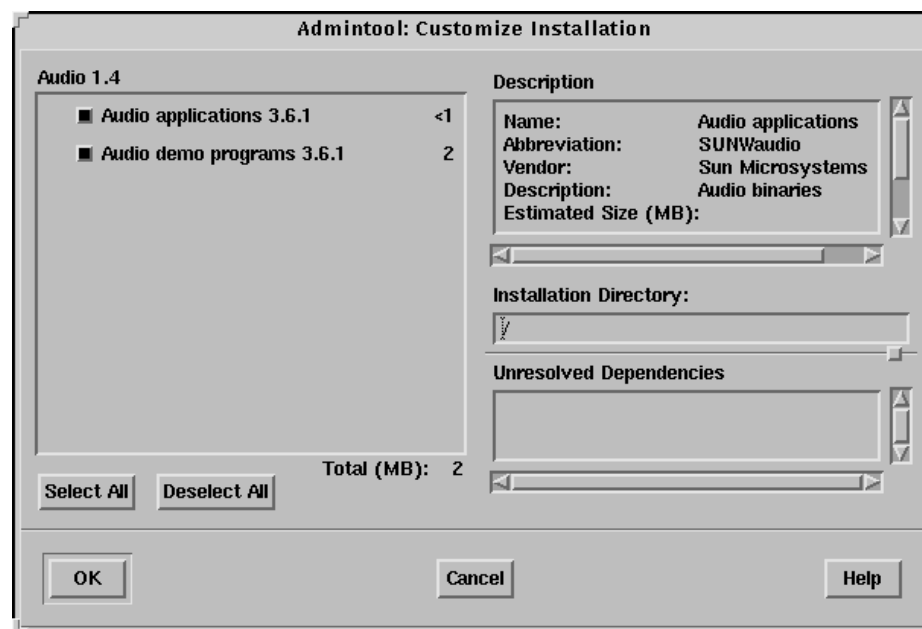


Add Software





Customize Installation



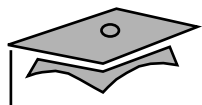


Using a Spool Directory

The `pkgadd` command, by default, looks in the `/var/spool/pkg` directory for any packages specified on the command line.

- To copy a package into the `/var/spool/pkg` directory:

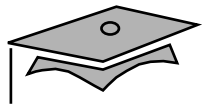
```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_8/Product -s spool  
SUNWaudio  
Transferring <SUNWaudio> package instance
```



Spooling Packages

A different directory location can be specified using the -s option.

```
# mkdir /export/pkg  
# pkgadd -d /cdrom/cdrom0/s0/Solaris_8/Product -s /export/  
pkgs SUNWaudio  
Transferring <SUNWaudio> package instance  
# ls /export/pkg  
SUNWaudio
```



To Remove Packages From Spool Directory

Software packages are removed from a spool directory using the `pkgrm` command with the `-s` option.

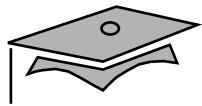
```
# pkgrm -s spool SUNWaudio
```

```
# pkgrm -s /export/pkgs SUNWaudio
```



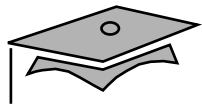
Exercise: Software Package Administration Commands

- Objectives
- Tasks
- Discussion
- Solutions



Check Your Progress

- Describe a software package
- View software package information using the `pkginfo` command
- Add a software package from the Solaris Software CDROM 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



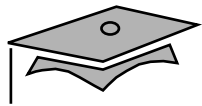
Check Your Progress

- View, add, and remove software packages using the `admintool`
- Add and remove a software package from a spool directory using the `pkgadd` and `pkgrm`



Module 16

Managing Software Patches



Objectives

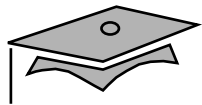
Upon completion of this module, you should be able to:

- 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



Objectives

- Demonstrate how to verify what patches are currently installed
- Remove a patch using the `patchrm` command



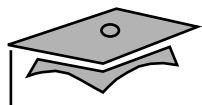
Patch Distribution

- To access patches via the World Wide Web, use the following URLs:

<code>http://sunsolve.sun.com</code>	United States
<code>http://sunsolve.sun.com.au</code>	Australia
<code>http://sunsolve.sun.fr</code>	France
<code>http://sunsolve.sun.de</code>	Germany
<code>http://sunsolve.sun.co.jp</code>	Japan
<code>http://sunsolve.sun.se</code>	Sweden
<code>http://sunsolve.sun.ch</code>	Switzerland
<code>http://sunsolve.sun.co.uk</code>	United Kingdom

- Publicly available patches can be accessed by using the URL:

`http://metalab.unc.edu/pub/sun-info/sun-patches/`



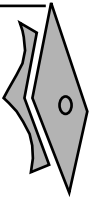
Patch Distribution

- To access patches via ftp
`sunsolve.sun.com`



Patch Informational Documents

Patch Document	Contents
Solaris8.PatchReport	A summary of all recommended patches for the Solaris 8 release.
8_Recommended.zip	A patch cluster containing all the recommended patches for the Solaris 8 release.
8_Recommended.README	Instructions for how to install the recommended patches for Solaris 8.



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

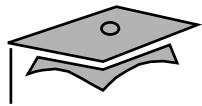
Changes incorporated in this version: 4303885 4308725

Date: Mar/16/00

Patch-ID# 108609-01

Synopsis: SunOS 5.8: Buttons/Dials Patch

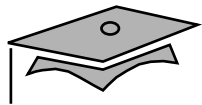
BugId''s fixed with this patch: 4299526 <output truncated>.....



Patch Formats

Patches come in three different formats depending on the Solaris version and where the patch had been retrieved.

- Solaris 8 and Solaris 7 patches are in `zip` format.
`105050-01.zip`
- Solaris 2.6 (and earlier) patches are compressed tar files in a `tar.Z` format.
`104040-01.tar.Z`
- The Solaris 2.6 (and earlier) Patch Update CDROM contains patches that are `gzip` compressed tar files.
`112340-01.tar.gz`



Preparing Patches for Installation

- For Solaris 8 and Solaris 7 the `unzip` command is used to extract the patch files.

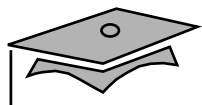
```
# /usr/bin/unzip 105050-01.zip
```

- For Solaris 2.6 patches the `zcat` command is used to uncompress the patch files and `tar` to create the patch directories.

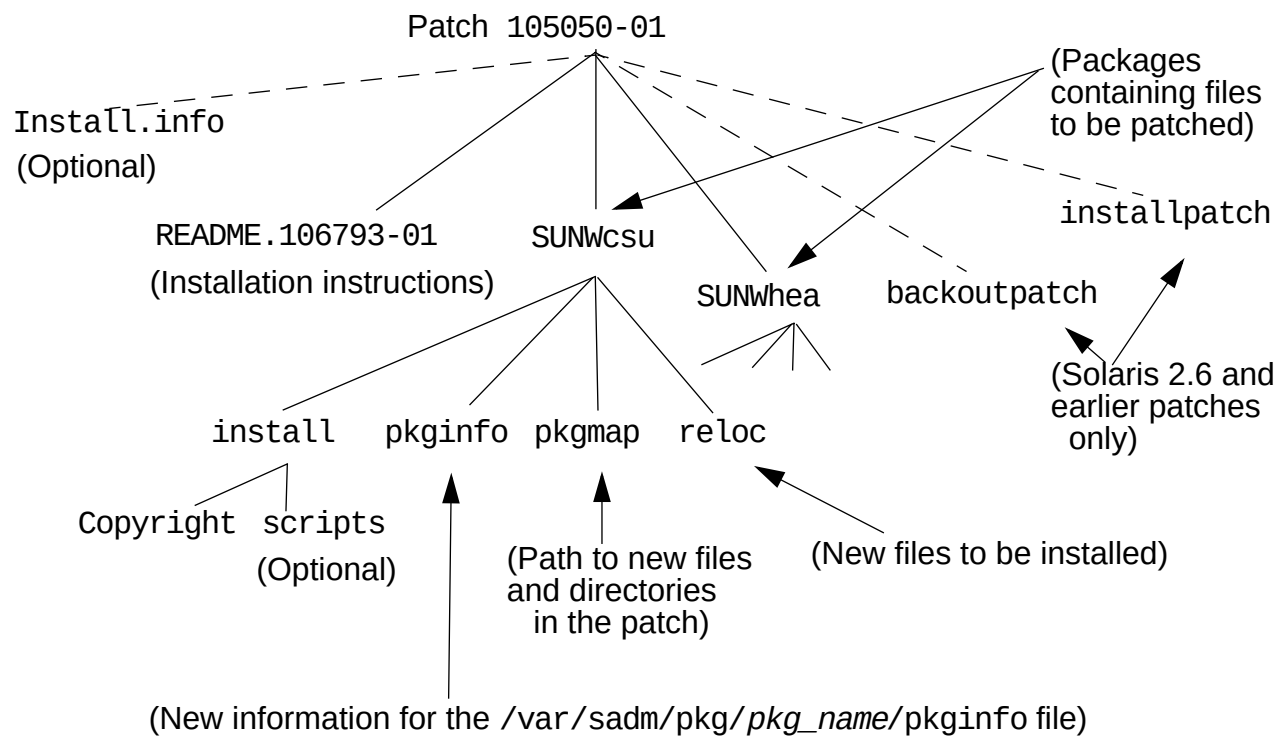
```
# /usr/bin/zcat 104040-01.tar.Z | tar xvf -
```

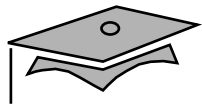
- For Solaris 2.6 patches retrieved from the Patch Update CDROM the `gzcat` command is used to uncompress and create patch directories.

```
# /usr/bin/gzcat 112340-01.tar.gz | tar xvf -
```



Patch Contents

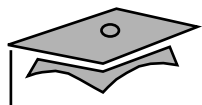




Managing Patches

There are two commands available to the system administrator for managing patches:

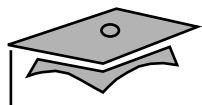
- `patchadd` — Used to install unpacked patches to a Solaris system.
- `patchrm` — Used to remove patches installed on a Solaris system.



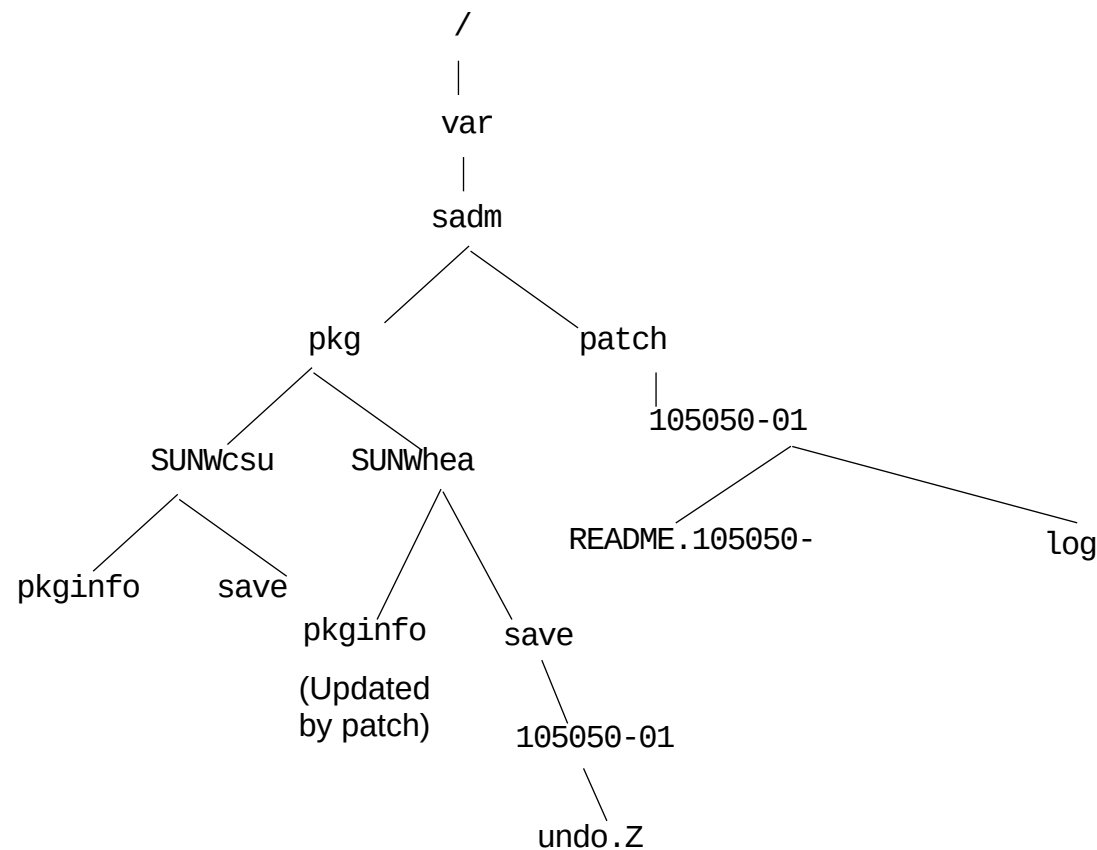
Patch Installation on Solaris 2.6 and Later Versions

For Solaris 2.6 and above, the `patchadd` command is available in the Solaris Operating Environment to install a patch.

```
# 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
```



The /var/sadm Directory





Checking Current Patch Status

There are two commands available that provide 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
```

```
. . .
```



Patch Removal on Systems at Solaris 2.6 and Later

For Solaris 2.6 and above, the `patchrm` command is available in the Solaris Operating Environment to remove a patch.

```
# patchrm 106793-01
```

```
Checking installed packages and patches...
```

```
Backing out patch 106793-01...
```

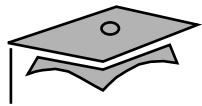
```
Patch 106793-01 has been backed out.
```

```
#
```



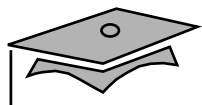

Exercise: Patch Maintenance

- Objectives
- Tasks
- Discussion
- Solutions



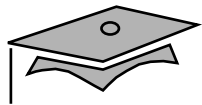
Check Your Progress

- 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



Module 17

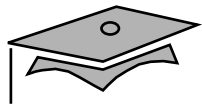
Backup and Recovery



Objectives

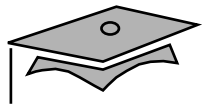
Upon completion of this module, you should be able to:

- Identify the logical device names for tape drives
- Define the two different types of file system backups
- Back up 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



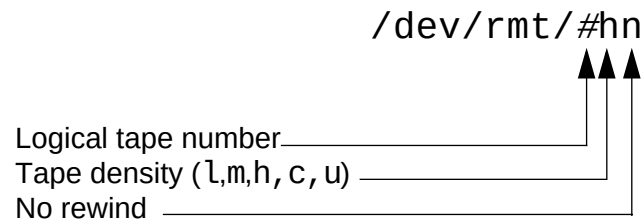
Objectives

- 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



Logical Tape Device Names

All tape devices have logical device names that are used to reference the device on the command line.

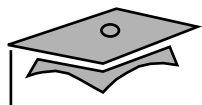




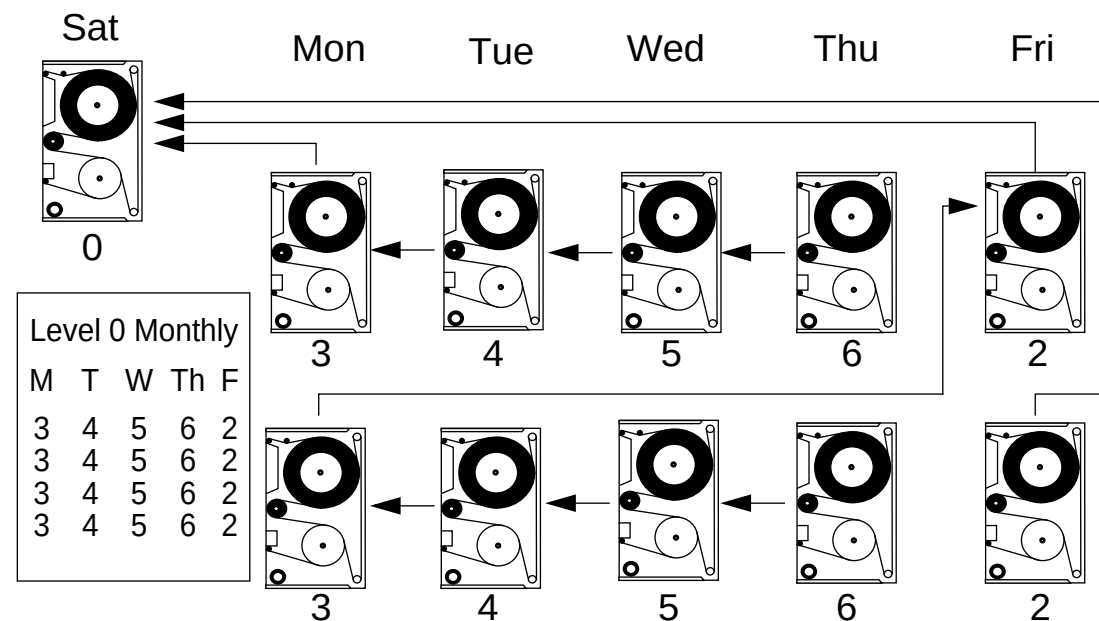
The /etc/dumpdates File

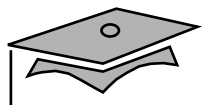
Each line in /etc/dumpdates file shows the file system backed up, the level of the last backup, and the day, date, and time of the backup.

```
# 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
```



A Sample Backup Strategy





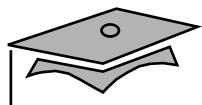
Planning File System Backups

Determining the Number of Tapes

- To determine the size of the file system to be backed up to tape:

```
# ufsdump 0S filesystem_name  
<number reported>
```

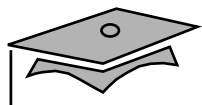
```
# ufsdump 3S filesystem_name  
<number reported>
```



Backing Up to Tape

To 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

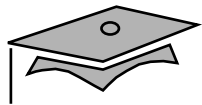
The `ufsdump` command can be used to perform a backup on a remote tape device.

```
# 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/rdsk/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
#
```



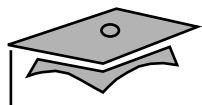
Exercise: Backup and Recovery

- Objectives
- Tasks
- Discussion
- Solutions



Check Your Progress

- Identify the logical device names for tape drives
- Define the two different types of file system backups
- Back up a file system to tape using the `ufsdump` command
- Describe how to back up 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



Check Your Progress

- Use the `tar` command to manage multiple archives
- Use the `mt` command to control the actions of the tape drive

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.

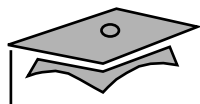
L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de 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.

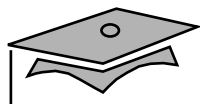
Le système X Window est un produit de X Consortium, Inc.

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.

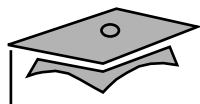




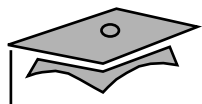
About This Course	Preface-i
Course Goal	Preface-ii
Course Overview	Preface-iii
Course Map	Preface-iv
Module-by-Module Overview	Preface-v
Course Objectives	Preface-viii
Skills Gained by Module	Preface-xi
Guidelines for Module Pacing	Preface-xiii
Topics Not Covered	Preface-xv
How Prepared Are You?	Preface-xvii
Introductions	Preface-xviii
How to Use Course Materials	Preface-xix
How to Use the Icons	Preface-xx
Typographical Conventions and Symbols	Preface-xxii
 Introducing the Solaris 8 Operating Environment System Administration	 1-1
Objectives	1-2
Roles of the System Administrator for a Standalone System	1-3
Roles of the System Administrator for Client-Server Systems	1-5
System Administration Terms	1-7
Distributed Computing Environment	1-8
Check Your Progress	1-9
 Adding Users	 2-1
Objectives	2-2
Setting Up User Accounts	2-4
Managing User Accounts	2-5
Managing User Accounts with <code>admintool</code>	2-6
Storing User and Group Account Information	2-7
Creating and Managing Accounts from the Command-line	2-8



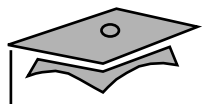
Creating Accounts with <code>useradd</code>	2-9
Modifying a User Account with <code>usermod</code>	2-10
Deleting User Accounts with the <code>userdel</code> Command	2-11
Creating Accounts with <code>groupadd</code>	2-12
Modifying a Group Account with <code>groupmod</code>	2-13
Deleting Group Accounts with the <code>groupdel</code> Command	2-14
Exercise: Adding Users and Groups	2-15
Using the Initialization Files	2-16
System-Wide Initialization Files	2-17
User Initialization Files	2-18
Initialization Files for the Six Shells	2-19
Shell Variables	2-20
Setting Environment Variables	2-21
Default User Initialization Files	2-22
Exercise: Using the Initialization Files	2-23
Check Your Progress	2-24
 System Security	3-1
Objectives	3-2
Managing System Security Overview	3-4
The <code>pwconv</code> Command	3-5
Recording Failed Login Attempts	3-6
Monitoring System Access	3-7
Displaying User Information	3-8
Displaying a Record of Login Activity	3-9
Displaying Users on Remote Systems	3-10
Accessing root Privileges	3-11
Effective User ID and Effective Group ID	3-12
Using the <code>whoami</code> Command	3-13
Using the <code>su</code> Command to Become root	3-14



Using the su Command to Become Another Regular User	3-17
Exercise: User Access	3-20
Check Your Progress	3-21
Determining a User's Group Membership	3-22
Identifying a User Account	3-23
Changing a File's Ownership	3-24
Changing Directory Ownership	3-25
Changing User and Group Ownership Simultaneously	3-26
Changing a File's Group Ownership	3-27
Special File Permissions	3-28
The setuid Permission	3-29
The setgid Permission	3-30
The Sticky Bit Permission	3-31
Exercise: File Owners, Groups, and Special Permissions	3-32
Check Your Progress	3-33
Access Control Lists	3-34
ACL Commands	3-35
ACL Entries	3-36
ACL Entry Examples	3-37
Adding ACL Permissions on a File	3-38
Modifying ACL Permissions on a File	3-39
Determining if a File has an ACL	3-40
Deleting an ACL Entry on a File	3-41
Replacing an Entire ACL on a File	3-42
Exercise: Access Control Lists	3-43
Check Your Progress	3-44
Managing Remote Access Issues	3-45
The /etc/hosts.equiv and \$HOME/.rhosts Files	3-46
Entries in /etc/hosts.equiv and \$HOME/.rhosts Files	3-48
The /etc/hosts.equiv File	3-49
The \$HOME/.rhosts File	3-50

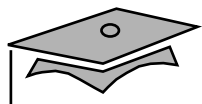


Restricting FTP Logins	3-51
The /etc/shells File	3-52
Exercise: Managing Remote Security Issues	3-53
Check Your Progress	3-54
 <i>The Directory Hierarchy</i>	<i>4-1</i>
Objectives	4-2
Solaris File Types	4-3
Identifying the File Types	4-4
File Types	4-5
File Names, Inodes, and Data Blocks	4-6
Regular Files	4-8
Directories	4-9
Symbolic Links	4-10
Device Files	4-11
A Device File Example	4-12
Two Categories of Device Files	4-13
Hard Links	4-14
A Hard Link Example	4-15
The root Subdirectories	4-16
Exercise: Identifying File Types	4-19
Check Your Progress	4-20
 <i>Device Configuration</i>	<i>5-1</i>
Objectives	5-2
Basic Architecture of a Disk	5-4
Components of a Disk Platter	5-5
Disk Slices Defined	5-6
The Boot Disk	5-7
How Slices Reside on Disk	5-8

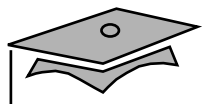


Disk Slice Naming Convention	5-9
An Embedded SCSI Configuration	5-10
An IDE Configuration	5-11
Device Naming Conventions	5-12
Logical Device Names	5-13
Physical Device Names	5-14
The /devices directory Structure	5-15
Instance Names	5-16
The /etc/path_to_inst File	5-17
Sample /etc/path_to_inst File	5-18
The prtconf Command	5-19
The format Command	5-20
The Solaris 8 Operating Environment Device Reconfiguration	5-21
Pre- Solaris 8 Operating Environment Device Reconfiguration	5-22
Exercise: Disk Configuration and Naming	5-23
Check Your Progress	5-24

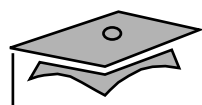
<i>Disks, Slices, and Format</i>	<i>6-1</i>
Objectives	6-2
Disk Slices and the format Utility	6-4
Disk Labels and Partition Tables	6-5
Disk Partition Table	6-6
Defining Disk Slices	6-7
Disk Space and Undesirable Conditions	6-8
Wasted Disk Space	6-9
Overlapping Disk Slices	6-10
The /etc/format.dat File	6-11
Viewing a Disk's VTOC	6-12
Reading a Disk's VTOC with the verify Command	6-13
Reading a Disk's VTOC with the prtvtoc Command	6-14



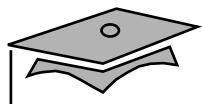
Exercise: Disks, Slices, and Format	6-15
Check Your Progress	6-16
 <i>The Solaris Operating Environment ufs File System</i>	<i>7-1</i>
Objectives	7-2
File System Types Supported by the Solaris Operating Environment.....	7-3
The Solaris ufs File System	7-4
Basic Disk Structures	7-6
Data Blocks and Fragmentation	7-9
Shadow Inode	7-11
Creating ufs File Systems	7-12
Exercise: Disks, Slices, and Format	7-13
Check Your Progress	7-14
 <i>Mounting File Systems</i>	<i>8-1</i>
Objectives	8-2
Working with File Systems	8-4
Identifying Mounted File Systems	8-6
Mounting File Systems	8-7
The Virtual File System Table	8-8
The /usr/sbin/mountall Command	8-9
Unmounting File Systems	8-10
The /usr/sbin/umountall Command	8-11
Commands to Unmount a Busy File System	8-12
Removable Media Device Management	8-13
Accessing Mounted Diskettes and CD-ROMs	8-14
Administering Volume Management	8-16
Mounting Different Types of File Systems	8-17
Determining a File System's Type	8-18
Exercise: Mounting File Systems	8-19
Check Your Progress	8-20



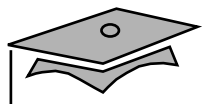
Maintaining File Systems	9-1
Objectives	9-2
The File System Check Program	9-3
Data Inconsistencies Checked by fsck	9-4
Phases of fsck	9-5
Using fsck	9-6
Troubleshooting with fsck	9-7
Using Backup Superblocks	9-8
Monitoring File System Usages	9-9
The df Command	9-10
The du Command	9-11
The ff Command	9-12
The quot Command	9-13
Exercise: Maintaining File Systems	9-14
Check Your Progress	9-15
 Scheduled Process Control	 10-1
Objectives	10-2
Viewing Processes and PIDs	10-4
CDE Process Manager	10-5
The prstat Command	10-6
Scheduling the Automatic Execution of Commands	10-7
The crontab File	10-8
Crontab for the root User	10-9
Controlling crontab Access	10-10
The at Command	10-11
Controlling at Access	10-13
Exercise: Process Control	10-14
Check Your Progress	10-15



<i>The Solaris Operating Environment LP Print Service</i>	<i>11-1</i>
Objectives	11-2
Print Management Tools	11-4
Remote Printing in a Solaris 2.6 to Solaris 8 Environment	11-12
The accept and reject Commands	11-13
The enable and disable Commands	11-14
The lpmove Command	11-15
Configuring the LP Print Service Using lpadmin Command	11-16
Creating a Printer Class	11-17
Setting or Changing a System's Default Printer	11-18
Halt and Restart the LP Print Service	11-20
Exercise: LP Print Service	11-21
Check Your Progress	11-22
 <i>The Boot PROM</i>	 <i>12-1</i>
Objectives	12-2
Basic Elements of the Boot PROM and NVRAM	12-5
Power On Self Test (POST)	12-6
The OpenBoot Goal	12-7
Third Party Device Configuration	12-8
Basic BootPROM Configurations	12-9
Controlling the POST Phase	12-11
Basic Boot PROM Commands	12-12
Basic Boot PROM Commands	12-13
The banner Command	12-14
The boot Command	12-15
The help Command	12-17
The printenv Command	12-18
The setenv Command	12-19
The set-defaults Command	12-21



Boot Disk Device Path Example	12-25
Boot Disk Device Path Example	12-26
Using probe- Commands to Identify Devices	12-27
The probe-scsi Command	12-28
The probe-scsi-all Command	12-29
The probe-ide Command	12-30
Identifying the Boot Device	12-31
Creating Custom Device Aliases	12-32
Removing a Custom Device Aliases	12-33
Sample nvedit Session	12-34
The eeeprom Command	12-35
Interrupting an Unresponsive System	12-36
Exercise: OpenBoot Prom	12-37
 System Boot Process	 13-1
Objectives	13-2
Exercise: The Boot Process	13-23
Check Your Progress	13-24
Check Your Progress	13-25
 Installing the Solaris 8 Operating Environment on a Standalone System	 14-1
Objectives	14-2
Power Management	14-20
Proxy Server Configuration	14-21
Confirmation Information	14-22
Insert CD	14-24
Select Type of Install	14-25
Select Software Localizations	14-26
Select System Locale	14-27
Select Products	14-28



Additional Products	14-29
64 Bit Selection	14-30
Select Solaris Cluster Configuration	14-31
Disk Selection	14-32
Layout File Systems	14-33
Disk	14-34
Ready to Install	14-35
Installing	14-36
Installation Summary	14-37
Details	14-38
Exercise: Solaris Operating Environment	14-39
Check Your Progress	14-40
 <i>Administration of Software Packages</i>	<i>15-1</i>
Objectives	15-2
Customize Installation	15-21
Exercise: Software Package Administration Commands	15-25
Check Your Progress	15-26
Check Your Progress	15-27
 <i>Managing Software Patches</i>	<i>16-1</i>
Objectives	16-2
Exercise: Patch Maintenance	16-16
Check Your Progress	16-17
 <i>Backup and Recovery</i>	<i>17-1</i>
Objectives	17-2
Exercise: Backup and Recovery	17-10
Check Your Progress	17-11