System Administration for the Solaris 10 OS Part 1

Student Guide

SA-200-S10 Rev C.1

D61734GC11
Edition 1.1
D65080 and D65081

ORACLE

This page intentionally left blank.

This page intentionally left blank.

Table of Contents

Preface

About This Course

Course Goals

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

- Manage file systems
- Manage disk labels and system devices
- Control and monitor network interfaces
- Perform mounts and unmounts
- Administer packages
- Obtain and use patches
- Perform system boot procedures
- Use SMF administrative commands
- Perform user administration
- Control System Processes
- Install software

Course Map

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

Outline

Introducing the Solaris 10 OS Directory Hierarchy	Managing Local Disk Devices	Interface Configurations
Managing the Solaris OS UFS	ZFS File System Introduction	Performing Mounts and Unmounts
Performing Solaris 10 OS Package Administration	Managing Software Patches on the Solaris 10 OS	Using Boot PROM Commands
Using the Grand Unified Bootloader	Performing Legacy Boot and Shutdown Procedures	Service Management Facility
Performing User Administration	Controlling System Processes	Installation and Upgrades

Topics Not Covered

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

- Basic UNIX® commands – Covered in SA-100-S10: *UNIX® Essentials Featuring the Solaris™ 10 Operating System*
- The `vi` editor – Covered in SA-100-S10: *UNIX® Essentials Featuring the Solaris™ 10 Operating System*
- Basic UNIX file security – Covered in SA-100-S10: *UNIX® Essentials Featuring the Solaris™ 10 Operating System*
- Network File System (NFS) environment configuration – Covered in SA-202-S10: *System Administration for the Solaris™ 10 Operating System. Part 2*
- All the new features in Solaris 10 – Covered in SA-225-S10: *Solaris™ 10 for Experienced System Administrators*
- Naming services – Covered in SA-202-S10: *System Administration for the Solaris™ 10 Operating System, Part 2*
- Troubleshooting – Covered in ST-350: *Sun™ Systems Fault Analysis Workshop*
- System tuning – Covered in SA-400: *Solaris™ System Performance Management*

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

How Prepared Are You?

To succeed in this course you should be able to create procedural application programs.

Introductions

Please introduce yourself to the other students and to the instructor. Please include the following when you introduce yourself:

- Your name, title, and job responsibility
- Your company affiliation
- Your experience as it relates to topics presented in this course
- Your reasons for enrolling in this course
- Your expectations of this course.

How to Use the Course Materials

The course materials contain learning modules that are composed of the following components:

- Goals – You should be able to accomplish the goals after finishing this course and meeting all of its objectives.
- Objectives – You should be able to accomplish the objectives after completing a portion of instructional content. Objectives support goals and can support other higher-level objectives.
- Lecture – The instructor presents information specific to the objective of the module. This information helps you obtain the knowledge and skills necessary to succeed with the activities.
- Activities include exercises, self-checks, discussions, and demonstrations. Activities help you master an objective.
- Visual aids – The instructor might use visual aids to convey a concept, such as a process, in a visual form. Visual aids commonly contain graphics, animation, and video.

Conventions

The following conventions are used in the course materials.

Icons

Additional resources – Indicates other references that provide additional information on the topics described in the module.

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

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

Typographical Conventions

Typeface	Meaning	Example
`AaBbCc123`	command, file, method, class and directory names, computer output	Edit the `.login` file

`AaBbCc123`	What you type and to highlight topic discussions	`%` **`su`**
		
		`1 import java.io.*;`
		`2 import javax.servlet.*;`
		`3 import javax.servlet.http.*;`
		
		`The javax.servlet interface is imported to access life cycle methods (line 2).`

`AaBbCc123`	placeholder, replace with real name	The command to remove a file is `rm` *`filename`*
`AaBbCc123`	What you enter in class activities	Type **`chmod a+rwx`** ***`filename`*** to grant read, write, and execute rights for *`filename`* to world, group, and users.
Palatino italics	book titles, new terms, emphasis	Read Chapter 6 in the *User's Guide*. These are called *class* options.

Module 1

Introducing the Solaris™ 10 OS Directory Hierarchy

Objectives

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

- Describe file systems
- Identify file components
- Identify file types
- Describe hard links

Introducing File Systems

A *file system* is a structure of directories you use to organize and store files. The term *file system* describes the following:

- A particular type of file system: disk-based, network-based, or virtual
- An entire file tree, beginning with the root (/) directory
- The data structure of a disk slice or other media storage device
- A portion of a file tree structure that is attached to a mount point on the main file tree so that the files are accessible

Usually, you know from the context which meaning is intended.

The Solaris™ Operating System (OS) uses the virtual file system (VFS) architecture, which provides a standard interface for different file system types. The VFS architecture enables the kernel to handle basic operations, such as reading, writing, and listing files. The VFS architecture also makes it easier to add new file systems.

Default Solaris File Systems

The Solaris Unix file system (UFS) is hierarchical. It starts with the root directory (/) and continues down through a number of directories. The Solaris installation process enables you to install a default set of directories and use a set of conventions to group similar file types together.

The Solaris OS includes an alternate file system called the Z file system (ZFS). You can use the ZFS with the UFS or as the primary file system. The ZFS uses storage pools to manage physical storage. The ZFS is also a hierarchical file system, starting with the root directory (/). See the *Solaris ZFS Administration Guide* for more information on ZFS.

Introducing / (root) Subdirectories

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

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

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

Introducing Important System Directories

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

Table 1-1 Critical Directories

`/`	The root of the overall file system namespace.
`/bin`	A symbolic link to the `/usr/bin` directory. It is the directory location for the binary files of standard system commands.
`/dev`	The primary directory for logical device names. The contents of this directory are symbolic links that point to device files in the `/devices` directory.
`/etc`	The directory that holds host-specific configuration files and databases for system administration.
`/export`	The default directory for commonly shared file systems, such as users' `home` directories, application software, or other shared file systems.
`/home`	The default directory or mount point for a user's `home` directory.
`/kernel`	The directory of platform-independent loadable `kernel` modules that are required as part of the boot process.
`/lib`	The contents of this directory are shared executable files and Service Management Facility executables.

Table 1-1 Critical Directories (Continued)

/mnt	A convenient, temporary mount point for file systems.
/opt	The default directory or mount point for add-on application packages.
/platform	The directory of platform-dependent loadable kernel modules.
/sbin	The single-user bin directory that contains essential executables that are used during the booting process and in manual system-failure recovery.
/usr	The directory that contains programs, scripts, and libraries that are used by all system users.
/var	The directory for varying files, which usually includes temporary, logging, or status files. Following the introduction of the Service Management Facility in the Solaris 10 OS, the /var directory hierarchy is more heavily used than in previous releases. It is important that the /var directory has sufficient disk space available to store software package information, log files, spool files, and so on.

Introducing Important In-Memory System Directories

Table 1-2 lists of some of the important in-memory system directories and subdirectories that are found in the Solaris 10 OS.

Table 1-2 In-Memory System Directories

/dev/fd	The directory that contains special files relating to current file-descriptors in use by the system.
/devices	The primary directory for physical device names.
/etc/mnttab	A memory-based file, in its own file system, that contains details of current file system mounts.

Table 1-2 In-Memory System Directories (Continued)

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

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

The following tables list primary subdirectories under key directories.

Table 1-3 Primary Subdirectories Under the `/dev` Directory

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

Table 1-4 Primary Subdirectories Under the `/etc` Directory

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

Table 1-4 Primary Subdirectories Under the `/etc` Directory (Continued)

Subdirectory	Description
`/etc/security`	Control files for Role Based Access Control and security privileges
`/etc/skel`	Default shell initialization files for new user accounts
`/etc/svc`	The Service Management Facility database and log files

Table 1-5 Contents of the `/usr` Directory

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

Table 1-6 Primary Subdirectories Under the `/var` Directory

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

Introducing File Components

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

Figure 1-1 illustrates the relationship between Filenames, Inodes, and Data Blocks.

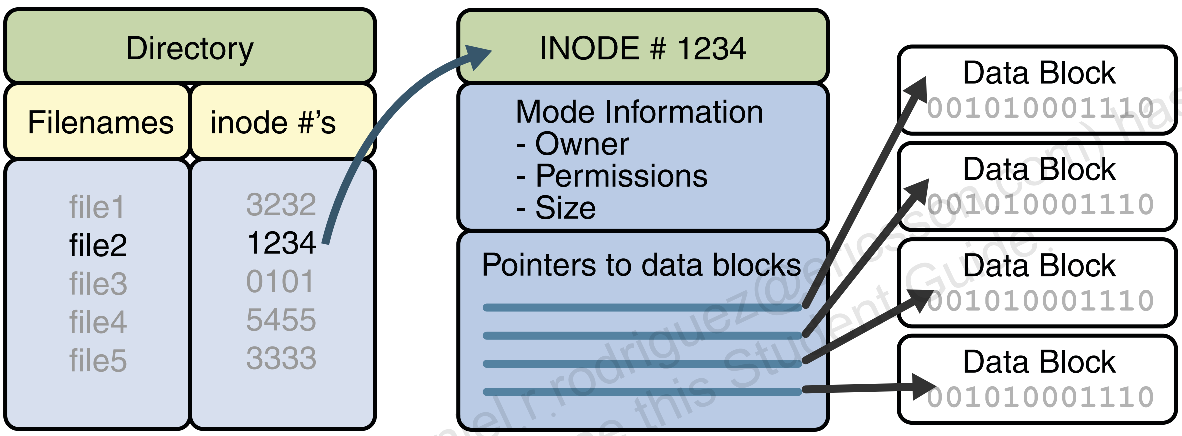


Figure 1-1 Filename, Inode, and Data Block Relationship

File Names

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

Inodes

Inodes are the objects that the Solaris OS uses to record information about a file. In general, inodes contain two parts:

1. *File information*, including the file owner, permissions, and size.
2. *Pointers* to data blocks associated with the file content.

Inodes are numbered, and each file system contains its own inode list. When you create a new UFS, a complete new inode list is also created for that UFS.

Data Blocks

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

ZFS uses the following data structures:

- *Dnode data structure*: this data structure contains pointers to data blocks.
- *Znode data structure*: this data structure contains information about the file, including its owner, permissions, and size.

Dnode and znode data structures are called metadata and are stored dynamically by the ZFS. ZFS implements the *ZFS POSIX layer* (ZPL) which is a primary interface for ZFS. The ZPL allows commands you use with the UFS to work with ZFS. ZFS emulates the UFS inode number.

Identifying File Types

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

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

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

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

The following examples, taken from an Ultra™ 20 and Sun Fire™ V40z server, show partial listings of directories that contain a variety of different file types:

```
# cd /etc
# ls -l
total 618
drwxr-xr-x   2 adm      adm          512 Dec 17 19:08 acct
-rw-r--r--   1 root     sys          253 Nov  9 09:48 aggregation.conf
lrwxrwxrwx   1 root     root          14 Dec 17 19:04 aliases ->
./mail/aliases
drwxr-xr-x   6 root     other        512 Dec 17 19:02 amd64
drwxr-xr-x   7 root     bin          512 Dec 17 19:07 apache
drwxr-xr-x   2 root     bin          512 Dec 17 19:09 apache2
(output truncated)

# cd /devices/pci@0,0/pci-ide@7,1/ide@1
# ls -l
total 2
drwxr-xr-x   2 root     sys          512 Dec 29 19:18 sd@0,0
brw-r-----   1 root     sys       28,  0 Dec 30 18:45 sd@0,0:a
crw-r-----   1 root     sys       28,  0 Dec 30 18:30 sd@0,0:a,raw
(output truncated)
```

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

Table 1-7 Characters in the First Column of `ls -l` Command

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

Directories

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

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

Regular Files

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

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

Symbolic Links

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

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

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

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

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

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

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

```
# ls -l
total 0
-rw-r--r--   1 root     other          0 Oct 26 10:15 file1
#
# ln -s file1 link1
# ls -l
total 16
-rw-r--r--   1 root     other          0 Oct 26 10:15 file1
lrwxrwxrwx   1 root     other          5 Oct 26 10:15 link1 -> file1
#
```

Symbolic links direct read and write operations to the file to which they point. The listing above shows that using `link1` as a command-line argument would cause a command to refer to the file called `file1`.

Figure 1-2 shows a symbolic link file called `link1`. As shown in the following figure, the `link1` file is associated with inode number 3561. The data block for the `link1` file contains the path name (`./file1`) to `file1`.

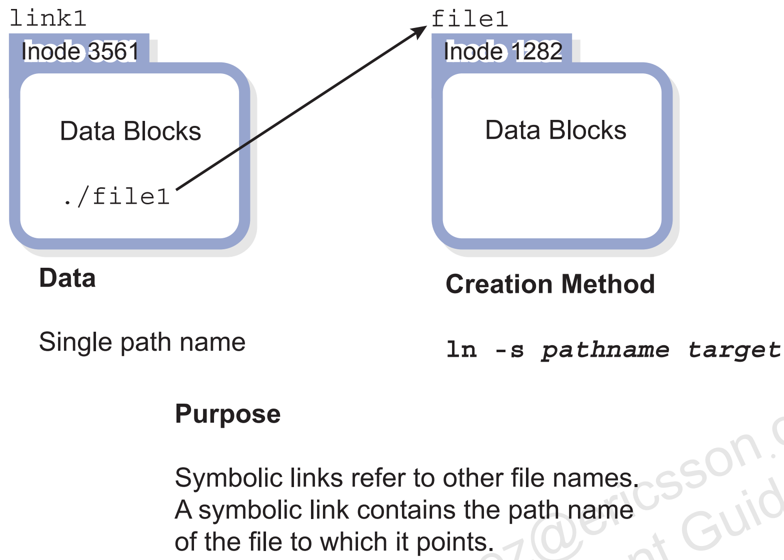


Figure 1-2 Symbolic Links

Device Files

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

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

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

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

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

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

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

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

Figure 1-3 shows the relationship between the device file `dad@0,0:a` and the disk device it controls. The inode information for `dad@0,0:a` contains major number 136 and minor number 8. Major number 136 identifies the `dad` device driver. The `dad` device driver controls integrated device electronics (IDE) disk drives. Minor number 8, in this case, identifies Slice 0.

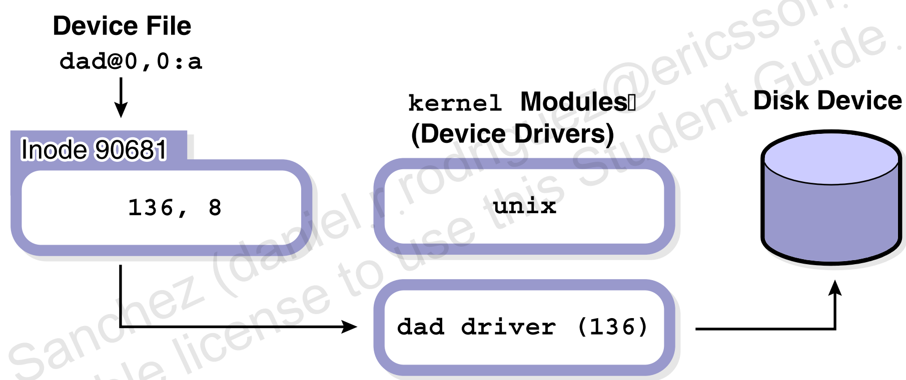


Figure 1-3 Device File Example

You can verify that the device driver is available as a kernel module:

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

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

Character-Special Device Files

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

The following example shows a character-special device file.

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

Block-Special Device Files

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

The following example shows a block-special device file.

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

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

Introducing Hard Links

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

In Figure 1-4, the file called `file1` is listed in the directory `dir1`. In `dir1`, the name `file1` is associated with inode number 1282. The hard link is the association between the name `file1`, and inode number 1282.

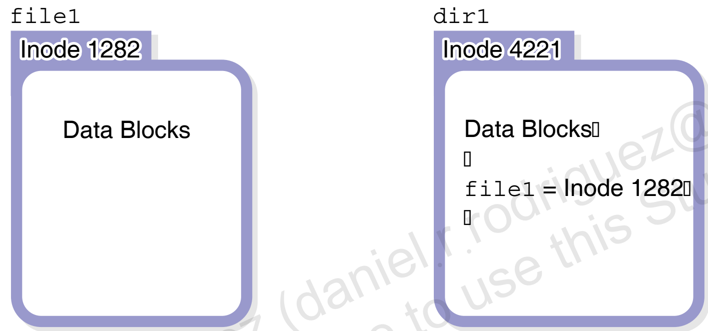


Figure 1-4 Hard Link

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

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

Creating New Hard Links

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

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

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

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

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

Figure 1-5 shows the result of the ln command. In `dir1`, two file names are associated with inode number 1282. Unlike symbolic links, hard links cannot span file systems.

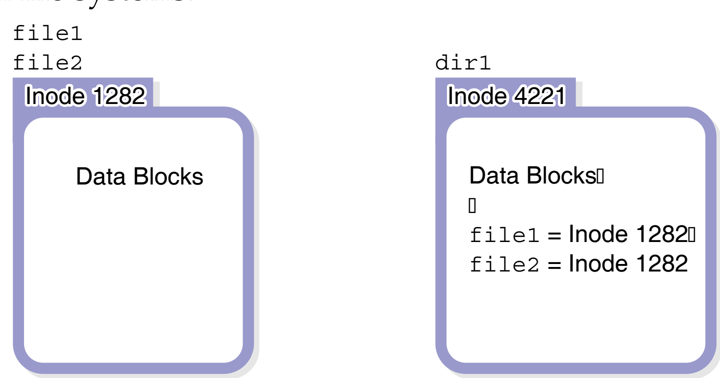


Figure 1-5 File Names Associated With an Inode Number

Removing Hard Links

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

The following example shows how deleting `file1` from the previous example has no effect on `file2`, except to decrement its link count.

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

Exercise: Identifying File Types

In this exercise, you complete the following tasks:

- Use basic Solaris commands to identify and work with the four major file types in the Solaris OS.

Preparation

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

Task

Complete the following steps:

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

   ____

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

   ____

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

   ____

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

   ____

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

   ____

   ____

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

   ____

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

   ____

   Find the number of files in the `/etc` directory or below that have the same inode number as that used by the `/etc/init.d/apache` file.

   ____

8. Create a new directory called `/testdir`. Create a file in this directory called `file1`. Create a symbolic link called `link1` that points to `file1`.
9. List `file1` and the `link1` symbolic link. Do these files use the same or different inodes?

   ____

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

    ______________________________.

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

    ____

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

    ____

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

    ____

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Identifying File Types

This section provides task steps and their solutions.

Task

Complete the following steps:

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

```
# cd /
# ls -l
total 599
lrwxrwxrwx   1 root     root           9 Apr 27
17:51 bin -> ./usr/bin/
drwxr-xr-x   3 root     sys          512 Apr 27
17:52 boot/
drwxr-xr-x   2 root     root         512 Apr 28
08:36 Desktop/
drwxr-xr-x  18 root     sys         5120 Apr 27
18:43 dev/
drwxr-xr-x   2 root     sys          512 Apr 27
18:39 devices/
drwxr-xr-x   2 root     root         512 Apr 28
08:36 Documents/
drwxr-xr-x  87 root     other       4608 Apr 27
18:45 etc/
drwxr-xr-x   2 root     sys          512 Apr 27
18:03 export/
dr-xr-xr-x   1 root     root           1 Apr 27
18:43 home/
drwxr-xr-x  15 root     sys          512 Apr 27
18:16 kernel/
```

   In the italicized text, "The /bin symbolic link ... directory.", remove '/' from '/bin'.

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

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

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

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

```
# cd /dev/dsk
# ls -l
total 126
lrwxrwxrwx   1 root     root          46 Dec 17 19:16 c0t0d0p0 ->
../../devices/pci@0,0/pci-ide@6/ide@0/sd@0,0:q
lrwxrwxrwx   1 root     root          46 Dec 17 19:16 c0t0d0p1 ->
../../devices/pci@0,0/pci-ide@6/ide@0/sd@0,0:r
lrwxrwxrwx   1 root     root          46 Dec 17 19:16 c0t0d0p2 ->
../../devices/pci@0,0/pci-ide@6/ide@0/sd@0,0:s
...
```

The /dev/dsk *directory contains symbolic links.*

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

```
# ls -lL
total 0
brw-r-----   1 root     sys       33, 16 Dec 31 15:33 c0t0d0p0
brw-r-----   1 root     sys       33, 17 Dec 31 15:33 c0t0d0p1
brw-r-----   1 root     sys       33, 18 Dec 31 15:33 c0t0d0p2
...
```

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

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

```
# cd /dev/pts
# ls -l
total 66
lrwxrwxrwx   1 root     root          28 Dec 17 19:16 0 ->
../../devices/pseudo/pts@0:0
lrwxrwxrwx   1 root     root          28 Dec 17 19:16 1 ->
../../devices/pseudo/pts@0:1
lrwxrwxrwx   1 root     root          29 Dec 17 19:23 10 ->
../../devices/pseudo/pts@0:10
...
# ls -lL
total 0
crw--w----   1 root     tty       24,  0 Dec 18 07:42 0
crw--w----   1 root     tty       24,  1 Dec 17 20:49 1
crw-r--r--   1 root     sys       24, 10 Dec 31 16:36 10
...
```

The /dev/pts *directory contains symbolic links.*

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

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

```
# cd /etc/init.d ; ls -l
total 128
-rwxr--r--   1 root     sys          625 Nov  9 09:48 acct
-rwxr--r--   1 root     sys          397 Nov  9 09:48 acctadm
-rwxr--r--   6 root     sys         1649 Sep  5 08:02 apache
-rwxr--r--   5 root     sys          824 Aug 16 09:47 appserv
...
```

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

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

```
# ls -li apache
     10142 -rwxr--r--   6 root     sys         1649 Sep  5 08:02 apache
#
```

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

8. Find the number of files in the /etc directory or below that have the same inode number as that used by the /etc/init.d/apache file. In this example, the inode number is 10142.

```
# ls -i /etc/init.d/apache
     10142 /etc/init.d/apache
# find /etc -inum 10142 -exec ls -i {} \;
     10142 /etc/init.d/apache
     10142 /etc/rc0.d/K16apache
     10142 /etc/rc1.d/K16apache
     10142 /etc/rc2.d/K16apache
     10142 /etc/rc3.d/S50apache
     10142 /etc/rcS.d/K16apache
#
```

Six files, including /etc/init.d/apache*, use the same inode number.*

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

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

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

```
# ls -li
total 2
    187998 -rw-r--r--   1 root     root            0 Dec 31 16:40 file1
    188110 lrwxrwxrwx   1 root     root            5 Dec 31 16:40 link1 ->
file1
#
```

These two files use two different inodes.

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

```
# mkdir newdir
# ls -ldi newdir
    188152 drwxr-xr-x   2 root     root          512 Dec 31 16:43 newdir
#
```

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

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

```
# ls -lia newdir
total 4
    188152 drwxr-xr-x   2 root     root          512 Dec 31 16:43 .
    187997 drwxr-xr-x   3 root     root          512 Dec 31 16:43 ..
#
```

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

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

```
# mkdir newdir/dir2
# ls -ldi newdir
    188152 drwxr-xr-x   3 root     root          512 Dec 31 16:45 newdir
#
```

The link count increases from two to three.

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

```
# ls -laRi newdir
newdir:
total 6
    188152 drwxr-xr-x   3 root     root          512 Dec 31 16:45 .
    187997 drwxr-xr-x   3 root     root          512 Dec 31 16:43 ..
    188154 drwxr-xr-x   2 root     root          512 Dec 31 16:45 dir2

newdir/dir2:
total 4
    188154 drwxr-xr-x   2 root     root          512 Dec 31 16:45 .
    188152 drwxr-xr-x   3 root     root          512 Dec 31 16:45 ..
#
```

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

___

___

___

Module 2

Managing Local Disk Devices

Objectives

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

- Describe the basic architecture of a disk
- Manage disk labels
- Describe the naming conventions for devices
- Describe support for iSCSI target devices
- List system devices
- Reconfigure devices
- Perform hard disk partitioning
- Describe EFI disk labels

Introducing the Basic Architecture of a Disk

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

Physical Disk Structure

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

Figure 2-1 identifies the parts of a disk.

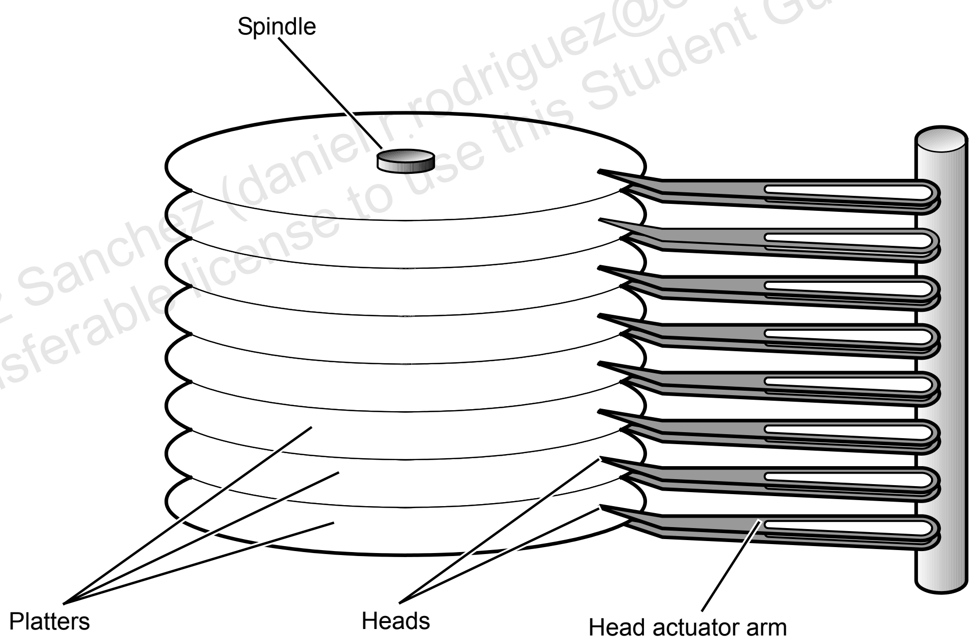


Figure 2-1 Components of a Disk

The following list describes the physical components of a disk:

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

Data Organization on Disk Platters

Figure 2-2 shows the logical components of a disk platter.

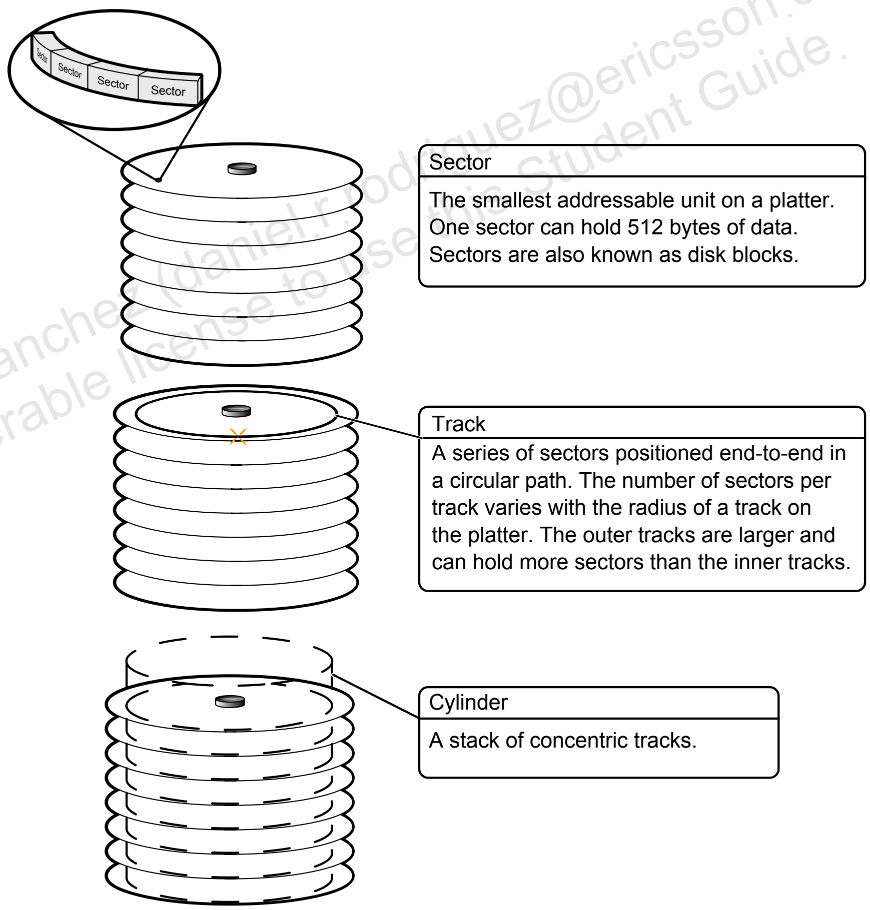


Figure 2-2 Data Organization on Disk Platters

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

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

The number of sectors per track varies with the radius of a track on the platter. The outermost tracks are larger and can hold more sectors than the inner tracks. Disks present a fixed number of sectors per track to the Solaris OS, even though physically the number of sectors per track varies by track location.Because a disk spins continuously and the read/write heads move as a single unit, the most efficient seeking occurs when the sectors to be read from or written to are located in a single cylinder.

Disk Labels and Partition Tables

Disk slices are groups of cylinders that are commonly used to organize data by function. For example, one slice can store critical system files and programs while another slice on the same disk can store user-created files. Grouping cylinders into slices is done to organize data, facilitate backups, and provide swap space. A starting cylinder and an ending cylinder define each slice. These cylinder boundaries determine the size of a slice.

An area of every disk is set aside for storing information about the disk controller, geometry, and slices. This information is called the disk label. Partition, cylinder, and slice information is stored in partition tables within disk labels. SPARC®-based systems maintain one partition table on each disk. To label a disk means to write slice information onto the disk. You usually label a disk after you change its slices.

The Solaris OS supports the following two disk labels:

- The Sun Microsystems, Inc. (SMI) disk label:
  - Is typically used in SPARC-based Solaris OSs.
  - Is the SPARC volume table of contents (VTOC) label for disks and is often called the SMI VTOC disk label.
  - Provides support for disks that are less than 1 terabyte.
  - Occupies the first sector of a disk in SPARC systems

- Includes a partition table in which you can define up to eight (0 through 7) disk partitions (slices). A starting cylinder and an ending cylinder define each slice. Whole-cylinder boundaries determine the sizes of a slices. Slice 2 represents the entire disk. Slice 2 maintains important data about the entire disk, such as the size of the actual disk and the total number of cylinders available for the storage of files and directories.
- Must be used for SPARC-based and Solaris x86/x64-based systems boot disks.
- Solaris OS systems for the x86/x64 platforms maintain two partition tables on each disk. The first sector of disks on x86/x64 systems contains a fixed disk (`fdisk`) partition table. The second sector of the Solaris `fdisk` partition on an x86/x64 system holds the partition table that defines slices within the Solaris `fdisk` partition. The label that holds this table is known as the x86/x64 VTOC.
- The `fdisk` partition table defines up to four `fdisk` partitions. One of these `fdisk` partitions may be used for the Solaris OS. Provision is made for up to sixteen slices within a Solaris `fdisk` partition, but generally only ten of these are used (eight, plus two used for platform-specific purposes).

- The extensible firmware interface (EFI) disk label:
  - Provides support for disks that are larger than 1 terabyte on systems that run a 64-bit Solaris kernel.
  - Provides support for virtual disk volumes.
  - Includes a partition table in which you can define up to ten (0 through 9) disk partitions (slices).
  - Is compatible with the UFS. You can create a UFS that is larger than 1 terabyte.
  - Solaris OS systems do not currently boot from disks that use EFI labels.

Use either SMI or EFI labels if a disk is not being used as a boot disk. Use either the `format` or `prtvtoc` command to check whether a disk has an SMI VTOC or EFI label.

Figure 2-3 shows how disk slices might reside on a disk.

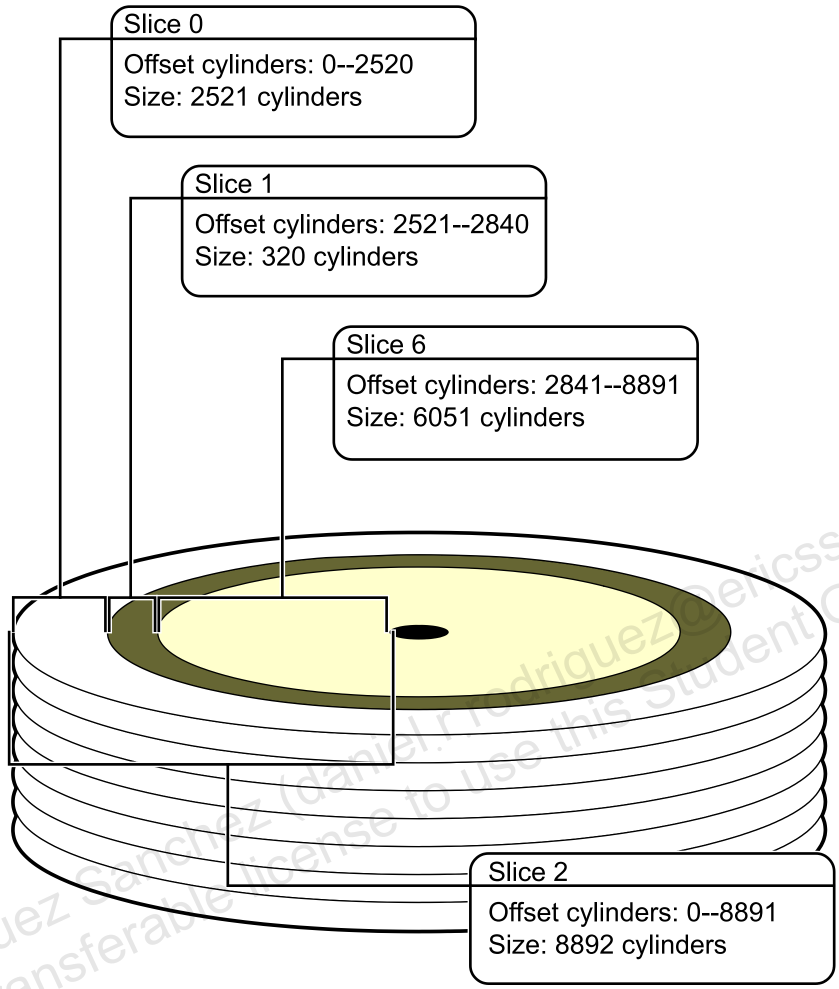


Figure 2-3 Cylinders and Slices

Table 2-1 shows disk slices and the different file systems they could hold.

Table 2-1 Disk Slices

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

Table 2-1 Disk Slices (Continued)

Slice	Name	Function
7	/export/home	User files and directories

Figure 2-4 shows a possible configuration convention for organizing data. The example disk is divided into slices that logically organize the data on the boot disk.

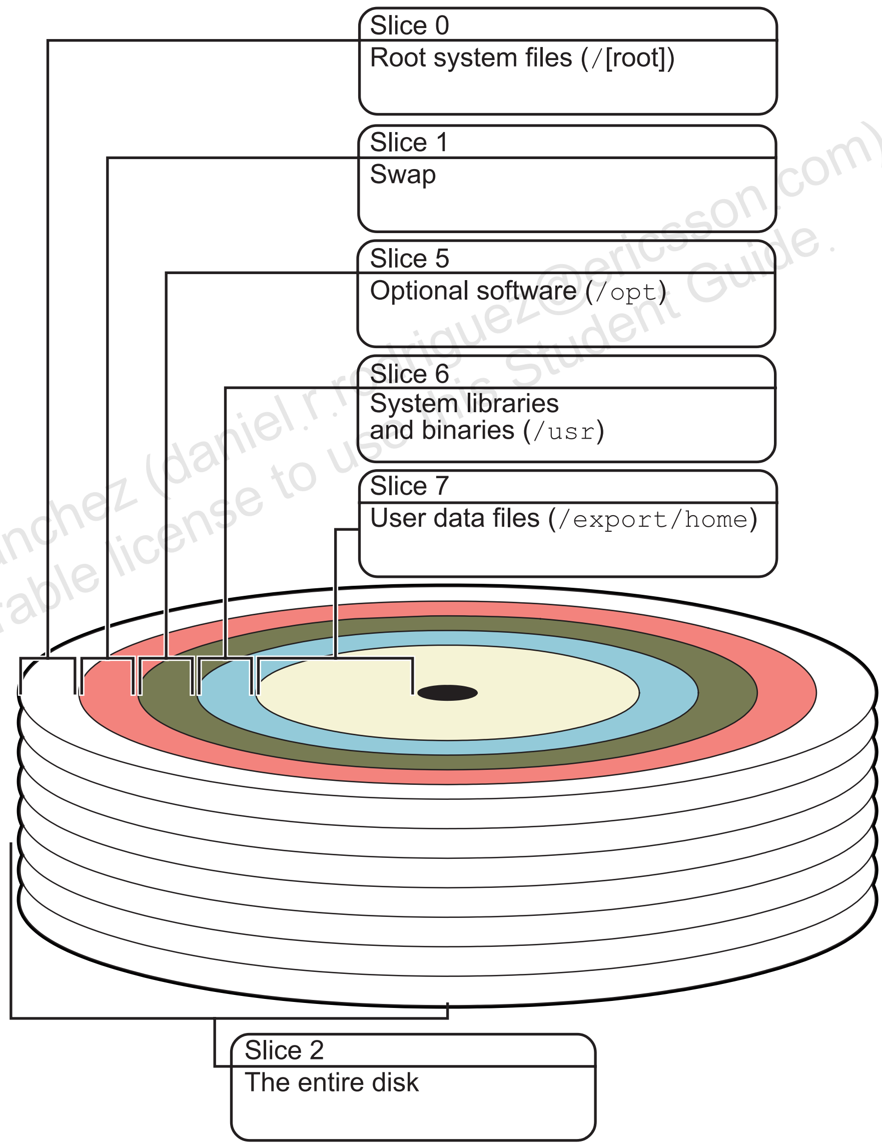


Figure 2-4 Top View of Five Configured Disk Slices

x86/x64 Partitions and Slices

On x86/x64 platform Solaris OS systems, disks typically use SMI labels. The boot disk must use an SMI label. Disks on x86/x64 systems maintain an `fdisk` partition table and a partition table for the Solaris `fdisk` partition.

Systems based on x86/x64 platforms use the `fdisk` partition table to identify parts of the disk reserved for different operating systems, and to identify the partition from which the system will boot. Systems boot from the `fdisk` partition that you designate the *active* `fdisk` partition. The sections that follow assume that the `fdisk` partition that holds the Solaris OS is the active `fdisk` partition.

The `fdisk` partition table defines up to four `fdisk` partitions. Disks on x86/x64 systems require at least one defined `fdisk` partition. You can assign only one `fdisk` partition on a disk to be used for Solaris.

The `fdisk` menu in the `format(1M)` utility is typically used to create or modify the `fdisk` partition table. The `fdisk(1M)` command also allows you to directly manipulate the `fdisk` partition table. It is recommended that you use the `fdisk` menu in the `format` utility to create or modify the `fdisk` partition table.

The second sector of the Solaris `fdisk` partition contains a partition table that defines slices within the Solaris `fdisk` partition. The label that contains this partition table is known as the x86/x64 VTOC.

Using the `format` utility, a Solaris `fdisk` partition can be divided into ten slices, labeled slice 0 through slice 9. Slices 0 through 7 are used for the same purposes as disk slices found on disks that use SPARC SMI labels. Slice 2 represents all of the space within the Solaris `fdisk` partition. Slices 8 and 9 are used for purposes specific to x86/x64 hardware.

The `format` utility automatically creates slices 8 and 9 when you define an x86/x64 VTOC within the Solaris `fdisk` partition. Slice 8 is always required. Slice 9 will exist, but remains unassigned, on SCSI or Fibre-Channel disks. The `format` utility prevents modifying slices 8 and 9.

Solaris OS systems on the x86/x64 platform create device files to support sixteen slices for each disk, numbered 0 through 15. The `format` utility, however, limits its operation to slices 0 through 7. Solaris x86/x64 systems also create device files that represent the four possible `fdisk` partitions.

Figure 2-5 describes a set of slices within a Solaris `fdisk` partition. For all slices marked with shaded title bars, cylinder values are expressed relative to the first cylinder in the Solaris `fdisk` partition. The three objects marked with unshaded title bars describe cylinder values relative to the first cylinder of the entire disk.

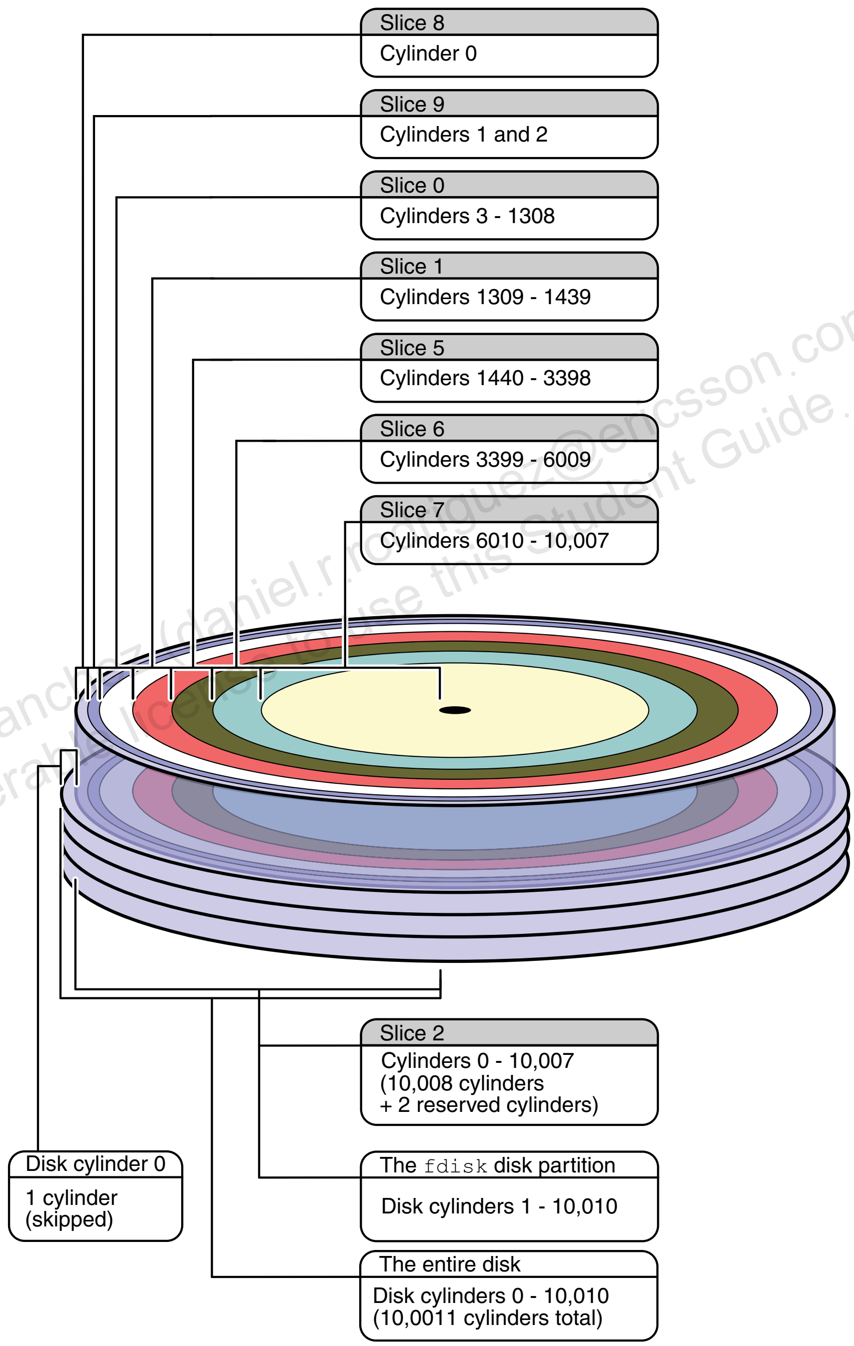


Figure 2-5 Solaris OS Slices on a Disk with a Single `fdisk` Partition

By default, slice 8 is the boot slice, and contains the GRUB `stage1` program in sector 0, the Solaris disk label and VTOC in sectors 1 and 2, and GRUB `stage2` program beginning at sector 50. Slice 8 occupies the first cylinder (cylinder 0) of the Solaris `fdisk` partition.

By convention on IDE and SATA disk drives, Slice 9 is tagged as the alternates slice. If it is defined, slice 9 occupies the second and third cylinders (cylinders 1 and 2) of the Solaris `fdisk` partition, and contains blocks used to store bad block information.

For an IDE or SATA disk attached to an x86/x64 system, the partition table you can display using the `format` utility shows that slices 8 and 9 are both defined, and that they use one and two cylinders respectively. For example:

```
Part      Tag    Flag     Cylinders        Size            Blocks
(output omitted)
8        boot    wu       0 -    0         7.84MB    (1/0/0)       16065
9 alternates    wm       1 -    2        15.69MB    (2/0/0)       32130
```

For a SCSI disk attached to an x86/x64 system, the partition table shows that slice 8 is defined and uses one cylinder (cylinder 0), but slice 9 has no space assigned. For example:

```
Part      Tag    Flag     Cylinders        Size            Blocks
(output omitted)
8        boot    wu       0 -    0         7.84MB    (1/0/0)       16065
9 unassigned    wm       0               0         (0/0/0)           0
```

Disk Device Naming Conventions

Solaris OS systems use a standard naming convention to identify disk devices. Disk device names identify a series of components that together specify a particular disk device. The disk device name includes the controller, target, disk (or LUN), `fdisk` partition, and slice that a particular disk device uses.

Examples of disk device names include the following:

- `c1t0d0s0` – A SCSI disk device name, that specifies controller 1, target 0, disk 0, and slice 0
- `c1d0p0` – An IDE or SATA disk name on an x86/x64 system that specifies controller 1, disk 0, and `fdisk` partition 0

- `c1d0s0` – An IDE or SATA disk name that specifies controller 1, disk 0, and slice 0
- `c5t12d0p0` – A SCSI disk device name on an x86/x64 system, that specifies controller 5, target 12, disk 0, and `fdisk` partition 0
- `c5t12d0s0` – A SCSI disk device name, that specifies controller 5, target 12, disk 0, and slice 0
- `c3t266000C0FFF7C140d31s2` – A Fibre-Channel attached LUN name that specifies controller 3, WWN 266000C0FFF7C140, LUN 31, and slice 2

Disk device names represent the full name of a slice or `fdisk` partition. These names identify the following components:

Controller number	Identifies the host bus adapter (HBA), which controls communication between the system and disk unit. The HBA takes care of transferring both commands and data to and from the device. Controller numbers are assigned in sequential order, such as `c0`, `c1`, `c2`, and so on.
Target number	Target numbers, such as t0, t1, t2, and t3, identify a unique hardware address assigned to the SCSI target controller of a disk, tape, or CD-ROM. Some external disk drives have an address switch located on the rear panel. Some internal disks have address pins that are jumpered to assign that disk's target number. Names for IDE or SATA disks typically do not use target numbers, although on some systems they do. Fibre-Channel attached disks may use a World Wide Name (WWN) instead of a target number.
Disk number	The disk number is also known as the logical unit number (LUN). This number may vary from `d0` if more than one disk can be attached at the target location.
`fdisk` Partition number	The `fdisk` partition number ranging from `p0` to `p4`. The number `p0` represents the whole physical disk. Device names that include `fdisk` partition IDs are found only on x86/x64 systems.
Slice number	A slice number ranging from `s0` to `s7` on SPARC systems, and from `s0` to `s15` on x86/x64 systems.

SCSI Disk Device Names

SCSI specifications allow multiple disks to be attached to a single target controller. In many cases however, a SCSI disk and target controller are combined into one physical unit. SCSI disks that are constructed this way are described as using an *embedded* SCSI configuration. The disk number is always set to d0 with embedded SCSI disks.

Figure 2-6 shows the string that represents the full name of a SCSI disk slice.

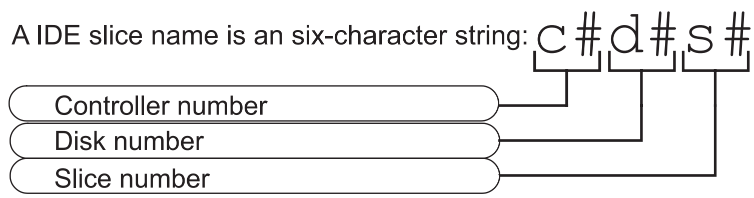


Figure 2-6 SCSI Disk Slice Naming Convention

Figure 2-7 shows the embedded SCSI configuration.

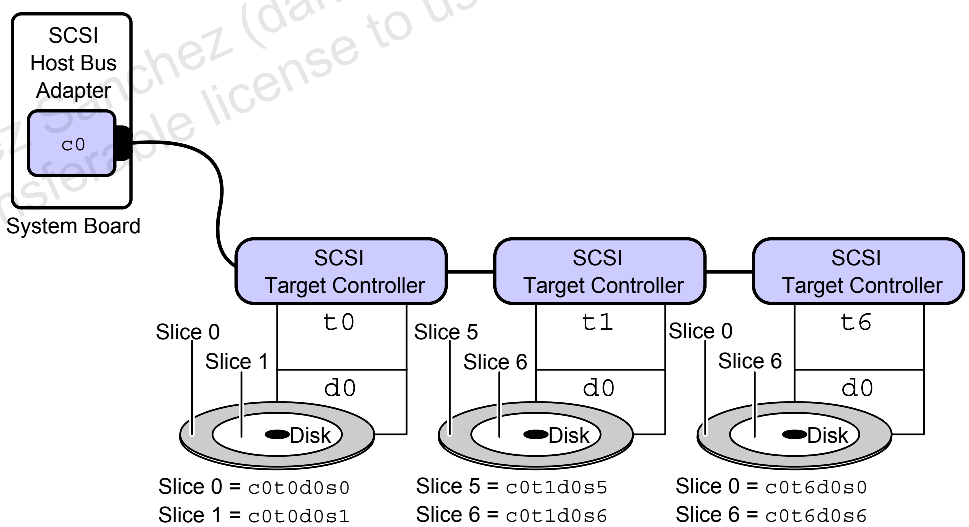


Figure 2-7 Embedded SCSI Configuration

Some external SCSI devices have multiple disks attached to a single target controller. Solaris OS systems use disk numbers to represent these disks, and so may present disk numbers other than d0 (such as d1, d2) in their device names.

Disk arrays that are capable of hardware RAID configurations typically present multiple LUNs for the target values they present to the Solaris OS. LUNs from arrays like these are also represented by `d0` - `d`*n* disk numbers in their device names.

iSCSI Target Devices

The internet small computer system interface (iSCSI) is an internet protocol (IP) based storage networking standard for linking data storage subsystems. By carrying SCSI commands over IP networks, the iSCSI protocol enables you to access block devices from across the network as if they were connected to the local system.

The Solaris 10 8/07 release provides support for iSCSI target devices. These target devices can be disk or tape. Releases prior to Solaris 10 8/07 provided support for iSCSI initiators. The advantage of setting up Solaris iSCSI targets is that you might have existing fibre-channel devices that can be connected to clients without the cost of fibre-channel host bus adapters (HBAs). In addition, systems with dedicated arrays can now export replicated storage with the ZFS or the UFS.

Setting Up iSCSI Target Devices

You can use the `iscsitadm` command to set up and manage your iSCSI target devices. For the disk device that you select as your iSCSI target, you'll need to provide an equivalently sized ZFS or UFS as the backing store for the iSCSI daemon. After the target device is set up, use the `iscsiadm` command to identify your iSCSI targets, which will discover and use the iSCSI target device.

Advantages of Using iSCSI Targets and Initiators

The benefits of using Solaris iSCSI targets and initiators are:

- The iSCSI protocol runs across existing Ethernet networks.
  - You can use any supported network interface card (NIC), Ethernet hub or switch.
  - One IP port can handle multiple iSCSI target devices.
  - You can use existing infrastructure and management tools for IP networks.

- You might have existing fibre-channel devices that can be connected to clients without the cost of fibre-channel HBAs. In addition, systems with dedicated arrays can now export replicated storage with ZFS or UFS file systems.
- There is no upper limit on the maximum number of configured iSCSI target devices.
- The protocol can be used to connect to Fibre Channel or iSCSI Storage Area Network (SAN) environments with the appropriate hardware.

IDE and SATA Disk Device Names

Integrated device electronics (IDE) and serial advanced technology attachment (SATA) disks do not use target controllers. Device names for these disks consequently represent the controller, disk, and slice or `fdisk` partition used by the IDE or SATA drive.

Figure 2-8 illustrates the IDE and SATA disk slice naming conventions.

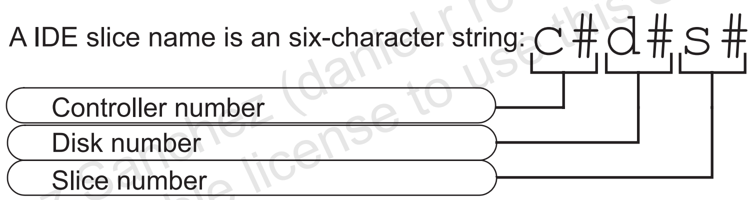


Figure 2-8 IDE and SATA Disk Slice Naming Conventions

For example, Ultra 20 systems support two internal SATA disks, where each disk is attached to a separate IDE controller.

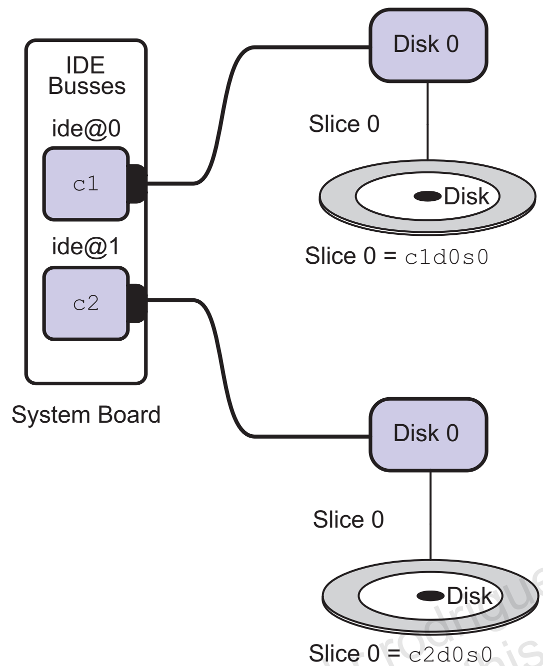
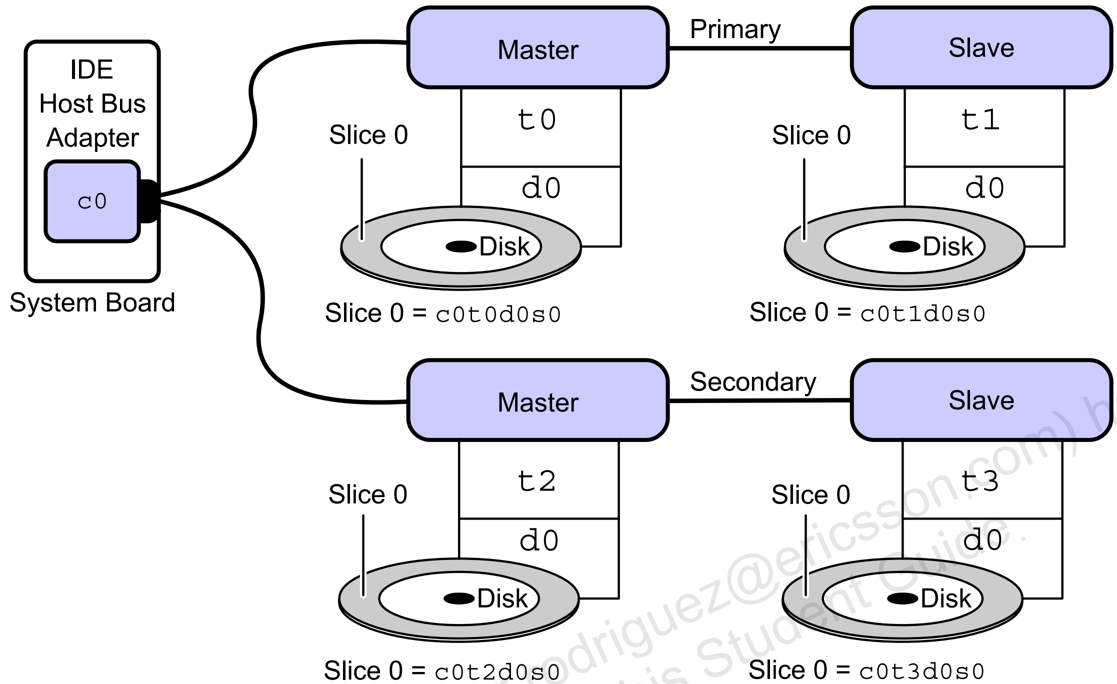


Figure 2-9 Ultra 20 IDE Configuration

Ultra 10 systems present a different example. Although IDE disks do not use target controllers, Ultra 10 systems use a target (`t`*n*) value to represent the identity of the disks on its primary and secondary IDE busses. Target values on these systems denote the following:

- `t0` - master device on primary IDE bus
- `t1` - slave device on primary IDE bus
- `t2` - master device on secondary IDE bus
- `t3` - slave device on secondary IDE bus

Figure 2-10 shows the configuration of the IDE architecture on an Ultra



10.

Figure 2-10 Ultra 10 IDE Configuration

Introducing Solaris OS Device Naming Conventions

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

- Logical device names
- Physical device names
- Instance names

Logical Device Names

Logical disk device names are symbolic links to the physical device names in the /devices directory. Logical device names are used primarily to specify a device when you enter commands on the command line. All logical device names are kept in the /dev directory. Logical device names contain the controller number, target number if required, disk number, fdisk partition number or slice number.

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

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

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

On an x86/x64 system, the list of logical device names for a disk will include names for `fdisk` partitions, and names for sixteen slices. For example:

```
# ls c1d0*
c1d0p0   c1d0p3   c1d0s1    c1d0s12  c1d0s15  c1d0s4   c1d0s7
c1d0p1   c1d0p4   c1d0s10   c1d0s13  c1d0s2   c1d0s5   c1d0s8
c1d0p2   c1d0s0   c1d0s11   c1d0s14  c1d0s3   c1d0s6   c1d0s9
#
```

- `c1d0p0` through `c1d0p4` – Identify the device names for `fdisk` partitions 0 through 4, for a disk that is attached to Controller 1, as Disk Unit 0.
- `c1d0s0` through `c1d0s15` – Identify the device names for slices 0 through 14, for a disk that is attached to Controller 1, as Disk Unit 0.

On an x86/x64 system, the list of logical device names for a LUN from a Fibre-Channel attached storage device will include names for `fdisk` partitions, and names for sixteen slices. For example:

```
# ls c3t266000C0FFF7C140d31*
c3t266000C0FFF7C140d31p0   c3t266000C0FFF7C140d31s10
c3t266000C0FFF7C140d31s3
c3t266000C0FFF7C140d31p1   c3t266000C0FFF7C140d31s11
c3t266000C0FFF7C140d31s4
c3t266000C0FFF7C140d31p2   c3t266000C0FFF7C140d31s12
c3t266000C0FFF7C140d31s5
c3t266000C0FFF7C140d31p3   c3t266000C0FFF7C140d31s13
c3t266000C0FFF7C140d31s6
c3t266000C0FFF7C140d31p4   c3t266000C0FFF7C140d31s14
c3t266000C0FFF7C140d31s7
c3t266000C0FFF7C140d31s0   c3t266000C0FFF7C140d31s15
c3t266000C0FFF7C140d31s8
c3t266000C0FFF7C140d31s1   c3t266000C0FFF7C140d31s2
c3t266000C0FFF7C140d31s9
#
```

- `c3t266000C0FFF7C140d31p0` through `c3t266000C0FFF7C140d31p4` – Identify the device names for `fdisk` partitions 0 through 4, for a LUN that is attached to Controller 3, through an array's Fibre-Channel port `266000C0FFF7C140` as LUN 31.
- `c3t266000C0FFF7C140d31s0` through `c3t266000C0FFF7C140d31s15` – Identify the device names for slices 0 through 15, for a LUN that is attached to Controller 3, through an array's Fibre-Channel port `266000C0FFF7C140` as LUN 31.

Physical Device Names

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

A physical device name contains the hardware information, represented as a series of node names, separated by slashes, that indicate the path through the system's physical device tree to the device. Listing the symbolic links that exist in the `/dev/dsk` and `/dev/rdsk` directories displays the names of the physical device files they point to.

To display a physical device name, use the ls -l command to list a logical device name. This example is from a Sun Blade™ 1500 workstation:

```
# ls -l /dev/dsk/c0t0d0s0
lrwxrwxrwx   1 root     root          43 Jan 23 16:06 /dev/dsk/c0t0d0s0 -
> ../../devices/pci@1e,600000/ide@d/dad@0,0:a
#
```

On an Ultra 20, listing the logical device `/dev/dsk/c1d0s0` displays the following physical device name information:

```
# ls -l /dev/dsk/c1d0s0
lrwxrwxrwx   1 root     root          48 Dec 17 19:16 /dev/dsk/c1d0s0 ->
../../devices/pci@0,0/pci-ide@7/ide@0/cmdk@0,0:a
#
```

Adding the `-L` option to the `ls -l` command displays the file information for the physical device file referenced by the logical device file. For example:

```
# ls -lL /dev/dsk/c1d0s0
brw-r-----   1 root     sys      102,  0 Dec 17 20:49 /dev/dsk/c1d0s0
#
```

Some systems use internal Fibre Channel Arbitrated Loop (FC-AL) disks. Their physical device names will appear slightly different because they include a World Wide Name (WWN). The following example was taken from a Sun™ Enterprise 3500 server:

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

In this example, the string `21000020375b9ab6` is the WWN.

Physical device names for external Fibre-Channel attached storage also display a World Wide Name (WWN). The following example was taken from a Sun™ V20z server attached to a Sun 3510 Fibre-Channel array:

```
# ls -l /dev/rdsk/c3t266000C0FFF7C140d31s2
lrwxrwxrwx   1 root     root          89 Jan 15 17:51
/dev/rdsk/c3t266000C0FFF7C140d31s2 ->
../../devices/pci@0,0/pci1022,7450@a/pci1077,10a@5/fp@0,0/disk@w266000c0f
ff7c140,1f:c,raw
#
```

In this example, the string `266000c0fff7c140` is the WWN. These physical device name examples describe the device locations in a system device tree.

Figure 2-11 shows a partial device tree from an Ultra 20 workstation.

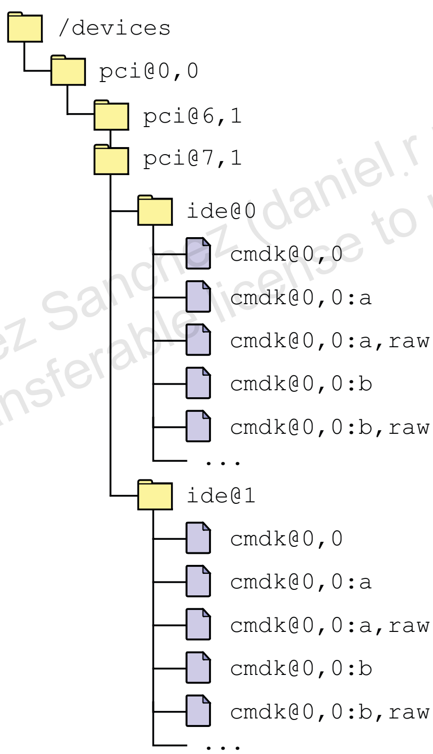


Figure 2-11 The `/devices` Directory Structure

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

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

Note – Different hardware platforms may have different device trees.

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

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

During system initialization, the `kernel` identifies the physical location of a device. The `kernel` associates a node with an address, *`nodename@address`*, which is the physical device name. In Figure 2-11, cmdk@0 is the common disk driver disk device at address 0. One of these disk device instances is attached to both ide@0 and ide@1 IDE busses.

Instance Names

Instance names are abbreviated names assigned by the `kernel` for each system device. Device-related messages that display on the system console or that are typically saved in the `/var/adm/messages` log file often use instance names to identify devices. An instance name is a shortened name for the physical device name. Four instance name examples are shown:

- `sd`*`n`*

  where `sd` (SCSI disk) is the disk name and *`n`* is the number, such as `sd0` for the first SCSI disk device.

- `cmdk`*`n`*

  where `cmdk` (common disk driver) is the disk name, and *`n`* is the number, such as `cmdk0` for the first SATA disk device in an Ultra 20

- `dad`*`n`*

  where `dad` (direct access device) is the disk name and *`n`* is the number, such as `dad0` for the first IDE disk device.

- `ata`*`n`*

  where `ata` (Advanced Technology Attachment), also known as Integrated Drive Electronics (IDE), is the disk name and *`n`* is the number, such as `ata0` for the first IDE disk device

Listing System Devices

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

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

The /etc/path_to_inst File

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

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

The following example from an Ultra 20 shows entries in the /etc/path_to_inst file. The text within the parentheses indicates what device the entry identifies, and does not appear in the actual file.

```
# cat /etc/path_to_inst
#
#       Caution! This file contains critical kernel state
#
"/pseudo" 0 "pseudo"
"/options" 0 "options"
"/xsvc" 0 "xsvc"
"/objmgr" 0 "objmgr"
"/scsi_vhci" 0 "scsi_vhci"
"/isa" 0 "isa"
"/isa/asy@1,3f8" 0 "asy"
"/ramdisk" 0 "ramdisk"
"/pci@0,0" 0 "npe"
"/pci@0,0/pci108e,5347@a" 0 "nge"
"/pci@0,0/pci108e,5347@2" 0 "ohci"
"/pci@0,0/pci108e,5347@2/mouse@1" 0 "hid"
"/pci@0,0/pci108e,5347@2/hub@2" 0 "hubd"
"/pci@0,0/pci108e,5347@2/hub@2/keyboard@4" 1 "hid"
"/pci@0,0/pci108e,5347@2/hub@1" 1 "hubd"
"/pci@0,0/pci108e,5347@2/hub@1/keyboard@4" 2 "hid"
```

```
"/pci@0,0/pci108e,5347@2/mouse@2" 3 "hid"
"/pci@0,0/pci108e,5347@2,1" 0 "ehci"
"/pci@0,0/pci108e,5347@4" 0 "audio810"
"/pci@0,0/pci10de,5d@e" 3 "pcie_pci"
"/pci@0,0/pci10de,5d@e/display@0" 0 "vgatext"
"/pci@0,0/pci-ide@6" 0 "pci-ide"
"/pci@0,0/pci-ide@6/ide@0" 0 "ata"
"/pci@0,0/pci-ide@6/ide@0/sd@0,0" 0 "sd" (DVD/CDROM)
"/pci@0,0/pci-ide@7" 1 "pci-ide"
"/pci@0,0/pci-ide@7/ide@0" 2 "ata"
"/pci@0,0/pci-ide@7/ide@0/cmdk@0,0" 0 "cmdk" (Disk 0)
"/pci@0,0/pci-ide@7/ide@1" 3 "ata"
"/pci@0,0/pci-ide@7/ide@1/cmdk@0,0" 1 "cmdk" (Disk 1)
"/pci@0,0/pci-ide@8" 2 "pci-ide"
"/pci@0,0/pci10de,5c@9" 0 "pci_pci"
"/pci@0,0/pci10de,5c@9/pci1106,3044@6" 0 "hci1394"
"/pci@0,0/pci1022,1101@18,1" 0 "mc-amd"
"/pci@0,0/pci1022,1102@18,2" 1 "mc-amd"
"/pci@0,0/pci1022,1103@18,3" 0 "amd64_gart"
"/iscsi" 0 "iscsi"
"/agpgart" 0 "agpgart"
#
```

Instance names correlate to values in the last two fields in `/etc/path_to_inst` entries. For example, the entry for disk 0 on an Ultra 20 contains this information:

```
"/pci@0,0/pci-ide@7/ide@0/cmdk@0,0" 0 "cmdk"
```

This line and the instance name `cmdk0` identify the same device. The field values in bold match the disk name and number in the instance name. The physical device path `/pci@0,0/pci-ide@7/ide@0/cmdk@0,0` identifies the path through the device tree to the device.

Different systems have different physical device paths. Entries in the `/etc/path_to_inst` file reflect those differences. This partial example is from a Sun Blade 1500 workstation:

```
# cat /etc/path_to_inst
#
#       Caution! This file contains critical kernel state
#
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/pci@1e,600000" 0 "pcisch"
```

```
"/pci@1e,600000/pmu@6" 0 "pmubus"
"/pci@1e,600000/pmu@6/ppm@0,b3" 0 "m1535ppm"
"/pci@1e,600000/pmu@6/beep@0,b2" 0 "grbeep"
"/pci@1e,600000/pmu@6/i2c@0,0" 0 "smbus"
"/pci@1e,600000/pmu@6/i2c@0,0/card-reader@40" 0 "scmi2c"
"/pci@1e,600000/isa@7" 0 "ebus"
"/pci@1e,600000/isa@7/i2c@0,320" 0 "pcf8584"
"/pci@1e,600000/isa@7/i2c@0,320/clock-generator@0,d2" 0 "ics951601"
"/pci@1e,600000/isa@7/i2c@0,320/hardware-monitor@0,5c" 0 "adm1031"
"/pci@1e,600000/isa@7/i2c@0,320/motherboard-fru-prom@0,a8" 0 "seeprom"
"/pci@1e,600000/isa@7/i2c@0,320/dimm-spd@0,a0" 1 "seeprom"
"/pci@1e,600000/isa@7/i2c@0,320/dimm-spd@0,a2" 2 "seeprom"
"/pci@1e,600000/isa@7/power@0,800" 0 "power"
"/pci@1e,600000/isa@7/serial@0,3f8" 0 "su"
"/pci@1e,600000/isa@7/serial@0,2e8" 1 "su"
"/pci@1e,600000/isa@7/dma@0,0" 0 "isadma"
"/pci@1e,600000/isa@7/dma@0,0/parallel@0,378" 0 "ecpp"
"/pci@1e,600000/usb@a" 0 "ohci"
"/pci@1e,600000/usb@b" 1 "ohci"
"/pci@1e,600000/usb@b/keyboard@1" 2 "hid"
"/pci@1e,600000/usb@b/mouse@2" 0 "hid"
"/pci@1e,600000/pci@2" 0 "pci_pci"
"/pci@1e,600000/pci@2/usb@8,2" 0 "ehci"
"/pci@1e,600000/pci@2/usb@8" 2 "ohci"
"/pci@1e,600000/pci@2/usb@8,1" 3 "ohci"
"/pci@1e,600000/pci@2/firewire@b" 0 "hci1394"
"/pci@1e,600000/sound@8" 0 "audiots"
"/pci@1e,600000/ide@d" 0 "uata"
"/pci@1e,600000/ide@d/sd@2,0" 1 "sd" (DVD/CDROM)
"/pci@1e,600000/ide@d/dad@0,0" 1 "dad" (Disk 0)
"/pci@1e,600000/ide@d/dad@1,0" 0 "dad" (Disk 1)
```

Note – This example illustrates that the numeric values in instance names may not necessarily match the values found in logical or physical device names. On the Sun Blade 1500, the disk 0 (`c0t0d0`) is referenced by an entry in `/etc/path_to_inst` for the `dad1` instance name.

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

```
"/sbus@2,0/SUNW,socal@d,10000/sf@0,0/ssd@w21000020375b9ab6,0" 0 "ssd"
```

The following example is from an /etc/path_to_inst file with an entry for an external Fibre-Channel attached LUN:

```
"/pci@0,0/pci1022,7450@a/pci1077,10a@5/fp@0,0/disk@w266000c0fff7c140,1f"
212 "sd"
```

The prtconf Command

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

The prtconf command lists all possible instances of devices, whether the device is attached or not attached to the system. Use the prtconf and grep commands to list only the attached devices on the system. This example is from an Ultra 20 system:

```
# prtconf | grep -v not
System Configuration:  Sun Microsystems  i86pc
Memory size: 4095 Megabytes
System Peripherals (Software Nodes):

i86pc
    scsi_vhci, instance #0
    isa, instance #0
        asy, instance #0
    pci, instance #0
        pci108e,5347, instance #0
            hub, instance #1
                keyboard, instance #2
            mouse, instance #3
        pci108e,5347, instance #0
        pci108e,5347, instance #0
        pci-ide, instance #0
            ide, instance #0
                sd, instance #0
        pci-ide, instance #1
            ide, instance #2
                cmdk, instance #0
            ide, instance #3
                cmdk, instance #1
```

```
        pci-ide, instance #2
        pci10de,5c, instance #0
            pci1106,3044, instance #0
        pci108e,5347, instance #0
(output truncated)
```

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

The `format` Utility

Use the `format` utility to display both logical and physical device names for all currently available disks. This example output is from a Sun Blade 1500 system:

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


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@0,0
       1. c0t1d0 <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@1,0
Specify disk (enter its number): ^D
#
```

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

Here is an example of the `format` utility output from an Ultra 20 system:

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


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 39887 alt 2 hd 64 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
       1. c2d0 <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
Specify disk (enter its number): ^D
#
```

The `prtdiag` command

Use the `prtdiag` command to display system configuration and diagnostic information. See the `prtdiag(1M)` man page for more information.

```
# prtdiag
System Configuration:  Sun Microsystems  sun4v Sun Fire T200
Memory size: 32640 Megabytes
================================ Virtual CPUs
================================

CPU ID Frequency Implementation         Status
------ --------- ---------------------- -------
0      1200 MHz  SUNW,UltraSPARC-T1     on-line
1      1200 MHz  SUNW,UltraSPARC-T1     on-line
2      1200 MHz  SUNW,UltraSPARC-T1     on-line
3      1200 MHz  SUNW,UltraSPARC-T1     on-line
4      1200 MHz  SUNW,UltraSPARC-T1     on-line
5      1200 MHz  SUNW,UltraSPARC-T1     on-line
6      1200 MHz  SUNW,UltraSPARC-T1     on-line
7      1200 MHz  SUNW,UltraSPARC-T1     on-line
8      1200 MHz  SUNW,UltraSPARC-T1     on-line
9      1200 MHz  SUNW,UltraSPARC-T1     on-line
10     1200 MHz  SUNW,UltraSPARC-T1     on-line
11     1200 MHz  SUNW,UltraSPARC-T1     on-line
12     1200 MHz  SUNW,UltraSPARC-T1     on-line
13     1200 MHz  SUNW,UltraSPARC-T1     on-line
14     1200 MHz  SUNW,UltraSPARC-T1     on-line
15     1200 MHz  SUNW,UltraSPARC-T1     on-line
16     1200 MHz  SUNW,UltraSPARC-T1     on-line
```

```
17      1200 MHz  SUNW,UltraSPARC-T1     on-line
18      1200 MHz  SUNW,UltraSPARC-T1     on-line
19      1200 MHz  SUNW,UltraSPARC-T1     on-line
20      1200 MHz  SUNW,UltraSPARC-T1     on-line
21      1200 MHz  SUNW,UltraSPARC-T1     on-line
22      1200 MHz  SUNW,UltraSPARC-T1     on-line
23      1200 MHz  SUNW,UltraSPARC-T1     on-line
24      1200 MHz  SUNW,UltraSPARC-T1     on-line
25      1200 MHz  SUNW,UltraSPARC-T1     on-line
26      1200 MHz  SUNW,UltraSPARC-T1     on-line
27      1200 MHz  SUNW,UltraSPARC-T1     on-line
28      1200 MHz  SUNW,UltraSPARC-T1     on-line
29      1200 MHz  SUNW,UltraSPARC-T1     on-line
30      1200 MHz  SUNW,UltraSPARC-T1     on-line
31      1200 MHz  SUNW,UltraSPARC-T1     on-line

======================= Physical Memory Configuration
========================
Segment Table:
--------------------------------------------------------------
Base           Segment  Interleave  Bank     Contains
Address        Size     Factor      Size     Modules
--------------------------------------------------------------
0x0            32 GB    4           4 GB     MB/CMP0/CH0/R0/D0
                                             MB/CMP0/CH0/R0/D1
                                    4 GB     MB/CMP0/CH0/R1/D0
                                             MB/CMP0/CH0/R1/D1
                                    4 GB     MB/CMP0/CH1/R0/D0
                                             MB/CMP0/CH1/R0/D1
                                    4 GB     MB/CMP0/CH1/R1/D0
                                             MB/CMP0/CH1/R1/D1
                                    4 GB     MB/CMP0/CH2/R0/D0
                                             MB/CMP0/CH2/R0/D1
                                    4 GB     MB/CMP0/CH2/R1/D0
                                             MB/CMP0/CH2/R1/D1
                                    4 GB     MB/CMP0/CH3/R0/D0
                                             MB/CMP0/CH3/R0/D1
                                    4 GB     MB/CMP0/CH3/R1/D0
                                             MB/CMP0/CH3/R1/D1


======================== IO Configuration ========================

            IO
Location   Type  Slot Path                                 Name
Model
```

```
----------- ----- ---- ------------------------------------------- ----
---------------------- ---------
IOBD/NET0    PCIE IOBD                /pci@780/pci@0/pci@1/network@0
network-pciex8086,105e
IOBD/NET1    PCIE IOBD              /pci@780/pci@0/pci@1/network@0,1
network-pciex8086,105e
IOBD/PCIE   PCIE IOBD                 /pci@780/pci@0/pci@9/scsi@0
scsi-pciex1000,56 LSI,1064E
IOBD/NET2    PCIE IOBD                /pci@7c0/pci@0/pci@2/network@0
network-pciex8086,105e
IOBD/NET3    PCIE IOBD              /pci@7c0/pci@0/pci@2/network@0,1
network-pciex8086,105e
IOBD/PCIX  PCIX IOBD       /pci@7c0/pci@0/pci@1/pci@0/isa@2
isa
IOBD/PCIX    PCIX IOBD             /pci@7c0/pci@0/pci@1/pci@0/usb@5
usb-pciclass,0c0310
IOBD/PCIX    PCIX IOBD             /pci@7c0/pci@0/pci@1/pci@0/usb@6
usb-pciclass,0c0310
IOBD/PCIX   PCIX IOBD            /pci@7c0/pci@0/pci@1/pci@0/ide@8
ide-pci10b9,5229

============================ Environmental Status
============================
Fan sensors:
All fan sensors are OK.

Temperature sensors:
All temperature sensors are OK.

Current sensors:
All current sensors are OK.

Current indicators:
All current indicators are OK.

Voltage sensors:
All voltage sensors are OK.

============================ FRU Status ============================
Location                           Name      Status
-----------------------------------------------------
0651NNN1R6:CH                      MB        disabled
0651NNN1R6:CH/PS1                  PS        disabled
```

Reconfiguring Devices

You can cause a system to recognize a newly-added device in a number of ways, including invoking a reconfiguration boot, or running the devfsadm command.

Performing a Reconfiguration Boot

The reconfiguration boot process adds new device entries to a system's /etc/path_to_inst file, and new symbolic links and device files to the /dev and /devices directories.

For example, the following steps reconfigure a system to recognize a new external disk.

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

```
# touch /reconfigure
#
```

2. Use the init 5 command to shut down the system. This command safely powers off the system, allowing for addition or removal of devices. (If the device is already attached to your system, you can use the init 0 command to shut down without powering off.)

```
# init 5
```

3. Install the disk device. Make sure that the address of the device you add does not conflict with the address of other devices on the system.
4. Turn on the power to all external devices.
5. Power on and boot the system.
6. Log in, and use either the prtconf command or the format utility to verify that the disk device has been added.

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

Note – On a SPARC-based system, if the `/reconfigure` file was not created before the system was shut down, you can invoke a manual reconfiguration boot with the open boot PROM (OBP) level command: `boot -r`. On an x86/x64 system that boots using GRUB, you can edit the `kernel` command associated with the desired boot selection to add the `-r` option, and then boot the system.

Using the `devfsadm` Command

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

The `devfsadm` command attempts to load every driver in the system and attach all possible device instances. For new devices it finds, the `devfsadm` command then creates the appropriate physical device files in the `/devices` directory and symbolic links in the `/dev` directory. In addition to managing these files, the `devfsadm` command also maintains the `/etc/path_to_inst` file.

```
# devfsadm
```

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

```
# devfsadm -c device_class
```

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

```
# devfsadm -c disk
```

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

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

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

```
# devfsadm -i driver_name
```

The following examples use the -i option.

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

  ```
  # devfsadm -i sd
  ```

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

  ```
  # devfsadm -i st
  ```

You can add the -v option to devfsadm commands to display verbose output of changes to the device tree.

```
# devfsadm -v
```

To invoke cleanup routines that remove unreferenced symbolic links and device files for devices no longer attached to the system, perform the command:

```
# devfsadm -C
```

Partitioning a Hard Disk

Use the `format` utility to prepare hard disk drives for use with the Solaris OS. Although the `format` utility performs other disk management activities, the main function of the `format` utility is to manage disk partitions and slices.

The procedures used to partition disks on x86/x64 systems require the additional steps to create a Solaris `fdisk` partition before you can create slices within that partition. SPARC systems do not use `fdisk` partitions.

The same principles of allocating disk space apply to creating slices within a Solaris `fdisk` partition on an x86/x64 system as apply to creating slices on a disk in a SPARC system. This section describes disk space concepts, and possible undesirable conditions you can create when you define slices using the `format` utility.

Note – You do not need to partition the disk before you install the Solaris OS. Disk partitioning is part of the installation process.

Introducing Disk Partitioning

To divide a disk into `fdisk` partitions or slices:

1. Identify the correct disk.
2. Plan the layout of the disk.
3. On x86/x64 systems, use the `fdisk` menu in the `format` utility to create the `fdisk` partitions you require, and save the `fdisk` partition table to disk.
4. Use the `partition` menu in the `format` utility to divide the disk or Solaris `fdisk` partition into slices.
5. Label the disk or Solaris `fdisk` partition with new slice information.

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

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

Recognizing Disk Space and Undesirable Conditions

Slices are defined by an offset and a size, expressed in whole cylinders. On SPARC systems, the offset is the distance from the disk's first cylinder, cylinder 0. On x86/x64 systems, slice offsets are specified from the first cylinder of the Solaris `fdisk` partition.

Figure 2-12 shows an example of disk slice sizes and offsets on a SPARC system.

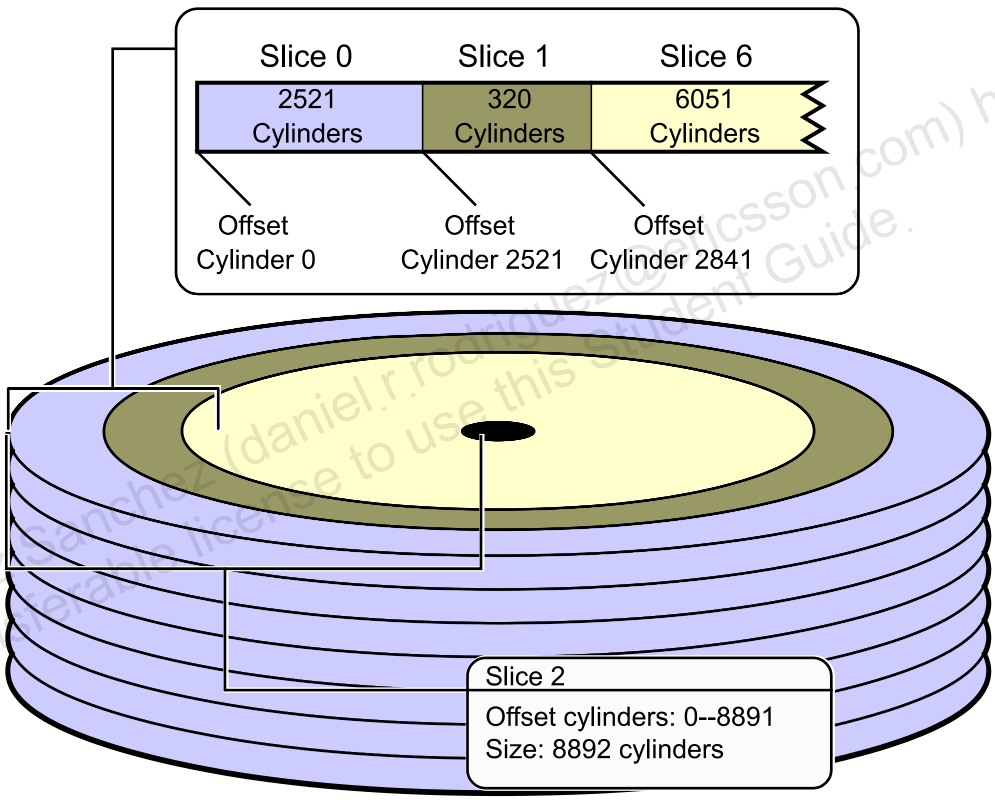


Figure 2-12 Offsets and Sizes for Disk Partitions on SPARC Systems

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

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

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

As viewed with the `format` utility, slices within Solaris `fdisk` partitions on an x86/x64 systems are also offset from cylinder 0. On these systems however, the `format` utility presents the first cylinder of the Solaris `fdisk` partition as cylinder 0.

On IDE or SATA disks in x86/x64 systems, slice 8 occupies cylinder 0, and slice 9 occupies cylinders 1 and 2 of the Solaris `fdisk` partition. The `format` utility prevents you from changing these two slice definitions.

Figure 2-13 shows an example of disk slice sizes and offsets on an x86/x64 system.

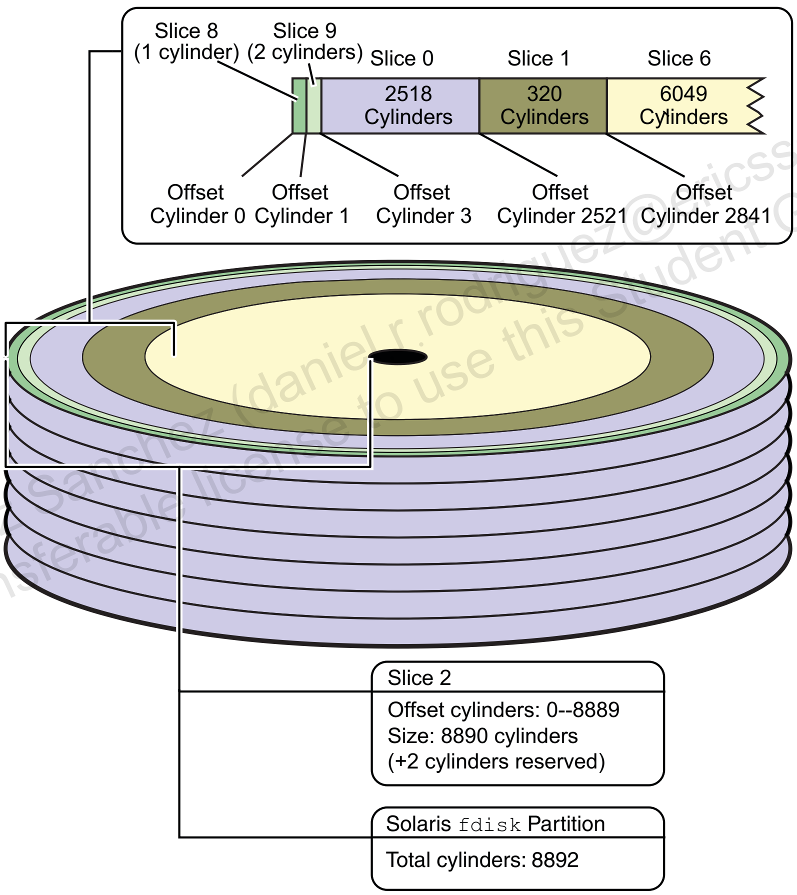


Figure 2-13 Offsets and Sizes for Disk Partitions on x86/x64 Systems

Additional slices you specify within the Solaris `fdisk` partition use the same offset and cylinder count method to specify their positions and sizes as used on SPARC systems. Because slice 8 and 9 occupy the first three cylinders of the Solaris `fdisk` partition, the first available cylinder for additional slice definitions is cylinder 3.

Slice 2 starts at cylinder 0 and ends on the last available cylinder of the Solaris `fdisk` partition. Typically two cylinders are reserved from the total number of cylinders that define the Solaris `fdisk` partition size.

Recognizing Wasted Disk Space

Wasted disk space occurs when one or more cylinders are not allocated to any disk slice. This undesirable space allocation can exist on SPARC and x86/x64 systems.

Figure 2-14 shows a disk with cylinders that are not allocated.

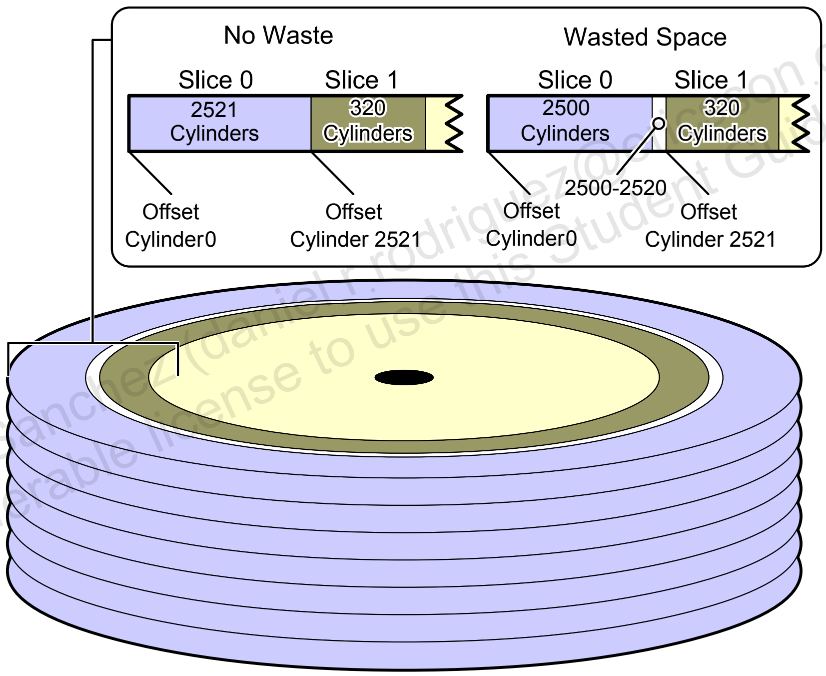


Figure 2-14 A Disk Slice With Wasted Space

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

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

Recognizing Overlapping Disk Slices

Overlapping disk slices occur when one or more cylinders are allocated to more than one disk slice. This undesirable allocation of space can exist on both SPARC and x86/x64 systems.

Figure 2-15 shows a disk with cylinders allocated to more than one disk slice.

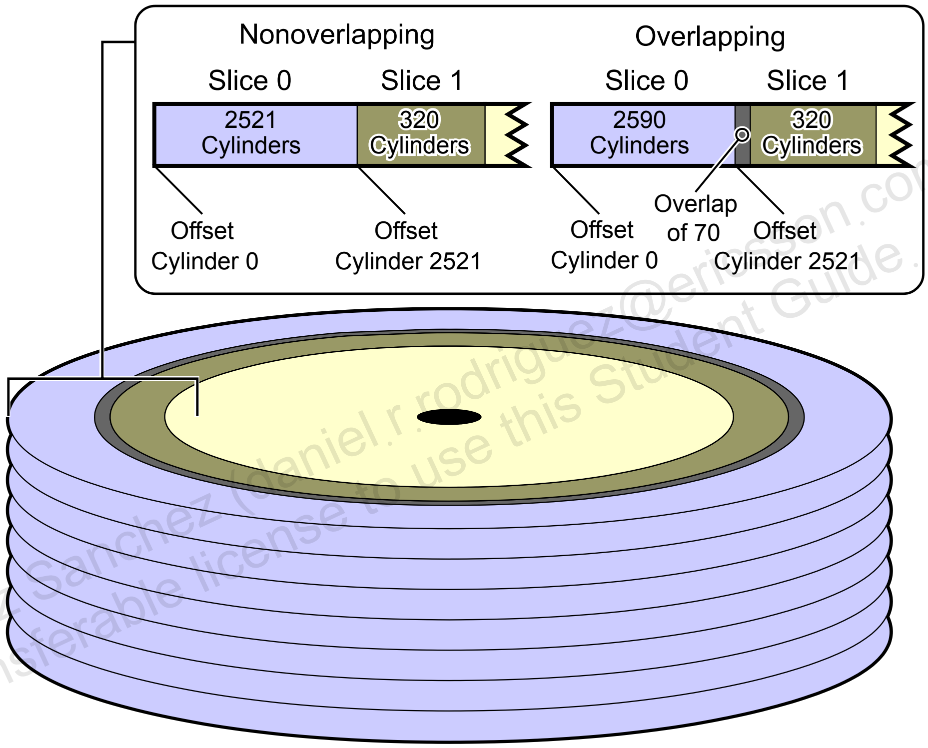


Figure 2-15 Disk Slices With Overlapping Cylinders

In Figure 2-15, Cylinders 2521 through 2590 are overlapping two disk slices.

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

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

```
Warning: Overlapping partition (1) in table.
Warning: Fix, or select a different partition table.
```

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

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

The `format` Utility and Disk Partition Tables

The `format` utility allows you to modify two partition tables on disk;

- `fdisk` partition tables
- Solaris OS partition tables (SPARC VTOC and x86/x64 VTOC)

`fdisk` Partition Tables

The `fdisk` partition table defines up to four `fdisk` partitions on disk. Solaris OS systems on the x86/x64 platform use the `fdisk` partition table to identify parts of the disk reserved for different operating systems, and to identify the partition from which the x86/x64 system will boot.

Only x86/x64 systems use `fdisk` partition tables. You use the `fdisk` menu in the `format` utility to view and modify `fdisk` partition tables.

Solaris OS Partition Tables

The SPARC VTOC and x86 and x64 VTOC define the slices that the Solaris OS will use on a disk. You use the `partition` menu in the `format` utility to view and modify these partition tables.

As the `root` user, when you use the `format` utility and select a disk to modify, a copy of the VTOC is read into memory, and is displayed as the current partition table in the `partition` menu.

The source of this partition table differs between SPARC and x86/x64 systems:

- SPARC systems read the VTOC from the first sector of the disk.
- x86/x64 systems read the VTOC from the second sector of the Solaris `fdisk` partition.

The `partition` menu in the `format` utility allows you to define slices and then save the slice definitions into the VTOC on disk.

Note – The `partition` menu describes writing the VTOC as *labeling* the disk. To label a disk means to write slice information into the VTOC. Keep in mind that the `partition` menu affects the VTOC that defines Solaris OS slices, and does not affect the `fdisk` partition table.

If you fail to label the disk or Solaris `fdisk` partition after defining slices, the slice information is lost.

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

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

Note – The `partition` menu uses the term *partition* exclusively, instead of *slice*. In describing working with this menu, the terms *partition* and *slice* mean the same thing.

Figure 2-16 shows the relationship among the VTOC, the current label in memory, and the predefined partition table in the `/etc/format.dat` file.

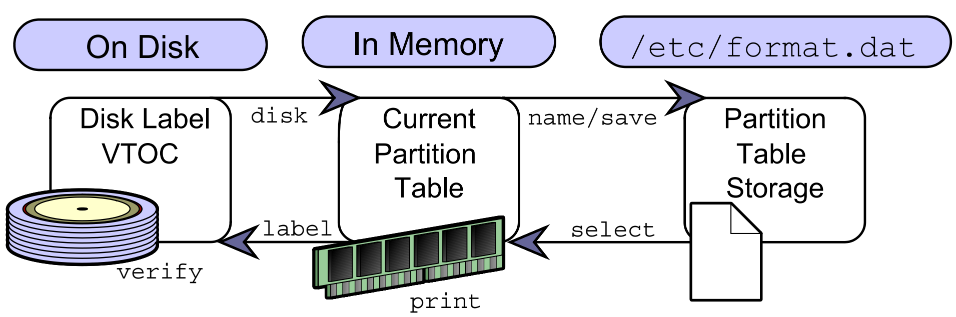


Figure 2-16 Solaris OS Partition Table Locations

The `save` function in the `format` utility writes partition table information into `./format.dat` by default. To save partition table information into the `/etc/format.dat` file, you must either specify the absolute pathname of the `/etc/format.dat` file when prompted, or run the `format` utility while located in the `/etc` directory and accept the default file name presented by the `save` function. A `format.dat` file exists in the `/etc` directory in the Solaris OS by default.

Table 2-2 describes terms presented by the `partition` menu in the `format` utility.

Table 2-2 Terms in the `partition` Menu

Term	**Description**
`Part`	The slice number. The `format` utility only allows you to modify slice numbers `0` through `7`.

`Tag`	A value that indicates how the slice is being used. This field is no longer used by Solaris, but Veritas Volume Manager assigns specific tags to slices it uses.
	`0 = unassigned`
	`1 = boot`
	`2 = root`
	`3 = swap`
	`4 = usr`
	`5 = backup`
	`6 = stand`
	`8 = home`
	9 = alternates
	
	Veritas Volume Manager array tags:
	`14 = public` region
	`15 = private` region

`Flag`	This field is no longer used by Solaris. Historical values are defined as follows: `00 wm` = The disk slice is writable and mountable. `01 wu` = The disk slice is writable and unmountable. *This is the default state of slices dedicated for swap areas.* `10 rm` = The disk slice is read-only and mountable. `11 ru` = The disk slice is read-only and unmountable.
`Cylinders`	The starting and ending cylinder numbers for the slice.
`Size`	The slice size: Mbytes (`MB`), Gbytes (`GB`), blocks (`b`), or cylinders (`c`).

Table 2-2 Terms in the `partition` Menu

`Blocks`	The space assigned to the slice, expressed as cylinders/tracks/sectors assigned to the slice. This value typically only reflects a whole number of cylinders.

Using the `format` Utility

The `format` utility presents a menu-driven, command-line interface. that initially presents the `FORMAT` menu. Commands in the `FORMAT` menu allow you to, among other functions, select a disk, select the `fdisk` menu, select the `partition` menu, save new disk and partition definitions to a file, and label the disk. The `FORMAT` menu displays the `format>` prompt.

Typing the name of a command or menu runs the command or accesses the menu you specify. Commands and menus indicate how to return to the FORMAT menu. For most, entering `quit` or `q` returns to the FORMAT menu.

For example, from the `format>` prompt, entering `partition`, or just `part` takes you to the `PARTITION` menu. The `partition>` prompt displays. This menu allows you to define Solaris OS slices, print the existing partition table, and write the VTOC to disk. Entering `quit` or `q` returns you to the `FORMAT` menu.

Defining Slices on a SPARC System

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

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

1. As the `root` user, run the `format` utility.

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


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@0,0
       1. c0t1d0 <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@1,0
Specify disk (enter its number):
```

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

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

```
Specify disk (enter its number): 1
selecting c0t1d0
[disk formatted, no defect list found]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
```

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

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

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

3. Type `partition` at the `format>` prompt. The `PARTITION` menu displays.

```
format> partition


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

The `PARTITION` menu enables you to perform the following functions:

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

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

```
partition> print
Current partition table (original):
Total disk cylinders available: 57459 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0 unassigned    wm       0                 0         (0/0/0)            0
  1 unassigned    wm       0                 0         (0/0/0)            0
  2     backup    wm       0 - 57458       111.79GB    (57459/0/0) 234432720
  3 unassigned    wm       0                 0         (0/0/0)            0
  4 unassigned    wm       0                 0         (0/0/0)            0
  5 unassigned    wm       0                 0         (0/0/0)            0
  6 unassigned    wu       0                 0         (0/0/0)            0
  7 unassigned    wu       0                 0         (0/0/0)            0

partition>
```

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

The columns of the table have the following meanings:

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

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

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

Enter partition id tag[unassigned]:
```

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

```
Enter partition id tag[unassigned]: ?
Expecting one of the following: (abbreviations ok):
        unassigned    boot          root          swap
        usr           backup        stand         var
        home          alternates    reserved

Enter partition id tag[unassigned]:
```

7. Type the tag `alternates`, and press Return.

```
Enter partition id tag[unassigned]: alternates
Enter partition permission flags[wm]:
```

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

```
Enter partition permission flags[wm]: ?
Expecting one of the following: (abbreviations ok):
        wm    - read-write, mountable
        wu    - read-write, unmountable
        rm    - read-only, mountable
        ru    - read-only, unmountable

Enter partition permission flags[wm]:
```

9. Press Return to accept the default flag.

```
Enter partition permission flags[wm]:  <return>
Enter new starting cyl[0]:
```

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

```
Enter new starting cyl[0]: <return>
Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]:
```

11. Enter 980mb for the new partition size for Slice 0.

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

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

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 57459 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0 alternates    wm       0 -   491      980.16MB    (492/0/0)     2007360
  1 unassigned    wm       0                0         (0/0/0)             0
  2     backup    wm       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wm       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0

partition>
```

The current partition table shows the change to Slice 0.

Now adjust the starting cylinder for Slice 1.

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

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

Enter partition id tag[unassigned]:
```

14. Type the tag swap, and press Return.

```
Enter partition id tag[unassigned]: swap
Enter partition permission flags[wm]:
```

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

```
Enter partition permission flags[wm]: wu
Enter new starting cyl[0]:
```

16. Enter the new starting cylinder for Slice 1. Specify the cylinder that follows the last cylinder used by slice 0.

```
Enter new starting cyl[0]: 492
Enter partition size[0b, 0c, 492e, 0.00mb, 0.00gb]:
```

17. Enter the new partition size for Slice 1.

```
Enter partition size[0b, 0c, 492e, 0.00mb, 0.00gb]: 512mb
partition>
```

18. Type print, and press Return.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 57459 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0 alternates    wm       0 -   491      980.16MB    (492/0/0)     2007360
  1       swap    wu     492 -   749      513.98MB    (258/0/0)     1052640
  2     backup    wm       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wm       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0

partition>
```

The current partition table shows the change to Slice 1.

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

Now adjust the starting cylinder for Slice 7.

19. Type **7** to select Slice 7.

```
partition> 7
Part      Tag    Flag     Cylinders         Size            Blocks
  7 unassigned    wu       0                0          (0/0/0)
0

Enter partition id tag[unassigned]:
```

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

```
Enter partition id tag[unassigned]: home
Enter partition permission flags[wu]:
```

21. Press Return to select the default flag.

```
Enter partition permission flags[wu]: <return>
Enter new starting cyl[0]:
```

22. Type the new starting cylinder for Slice 7.

```
Enter new starting cyl[0]: 750
Enter partition size[0b, 0c, 750e, 0.00mb, 0.00gb]:
```

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

```
Enter partition size[0b, 0c, 750e, 0.00mb, 0.00gb]:  $
partition>
```

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

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

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 57459 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0 alternates    wm       0 -   491      980.16MB    (492/0/0)     2007360
  1       swap    wu     492 -   749      513.98MB    (258/0/0)     1052640
  2     backup    wm       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wm       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7       home    wu     750 - 57458      110.33GB    (56709/0/0) 231372720

partition>
```

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

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

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

partition>
```

26. Enter `quit` or `q` to exit the `PARTITION` menu and exit the `format` utility.

```
partition> q


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format> q
#
```

Using the `format` Utility to Create `fdisk` Partitions

On x86/x64 systems, you must establish at least one `fdisk` partition on each disk you intend to use. This procedure is not necessary or available on SPARC-based systems.

On x86/x64 systems, the `format` utility provides access to the `fdisk` menu. The `fdisk` menu allows you to create `fdisk` partitions, to specify a Solaris `fdisk` partition, and choose an `fdisk` partition to use as the active (boot) partition. Only x86/x64 systems require `fdisk` partitions.

The `fdisk` menu allows you to create up to four `fdisk` partitions, however, only one Solaris OS `fdisk` partition can exist on a disk.

To create `fdisk` partitions, run the `format` utility and select an appropriate disk (in this example `c2d0`):

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


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 39887 alt 2 hd 64 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
       1. c2d0 <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
Specify disk (enter its number): 1
```

The `format` menu displays. To select another disk to view or modify, enter `disk` while in the `FORMAT` menu.

```
Controller working list found
[disk formatted, defect list found]

FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        fdisk      - run the fdisk program
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
```

```
        backup      - search for backup labels
        verify      - read and display labels
        save        - save new disk/partition definitions
        volname     - set 8-character volume name
        !<cmd>      - execute <cmd>, then return
        quit
format>
```

Enter the `fdisk` command to display the `fdisk` menu. If the disk has no `fdisk` partition defined, the `fdisk` menu presents the option to create a single `fdisk` partition that uses the entire disk, and that is identified as the Solaris `fdisk` partition on this disk. Type `n` to edit the `fdisk` partition table.

Note – Accepting this option makes the single `fdisk` partition it creates the *active* `fdisk` partition. This probably is not desirable on most systems.

```
format> fdisk
No fdisk table exists. The default partition for the disk is:

  a 100% "SOLARIS System" partition

Type "y" to accept the default partition,  otherwise type "n" to edit the
 partition table.

n
```

The `fdisk` menu displays information about the disk size, cylinder size, and current `fdisk` partitions. In this example, no `fdisk` partition is yet defined on the disk:

```
             Total disk size is 10011 cylinders
             Cylinder size is 16065 (512 byte) blocks

                                                     Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
```

```
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection:
```

To create a single `fdisk` partition that uses the whole disk, select option 1 - `Create a partition.` Enter the number that selects the type of disk partition that you would like to create. Select option 1 to create a SOLARIS2 `fdisk` partition.

```
Enter Selection: 1
Select the partition type to create:
   1=SOLARIS2  2=UNIX        3=PCIXOS     4=Other
   5=DOS12     6=DOS16       7=DOSEXT     8=DOSBIG
   9=DOS16LBA  A=x86 Boot    B=Diagnostic C=FAT32
   D=FAT32LBA  E=DOSEXTLBA   F=EFI        0=Exit? 1
```

Note – Earlier Solaris OS releases may have listed the SOLARIS partition type. The SOLARIS2 partition type simply uses a different identifying value in the disk label than the SOLARIS type did. This change prevents potential conflicts with other partition types.

The `fdisk` menu then prompts you to enter the percentage of the disk you would like to use for this `fdisk` partition. Enter 100 to use the whole disk:

```
Specify the percentage of disk to use for this partition
(or type "c" to specify the size in cylinders). 100
```

The `fdisk` menu then asks if this should be the active `fdisk` partition. Only the `fdisk` partition used to boot the system should be marked as the active `fdisk` partition. Because this is going to be a non-bootable partition, enter `n` for no.

```
Should this become the active partition? If yes, it  will be activated
each time the computer is reset or turned on.
Please type "y" or "n". n
```

The fdisk menu displays the list of fdisk partitions. The empty Status column indicates that the Solaris2 fdisk partition is not active.

```
        Total disk size is 10011 cylinders
        Cylinder size is 16065 (512 byte) blocks

                                                   Cylinders
   Partition   Status    Type          Start   End   Length    %
   =========   ======    ============  =====   ===   ======   ===
       1                 Solaris2          1  10010    10010    100


SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection:
```

Select option 5 to save the fdisk partition configuration.

```
Enter Selection: 5


format>
```

Note – To install multiple operating systems on a disk, create multiple fdisk partitions using fdisk option 1.

The following is an example of a disk with three fdisk partitions supporting three different operating systems:

```
        Total disk size is 10011 cylinders
        Cylinder size is 16065 (512 byte) blocks

                                                   Cylinders
   Partition   Status    Type          Start   End   Length    %
   =========   ======    ============  =====   ===   ======   ===
       1                 Win95 FAT32       1   3303    3303     33
       2                 UNIX System    3304   6606    3303     33
       3                 Solaris2       6607   9909    3303     33
```

Defining Slices on an x86/x64 System

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

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

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

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


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 39887 alt 2 hd 64 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
       1. c2d0 <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
Specify disk (enter its number):
```

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

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

```
Specify disk (enter its number): 1
selecting c2d0
Controller working list found
[disk formatted, defect list found]

FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        fdisk      - run the fdisk program
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
```

```
        save       - save new disk/partition definitions
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format>
```

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

partition	Displays the PARTITION menu
label	Writes the current partition definition to the VTOC of the Solaris fdisk partition
verify	Reads and displays the disk label
quit	Exits the format utility

3. Type partition at the format prompt. The PARTITION menu displays.

```
format> partition


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

The Partition menu enables you to perform the following functions:

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

4. Type `print` at the `partition` prompt to display the disk label that was copied to random access memory (RAM) when the `format` utility was invoked. This example is from a SATA disk drive in an Ultra 20.

```
partition> print
Current partition table (original):
Total disk cylinders available: 10008 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0 unassigned    wm       0                  0         (0/0/0)            0
  1 unassigned    wm       0                  0         (0/0/0)            0
  2     backup    wu       0 - 10007      76.67GB    (10008/0/0) 160778520
  3 unassigned    wm       0                  0         (0/0/0)            0
  4 unassigned    wm       0                  0         (0/0/0)            0
  5 unassigned    wm       0                  0         (0/0/0)            0
  6 unassigned    wm       0                  0         (0/0/0)            0
  7 unassigned    wm       0                  0         (0/0/0)            0
  8       boot    wu       0 -     0       7.84MB    (1/0/0)          16065
  9 alternates    wm       1 -     2      15.69MB    (2/0/0)          32130

partition>
```

The partition table name appears in parentheses in the first line of the table. Slice 8 is the boot slice. The boot slice uses the first cylinder (cylinder 0) of the Solaris OS `fdisk` partition. Slice 9 uses the next two cylinders, and the `alternates` tag. Slice 9 contains blocks used to store bad block information. This slice is not found or used for this purpose on SCSI or Fibre disks.

The columns of the table have the following meanings:

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

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

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

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

```
Enter partition id tag[unassigned]: ?
Expecting one of the following: (abbreviations ok):
        unassigned    boot          root          swap
        usr           backup        stand         var
        home          alternates    reserved

Enter partition id tag[unassigned]:
```

7. Type the tag `reserved`, and press Return.

```
Enter partition id tag[unassigned]: reserved
Enter partition permission flags[wm]:
```

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

```
Enter partition permission flags[wm]: ?
Expecting one of the following: (abbreviations ok):
        wm    - read-write, mountable
        wu    - read-write, unmountable
        rm    - read-only, mountable
        ru    - read-only, unmountable

Enter partition permission flags[wm]:
```

9. Press Return to accept the default flag.

```
Enter partition permission flags[wm]: <return>
```

10. Enter 3 to accept the starting cylinder of 3. Slices 8 and 9 are already using cylinders 0-2.

```
Enter new starting cyl[3]:  <return>
Enter partition size[0b, 0c, 3e, 0.00mb, 0.00gb]:
```

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

```
Enter partition size[0b, 0c, 3e, 0.00mb, 0.00gb]:  400mb
partition>
```

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

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 10008 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 -    53      400.06MB    (51/0/0)       819315
  1 unassigned    wm       0                0         (0/0/0)             0
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wm       0                0         (0/0/0)             0
  6 unassigned    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130

partition>
```

The current partition table shows the change to Slice `0`. Adjust the starting cylinder for Slice 1.

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

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

Enter partition id tag[unassigned]:
```

14. Type the tag `swap`, and press Return.

```
Enter partition id tag[unassigned]: swap
Enter partition permission flags[wm]:
```

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

```
Enter partition permission flags[wm]: wu
Enter new starting cyl[3]:
```

16. Enter the new starting cylinder for Slice 1. Specify the cylinder that follows the ending cylinder for slice 0.

```
Enter new starting cyl[3]: 54
Enter partition size[0b, 0c, 54e, 0.00mb, 0.00gb]:
```

17. Enter the new partition size for Slice 1.

```
Enter partition size[0b, 0c, 54e, 0.00mb, 0.00gb]:  512mb
partition>
```

18. Type `print`, and press Return.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 10008 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 -    53      400.06MB    (51/0/0)       819315
  1       swap    wu      54 -   119      517.72MB    (66/0/0)      1060290
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wm       0                 0        (0/0/0)             0
  4 unassigned    wm       0                 0        (0/0/0)             0
  5 unassigned    wm       0                 0        (0/0/0)             0
  6 unassigned    wm       0                 0        (0/0/0)             0
  7 unassigned    wm       0                 0        (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130

partition>
```

The current partition table shows the change to Slice 1. The new starting cylinder for Slice 1 is one greater than the ending cylinder for Slice 0. Adjust the starting cylinder for Slice 7.

19. Type 7 to select Slice 7.

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

Enter partition id tag[unassigned]:
```

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

```
Enter partition id tag[unassigned]: home
Enter partition permission flags[wm]:
```

21. Press Return to select the default flag.

```
Enter partition permission flags[wm]: <return>
Enter new starting cyl[3]:
```

22. Type the new starting cylinder for Slice 7. Specify the cylinder that follows the last cylinder for slice 1.

```
Enter new starting cyl[3]:  120
Enter partition size[0b, 0c, 120e, 0.00mb, 0.00gb]:
```

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

```
Enter partition size[0b, 0c, 120e, 0.00mb, 0.00gb]:  $
partition>
```

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

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

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 10008 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0   reserved    wm       3 -    53      400.06MB    (51/0/0)       819315
  1       swap    wu      54 -   119      517.72MB    (66/0/0)      1060290
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wm       0                0         (0/0/0)             0
  6 unassigned    wm       0                0         (0/0/0)             0
  7       home    wm     120 - 10007       75.75GB    (9888/0/0)  158850720
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130

partition>
```

Add the cylinders in the Blocks column for Slice 0, Slice 1, Slice 7, Slice 8, and Slice 9. The number should equal the total number of cylinders contained in Slice 2.

25. After checking the partition table, to ensure that there are no errors, label the Solaris `fdisk` partition by typing `label`.

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

partition>
```

26. Enter `quit` or `q` to exit the `PARTITION` menu and exit the `format` utility.

```
partition> q

FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        fdisk      - run the fdisk program
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format> q
```

Note – To select another disk to view or modify, enter `disk` in the `FORMAT` menu.

EFI Label and VTOC Label Comparison

The EFI disk label differs from the VTOC disk label as follows:

- The EFI label provides support for disks greater than 1 terabyte.
- The EFI label provides usable slices 0-6, where slice 2 is just another slice.
- Partitions (or slices) cannot overlap with the primary or backup label, nor with any other partition. The EFI label size is usually 34 sectors, so partitions start at sector 34. This feature means that no partition can start at sector zero.
- No cylinder, head, or sector information is stored in the EFI label. Sizes are reported in blocks.
- Information that was stored in alternate cylinder areas (the last two cylinders of the disk) is now stored in slice 8.
- If you use the `format` utility to change partition sizes, the unassigned partition tag is assigned to partitions with sizes equal to zero. By default, the `format` utility assigns the `usr` partition tag to any partition with a size greater than zero. Use the partition change menu to reassign partition tags after the partitions are changed. You cannot change a partition with a non-zero size to the unassigned partition tag.

Note – For SMI/VTOC labeled disks, you have 8 slices (0 to 7). For EFI labeled disks, you have 10 slices (0 to 9). How you use slices depends on your requirements. You can allocate some space to all the slices, leave some slices blank, or allocate one slice as the entire disk space.

The default behavior of both EFI and SMI/VTOC labeled disks are that slice 2 always occupies the entire disk space. For EFI labeled disks, you can resize slice 2. You cannot do this with SMI/VTOC labeled disks.

EFI Disk Label Restrictions

If you have disks in your environment that contain more than 1 terabyte, you must consider the following restrictions:

- Layered software products intended for systems with VTOC-labelled disks might be incapable of accessing a disk with an EFI disk label.
- A disk with an EFI label is not recognized on systems running previous Solaris releases.
- You cannot boot from a disk with an EFI disk label.
- You cannot use the `fdisk` command on a disk with an EFI label that is greater than 1 terabyte.
- The EFI specification prohibits overlapping slices. The entire disk is represented by *`cxtydz`*.
- The EFI disk label provides information about disk or partition sizes in sectors and blocks, but not in cylinders and heads.

The following `format` options are either not supported or are not applicable on disks with EFI labels:

- The `save` option is not supported because disks with EFI labels do not need an entry in the `format.dat` file.
- The `backup` option is not applicable because the disk driver finds the primary label and writes it back to the disk.

Viewing the VTOC

You can use two methods to view the SPARC or x86/x64 VTOC on disk:

- Use the `verify` command in the `format` utility
- Run the `prtvtoc` command from the command line

Note – The VTOC on SPARC systems is found in the first sector on the disk, and the VTOC on x86/x64 systems is found in the second sector of the Solaris `fdisk` partition on the disk.

Reading a Disk VTOC Using the `verify` Command

The `verify` command enables you to view a SPARC or x86/x64 VTOC from within the `format` utility. To read a disk's VTOC, perform the following steps:

1. Run the `format` utility and select the disk you want to use.
2. At the `format>` prompt, enter the `verify` command, and press Return.

   This example is from a Sun Blade 1500:

```
format> verify
Primary label contents:

Volume name = <        >
ascii name  = <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
pcyl        = 57461
ncyl        = 57459
acyl        =    2
nhead       =   16
nsect       =  255
Part      Tag    Flag     Cylinders         Size            Blocks
  0 alternates    wm       0 -   491      980.16MB    (492/0/0)
2007360
  1       swap    wu     492 -   749      513.98MB    (258/0/0)
1052640
  2     backup    wm       0 - 57458      111.79GB    (57459/0/0)
234432720
  3 unassigned    wm       0                0         (0/0/0)
0
  4 unassigned    wm       0                0         (0/0/0)
0
  5 unassigned    wm       0                0         (0/0/0)
0
  6 unassigned    wm       0                0         (0/0/0)
0
  7       home    wm     750 - 57458      110.33GB    (56709/0/0)
231372720

format>
```

This example is from an Ultra 20:

```
format> verify

Primary label contents:

Volume name = <        >
ascii name  = <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
pcyl        = 10010
ncyl        = 10008
acyl        =    2
bcyl        =    0
nhead       =  255
nsect       =   63
Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 -    53      400.06MB    (51/0/0)
819315
  1       swap    wu      54 -   119      517.72MB    (66/0/0)
1060290
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0)
160778520
  3 unassigned    wm       0                0         (0/0/0)
0
  4 unassigned    wm       0                0         (0/0/0)
0
  5 unassigned    wm       0                0         (0/0/0)
0
  6 unassigned    wm       0                0         (0/0/0)
0
  7       home    wm     120 - 10007       75.75GB    (9888/0/0)
158850720
  8       boot    wu       0 -     0        7.84MB    (1/0/0)
16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)
32130

format>
```

3. Type `quit` or `q`, and press Return to exit the `format` utility.

Reading a Disk VTOC Using the `prtvtoc` Command

The `prtvtoc` command enables you to view a SPARC or x86/x64 VTOC from the command line.

Use the prtvtoc command as described in these examples to view the VTOC. This example is from a Sun Blade 1500:

```
# prtvtoc /dev/rdsk/c0t1d0s2
* /dev/rdsk/c0t1d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*     255 sectors/track
*      16 tracks/cylinder
*    4080 sectors/cylinder
*   57461 cylinders
*   57459 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      9    00          0   2007360   2007359
       1      3    01    2007360   1052640   3059999
       2      5    00          0 234432720 234432719
       7      8    00    3060000 231372720 234432719
```

This example is from an Ultra 20 workstation:

```
# prtvtoc /dev/rdsk/c2d0s2
* /dev/rdsk/c2d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*   10010 cylinders
*   10008 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0     11    00      48195    819315    867509
       1      3    01     867510   1060290   1927799
       2      5    01          0 160778520 160778519
       7      8    00    1927800 158850720 160778519
       8      1    01          0     16065     16064
       9      9    00      16065     32130     48194
#
```

The VTOC information includes the following fields:

`Dimensions`	Describes the logical dimensions of the disk.
`Partition`	A number that identifies the slice.
`Tag`	A value that indicates how the slice is being used. The Solaris OS no longer uses this field.
`Flags`	The Solaris OS no longer uses this field.
`First Sector`	Defines the first sector of the slice.
`Sector Count`	Defines the total number of sectors in the slice.
`Last Sector`	Defines the last sector number in the slice.
`Mount Directory`	If the field is empty, the slice is currently not mounted.

Replacing a SPARC or x86/x64 VTOC on Disk

Use the `prtvtoc` command to save VTOC information to a file. This enables you to use the `fmthard` command to replace the SPARC or x86/x64 VTOC to recover from events such as these:

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

Note – The `prtvtoc` and `fmthard` commands read and write the SPARC or x86/x64 VTOC on disks. They do not affect the `fdisk` partition table on x86/x64 systems. Use the `fdisk` menu in the `format` utility to establish `fdisk` partitions on x86/x64 systems.

To save VTOC information to a file, use the `prtvtoc` command as in the following examples. Use the device name for the correct disk, and save the `prtvtoc` output to a file named as you wish.

```
# prtvtoc /dev/rdsk/c0t1d0s2 > /var/tmp/c0t1d0s2.vtoc

# prtvtoc /dev/rdsk/c2d0s2 > /var/tmp/c2d0s2.vtoc
```

The `fmthard` Command

To replace a SPARC or x86/x64 VTOC on disk, you can use the saved output from the `prtvtoc` command as input to the `fmthard` command. In this syntax example, you would replace *`datafile`* with the name of the file that contains the `prtvtoc` command output. Use slice 2 in the name of the target disk or Solaris `fdisk` partition.

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

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

If the need to replace a SPARC or x86/x64 VTOC arises and the VTOC was previously saved to a file, the following options are available:

- Run `format`, select the disk, and label it with the default partition table, or define slices and label the disk as required.
- Use the `fmthard` command to write saved VTOC information back to the disk.

This example is from a Sun Blade 1500:

```
# fmthard -s /var/tmp/c0t1d0s2.vtoc /dev/rdsk/c0t1d0s2
fmthard:  New volume table of contents now in place.
#
```

This example is from an Ultra 20:

```
# fmthard -s /var/tmp/c2d0s2.vtoc /dev/rdsk/c2d0s2
fmthard:  New volume table of contents now in place.
#
```

- Use the `fmthard` command to initialize the VTOC of a disk.

```
# fmthard -s /dev/null /dev/rdsk/c1t3d0s2
fmthard:  New volume table of contents now in place.
#
```

x86/x64 VTOC Information and Solaris `fdisk` Partitions

On x86/x64 systems, the Solaris `fdisk` partition size must match the size described in the saved VTOC information. If the Solaris `fdisk` partition size differs from the size described in the `prtvtoc` command output, `fmthard` reports errors similar to these:

```
# fmthard -s /var/tmp/c2d0s2.vtoc /dev/rdsk/c2d0s2
fmthard: Partition 2 specifies the full disk and is not equal
full size of disk.  The full disk capacity is 80373195 sectors.
fmthard: Partition 2 specified as 160778520 sectors starting at 0
        does not fit. The full disk contains 80373195 sectors.
fmthard: Partition 7 specified as 158850720 sectors starting at 1927800
        does not fit. The full disk contains 80373195 sectors.
fmthard:  New volume table of contents now in place.
#
```

These messages serve only as a warning. The erroneous partition information is written to the x86/x64 VTOC.

To correct this situation, you can:

- Use the `partition` menu in the `format` utility to re-define the slices in the Solaris `fdisk` partition.
- Use the `fdisk` menu in the `format` utility to re-define the `fdisk` partitions to match the size defined in the saved `prtvtoc` output, and use `fmthard` to write the x86/x64 VTOC information to disk.
- Use the `fdisk` command to replace the `fdisk` partition table with saved `fdisk` partition information, and use `fmthard` to write the x86/x64 VTOC information to disk.

Replacing `fdisk` Partition Tables on x86/x64 Systems

The `fdisk` command enables you to view and modify `fdisk` partition tables on disk. The `fdisk` command allows you to save a disk's `fdisk` partition table to a file. You can then use the `fdisk` command to replace the `fdisk` partition table on disk with the saved information. This procedure only pertains to x86/x64 systems.

To view an `fdisk` partition table, use the `fdisk` command as in the following example from an Ultra 20 workstation:

```
# fdisk -W - /dev/rdsk/c2d0p0

* /dev/rdsk/c2d0p0 default fdisk table
* Dimensions:
*    512 bytes/sector
*     63 sectors/track
*    255 tracks/cylinder
*   10011 cylinders
*
* systid:
*    1: DOSOS12
*    2: PCIXOS
*    4: DOSOS16
*    5: EXTDOS
*    6: DOSBIG
*    7: FDISK_IFS
*    8: FDISK_AIXBOOT
*    9: FDISK_AIXDATA
*   10: FDISK_0S2BOOT
*   11: FDISK_WINDOWS
*   12: FDISK_EXT_WIN
*   14: FDISK_FAT95
*   15: FDISK_EXTLBA
*   18: DIAGPART
*   65: FDISK_LINUX
*   82: FDISK_CPM
*   86: DOSDATA
*   98: OTHEROS
*   99: UNIXOS
*  101: FDISK_NOVELL3
*  119: FDISK_QNX4
*  120: FDISK_QNX42
*  121: FDISK_QNX43
*  130: SUNIXOS
*  131: FDISK_LINUXNAT
```

```
*   134: FDISK_NTFSVOL1
*   135: FDISK_NTFSVOL2
*   165: FDISK_BSD
*   167: FDISK_NEXTSTEP
*   183: FDISK_BSDIFS
*   184: FDISK_BSDISWAP
*   190: X86BOOT
*   191: SUNIXOS2
*   238: EFI_PMBR
*   239: EFI_FS
*

* Id    Act  Bhead  Bsect  Bcyl    Ehead  Esect  Ecyl    Rsect    Numsect
  191   0    0      1      1       254    63     1023    16065
160810650
#
```

The `fdisk` command lists the disk geometry, all of the possible partition types and their corresponding ID values, and the currently defined `fdisk` partitions. In this example, only one `fdisk` partition is defined. Its type 191 indicates that it is a Solaris OS (`SUNIXOS2` = `SOLARIS2`) partition.

To save `fdisk` partition table information to a file, use the `fdisk` command as in the following example. Replace the argument to the `-W` option with the name of the file you wish to use, and specify `fdisk` partition 0 of the correct disk.

```
# fdisk -W /var/tmp/c2d0p0.fdisk /dev/rdsk/c2d0p0
#
```

To replace the `fdisk` partition table on disk with the saved information, use the `fdisk` command as in the following example. Replace the argument to the `-F` option with the name of the file that contains the saved partition information, and specify `fdisk` partition 0 of the correct disk.

```
# fdisk -F /var/tmp/c2d0p0.fdisk /dev/rdsk/c2d0p0
#
```

The `-F` option will zero out the VTOC on the Solaris partition if the `fdisk` partition table changes.

Extended Volume Table of Contents

Prior to the Solaris 10 10/09 release, the Sun disk label limited the size of the bootable disk to less than 1 Tbyte. Now, installing and booting from disks in the 1 to 2 Tbytes range is supported with the 64-bit Solaris kernel.

The Extended VTOC feature is available after a standard installation of Solaris 10 10/09 release, and can be used immediately as it is on by default.

Note – There is currently no support for disks over 1 Tbyte on 32-bit kernels.

Previously, an Extensible Firmware Interface (EFI) label for a disk that is larger than 1 Tbyte was always used. Now the Volume Table of Contents (VTOC) label is extended to any size disk, but the addressable space by the VTOC is limited to 2 Tbytes.

Features of the new Extended VTOC

- The `fdisk` utility can now be used on a disk that is greater than 1 Tbyte on x86 systems. Support is added for up to 2 Tbyte partitions in the Master Boot Record (MBR) for non-EFI partition types. Other non-EFI partitions may be subject to a limit depending on partition type.
- The `format -e` utility can be used to label a disk of any size with a VTOC label, but the addressable space is limited to 2 Tbytes.
- In addition to using a VTOC label, the `format -e` command can be used to label a disk that is 2 Tbytes in size or less with an EFI label.
- When the `fdisk` utility is run on a disk that is greater than 2 Tbytes in size, a warning message is displayed indicating that a non-EFI partition greater than 2 Tbytes cannot be created.

Note – The Solaris Volume Manager software in the Solaris 10/09 release is modified to create meta-devices that support physical disks with VTOC labels up to 2 Tbytes in size.

Extended VTOC Requirements and Dependencies

- The Extended VTOC is supported on both x86 (64-bit) and SPARC platforms, and utilizes the Newboot feature.
- The Extended VTOC works with target and HBA drivers that support disks larger than 1 Tbyte.
- The DAD and IDE interfaces on SPARC, and the PCMCIA device on x86 systems do not support disks that are larger than 1 Tbyte.

Note – Disks with existing EFI labels continue to work as in previous Solaris releases.

Exercise: Managing Local Disk Devices

In this exercise, you complete the following tasks:

- Identify logical, physical, and instance names for disk devices
- Add a new disk or tape drive to a system
- Create and manage disk slices on Solaris systems on the SPARC platform
- Create and manage fdisk partitions and disk slices on x86/x64 Solaris systems
- Use Solaris Management Console to manage local disks on SPARC-based systems

Preparation

This exercise requires one unused spare disk.

This exercise applies to both SPARC and x86/x64 systems. Keep in mind that these different systems typically use different disk device names. For example:

- The boot disk of a Sun Blade 1500 (SPARC) is typically c0t0d0.
- The boot disk of an Ultra 20 (x86/x64) is typically c1d0.

Throughout this exercise, be certain to use the correct device names for your particular system.

Task 2 in this exercise is optional. Task 2 requires a system that is configured with an external disk or tape drive. In order to perform Task 2, one of these two conditions must be met:

- The external device was turned off during system installation, so no device files for the device will exist in the `/dev` structure.
- Device files for the external device are removed before starting Task 2. To remove device files:
  - Shut down the system to run state 0 or 5, and power off the external device.
  - Boot the system to the default run state.
  - Login as `root` and run `devfsadm -C` to clear any device files related to the external device.

Exercise Summary

This exercise is composed of five tasks that illustrate how to manage local disk devices. The tasks in this exercise describe the following concepts:

- Identifying Device Files - demonstrates the relationships among logical and physical device files, and device instance names.
- Adding a New Disk or Tape Device - describes the procedure required to build device files for new devices added to your system. This task is optional, depending on the peripherals available on your system.
- Working With Disks and Partitions on SPARC Systems - describes using the `format` utility, and the `prtvtoc` and `fmthard` commands, to create and manage disk slices on disks attached to Solaris OS systems on the SPARC platform.
- Working With Disks and Partitions on x86/x64 Systems - describes using the `format` utility, and the `fdisk`, `prtvtoc` and `fmthard` commands, to create and manage `fdisk` partitions and disk slices on disks attached to x86/x64 Solaris systems.

Task 1 – Identifying Device Files

Complete the following steps:

1. Log in as the `root` user and complete the following steps:
   a. Open a terminal window.
   b. Expand the window so that it occupies the entire screen area.
   c. Change to the `/dev/dsk` directory.
2. List the files in this directory and complete the following steps:
   a. Identify the files related to the boot disk of your system.

      Systems typically use `c0t0d0` or `c1d0`, but this may vary.
   b. Locate the file that represents slice 0 on this disk and display a long listing of it.

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

   ____

   c. Record the full path name to which this file points.

   ____

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

   Which type of file is this?

   ____

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

4. Change to the `/dev/rdsk` directory and complete the following steps:

   a. Display a long listing of the same file name you selected in Step 2.

   Which type of file is this?

   ____

   b. Record the full path name to which this file points.

   ____

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

   ____

   Which type of file is this?

   ____

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

6. Change to the `/etc` directory and display the contents of the `path_to_inst` file.

7. Use the information from the previous steps to locate and record the entry for your boot disk.

   ____

   What is the instance name for the device listed in this step?

   ____

Task 2 – Adding a New Disk or Tape Device

Complete the following steps:

Note – This task is optional. Complete the steps in this task if an external disk or tape device is available on your system.

---

1. In the /dev/dsk or /dev/rmt directories, confirm that no files exist for your external disk or tape device; for example, /dev/dsk/c1t0d0s0 or /dev/rmt/0.

   If files for the external device do exist, ask your instructor for guidance. Do not simply remove device files that exist. Refer to the Preparation section for advice about removing device files.
2. If you are using a SPARC system, shut it down to run state 0.
3. If you are using an x86/x64 system, shut it down to run state 5. This shuts down the system and turns off the power.
4. Power on the external disk or tape drive attached to your system.
5. If you are using an x86/x64 system, power on your system and allow it to boot to its default run state.
6. If you are using a SPARC system, boot the system to its default run state.
7. Log in as the root user and complete the following steps:
   a. Open a terminal window.
   b. Run the devfsadm command with the -v option to create new links and device files for the new disk or tape drive.
   c. Observe the messages that the devfsadm command displays.
8. Confirm that new links and device files exist in the /dev/dsk and /dev/rdsk directories for disks or /dev/rmt for tape drives.

   Use the devfsadm -v output to determine what files to look for.

Task 3 - Working With Disks and Partitions on SPARC

Complete the following steps:

1. Log in as the `root` user and complete the following steps:
   a. Open a terminal window.
   b. Run the `format` utility.
2. Complete the following steps:
   a. Record the list of disks presented by the `format` utility. For example, `c0t0d0` and `c0t1d0`.

   ____

   b. Press **`Control-D`** to exit the `format` utility.
3. Use the `prtvtoc` command to list the VTOC for each of the disks you found in the previous step and complete the following steps:
   a. Examine the `Mount Directory` field in the information that the `prtvtoc` command displays.
   b. Record the name of a disk that has no mount directory listed.

   For your classroom environment, this is an unused disk.

Note – Systems in production can use unmounted slices to store production data. Do not assume that unmounted slices are unused on systems in production. Systems will display various lists of partitions, so the list you see may differ from the list the examples.

Unused disk:

____

4. Run the `format` utility again and select the unused disk from the list of disks presented.
5. Display the `partition` menu and complete the following steps:
   a. Print the current partition table.
   b. Record the number of gigabytes (Gbytes) assigned to Slice 2.

   Gbytes:

   ____

Note – The list of defined partitions will vary, depending on how the spare disk was last used.

6. Divide the number of Gbytes by 4 and complete the following steps:
   a. Use the result as the number of Gbytes to assign as disk space to four slices.
   b. Round *down* to the next whole Gbyte if the result includes a fraction.

      For example, if the slice 2 is 111.79 Gbytes, 111.79 / 4 = 27.94 Gbytes. In this case use 27 Gbytes as the size for the four slices.

      Gbytes/4:

   ____________________________________.

7. Display the Partition menu again and complete the following steps:
   a. Select Slice 0.
   b. Accept the defaults for tags and flags.
   c. Start this first slice on Cylinder 0.
   d. Enter the resulting number of Gbytes from the previous step for the slice size.
   e. Print the partition table again to verify the change.
8. Set the sizes of Slices 1, 3, and 4 so that they are the same as Slice 0.

   You should begin each successive slice on the cylinder that follows the ending cylinder of the previous slice.
9. Set Slices 5, 6, and 7 to start at Cylinder 0, and then assign them 0 Mbytes.
10. Complete the following steps:
    a. Print the partition table.

    Except for slice 2, is there any overlap of ending and beginning cylinders for any of the slices listed?

    ____

    b. Proceed to the following steps to introduce this problem.
11. Complete the following steps:
    a. Add 1 to the number Gbytes/4 value listed in Step 6.

       (Gbytes/4) + 1:

    ____

b. Change Slice 0 so that it uses the new size.

The Cylinders column in the partition table should now indicate that Slice 0 ends after Slice 1 begins.

12. Complete the following steps:

a. Use the modify command from the partition menu to attempt to fix this problem.

b. Select Item 0 to modify the current partition table.

Which warnings display?

____

____

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

The partition table displays.

14. Observe the Cylinders and Size columns, and then verify that they are all zero except for slice 2.

15. Respond to the prompts to continue the process and complete the following steps:

a. Select Slice 4 as the All Free Hog partition.

b. Use the size listed in Step 6 for Slices 0, 1, and 3.

c. Enter Return to set the other slices to size 0.

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

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

17. Save your new partition table to the /etc/format.dat file and complete the following steps:

a. Carefully read the message that the format utility displays, and then enter the correct file name.

b. Quit the format utility when you have finished.

c. Use the tail command to view the contents of the /etc/format.dat file.

d. Note that your new partition information is appended to the file.

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

19. Create a directory called /vtoc.

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

```
# dd if=/dev/zero of=/dev/rdsk/c0t1d0s2 bs=512 count=1
1+0 records in
1+0 records out
#
```

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

    What happens?

    ____

    ____

    ____

23. Complete one of the following steps:
    - If the prtvtoc command reported an "Unable to read Disk geometry" message, use the format utility to place a default label on the disk for which you destroyed the label earlier.
    - If the prtvtoc command reports that only slice 2 exists, or slice 0 and 2 exist on the disk, proceed to step 24.
24. Use the fmthard command to write to the disk the label information you saved earlier.
25. Attempt to read the label from the same disk.

    Was this successful?

    ____

Task 4 - Working With Disks and Partitions on x86/x64

Complete the following steps:

1. Log in as the `root` user and complete the following steps:
   a. Open a terminal window.
   b. Run the `format` utility.
2. Complete the following steps:
   a. Record the list of disks presented by the `format` utility. For example, `c1d0` and `c2d0`.
   b. Press **`Control-D`** to exit the `format` utility.
3. Use the `prtvtoc` command to list the VTOC for each of the disks you found in the previous step, and complete the following steps:
   a. If the `prtvtoc` command displays an error similar to this example, you have identified the unused disk:

```
# prtvtoc /dev/rdsk/c2d0s2
prtvtoc: /dev/rdsk/c2d0s2: Unable to read Disk geometry errno = 0x5
#
```

   If this error displays, record the name of this disk and skip steps b and c.

   b. If the `prtvtoc` command displays a partition table for both disks, examine the `Mount Directory` field in the command output.
   c. Record the name of a disk that has no mount directory listed.

   For your classroom environment, this is an unused disk.

Note – Systems in production can use unmounted slices to store production data. Do not assume that unmounted slices are unused on systems in production.

Unused disk:

____

4. Run the `format` utility again, and select the unused disk from the list of disks presented.
   a. If the disk has no `fdisk` partition table defined, `format` asks you to select a drive type. Select the `DEFAULT` entry. The `FORMAT` menu displays.

   b. If the disk has a defined `fdisk` partition table, the FORMAT menu displays:

5. Display the `fdisk` menu and complete the following steps to create one Solaris `fdisk` partition that uses the whole disk:

   a. If the disk has no defined `fdisk` partition table, the `fdisk` menu asks if you want to accept the default Solaris partition. Enter `n` to decline, and then edit the partition table.

   b. If the disk has a defined `fdisk` partition table, the table displays when you enter the `fdisk` menu.

   c. Delete all `fdisk` partitions that exist.

6. Use the `fdisk` menu to create one Solaris `fdisk` partition that uses the whole disk.

   a. Select the `SOLARIS2` partition type

   b. Enter 100 to specify using 100% of the disk

   c. Enter `n` to prevent this partition from being the active partition

   d. Select item 5 from the `fdisk` menu to update the disk configuration, and exit the `fdisk` menu.

7. Display the `partition` menu and complete the following steps:

   a. Print the current partition table.

   b. Record the number of gigabytes (Gbytes) assigned to Slice 2.

Note – Various slices may already be defined. This task replaces all slices that already exist.

   Gbytes:

   ____

8. Divide the number of Gbytes by 4 and complete the following steps:

   a. Use the result as the number of Gbytes to assign as disk space to four slices.

   b. Round *down* to the next whole Gbyte if the result includes a fraction.

      For example, if the slice 2 is 76.67 Gbytes, 76.67 / 4 = 19.16 Gbytes. In this case use 19 Gbytes as the size for the four slices.

      Gbytes/4:

   ____

9. Display the PARTITION menu again and complete the following steps:
   a. Select Slice 0.
   b. Accept the defaults for tags and flags.
   c. Use the default starting cylinder. Typically this is cylinder 3.
   d. Enter the resulting number of Gbytes from the previous step for the slice size.
   e. Print the partition table again to verify the change.
10. Attempt to select slice 8 and 9 for modification. What happens?

    ____

11. Set the sizes of Slices 1, 3, and 4 so that they are the same as Slice 0.

    You should begin each successive slice on the cylinder that follows the ending cylinder of the previous slice. For example:
12. Set Slices 5, 6, and 7 to start at cylinder 0, and then assign them 0 Mbytes.
13. Complete the following steps:
    a. Print the partition table.

    Except for slice 2, is there any overlap of ending and beginning cylinders for any of the slices listed?

    ____

    b. Proceed to the following steps to introduce this problem.
14. Complete the following steps:
    a. Add 1 to the number Gbytes/4 value listed in Step 6.

       (Gbytes/4) + 1:

    ____

    b. Change Slice 0 so that it uses the new size.

    The Cylinders column in the partition table should now indicate that Slice 0 ends after Slice 1 begins.
15. Complete the following steps:
    a. Use the modify command from the partition menu to attempt to fix this problem.
    b. Select Item 0 to modify the current partition table.

    Which warnings display?

    ____

16. Modify the partition table and select Item `1` to use the `All Free Hog` option.

    The partition table displays.

17. Observe the `Cylinders` and `Size` columns, and then verify that they are all zero except for slice 2, 8, and possibly 9.

18. Respond to the prompts to continue the process and complete the following steps:

    a. Select Slice 4 as the `All Free Hog` partition.

    b. Use the size listed in Step 8 for Slices 0, 1, and 3.

    c. Enter Return to set the other slices to size 0.

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

19. Name the partition table `"MYDISKpartition"`, and then label the disk.

20. Save your new partition table to the `/etc/format.dat` file and complete the following steps:

    a. Carefully read the message that the `format` utility displays, and then enter the correct file name.

    b. Quit the `format` utility when you have finished.

    c. Use the `tail` command to view the contents of the `/etc/format.dat` file.

    d. Note that your new partition information is appended to the file.

21. Verify your new partition table with the `prtvtoc` command.

22. Create two directories, one called `/vtoc` and another called `/fdisk`.

23. Use the `prtvtoc` command to print the partition table that you just created and complete the following steps:

    a. Save its output to a file in the `/vtoc` directory.

    b. Name the file so that it corresponds with the disk you are examining.

    c. Use the `cat` command to verify that valid information exists in the file that you create.

24. Verify the `fdisk` partition table for your spare disk with the `fdisk` command.

25. Use the fdisk command to print the fdisk partition table and complete the following steps:
    a. Save its output to a file in the /fdisk directory.
    b. Name the file so that it corresponds with the disk you are examining.
    c. Use the cat command to verify that valid information exists in the file that you create.
26. Use the following dd command to destroy the x86/x64 VTOC and complete the following steps:
    a. Be certain to specify the correct disk device name for the of= argument.
    b. Enter all other arguments exactly as listed.

```
# dd if=/dev/zero of=/dev/rdsk/c2d0s2 bs=512 count=2
2+0 records in
2+0 records out
#
```

27. Attempt to read the label from the same disk by using the prtvtoc command. What happens?

    ____

28. Run the format utility, and perform the following steps:
    a. Select your spare disk.
    b. Display the fdisk menu.
    c. Delete the defined Solaris fdisk partition
    d. Save the fdisk partition table, exit the fdisk menu, and quit the format utility.
29. Attempt to use the fmthard command to write to the disk the x86/x64 VTOC information you saved earlier. What happens?

    ____

30. Use the fdisk command to replace the fdisk partition table with the fdisk information you saved earlier.
31. Use the fmthard command to write to the disk the label information you saved earlier.
32. Attempt to read the label from the same disk.

    Was this successful?

    ____

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions - Managing Local Disk Devices

This section contains solutions to the exercise.

Task 1 – Identifying Device Files

Complete the following steps:

1. Log in as the `root` user and complete the following steps:
   a. Open a terminal window.
   b. Expand the window so that it occupies the entire screen area.
   c. Change to the `/dev/dsk` directory.

   ```
   # cd /dev/dsk
   ```
2. List the files in this directory and complete the following steps:
   a. Identify the files related to the boot disk of your system.

      Systems typically use `c0t0d0` or `c1d0`, but this may vary.

   b. Locate the file that represents slice 0 on this disk and display a long listing of it.

   ```
   # ls
   ```

   Sun Blade 1500 example:

   ```
   # ls -l c0t0d0s0
   ```

   Ultra 20 example:

   ```
   # ls -l c1d0s0
   ```

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

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

   c. Record the full path name to which this file points.

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

   Sun Blade 1500 example:

   ```
   ../../devices/pci@1e,600000/ide@d/dad@0,0:a
   ```

   Ultra 20 example:

   ```
   ../../devices/pci@0,0/pci-ide@7/ide@0/cmdk@0,0:a
   ```

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

   Sun Blade 1500 example:

```
# ls -l ../../devices/pci@1e,600000/ide@d/dad@0,0:a
```

   Ultra 20 example:

```
# ls -l ../../devices/pci@0,0/pci-ide@7/ide@0/cmdk@0,0:a
```

   Which type of file is this?

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

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

4. Change to the `/dev/rdsk` directory and complete the following steps:

   a. Display a long listing of the same file name you selected in Step 2.

   ```
   # cd /dev/rdsk
   ```

   Sun Blade 1500 example:

   ```
   # ls -l c0t0d0s0
   ```

   Ultra 20 example:

   ```
   # ls -l c1d0s0
   ```

   Which type of file is this?

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

   b. Record the full path name to which this file points.

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

   Sun Blade 1500 example:

   ```
   ../../devices/pci@1e,600000/ide@d/dad@0,0:a,raw
   ```

   Ultra 20 example:

   ```
   ../../devices/pci@0,0/pci-ide@7/ide@0/cmdk@0,0:a,raw
   ```

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

   Sun Blade 1500 example:

```
# ls -l ../../devices/pci@1e,600000/ide@d/dad@0,0:a,raw
```

   Ultra 20 example:

```
# ls -l ../../devices/pci@0,0/pci-ide@7/ide@0/cmdk@0,0:a,raw
```

   Which type of file is this?

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

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

6. Change to the `/etc` directory and display the contents of the `path_to_inst` file.

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

7. Use the information from the previous steps to locate and record the entry for your boot disk.

   A Sun Blade 1500 workstation, for example, would typically use `c0t0d0` as its boot disk. This relates to the device file called `dad@0,0` and is listed in the `/etc/path_to_inst` file.

   Systems that use PCI bus architectures may have entries in `/etc/path_to_inst` *similar to the following example from a Sun Blade 1500:*

```
"/pci@1e,600000/ide@d/dad@0,0" 1 "dad"
```

   Instance names are composed of the last two fields of each line in `/etc/path_to_inst`. Examples include the `dad`, `cmdk`, or `sd` tags, and the number that precedes them in `/etc/path_to_inst`.

   What is the instance name for the device listed in this step?

   The instance name for the device could be `dad1`, `dad0`, `cmdk0`, `sd3`, *or* `sd0`, *depending on the system architecture.*

Task 2 – Adding a New Disk or Tape Device

Complete the following steps:

Note – This task is optional. Perform the steps in this task if an external disk or tape device is available on your system.

---

1. In the `/dev/dsk` or `/dev/rmt` directories, confirm that no files exist for your external disk or tape device; for example, `/dev/dsk/c1t0d0s0` or `/dev/rmt/0`.

   If files for the external device do exist, ask your instructor for guidance. Do not simply remove device files that exist. Refer to the Preparation section for advice about removing device files.

2. If you are using a SPARC system, shut it down to run state 0.

   ```
   # init 0
   ```

3. If you are using an x86/x64 system, shut it down to run state 5. This shuts down the system and turns off the power.

   ```
   # init 5
   ```

4. Power on the external disk or tape drive attached to your system.

5. If you are using an x86/x64 system, power on your system and allow it to boot to its default run state.

6. If you are using a SPARC system, boot the system to its default run state.

   ```
   ok boot
   ```

7. Log in as the `root` user and complete the following steps:

   a. Open a terminal window.

   b. Run the `devfsadm` command with the `-v` option to create new links and device files for the new disk or tape drive.

   ```
   # devfsadm -v
   ```

   c. Observe the messages that the `devfsadm` command displays.

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

   Use the `devfsadm -v` output to determine what files to look for.

Task 3 - Working With Disks and Partitions on SPARC

Complete the following steps:

1. Log in as the `root` user and complete the following steps:
   a. Open a terminal window.
   b. Run the `format` utility.

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


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@0,0
       1. c0t1d0 <DEFAULT cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@1,0
Specify disk (enter its number):
```

2. Complete the following steps:
   a. Record the list of disks presented by the `format` utility. For example, `c0t0d0` and `c0t1d0`.
   b. Press **`Control-D`** to exit the `format` utility.

```
Specify disk (enter its number): ^D
#
```

3. Use the `prtvtoc` command to list the VTOC for each of the disks you found in the previous step and complete the following steps:
   a. Examine the `Mount Directory` field in the information that the `prtvtoc` command displays.
   b. Record the name of a disk that has no mount directory listed.

      For your classroom environment, this is an unused disk.

Note – Systems in production can use unmounted slices to store production data. Do not assume that unmounted slices are unused on systems in production. Systems will display various lists of partitions, so the list you see may differ from the list the examples.

```
# prtvtoc /dev/rdsk/c0t1d0s2
* /dev/rdsk/c0t1d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*     255 sectors/track
*      16 tracks/cylinder
*    4080 sectors/cylinder
*   57461 cylinders
*   57459 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00          0    265200    265199
       1      3    01     265200    265200    530399
       2      5    01          0 234432720 234432719
       6      4    00     530400 233902320 234432719
#
```

Unused disk: *The disk you identify will depend on your specific system.*

4. Run the `format` utility again and select the unused disk from the list of disks presented.

```
# format

Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@0,0
       1. c0t1d0 <DEFAULT cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@1,0
Specify disk (enter its number):1
selecting c0t1d0
[disk formatted, no defect list found]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
```

```
        format     - format and analyze the disk
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format>
```

5. Display the `partition` menu and complete the following steps:
   a. Print the current partition table.
   b. Record the number of gigabytes (Gbytes) assigned to Slice 2.

```
format> part

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

```
partition> print
Current partition table (original):
Total disk cylinders available: 57459 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 -    64      129.49MB    (65/0/0)        265200
  1       swap    wu      65 -   129      129.49MB    (65/0/0)        265200
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  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     130 - 57458      111.53GB    (57329/0/0) 233902320
  7 unassigned    wm       0                0         (0/0/0)              0

partition>
```

Note – The list of defined partitions will vary, depending on how the spare disk was last used.

Gbytes: *Your entry will depend on the disks present in your system.*

6. Divide the number of Gbytes by 4 and complete the following steps:
   a. Use the result as the number of Gbytes to assign as disk space to four slices.
   b. Round *down* to the next whole Gbyte if the result includes a fraction.

      For example, if the slice 2 is 111.79 Gbytes, 111.79 / 4 = 27.94 Gbytes. In this case use 27 Gbytes as the size for the four slices.

      Gbytes/4: *Your entry will depend on your system.*
7. Display the Partition menu again and complete the following steps:
   a. Select Slice 0.
   b. Accept the defaults for tags and flags.
   c. Start this first slice on Cylinder 0.
   d. Enter the resulting number of Gbytes from the previous step for the slice size.

e. Print the partition table again to verify the change.

```
partition> 0
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 -    64      129.49MB    (65/0/0)
265200

Enter partition id tag[root]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 0
Enter partition size[265200b, 65c, 64e, 129.49mb, 0.13gb]: 27gb
partition> print
Current partition table (unnamed):
Total disk cylinders available: 57459 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)  56626320
  1       swap    wu      65 -   129      129.49MB    (65/0/0)       265200
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  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     130 - 57458      111.53GB    (57329/0/0) 233902320
  7 unassigned    wm       0                0         (0/0/0)             0

partition>
```

8. Set the sizes of Slices 1, 3, and 4 so that they are the same as Slice 0.

You should begin each successive slice on the cylinder that follows the ending cylinder of the previous slice. For example:

```
partition> 1
Part      Tag    Flag     Cylinders         Size            Blocks
  1       swap    wu      65 -   129      129.49MB    (65/0/0)
265200

Enter partition id tag[swap]: <Return>
Enter partition permission flags[wu]: <Return>
Enter new starting cyl[65]: 13879
Enter partition size[265200b, 65c, 13943e, 129.49mb, 0.13gb]: 27gb
partition> print
Current partition table (unnamed):
Total disk cylinders available: 57459 + 2 (reserved cylinders)
```

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)  56626320
  1       swap    wu   13879 - 27757       27.00GB    (13879/0/0)  56626320
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  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     130 - 57458      111.53GB    (57329/0/0) 233902320
  7 unassigned    wm       0                0         (0/0/0)              0

partition>
```

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

```
partition> 6
Part      Tag    Flag     Cylinders         Size            Blocks
  6       usr     wm     130 - 57458      111.53GB    (57329/0/0)
233902320

Enter partition id tag[usr]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[130]: 0
Enter partition size[233902320b, 57329c, 57328e, 114210.12mb, 111.53gb]:
0
partition>
```

10. Complete the following steps:
    a. Print the partition table.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 57459 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)  56626320
  1       swap    wu   13879 - 27757       27.00GB    (13879/0/0)  56626320
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm   27758 - 41636       27.00GB    (13879/0/0)  56626320
  4 unassigned    wm   41637 - 55515       27.00GB    (13879/0/0)  56626320
  5 unassigned    wm       0                0         (0/0/0)              0
  6 unassigned    wm       0                0         (0/0/0)              0
  7 unassigned    wm       0                0         (0/0/0)              0

partition>
```

Except for slice 2, is there any overlap of ending and beginning cylinders for any of the slices listed?

No.

b. Proceed to the following steps to introduce this problem.

11. Complete the following steps:

a. Add 1 to the number Gbytes/4 value listed in Step 6.

(Gbytes/4) + 1: *Your entry will depend on your system.*

b. Change Slice 0 so that it uses the new size.

```
partition> 0
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)
56626320

Enter partition id tag[root]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 0
Enter partition size[56626320b, 13879c, 13878e, 27649.57mb, 27.00gb]:
28gb
partition> print
Current partition table (unnamed):
Total disk cylinders available: 57459 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 14392       28.00GB    (14393/0/0)  58723440
  1       swap    wu   13879 - 27757       27.00GB    (13879/0/0)  56626320
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm   27758 - 41636       27.00GB    (13879/0/0)  56626320
  4 unassigned    wm   41637 - 55515       27.00GB    (13879/0/0)  56626320
  5 unassigned    wm       0                0         (0/0/0)             0
  6 unassigned    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0

partition>
```

The `Cylinders` column in the partition table should now indicate that Slice 0 ends after Slice 1 begins.

12. Complete the following steps:

a. Use the `modify` command from the `partition` menu to attempt to fix this problem.

b. Select Item `0` to modify the current partition table.

```
partition> ?
Expecting one of the following: (abbreviations ok):
        0      - change '0' partition
        1      - change '1' partition
        2      - change '2' partition
        3      - change '3' partition
```

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

partition> modify
Select partitioning base:
        0. Current partition table (unnamed)
        1. All Free Hog
Choose base (enter number) [0]? 0
Warning: Overlapping partition (1) in table.
Warning: Fix, or select a different partition table.
partition>
```

Which warnings display?

`Warning: Overlapping partition (1) in table.`
`Warning: Fix, or select a different partition table.`

13. Modify the partition table and select Item `1` to use the `All Free Hog` option.

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

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0                0         (0/0/0)            0
  1       swap    wu       0                0         (0/0/0)            0
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm       0                0         (0/0/0)            0
  4 unassigned    wm       0                0         (0/0/0)            0
  5 unassigned    wm       0                0         (0/0/0)            0
  6        usr    wm       0                0         (0/0/0)            0
  7 unassigned    wm       0                0         (0/0/0)            0

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

The partition table displays.

14. Observe the `Cylinders` and `Size` columns, and then verify that they are all zero except for slice 2.
15. Respond to the prompts to continue the process and complete the following steps:
    a. Select Slice 4 as the `All Free Hog` partition.
    b. Use the size listed in Step 6 for Slices 0, 1, and 3.
    c. Enter Return to set the other slices to size 0.

```
table based on above table[yes]? y
Free Hog partition[6]? 4
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 27gb
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 27gb
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]: 27gb
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]: <Return>
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]: <Return>
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]: <Return>

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)  56626320
  1       swap    wu   13879 - 27757       27.00GB    (13879/0/0)  56626320
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm   27758 - 41636       27.00GB    (13879/0/0)  56626320
  4 unassigned    wm   41637 - 57458       30.78GB    (15822/0/0)  64553760
  5 unassigned    wm       0                0         (0/0/0)             0
  6        usr    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0

Okay to make this the current partition table[yes]?
```

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

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

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

Ready to label disk, continue? y

partition> q


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format>
```

17. Save your new partition table to the /etc/format.dat file and complete the following steps:

   a. Carefully read the message that the format utility displays, and then enter the correct file name.

   b. Quit the format utility when you have finished.

   c. Use the tail command to view the contents of the /etc/format.dat file.

d. Note that your new partition information is appended to the file.

```
format> save
Saving new disk and partition definitions
Enter file name["./format.dat"]: /etc/format.dat
format> quit
# tail /etc/format.dat
disk_type = "DEFAULT" \
         : ctlr = ata : ncyl = 57459 : acyl = 2 : pcyl = 57461 \
         : nhead = 16 : nsect = 255 : rpm = 5400

partition = "MYDISKpartition" \
         : disk = "DEFAULT" : ctlr = ata \
         : 0 =  0, 56626320 : 1 =  13879, 56626320 : 2 =  0, 234432720 \
         : 3 =  27758, 56626320 : 4 =  41637, 64553760

#
```

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

```
# prtvtoc /dev/rdsk/c0t1d0s2
* /dev/rdsk/c0t1d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*     255 sectors/track
*      16 tracks/cylinder
*    4080 sectors/cylinder
*   57461 cylinders
*   57459 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00          0   56626320  56626319
       1      3    01   56626320  56626320 113252639
       2      5    01          0 234432720 234432719
       3      0    00  113252640  56626320 169878959
       4      0    00  169878960  64553760 234432719
#
```

19. Create a directory called /vtoc.

```
# mkdir /vtoc
```

20. Use the prtvtoc command to print the partition table that you just created and complete the following steps:

    a. Save its output to a file in the /vtoc directory.

    b. Name the file so that it corresponds with the disk you are examining.

    c. Use the cat command to verify that valid information exists in the file that you create.

```
# prtvtoc /dev/rdsk/c0t1d0s2 > /vtoc/c0t1d0
# cat /vtoc/c0t1d0
* /dev/rdsk/c0t1d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*     255 sectors/track
*      16 tracks/cylinder
*    4080 sectors/cylinder
*   57461 cylinders
*   57459 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00          0  56626320  56626319
       1      3    01   56626320  56626320 113252639
       2      5    01          0 234432720 234432719
       3      0    00  113252640  56626320 169878959
       4      0    00  169878960  64553760 234432719
#
```

21. Use the following dd command to destroy the disk label and complete the following steps:

    a. Be certain to specify the correct disk device name for the of= argument.

    b. Enter all other arguments exactly as listed.

```
# dd if=/dev/zero of=/dev/rdsk/c0t1d0s2 bs=512 count=1
1+0 records in
1+0 records out
#
```

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

```
# prtvtoc /dev/rdsk/c0t1d0s2
```

What happens?

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

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

IDE disks might report a partition table where only slice 2, or slice 0 and 2 remain defined. For example:

```
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       2      5    01          0  17801280  17801279
```

Example from a Sun Blade 1500:

```
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      0    00          0 234432720 234432719
       2      5    01          0 234432720 234432719
#
```

23. Complete one of the following steps:

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

Note – The device in this example, `c1t3d0`, is a disk of the type that would match the condition described above.

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

c1t3d0: configured with capacity of 8.43GB


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

```
[disk formatted]
Disk not labeled.  Label it now? y

(format menu)
format> q
#
# prtvtoc /dev/rdsk/c1t3d0s2
(output omitted)
```

- If the prtvtoc command reports that only slice 2 exists, or slice 0 and 2 exist on the disk, proceed to step 24.

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

```
# fmthard -s /vtoc/c0t1d0 /dev/rdsk/c0t1d0s2
fmthard:  New volume table of contents now in place.
#
```

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

```
# prtvtoc /dev/rdsk/c0t1d0s2
* /dev/rdsk/c0t1d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*     255 sectors/track
*      16 tracks/cylinder
*    4080 sectors/cylinder
*   57461 cylinders
*   57459 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00          0  56626320  56626319
       1      3    01   56626320  56626320 113252639
       2      5    01          0 234432720 234432719
       3      0    00  113252640  56626320 169878959
       4      0    00  169878960  64553760 234432719
#
```

Was this successful?

This command should successfully read the disk label.

Task 4 - Working With Disks and Partitions on x86/x64

Complete the following steps:

1. Log in as the `root` user and complete the following steps:
   a. Open a terminal window.
   b. Run the `format` utility.

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


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 39887 alt 2 hd 64 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
       1. c2d0 <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
Specify disk (enter its number):
```

2. Complete the following steps:
   a. Record the list of disks presented by the `format` utility. For example, `c1d0` and `c2d0`.
   b. Press **`Control-D`** to exit the `format` utility.

```
Specify disk (enter its number): ^D
#
```

3. Use the `prtvtoc` command to list the VTOC for each of the disks you found in the previous step, and complete the following steps:
   a. If the `prtvtoc` command displays an error similar to this example, you have identified the unused disk:

```
# prtvtoc /dev/rdsk/c2d0s2
prtvtoc: /dev/rdsk/c2d0s2: Unable to read Disk geometry errno = 0x5
#
```

   If this error displays, record the name of this disk and skip steps b and c.

   b. If the `prtvtoc` command displays a partition table for both disks, examine the `Mount Directory` field in the command output.
   c. Record the name of a disk that has no mount directory listed.

   For your classroom environment, this is an unused disk.

Note – Systems in production can use unmounted slices to store production data. Do not assume that unmounted slices are unused on systems in production.

```
# prtvtoc /dev/rdsk/c2d0s2
* /dev/rdsk/c2d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*   10010 cylinders
*   10008 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0     11    00      48195    819315    867509
       1      3    01     867510   1060290   1927799
       2      5    01          0 160778520 160778519
       7      8    00    1927800 158850720 160778519
       8      1    01          0     16065     16064
       9      9    00      16065     32130     48194
#
```

Unused disk: *The disk you identify will depend on your specific system.*

4. Run the `format` utility again, and select the unused disk from the list of disks presented.

   a. If the disk has no `fdisk` partition table defined, `format` asks you to select a drive type. Select the `DEFAULT` entry. The `FORMAT` menu displays.

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


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 10002 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
       1. c2d0 <drive type unknown>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
```

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



AVAILABLE DRIVE TYPES:
        0. DEFAULT
        1. other
Specify disk type (enter its number): 0
selecting c2d0
No current partition list
No defect list found
[disk formatted, no defect list found]


FORMAT MENU:
        disk       - select a disk

(output omitted)

format>
```

b. If the disk has a defined `fdisk` partition table, the FORMAT menu displays:

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


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 39887 alt 2 hd 64 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
       1. c2d0 <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
Specify disk (enter its number): 1
selecting c2d0
Controller working list found
[disk formatted, defect list found]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
```

```
        fdisk      - run the fdisk program
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format>
```

5. Display the `fdisk` menu and complete the following steps to create one Solaris `fdisk` partition that uses the whole disk:

   a. If the disk has no defined `fdisk` partition table, the `fdisk` menu asks if you want to accept the default Solaris partition. Enter `n` to decline, and then edit the partition table.

```
format> fdisk
No fdisk table exists. The default partition for the disk is:

  a 100% "SOLARIS System" partition

Type "y" to accept the default partition,  otherwise type "n" to edit the
 partition table. n
             Total disk size is 10011 cylinders
             Cylinder size is 16065 (512 byte) blocks

                                               Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===


WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection:
```

b. If the disk has a defined `fdisk` partition table, the table displays when you enter the `fdisk` menu.

```
format> fdisk

             Total disk size is 10011 cylinders
             Cylinder size is 16065 (512 byte) blocks

                                                   Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1                 Solaris2          1  10010   10010    100

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection:
```

c. Delete all `fdisk` partitions that exist. For example:

```
Enter Selection: 3
Specify the partition number to delete (or enter 0 to exit): 1

Are you sure you want to delete partition 1? This will make all files and
programs in this partition inaccessible (type "y" or "n").

y

             Total disk size is 10011 cylinders
             Cylinder size is 16065 (512 byte) blocks

                                                   Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
```

```
Enter Selection:


Partition 1 has been deleted.
```

6. Use the fdisk menu to create one Solaris fdisk partition that uses the whole disk.

```
Enter Selection: 1
```

a. Select the SOLARIS2 partition type

```
Select the partition type to create: 1
   1=SOLARIS2  2=UNIX        3=PCIXOS     4=Other
   5=DOS12     6=DOS16       7=DOSEXT     8=DOSBIG
   9=DOS16LBA  A=x86 Boot    B=Diagnostic C=FAT32
   D=FAT32LBA  E=DOSEXTLBA   F=EFI        0=Exit?
```

b. Enter 100 to specify using 100% of the disk

```
Specify the percentage of disk to use for this partition
(or type "c" to specify the size in cylinders). 100
```

c. Enter n to prevent this partition from being the active partition

```
Should this become the active partition? If yes, it  will be activated
each time the computer is reset or turned on.
Please type "y" or "n". n

                 Total disk size is 10011 cylinders
                 Cylinder size is 16065 (512 byte) blocks

                                               Cylinders
        Partition   Status    Type          Start   End   Length    %
        =========   ======    ============  =====   ===   ======   ===
            1                 Solaris2          1  10010    10010    100


SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection:
```

d. Select item 5 from the fdisk menu to update the disk configuration, and exit the fdisk menu.

```
Enter Selection: 5
```

```
format>
```

7. Display the `partition` menu and complete the following steps:
   a. Print the current partition table.
   b. Record the number of gigabytes (Gbytes) assigned to Slice 2.

```
format> part


PARTITION MENU:
        0      - change '0' partition
        1      - change '1' partition
        2      - change '2' partition
        3      - change '3' partition
        4      - change '4' partition
        5      - change '5' partition
        6      - change '6' partition
        7      - change '7' partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        !<cmd> - execute <cmd>, then return
        quit
partition> print
Current partition table (original):
Total disk cylinders available: 10008 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 -    53      400.06MB    (51/0/0)       819315
  1       swap    wu      54 -   119      517.72MB    (66/0/0)      1060290
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7       home    wm     120 - 10007       75.75GB    (9888/0/0)  158850720
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130

partition>
```

Note – Various slices may already be defined. This task replaces all slices that already exist.

Gbytes: *Your entry will depend on the disks present in your system.*

8. Divide the number of Gbytes by 4 and complete the following steps:
   a. Use the result as the number of Gbytes to assign as disk space to four slices.
   b. Round *down* to the next whole Gbyte if the result includes a fraction.

      For example, if the slice 2 is 76.67 Gbytes, 76.67 / 4 = 19.16 Gbytes. In this case use 19 Gbytes as the size for the four slices.

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

9. Display the PARTITION menu again and complete the following steps:
   a. Select Slice 0.
   b. Accept the defaults for tags and flags.
   c. Use the default starting cylinder. Typically this is cylinder 3.
   d. Enter the resulting number of Gbytes from the previous step for the slice size.
   e. Print the partition table again to verify the change.

```
partition> 0
Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 -    53      400.06MB    (51/0/0)
819315

Enter partition id tag[reserved]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[3]: 3
Enter partition size[819315b, 51c, 53e, 400.06mb, 0.39gb]: 19gb
partition> print
Current partition table (unnamed):
Total disk cylinders available: 10008 + 2 (reserved cylinders)
```

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 -  2483       19.01GB    (2481/0/0)   39857265
  1       swap    wu      54 -   119      517.72MB    (66/0/0)      1060290
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7       home    wm     120 - 10007       75.75GB    (9888/0/0)  158850720
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130

partition>
```

10. Attempt to select slice 8 and 9 for modification. What happens?

```
partition> 8
'8' is not expected.
partition> 9
'9' is not expected.
partition>
```

The `format` utility prevents you from modifying these two pre-defined slices.

11. Set the sizes of Slices 1, 3, and 4 so that they are the same as Slice 0.

You should begin each successive slice on the cylinder that follows the ending cylinder of the previous slice. For example:

```
partition> 1
Part      Tag    Flag     Cylinders         Size            Blocks
  1      swap     wu      54 -   119      517.72MB    (66/0/0)
1060290

Enter partition id tag[swap]: <Return>
Enter partition permission flags[wu]: <Return>
Enter new starting cyl[54]: 2484
Enter partition size[1060290b, 66c, 2549e, 517.72mb, 0.51gb]: 19gb
partition> print
Current partition table (unnamed):
Total disk cylinders available: 10008 + 2 (reserved cylinders)
```

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm        3 -  2483       19.01GB    (2481/0/0)   39857265
  1       swap    wu     2484 -  4964       19.01GB    (2481/0/0)   39857265
  2     backup    wu        0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wu        0                0         (0/0/0)             0
  4 unassigned    wu        0                0         (0/0/0)             0
  5 unassigned    wu        0                0         (0/0/0)             0
  6 unassigned    wu        0                0         (0/0/0)             0
  7       home    wm      120 - 10007       75.75GB    (9888/0/0)  158850720
  8       boot    wu        0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm        1 -     2       15.69MB    (2/0/0)         32130

partition>
```

12. Set Slices 5, 6, and 7 to start at cylinder 0, and then assign them 0 Mbytes.

```
partition> 7
Part      Tag    Flag     Cylinders         Size            Blocks
  7       home    wm      120 - 10007       75.75GB    (9888/0/0)
158850720

Enter partition id tag[home]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[120]: 0
Enter partition size[158850720b, 9888c, 9887e, 77563.83mb, 75.75gb]: 0
partition>
```

13. Complete the following steps:

    a. Print the partition table.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 10008 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm        3 -  2483       19.01GB    (2481/0/0)   39857265
  1       swap    wu     2484 -  4964       19.01GB    (2481/0/0)   39857265
  2     backup    wu        0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wu     4965 -  7445       19.01GB    (2481/0/0)   39857265
  4 unassigned    wu     7446 -  9926       19.01GB    (2481/0/0)   39857265
  5 unassigned    wu        0                0         (0/0/0)             0
  6 unassigned    wu        0                0         (0/0/0)             0
  7 unassigned    wm        0                0         (0/0/0)             0
  8       boot    wu        0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm        1 -     2       15.69MB    (2/0/0)         32130

partition>
```

Except for slice 2, is there any overlap of ending and beginning cylinders for any of the slices listed?

No.

b. Proceed to the following steps to introduce this problem.

14. Complete the following steps:

a. Add 1 to the number Gbytes/4 value listed in Step 6.

(Gbytes/4) + 1: *Your entry will depend on your system.*

b. Change Slice 0 so that it uses the new size.

```
partition> 0
Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 -  2483       19.01GB    (2481/0/0)
39857265

Enter partition id tag[reserved]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[3]: 3
Enter partition size[39857265b, 2481c, 2483e, 19461.55mb, 19.01gb]: 20gb
partition> print
Current partition table (unnamed):
Total disk cylinders available: 10008 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 -  2613       20.00GB    (2611/0/0)   41945715
  1       swap    wu    2484 -  4964       19.01GB    (2481/0/0)   39857265
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wu    4965 -  7445       19.01GB    (2481/0/0)   39857265
  4 unassigned    wu    7446 -  9926       19.01GB    (2481/0/0)   39857265
  5 unassigned    wu       0                0         (0/0/0)              0
  6 unassigned    wu       0                0         (0/0/0)              0
  7 unassigned    wm       0                0         (0/0/0)              0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)          16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)          32130

partition>
```

The `Cylinders` column in the partition table should now indicate that Slice 0 ends after Slice 1 begins.

15. Complete the following steps:

   a. Use the `modify` command from the `partition` menu to attempt to fix this problem.

   b. Select Item `0` to modify the current partition table.

```
partition> ?
Expecting one of the following: (abbreviations ok):
        0      - change '0' partition
        1      - change '1' partition
        2      - change '2' partition
        3      - change '3' partition
        4      - change '4' partition
        5      - change '5' partition
        6      - change '6' partition
        7      - change '7' partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        !<cmd> - execute <cmd>, then return
        quit

partition> modify
Select partitioning base:
        0. Current partition table (unnamed)
        1. All Free Hog
Choose base (enter number) [0]? 0
Warning: Overlapping partition (1) in table.
Warning: Fix, or select a different partition table.
partition>
```

   Which warnings display?

   ```
   Warning: Overlapping partition (1) in table.
   Warning: Fix, or select a different partition table.
   ```

16. Modify the partition table and select Item `1` to use the `All Free Hog` option.

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

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0                 0         (0/0/0)            0
  1       swap    wu       0                 0         (0/0/0)            0
  2     backup    wu       0 - 10007        76.67GB    (10008/0/0) 160778520
  3 unassigned    wm       0                 0         (0/0/0)            0
  4 unassigned    wm       0                 0         (0/0/0)            0
  5 unassigned    wm       0                 0         (0/0/0)            0
  6        usr    wm       0                 0         (0/0/0)            0
  7 unassigned    wm       0                 0         (0/0/0)            0
  8       boot    wu       0 -     0         7.84MB    (1/0/0)        16065
  9 alternates    wm       1 -     2        15.69MB    (2/0/0)        32130

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

The partition table displays.

17. Observe the `Cylinders` and `Size` columns, and then verify that they are all zero except for slice 2, 8, and possibly 9.

18. Respond to the prompts to continue the process and complete the following steps:
    a. Select Slice 4 as the `All Free Hog` partition.
    b. Use the size listed in Step 8 for Slices 0, 1, and 3.
    c. Enter Return to set the other slices to size 0.

```
table based on above table[yes]? y
Free Hog partition[6]? 4
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 19gb
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 19gb
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]: 19gb
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]: <Return>
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]: <Return>
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]: <Return>

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       3 -  2483      19.01GB    (2481/0/0)   39857265
  1       swap    wu    2484 -  4964      19.01GB    (2481/0/0)   39857265
  2     backup    wu       0 - 10007      76.67GB    (10008/0/0) 160778520
  3 unassigned    wm    4965 -  7445      19.01GB    (2481/0/0)   39857265
  4 unassigned    wm    7446 - 10007      19.63GB    (2562/0/0)   41158530
  5 unassigned    wm       0               0         (0/0/0)             0
  6        usr    wm       0               0         (0/0/0)             0
  7 unassigned    wm       0               0         (0/0/0)             0
  8       boot    wu       0 -     0       7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2      15.69MB    (2/0/0)         32130

Okay to make this the current partition table [yes]?
```

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

19. Name the partition table "MYDISKpartition", and then label the disk.

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

Ready to label disk, continue? y

partition> q


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        fdisk      - run the fdisk program
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format>
```

20. Save your new partition table to the /etc/format.dat file and complete the following steps:

    a. Carefully read the message that the format utility displays, and then enter the correct file name.

    b. Quit the format utility when you have finished.

    c. Use the tail command to view the contents of the /etc/format.dat file.

d. Your new partition information is appended to the file.

```
format> save

Saving new disk and partition definitions
Enter file name["./format.dat"]: /etc/format.dat
format> quit
# tail /etc/format.dat
        : ctlr = ata : ncyl = 10008 : acyl = 2 : pcyl = 10010 \
        : nhead = 255 : nsect = 63 : rpm = 3600

partition = "MYDISKpartition" \
        : disk = "DEFAULT" : ctlr = ata \
        : 0 =  reserved, wm, 3, 39857265 : 1 =  2484, 39857265 \
        : 2 =  0, 160778520 : 3 =  unassigned, wu, 4965, 39857265 \
        : 4 =  unassigned, wu, 7446, 41158530 : 8 =  0, 16065 \
        : 9 =  1, 32130

#
```

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

```
# prtvtoc /dev/rdsk/c2d0s2
* /dev/rdsk/c2d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*   10010 cylinders
*   10008 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0     11    00       48195  39857265  39905459
       1      3    01    39905460  39857265  79762724
       2      5    01           0 160778520 160778519
       3      0    01    79762725  39857265 119619989
       4      0    01   119619990  41158530 160778519
       8      1    01           0     16065     16064
       9      9    00       16065     32130     48194
#
```

22. Create two directories, one called /vtoc and another called /fdisk.

```
# mkdir /vtoc
# mkdir /fdisk
```

23. Use the prtvtoc command to print the partition table that you just created and complete the following steps:

   a. Save its output to a file in the /vtoc directory.

   b. Name the file so that it corresponds with the disk you are examining.

   c. Use the cat command to verify that valid information exists in the file that you create.

```
# prtvtoc /dev/rdsk/c2d0s2 > /vtoc/c2d0
# cat  /vtoc/c2d0
* /dev/rdsk/c2d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*   10010 cylinders
*   10008 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0     11    00       48195  39857265  39905459
       1      3    01    39905460  39857265  79762724
       2      5    01           0 160778520 160778519
       3      0    01    79762725  39857265 119619989
       4      0    01   119619990  41158530 160778519
       8      1    01           0     16065     16064
       9      9    00       16065     32130     48194
#
```

24. Verify the fdisk partition table for your spare disk with the fdisk command.

```
# fdisk -W - /dev/rdsk/c2d0p0

* /dev/rdsk/c2d0p0 default fdisk table
* Dimensions:
*    512 bytes/sector
```

```
*      63 sectors/track
*     255 tracks/cylinder
*   10011 cylinders
*
* systid:
*     1: DOSOS12
*     2: PCIXOS
*     4: DOSOS16
*     5: EXTDOS
*     6: DOSBIG
*     7: FDISK_IFS
*     8: FDISK_AIXBOOT
*     9: FDISK_AIXDATA
*    10: FDISK_0S2BOOT
*    11: FDISK_WINDOWS
*    12: FDISK_EXT_WIN
*    14: FDISK_FAT95
*    15: FDISK_EXTLBA
*    18: DIAGPART
*    65: FDISK_LINUX
*    82: FDISK_CPM
*    86: DOSDATA
*    98: OTHEROS
*    99: UNIXOS
*   101: FDISK_NOVELL3
*   119: FDISK_QNX4
*   120: FDISK_QNX42
*   121: FDISK_QNX43
*   130: SUNIXOS
*   131: FDISK_LINUXNAT
*   134: FDISK_NTFSVOL1
*   135: FDISK_NTFSVOL2
*   165: FDISK_BSD
*   167: FDISK_NEXTSTEP
*   183: FDISK_BSDIFS
*   184: FDISK_BSDISWAP
*   190: X86BOOT
*   191: SUNIXOS2
*   238: EFI_PMBR
*   239: EFI_FS
*

* Id    Act  Bhead  Bsect  Bcyl    Ehead  Esect  Ecyl    Rsect    Numsect
  191   0    0      1      1       254    63     1023    16065
160810650
#
```

25. Use the `fdisk` command to print the `fdisk` partition table and complete the following steps:
    a. Save its output to a file in the `/fdisk` directory.
    b. Name the file so that it corresponds with the disk you are examining.
    c. Use the `cat` command to verify that valid information exists in the file that you create.

```
# fdisk -W /fdisk/c2d0p0 /dev/rdsk/c2d0p0
# cat /fdisk/c2d0p0

* /dev/rdsk/c2d0p0 default fdisk table
* Dimensions:
*    512 bytes/sector
*     63 sectors/track
*    255 tracks/cylinder
*   10011 cylinders
*
(ooutput omitted)
```

26. Use the following `dd` command to destroy the x86/x64 VTOC and complete the following steps:
    a. Be certain to specify the correct disk device name for the `of=` argument.
    b. Enter all other arguments exactly as listed.

```
# dd if=/dev/zero of=/dev/rdsk/c2d0s2 bs=512 count=2
2+0 records in
2+0 records out
#
```

27. Attempt to read the label from the same disk by using the `prtvtoc` command. What happens?

```
# prtvtoc /dev/rdsk/c2d0s2
* /dev/rdsk/c2d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*   10010 cylinders
*   10008 accessible cylinders
*
* Flags:
*   1: unmountable
```

```
*  10: read-only
*
* Unallocated space:
*       First     Sector    Last
*       Sector     Count    Sector
*       48195 160730325 160778519
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       2      5    01          0 160778520 160778519
       8      1    01          0     16065     16064
       9      9    00      16065     32130     48194
#
```

Typically on x86/x64 systems, a default partition table displays. In this example from an Ultra 20, only slices 2, 8 and 9 remain defined. All other partitions have been destroyed.

28. Run the `format` utility, and perform the following steps:

    a. Select your spare disk.

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


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 39887 alt 2 hd 64 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
       1. c2d0 <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
Specify disk (enter its number): 1
selecting c2d0
Controller working list found
[disk formatted, defect list found]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        fdisk      - run the fdisk program
        repair     - repair a defective sector
        show       - translate a disk address
        label      - write label to the disk
        analyze    - surface analysis
```

```
        defect      - defect list management
        backup      - search for backup labels
        verify      - read and display labels
        save        - save new disk/partition definitions
        volname     - set 8-character volume name
        !<cmd>      - execute <cmd>, then return
        quit
format>
```

b. Display the fdisk menu.

```
format> fdisk
             Total disk size is 10011 cylinders
             Cylinder size is 16065 (512 byte) blocks

                                               Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ============  =====   ===   ======   ===
          1                 Solaris2          1  10010   10010    100




SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection:
```

c. Delete the defined Solaris fdisk partition

```
Enter Selection: 3

Specify the partition number to delete (or enter 0 to exit): 1

Are you sure you want to delete partition 1? This will make all files and
programs in this partition inaccessible (type "y" or "n").

y
```

d. Save the fdisk partition table, exit the fdisk menu, and quit the format utility.

```
          Total disk size is 10011 cylinders
          Cylinder size is 16065 (512 byte) blocks

                                              Cylinders
     Partition   Status    Type          Start   End   Length    %
     =========   ======    ============  =====   ===   ======   ===



WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
   1. Create a partition
   2. Specify the active partition
   3. Delete a partition
   4. Change between Solaris and Solaris2 Partition IDs
   5. Exit (update disk configuration and exit)
   6. Cancel (exit without updating disk configuration)
Enter Selection: 5


Solaris fdisk partition not found
No fdisk solaris partition found
format> q
#
```

29. Attempt to use the fmthard command to write to the disk the x86/x64 VTOC information you saved earlier. What happens?

```
# fmthard -s /vtoc/c2d0 /dev/rdsk/c2d0s2
/dev/rdsk/c2d0s2: Cannot get disk geometry
#
```

The attempt fails because the fmthard command can not find a slice 2 defined on the disk.

30. Use the fdisk command to replace the fdisk partition table with the fdisk information you saved earlier.

```
# fdisk -F /fdisk/c2d0p0 /dev/rdsk/c2d0p0
#
```

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

```
# fmthard -s /vtoc/c2d0 /dev/rdsk/c2d0s2
fmthard:  New volume table of contents now in place.
#
```

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

```
# prtvtoc /dev/rdsk/c2d0s2
* /dev/rdsk/c2d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*     255 tracks/cylinder
*   16065 sectors/cylinder
*   10010 cylinders
*   10008 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0     11    00       48195  39857265  39905459
       1      3    01   39905460  39857265  79762724
       2      5    01          0 160778520 160778519
       3      0    01   79762725  39857265 119619989
       4      0    01  119619990  41158530 160778519
       8      1    01          0     16065     16064
       9      9    00      16065     32130     48194
#
```

Was this successful?

This command should successfully read the disk label.

Notes:

Module 3

Interface Configurations

Objectives

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

- Control and monitor network interfaces
- Configure internet protocol version 4 (IPv4) interfaces at boot time

Controlling and Monitoring Network Interfaces

Network commands, such as `ifconfig`, `ping`, and `snoop`, control and monitor the functionality of network interfaces.

Displaying the MAC Address

The media access control (MAC) address is your computer's unique hardware address on a local area network (LAN). The MAC address is also the Ethernet address on an Ethernet LAN. When you are connected to a LAN, an address resolution table maps your computer's physical MAC address to an Internet Protocol (IP) address on the LAN. Two ways to display the MAC address or the Ethernet address are:

- Use the `ifconfig -a` command
- Use the boot programmable read-only memory (PROM) `banner` command

---

Note – The MAC address is displayed only if run as the `root` user.

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232
index 1 inet 127.0.0.1 netmask ff000000
nge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.30.41 netmask ffffff00 broadcast 192.168.30.255
        ether 8:0:20:93:c9:af
```

The MAC address is listed as `8:0:20:93:c9:af` in this example.

You can also retrieve the MAC address from a SPARC-based system that has not yet been booted by running the `banner` command at the `ok` prompt.

```
ok banner
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 300MHz), Keyboard Present
OpenBoot 3.31 256 MB (60ns) memory installed, Serial #9685423.
Ethernet address 8:0:20:93:c9:af, Host ID: 8093c9af.
```

Displaying the IP Address

The `ifconfig -a` command displays the current configuration for the network interfaces.

With the `-a` option, the `ifconfig` command displays the current configuration for all network interfaces in the system.

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232
index 1 inet 127.0.0.1 netmask ff000000
nge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.30.41 netmask ffffff00 broadcast 192.168.30.255
        ether 8:0:20:93:c9:af
```

The previous example shows that the loopback interface (`lo0`) is up, running, and configured with an IP address of `127.0.0.1`. The `nge0` interface is up, running, and configured with an IP address of `192.168.30.41`.

Marking an Ethernet Interface as Down

When an Ethernet interface is marked as down, it means that it cannot communicate. You can use the `ifconfig` command to mark an Ethernet interface as up or down. For example, to mark the `nge0` interface as down, perform the commands:

```
# ifconfig nge0 down
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232
index 1 inet 127.0.0.1 netmask ff000000
nge0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.30.41 netmask ffffff00 broadcast 192.168.30.255
        ether 8:0:20:93:c9:af
```

Note – The `UP` flag for `nge0` is no longer present. When an interface is flagged as `UP`, it is ready to communicate.

The following example shows that when you mark an interface as up, the `UP` status appears in the flags field of the `ifconfig` command output:

```
# ifconfig nge0 up
# ifconfig -a
```

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232
index 1 inet 127.0.0.1 netmask ff000000
nge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.30.41 netmask ffffff00 broadcast 192.168.30.255
        ether 8:0:20:93:c9:af
```

Sending ICMP `ECHO_REQUEST` Packets

To determine if you can contact another system over the network, enter the `ping` command:

```
# ping sys41
sys41 is alive
```

The previous response indicates the host name `sys41` is alive. A response of `no answer from sys41` indicates that you cannot contact host `sys41`. This implies a problem with host `sys41`, or a problem with the network.

For the `ping` command to succeed, the following conditions must be satisfied on both systems:

- The interface must be plumbed.
- The interface must be configured.
- The interface must be `up`.
- The interface must be physically connected.
- The interface must have valid routes configured.

For more information on `ifconfig` and plumbed, see the `ifconfig` man page.

Note – You can track the route a packet follows using `traceroute`. This utility traces the route that an IP packet follows to another internet host.

Capturing and Inspecting Network Packets

Use the `snoop` utility to capture and inspect network packets to determine what kind of data is transferred between systems. Use the `snoop` utility to see what happens when one system uses the `ping` command to communicate with another system. To view network traffic between two specific systems, perform the command:

```
# snoop sys41 sys42
sys41 -> sys42 ICMP Echo request (ID: 615 Sequence number: 0)
sys42 -> sys41 ICMP Echo reply (ID: 615 Sequence number: 0)
```

Use the `-a` option to enable audible clicks, which notify you of any network traffic. Although noisy, the clicks are useful when troubleshooting.

The following example shows how to turn on audible clicks for all network traffic related to a Dynamic Host Configuration Protocol (DHCP) boot:

```
# snoop -a dhcp
```

Additional `snoop` options include:

`snoop`	Summary output
`snoop -o` *`filename`*	Saves captured packets in filename as they are captured
`snoop -i` *`filename`*	Displays packets previously captured in *`filename`*
`snoop -i` *`filename`* `-v`	Filters output to verbose mode
`snoop -i` *`filename`* `-V`	Filters output to summary verbose mode
`snoop -i` *`filename`* `-x` *`offset`*	Displays packet data in hexadecimal and ASCII format.
`snoop -d` *device*	Receives packets from a network interface specified by *`device`*

Note – Press Control-C to stop the `snoop` utility.

Configuring IPv4 Interfaces at Boot Time

This section describes the files and scripts involved with configuring IPv4 network interfaces.

Introducing IPv4 Interface Files

You can get a basic understanding of network interfaces within the Solaris OS by learning the function of a few files and services. The services and files are the following:

- The `svc:/network/physical:default` service
- The `/etc/hostname.`*`xxn`* file
- The `/etc/inet/hosts` file
- The `/etc/inet/ipnodes` file

The `svc:/network/physical:default` Service

The `svc:/network/physical:default` service calls the `/lib/svc/method/net-physical` method script. It is one of the startup scripts that runs each time you boot the system. This script uses the `ifconfig` utility to configure each interface with an IP address and other required network information. The script searches for files called `hostname.`*`xxn`* in the `/etc` directory, where *`xx`* is an interface type and *`n`* is the instance of the interface. For every file named `/etc/hostname.`*`xxn`*, the script uses the `ifconfig` command with the `plumb` option to make the `kernel` ready to talk to this type of interface. The script then configures the named interface using other options to the `ifconfig` command. The `/etc/hostname.nge0` file is an example of an interface configuration file.

Note – In Solaris 8 and 9 OS, the `etc/rcS.d/S30network.sh` file is used to perform the same function. Before Solaris 8 OS, the `/etc/rcS.d/S30rootusr.sh` file was used.

The etc/hostname.*xxn* File

The /etc/hostname.*xxn* file contains an entry that configures a corresponding interface. The variable component of the file name is replaced by an interface type and a number that differentiates between multiple interfaces of the same type configured in the system. Table 3-1 shows examples of file entries for Ethernet interfaces commonly found in Sun systems.

Table 3-1 The /etc/hostname.*xxn* File Entries and Corresponding Interfaces

Entry	Interface
/etc/hostname.e1000g0	First e1000g (Intel PRO/1000 Gigabit family device driver) Ethernet interface in the system
/etc/hostname.e1000g1	Second e1000g Ethernet interface in the system
/etc/hostname.bge0	First bge (Broadcom Gigabit Ethernet device driver) Ethernet interface in the system
/etc/hostname.bge1	Second bge Ethernet interface in the system
/etc/hostname.ce0	First ce (Cassini Gigabit-Ethernet device driver) Ethernet interface in the system
/etc/hostname.qfe0	First qfe (Quad Fast-Ethernet device driver) Ethernet interface in the system
/etc/hostname.hme0	First hme (Fast-Ethernet device driver) Ethernet interface in the system
/etc/hostname.eri0	First eri (eri Fast-Ethernet device driver) Ethernet interface in the system
/etc/hostname.nge0	First nge (Nvidia Gigabit Ethernet driver) Ethernet interface in the system

The codes for the interface types are product codes. These codes originate from varying sources. For example, the e1000g code is for the Intel PRO/1000 Gigabit and the bge code is an abbreviation for Broadcom Gigabit Ethernet.

The /etc/hostname.nge0 file contains either the host name or the IP address of the system that contains the nge0 interface. The host name contained in the file must exist in the /etc/inet/hosts file so that it can be resolved to an IP address at system boot time. You can edit the /etc/hostname.nge0 file to contain either the host name or the IP address from the /etc/inet/hosts file.

```
# cat /etc/hostname.nge0
sys41
or
# cat /etc/hostname.nge0
192.168.30.41
```

The /etc/inet/hosts File

The /etc/inet/hosts file is a local database that associates the IP addresses of hosts with their names. You can use the /etc/inet/hosts file with, or instead of, other hosts databases, including the Domain Name System (DNS), the Network Information Service (NIS) hosts map, and the Network Information Service Plus (NIS+) hosts table. Programs use library interfaces to access information in the /etc/inet/hosts file.

The /etc/inet/hosts file contains at least the loopback and host information. The file has one entry for each IP address of each host. If a host has more than one IP address, this file will have one entry for each address, on separate lines. The format of each line is:

```
IP-address official-host-name [aliases] . . .
```

Items are separated by any number of spaces or tab characters. The first item on a line is the host's IP address. The second entry is the host's official name. Subsequent entries on the same line are alternative names for the same machine, or nicknames. Nicknames are optional.

```
# cat /etc/inet/hosts
.
<output truncated>
.
127.0.0.1 localhost
.
<output truncated>
.
192.168.30.41  sys41         loghost      #connection to nge0 interface
.
<output truncated>
```

Note – The /etc/inet/hosts file is the official (system V release 4) SVr4 name of the hosts file. The symbolic link /etc/inet/hosts exists for Berkeley Software Distribution (BSD) compatibility.

The /etc/ipnodes File

The ipnodes file is a symbolic link to the /etc/inet/hosts file. It associates the names of nodes with their internet protocol (IP) addresses. The ipnodes file can be used in conjunction with, or instead of, other ipnodes databases, including the domain name system (DNS), the NIS ipnodes map, and LDAP.

The ipnodes file has one entry for each IP address of each node, and can contain either IPv4 or IPv6 addresses.

If a node has more than one IP address, it will have one entry for each, on consecutive lines. The format of each line is:

```
IP-address official-node-name nicknames...
```

Items are separated spaces or tab characters. The first item on a line is the host IP address. The second entry is the host official name. Subsequent entries on the same line are alternative names for the same machine, or nicknames. Nicknames are optional.

```
# cat /etc/inet/ipnodes
#
# Internet host table
#
::1     localhost
127.0.0.1       localhost
192.168.30.41sys41       loghost
```

IP addresses can be defined in the ipnodes file or in the hosts file. The ipnodes file will be searched first, then the hosts file.

Changing the System Host Name

The host name of a system is contained in four files on the system. You must modify all of these files, and perform a reboot, to successfully change a system's host name. The files that contain the host name of a system are:

- The `/etc/nodename` file
- The `/etc/hostname.`*xxn* file
- The `/etc/inet/hosts` file

Note – `/etc/inet/nodes` was used for IPv6 addresses but now it is a link to `/etc/inet/hosts`.

Note – If crash dump is enabled on a system, the system name must be changed under `/var/crash`. Older versions of Solaris also had the `hostname` in files located under `/etc/net/tic*/*`.

Use the `uname -s` command is to change the system hostname on a temporarily.

Editing the `/etc/nodename` File

Each Solaris OS has a canonical name, which is the official name used when referring to a system. By convention, the system name is the same as the host name associated with the IP address of the primary network interface; for example, `hostname.nge0`.

The following example shows a system's `/etc/nodename` file:

```
# cat /etc/nodename
sys41
```

Change the canonical name by editing the `/etc/nodename` file, and rebooting the system.

If the machine network configuration is managed remotely and delivered by the DHCP or remote procedure call (RPC) bootparams protocols, the `/etc/nodename` file is not used. The file is not used because the remote service delivers the canonical name.

The `/etc/hostname.xxn` File

You do not have to edit the `/etc/inet/ipnodes` file. It *is* the `/etc/hosts` file.

Editing the `/etc/inet/hosts` File

Network addresses are written in the conventional decimal-dot notation.

Host names are text strings up to 24 characters. Alphabetic characters, numbers, the (-) sign, and a (.) are allowed in the host name. Periods are only allowed when they serve to delimit components of domain style names. Spaces are not allowed in the host name. The first character must be an alphabetic character. The last character must not be a (–) or a (.).

No distinction is made between uppercase and lowercase characters, unless the NIS naming service is used. Uppercase characters in names have been known to cause problems with NIS.

A (#) indicates the beginning of a comment. After a comment character, all characters, up to the end of the line, are not interpreted.

The `sys-unconfig` Command

Use the `/usr/sbin/sys-unconfig` command to undo a system configuration. Use the `/usr/sbin/sys-unconfig` command to restore a system configuration to an unconfigured state. It will be ready to be reconfigured.

The `sys-unconfig` command does the following:

- Saves the current `/etc/inet/hosts` file information in the `/etc/inet/hosts.saved` file.
- If the current `/etc/vfstab` file contains Network File System (NFS) mount entries, it saves the `/etc/vfstab` file to the `/etc/vfstab.orig` file.
- Restores the default `/etc/inet/hosts` file.
- Removes the default host name in the `/etc/hostname.xxn` files for all configured interfaces.
- Removes the default domain name in the `/etc/defaultdomain` file.
- Restores the time zone to `PST8PDT` in the `/etc/TIMEZONE` file.
- Resets naming services to local files.

- Removes the /etc/inet/netmasks file.
- Removes the /etc/defaultrouter file.
- Removes the password set for the root user in the /etc/shadow file.
- Removes the /etc/.rootkey file for NIS+.
- Executes all system configuration applications. These applications are defined by prior executions of a sysidconfig -a command.
- Removes the /etc/resolv.conf file for DNS clients.
- Disables Lightweight Directory Access Protocol (LDAP) by removing:
  - The /var/ldap/ldap_client_cache file
  - The /var/ldap/ldap_client_file file
  - The /var/ldap/ldap_client_cred file
  - The /var/ldap/cachemgr.log file
- Regenerates keys for the Secure Shell Daemon (sshd).

When the sys-unconfig command is finished, it performs a system shutdown. The sys-unconfig command is a potentially dangerous utility and can only be run by the root user.

When you restart the system, a configuration script prompts you to configure the system information. The sys-unconfig command is not available on diskless clients.

Exercise: The Solaris OS Network Commands

In this exercise, you use basic network-related commands.

Preparation

To prepare for this exercise, perform the following tasks:

- Check that you have two systems listed in each `/etc/inet/hosts` file on each system.
- Work with a partner for this exercise, and perform all steps on both systems, unless noted otherwise.

Task

Complete the following steps using the `ifconfig` utility, the `ping` command, and the `snoop` utility.

Note – Be sure to work closely with your partner during the lab to ensure you are both working on the same steps.

---

1. On both systems, log in as the `root` user, and open a terminal window. Using the `ifconfig -a` command, display basic configuration information about your network interfaces.

   For your primary interface (for example `nge0`), what does the `ifconfig` command report for the following attributes? Enter your values into Table 3-2.

Table 3-2 Primary Interface Values

Attribute	Value
IP address	
Ethernet address	
Interface up/down	

2. On both systems, open a new terminal window. In the new window, enter the `snoop` command to display the network traffic between your two systems only.

3. Use the `ping` command to verify that your system can contact the network interface on your partner's system.
4. Observe the output from the `snoop` command. Which protocol does the `ping` command use?

   ___

   Does the `snoop` output contain requests and replies (yes or no)?

   Requests: Replies:
5. On one system, use the `ifconfig` command to mark its primary interface as down and then again to display its configuration information. Warn your lab partner that the system's interface will go down.

   Has anything changed in the information that the `ifconfig` command reports?

   ___
6. On the system whose interface remains up, attempt to use the `ping` command to contact the system whose interface is down.

   What does the `ping` command report?

   ___
7. Observe the output from the `snoop` utility on both systems. How does the `snoop` output differ from the `ping` command output before and after you marked the interface as down?

   ___

   How many requests does the `ping` command send by default?

   ___

   Does the target system see the `ping` command requests? If so, how are these requests handled?

   ___
8. On the system whose interface is down, use the `ifconfig` command to mark its primary interface as up. Check that the change took place.
9. On the system whose interface remained up, use the `ping` command to contact the other system.

   What does the `ping` command report?

   ___

   Does the `snoop` utility report a reply from the target host?

   ___

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: The Solaris OS Network Commands

In this exercise, you use basic network-related commands.

Preparation

To prepare for this exercise, perform the following tasks:

- Check that you have two systems listed in each `/etc/inet/hosts` file on each system.
- Work with a partner for this exercise, and perform all steps on both systems, unless noted otherwise.

Task and Solutions

This section describes the tasks for you to perform, and lists the solutions. Complete the following steps using the ifconfig utility, the ping command, and the snoop utility.

Note – Be sure to work closely with your partner during the lab to ensure you are both working on the same steps.

1. On both systems, log in as the root user, and open a terminal window. Using the ifconfig -a command, display basic configuration information about your network interfaces.

   For your primary interface (for example nge0), what does the ifconfig command report for the following attributes? Enter your values into Table 3-3.

Table 3-3 Primary Interface Values

Attribute	Value
IP address	*It varies according to the system in use.*
Ethernet address	*It varies according to the system in use.*
Interface up/down	*The interface should be* UP.

2. On both systems, open a new terminal window. In the new window, enter the snoop command to display the network traffic between your two systems only.

```
# snoop host1 host2
```

3. Use the ping command to verify that your system can contact the network interface on your partner's system.

```
# ping host
```

4. Observe the output from the snoop command. Which protocol does the ping command use?

   ICMP

   Does the snoop output contain requests and replies (yes or no)?

   Requests: *Yes* Replies: *Yes*

5. On one system, use the `ifconfig` command to mark its primary interface as down and then again to display its configuration information. Warn your lab partner that the system's interface will go down.

```
# ifconfig nge0 down
# ifconfig nge0
```

Has anything changed in the information that the `ifconfig` command reports?

The `ifconfig` *command no longer lists the interface as* `UP`*.*

6. On the system whose interface remains up, attempt to use the `ping` command to contact the system whose interface is down.

   What does the `ping` command report?

   After a time-out period, the `ping` *command reports* `no answer from host`.

7. Observe the output from the `snoop` utility on both systems. How does the `snoop` output differ from the `ping` command output before and after you marked the interface as down?

   The `snoop` *utility only shows the* `ping` *command requests—no replies.*

   How many requests does the `ping` command send by default?

   Twenty

   Does the target system see the `ping` command requests? If so, how are these requests handled?

   Yes it does, but it does not send a reply.

8. On the system whose interface is down, use the `ifconfig` command to mark its primary interface as up. Check that the change took place.

```
# ifconfig nge0 up
# ifconfig nge0
```

9. On the system whose interface remained up, use the `ping` command to contact the other system.

   What does the `ping` command report?

   The host is alive.

   Does the `snoop` utility report a reply from the target host?

   Yes.

Module 4

Managing the Solaris OS UFS

Objectives

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

- Identify disk-based, distributed, and pseudo file systems in the Solaris OS
- Describe the Solaris OS Unix File System (UFS)
- Create a new UFS file system
- Check a file system using the `fsck` command
- Resolve file system inconsistencies
- Monitor file system use

Introducing Solaris OS File Systems

A file system is a collection of files and directories that make up a structured set of information. The Solaris OS supports three different types of file systems:

- Disk-based file systems
- Distributed file systems
- Pseudo file systems

Disk-Based File Systems

Disk-based file systems are found on hard disks, CD-ROMs, diskettes, and DVDs. The following are examples of disk-based file systems:

- ufs – The UNIX file system in the Solaris OS is based on the Berkeley fast file system. Enhancements in the Solaris 10 OS allow the `ufs` to grow to multiple terabytes in size.
- hsfs – The High Sierra file system is a special-purpose file system developed for use on CD-ROM media.
- pcfs – The PC file system is a UNIX implementation of the disk operating system (DOS) file allocation table (FAT32) file system. The pcfs file system allows the Solaris OS to access PC-DOS formatted file systems. Users can use UNIX commands for direct read and write access to PC-DOS files.
- udfs – The Universal Disk Format file system is used for optical storage targeted at DVD and CD-ROM media. The UDF file system allows universal data exchange and supports read and write operations.
- ZFS - A file system created by Sun that features simplified administration, pooled storage, checksums and self-healing data, snapshots, clones, and unparalleled scalability.

Distributed File Systems

Distributed file systems provide network access to file system resources.

- NFS – The network file system allows users to share files among many types of systems on the network. The NFS file system makes part of a file system on one system appear as though it were part of the local directory tree.

Pseudo File Systems

Pseudo file systems are memory based. These file systems provide for better system performance, in addition to providing access to `kernel` information and facilities. Pseudo file systems include:

- `tmpfs` – The temporary file system stores files in memory, which avoids the overhead of writing to a disk-based file system. The tmpfs file system is created and destroyed every time the system is rebooted.
- `swapfs` – The swap file system is used by the `kernel` to manage swap space on disks.
- `fdfs` – The file descriptor file system provides explicit names for opening files by using file descriptors (for example, `/dev/fd/0`, `/dev/fd/1`, `/dev/fd/2`) in the `/dev/fd` directory.
- `procfs` – The process file system contains a list of active processes in the `/proc` directory. The processes are listed by process number. Information in this directory is used by commands, such as the `ps` command.
- `lofs` - The loopback file system provides access to existing files using alternate pathnames.
- `mntfs` – The mount file system provides read-only information from the `kernel` about locally mounted file systems.
- `objfs` – The kernel object file system. This file system is used by the kernel to store details relating to the modules currently loaded by the kernel. The object file system is used for the `/system/object` directory.

- `devfs` – The device file system is used to manage the namespace of all devices on the system. This file system is used for the `/devices` directory.
- `ctfs` – The contract file system is associated with the `/system/contract` directory. This is used by the Service Management Facility to track the processes which compose a service, so that a failure in a part of a multi-process service can be identified as a failure of that service.

Note – The `mount` command with the `-p` or `-v` option displays the file system types currently in use.

Creating a New `ufs` File System

This section describes the `ufs` file system in the Solaris OS.

Viewing the Solaris OS `ufs` File System

The user views the `ufs` file system differently than the operating system does in the Solaris OS. To a user, a file system appears as a collection of files and directories used to store and organize data for access by the system and its users. To the operating system, a file system is a collection of control structures and data blocks that occupy the space defined by a partition, which allow for data storage and management.

The Solaris OS stores data in a logical file hierarchy often consisting of several file systems. This file hierarchy is referred to as the Solaris directory hierarchy.

Figure 4-1 shows the Solaris OS directory hierarchy beginning with the / (root) directory.

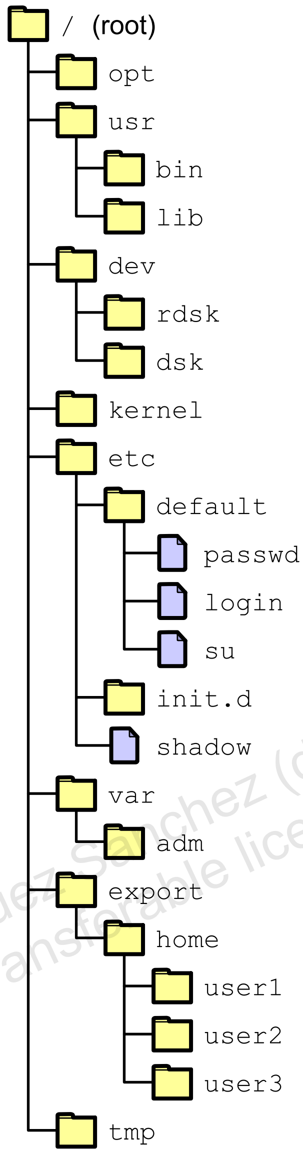


Figure 4-1 Solaris OS Directory Hierarchy

Note – Figure 4-1 is not a complete representation of a Solaris OS directory hierarchy.

A `ufs` file system must be created on disk before it can be used in the Solaris OS.

You can create ufs file systems on different types of storage resources, including disk slices found on a single disk, and slices on virtual storage objects, such as volumes created by software volume managers.

This description focuses on using ufs file systems on disk slices. Solaris systems typically use a number of ufs file systems to store the directories and files that make up the Solaris OS. The Solaris directory hierarchy may consist of data found on a number of separate ufs file systems.

Figure 4-2 shows ufs file systems that are located on different disk slices.

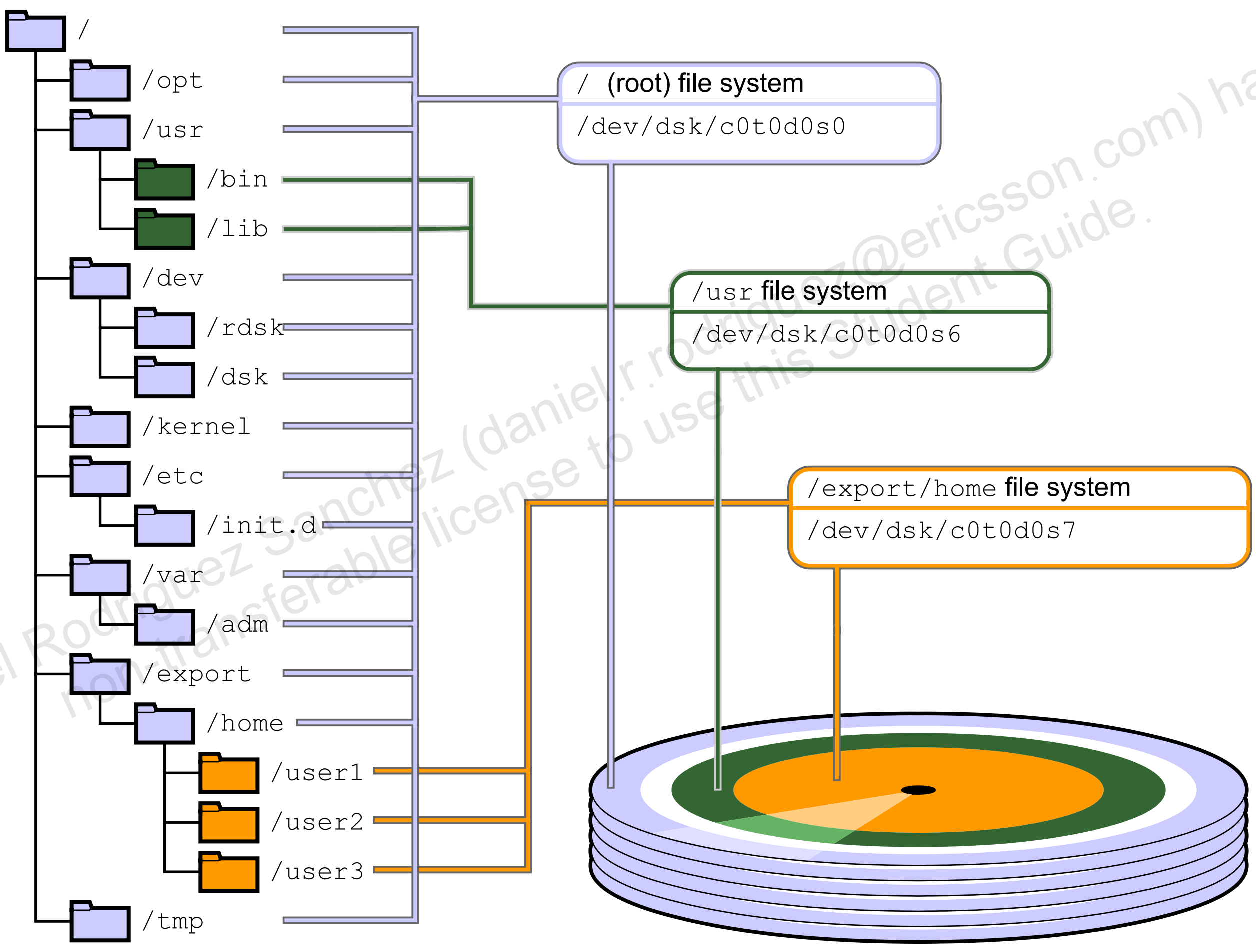


Figure 4-2 Solaris ufs File Systems Residing on Disk Slices

Partition Tables and Reserved Space

In order to create a ufs file system, slices on the disk must be defined in a partition table within a disk label. The space used to hold a `ufs` file system is provided by a disk slice.

- Disk labels record the locations and boundaries of disk slices.
- The label that contains these slice definitions is also known as the volume table of contents, or VTOC.
- On SPARC systems, the VTOC is located in sector 0 of the disk.
- On x86/x64 systems, the VTOC is located in the second sector of the Solaris `fdisk` partition. On x86/x64 systems, slice 8 occupies the first cylinder of the Solaris `fdisk` partition, and protects the x86/x64 VTOC.

The VTOC itself is not a file system structure, but the slices it defines are necessary for a `ufs` file system to be created on a disk.

Solaris OS `ufs` file systems do not use the following sectors of the slices in which they reside:

- Sector 0 - This sector is reserved for the VTOC on SPARC systems. If a `ufs` file system on a SPARC system occupies a slice that starts on the first disk cylinder, this sector would contain the SPARC VTOC. Every `ufs` file system on SPARC and x86/x64 systems skips sector 0.
- Sectors 1 through 15 - These sectors are reserved for the bootstrap program (`bootblk`) used by SPARC systems. Only a slice that contains a file system from which a SPARC system boots would contain a bootstrap program in these sectors. Every `ufs` file system on SPARC and x86/x64 systems skips sectors 1 through 15.

Solaris OS `ufs` file system structures begin at sector 16 of the slice that holds the file system. This is true of `ufs` file systems on SPARC and x86/x64 systems.

ufs File System Structures

Solaris ufs file systems use the structures described in this section.

Primary Superblock

The primary superblock resides in the 16 disk sectors (Sectors 16–31 relative to the first sector in the slice) that follow the space reserved for a SPARC boot block. The superblock is a table of information that describes the file system, including:

- The number of data blocks
- The number of cylinder groups
- The size of a data block and fragment
- A description of the hardware, derived from the label
- The name of the mount point
- File system state flag: clean, stable, active, logging, or unknown

Backup Superblocks

When the file system is created, a backup copy of the superblock is created beginning at sector 32, and an additional backup is created at the beginning of each cylinder group. This replication protects the critical data in the superblock against catastrophic loss.

Cylinder Groups

Each file system is divided into cylinder groups with a minimum default size of 16 cylinders per group. Cylinder groups improve disk access.

The file system constantly optimizes disk performance by attempting to place a file's data into a single cylinder group, which reduces the distance a head has to travel to access the file's data. The file system stores large files across several cylinder groups, if needed.

Cylinder Group Blocks

The cylinder group block is a table in each cylinder group that describes the cylinder group, including:

- The number of inodes
- The number of data blocks in the cylinder group
- The number of directories
- Free blocks, free inodes, and free fragments in the cylinder group
- The free block map
- The used inode map

Figure 4-3 shows a series of cylinder groups in a ufs file system. The sectors reserved for the SPARC disk label (VTOC) and SPARC bootblk program are skipped when you create a ufs file system on both SPARC and x86/64 systems.

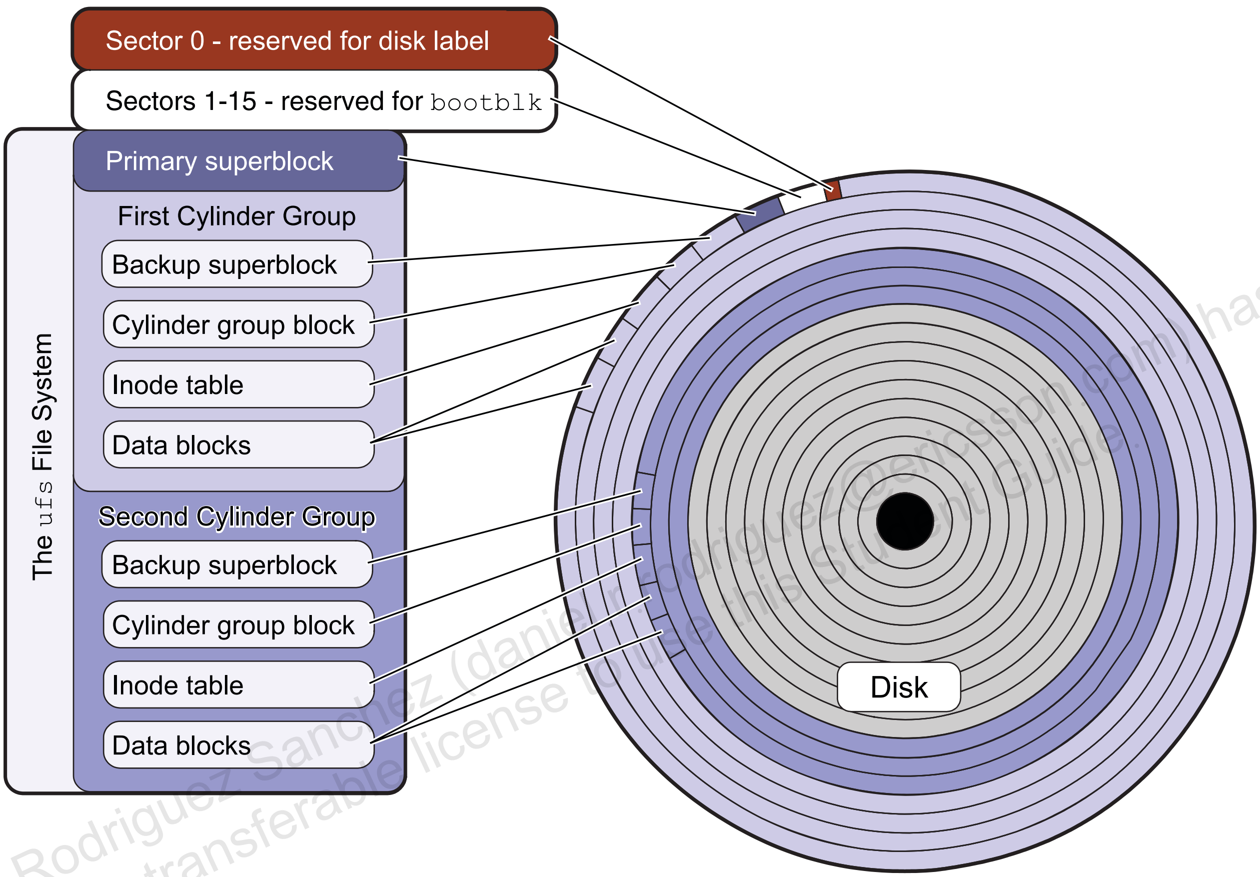


Figure 4-3 Solaris OS ufs File System Structure

The ufs Inode

An inode contains the following information about a file:

- The type of file and the access modes
- The user identification (UID) and group identification (GID) numbers of the file's owner and group
- The size of the file
- The link count
- The time the file was last accessed and modified and the inode changed
- The total number of data blocks used by or allocated to the file
- Two types of pointers: direct pointers and indirect pointers

Figure 4-4 shows some of the information contained in an inode.

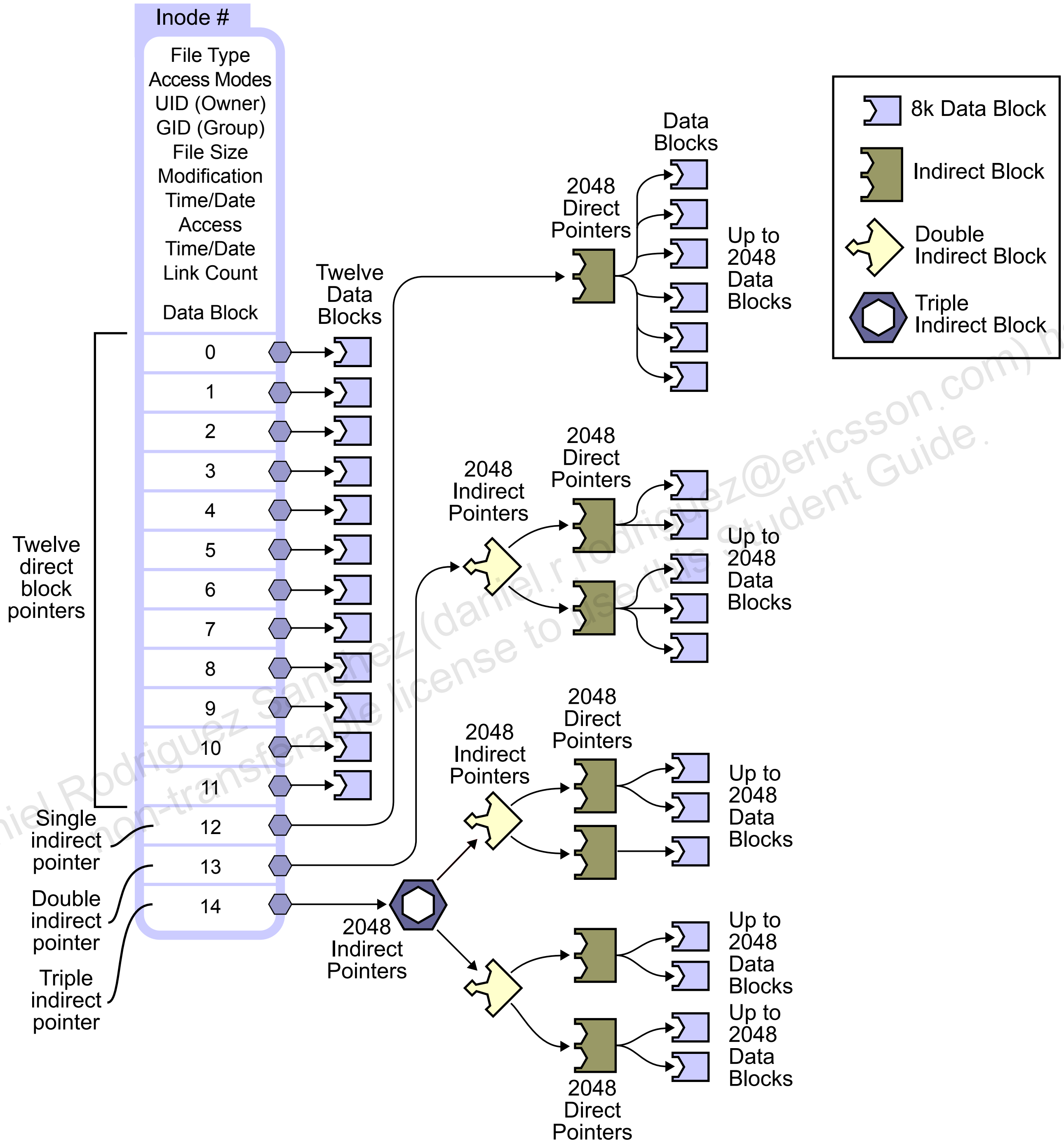


Figure 4-4 Structure of a `ufs` Inode

Note – To view some of the information contained in a file or directory inode, use the `ls -l` command. To view the inode number assigned to the file or directory, use the `ls -i` command.

Direct Pointers

Inside the inode there are 12 direct pointers, which contain addresses for the file's first 12 data blocks. The 12 direct pointers can each reference 8-Kbyte data blocks for a file that is up to 96 Kbytes.

Indirect Pointers

The three types of indirect pointers within an inode are:

- Single indirect pointer – Refers to a file system block that contains pointers to data blocks. This file system block contains 2048 additional addresses of 8-Kbyte data blocks, which can point to an additional 16 Mbytes of data.
- Double indirect pointer – Refers to a file system block that contains single indirect pointers. Each indirect pointer refers to a file system block that contains the data block pointers. Double indirect pointers point to an additional 32 Gbytes of data.
- Triple indirect pointer – Can reference up to an additional 64 Tbytes of data.

Data Blocks

The remaining space allocated to the `ufs` file system holds data blocks. Data blocks are allocated, by default, in 8-Kbyte logical block sizes. The blocks are further divided into 1-Kbyte fragments. For a regular file, the data blocks contain the contents of the file. For a directory, the data blocks contain entries that associate the inode numbers and the file names of the files and directories contained in that directory.

Within a file system, those blocks that are currently not being used as files, directories, indirect address blocks, or storage blocks are marked as free in the cylinder group map. This map also keeps track of fragments to prevent disk performance from degrading.

Fragmentation

Fragmentation is the method used by the ufs file system to allocate disk space efficiently. Files less than 96 Kbytes in size are stored using fragmentation.

By default, data blocks can be divided into eight fragments of 1024 bytes each. Fragments store files and pieces of files smaller than 8192 bytes. For files larger than 96 Kbytes, fragments are never allocated and full blocks are exclusively used.

Figure 4-5 shows a fragment in a data block.

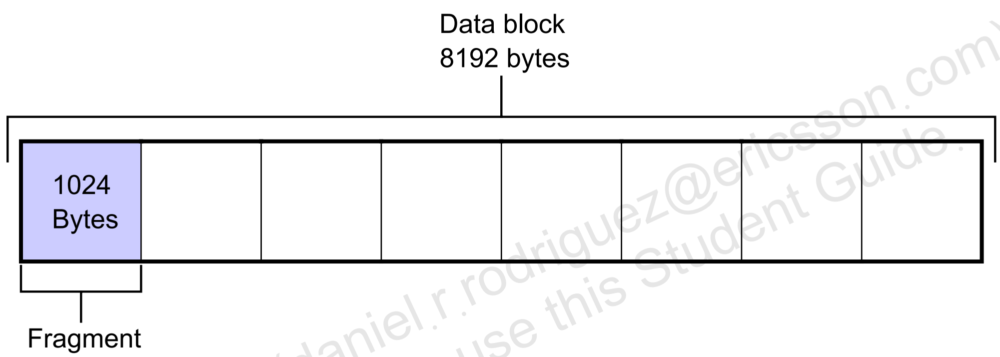


Figure 4-5 Divided Data Block

If a file contained in a fragment grows and requires more space, it is allocated one or more additional fragments in the same data block.

Figure 4-6 shows the contents of two different files stored in fragments in the same data block.

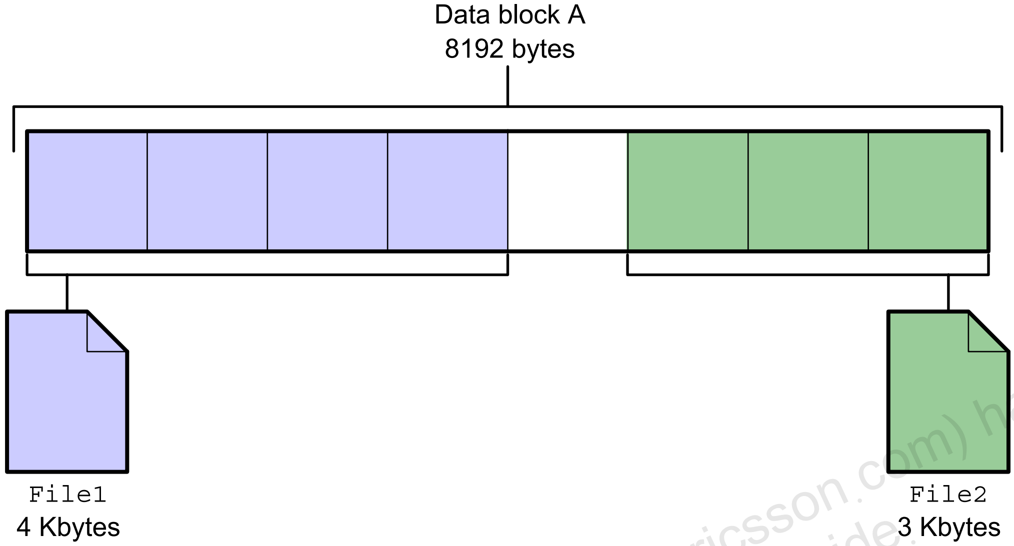


Figure 4-6 Two Files Stored in One Data Block

For example, if File1 requires more space than is currently available in the shared data block, the entire contents of that expanding file are moved by the ufs file system into a free data block. This requirement by the ufs file system assures that all of a file's fragments are contained in a whole data block. The ufs file system does not allow fragments of the same file to be stored in two different data blocks.

Using the newfs Command

To use the disk to store directories or files, a file system must be created on every disk partition. As the root user, you can construct a ufs file system on a disk slice by using the newfs command.

The newfs command is an easy-to-use front-end to the mkfs command, which you use to create file systems. The newfs command is located in the /usr/sbin directory.

Caution – Creating a new file system is destructive. The mkfs and newfs commands re-create the file system structure (new superblock, cylinder group blocks, and inode tables). Access to all existing data on the filesystem will be lost.

To create a ufs file system, by using the newfs command, perform the following steps:

1. As the root user, create a file system on a slice of a newly partitioned disk by entering the command:

   ```
   # newfs /dev/rdsk/c2d0s0
   ```

2. The newfs command asks for confirmation before continuing. Verify that the correct disk slice on the correct disk is selected. To proceed, type y, to terminate the process, type n.

```
newfs: construct a new file system /dev/rdsk/c2d0s0: (y/n)? y
```

The newfs command displays information about the new file system being created.

```
Warning: 5008 sector(s) in last cylinder unallocated
/dev/rdsk/c2d0s0:       39857264 sectors in 6488 cylinders of 48 tracks,
128 sectors
        19461.6MB in 406 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488,
885920,
Initializing cylinder groups:
.......
super-block backups for last 10 cylinder groups at:
 38929952, 39028384, 39126816, 39225248, 39323680, 39422112, 39520544,
 39618976, 39717408, 39815840
#
```

Output printed by the newfs command describes the basic disk geometry and the ufs file system created in this slice. The beginning sector locations of backup superblocks also displays.

The newfs -i command is used to specify the density of the number of bytes per inode in the file system. To create more inodes, a specify a smaller number of bytes per inode.

Note – This process also creates a lost+found directory for the ufs file system, which is a directory that is used by the file system check and repair (fsck) utility.

---

3. Repeat Steps 1 and 2 for every slice on any newly partitioned disk that needs to contain a file system.

The newfs command reserves between 1 and 10 percent of the file system space, depending on the size of the file system, for maintenance. This free space, referred to as minfree, specifies the amount of space on the slice that is reserved or held back from regular users. You can use the newfs -m *%free* command to preset the percentage of free space when you create a new file system.

To show the value of minfree for a file system, use the fstyp command with the -v option.

```
# fstyp -v /dev/dsk/c2d0s0 | grep minfree
minfree 1%      maxbpg  2048    optim   time
# newfs -m 2 /dev/dsk/c2d0s0
newfs: construct a new file system /dev/rdsk/c2d0s0: (y/n)? y
(output omitted)
# fstyp -v /dev/dsk/c2d0s0 | grep minfree
minfree 2%      maxbpg  2048    optim   time
#
```

The following command shows the minfree value for the file system on the c0t0d0s0 device.

```
# fstyp -v /dev/rdsk/c0t0d0s0 | head
ufs
magic   11954   format  dynamic time    Fri Oct 22 10:09:11 2004
sblkno  16      cblkno  24      iblkno  32      dblkno  456
sbsize  5120    cgsize  5120    cgoffset 72     cgmask  0xffffffe0
ncg     110     size    3147511 blocks  3099093
bsize   8192    shift   13      mask    0xffffe000
fsize   1024    shift   10      mask    0xfffffc00
frag    8       shift   3       fsbtodb 1
minfree 2%      maxbpg  2048    optim   time
maxcontig 128   rotdelay 0ms    rps     120
```

To change the minimum percentage value of free space on an existing file system, you can use the tunefs -m *%free* command.

The following command changes the minimum percentage of free space on the /dev/rdsk/c0t0d0s0 device to 1 percent.

```
# tunefs -m 1 /dev/rdsk/c0t0d0s0
minimum percentage of free space changes from 2% to 1%
#
```

Checking the File System Using the fsck Command

A file system can become damaged if it is corrupted from a power failure, a software error in the kernel, a hardware failure, or an improper shutdown of the system. The file system check program, fsck, checks the data consistency of a file system and attempts to correct or repair any inconsistencies or damage found.

Caution – Never run the fsck command on a mounted file system. This could leave the file system in an unusable state. It could also delete data. The / (root), /usr, and /var file systems should have the fsck command run on them while in single-user mode.

Every time you boot a system, the operating system determines which file systems the fsck command should check. The fsck command checks and repairs any problems encountered in file systems before they are mounted.

Note – The status of a file system's state flag determines whether the file system needs to be scanned by the fsck command. When the state flag is "clean," "stable," or "logging," file system scans are not run.

Data Inconsistencies Checked by the fsck Command

The fsck command makes several passes through a file system. During each pass, the fsck command checks for several types of file system inconsistencies.

Superblock Consistency

The file system superblock is checked for inconsistencies involving such parameters as file system size, free block count, and free inode count.

Cylinder Group Block Consistency

The fsck command checks any unallocated data blocks claimed by inodes, the unallocated data block count, and the unallocated inode count.

Inode Consistency

The fsck command checks for the allocation state of inodes, as well as the type, the link count, duplicate blocks (blocks already claimed by another inode), bad blocks, the inode size, and the block count for each inode. Any unreferenced inode with a nonzero link count is linked to the file system's lost+found directory.

Data Block Consistency

The fsck command cannot check ordinary data blocks, but it can check directory data blocks. In directory data blocks, the fsck command checks for inodes that point to unallocated blocks, unallocated blocks tagged as in use, allocated blocks tagged as free (incorrect inodes for . and ..) and directories not connected to the file system. These directories are linked back to the file system in its lost+found directory.

The lost+found Directory

The fsck command puts files and directories that are allocated but unreferenced in the lost+found directory located in that file system. The inode number of each file is assigned as the file name. If the lost+found directory does not exist, the fsck command creates it. If not enough space exists in the lost+found directory, the fsck command increases the directory's size.

Non-Interactive Mode

During a normal system boot, the fsck command operates in noninteractive mode, which is often referred to as preen, or silent mode. In this mode, the fsck command addresses only minor inconsistency problems that can be corrected. If a more serious inconsistency is found and a decision has to be made, the fsck program terminates and requests the root password to enter single-user mode. Execute the fsck command in interactive mode to continue.

Interactive Mode

In interactive mode, the fsck command lists each problem it encounters, followed by a suggested corrective action in the form of a question that requires a yes or no response.

The following example shows how the fsck command displays a message that asks if you want to correct the block count.

```
# fsck /dev/rdsk/c0t0d0s7
** /dev/rdsk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
INCORRECT BLOCK COUNT I=743 (5 should be 2)
CORRECT?
```

If you respond with yes, the fsck command applies the corrective action and moves on. If you respond with no, the fsck command repeats the message about the original problem and suggests corrective action. It does not fix the inconsistency until you respond yes.

The following examples demonstrate how you as the system's root user can run the fsck command to check the integrity of file systems.

- To check a single unmounted file system, perform the command:

  ```
  # fsck /dev/rdsk/c0t0d0s7
  ```

  This is the only way to check a file system that has not been entered in the /etc/vfstab file.

- To check a file system using the mount point directory name as listed in the /etc/vfstab file, perform the command:

  ```
  # fsck /export/home
  ```

In the following example, the fsck command checks and repairs the file system with the force (f) and preen (p) options.

```
# fsck -o f,p /dev/rdsk/c0t0d0s7
/dev/rdsk/c0t0d0s7: 77 files, 9621 used, 46089 free
/dev/rdsk/c0t0d0s7: (4 frags, 57 blocks, 0.0% fragmentation)
```

The f option of the fsck command forces a file system check, regardless of the state of the file system's superblock state flag.

The p option checks and fixes the file system noninteractively (preen). The program exits immediately if a problem requiring intervention is found.

Resolving File System Inconsistencies

If problems are located in a file system, you are alerted by the `fsck` utility. Some of the more common file system errors that require interactive intervention are:

- Allocated unreferenced file
- Inconsistent link count
- Free block count corruption
- Superblock corruption

Reconnecting an Allocated Unreferenced File

If the `fsck` command discovers an inode that is allocated but unreferenced or not linked in any directory, the command sends a message that asks you if you want to reconnect the inode.

```
** Phase 3a - Check Connectivity
UNREF FILE I=788 OWNER=root MODE=100644
SIZE=19994 MTIME=Oct 18 10:49 2004
RECONNECT? y
```

A yes response causes the `fsck` command to save the file to the `lost+found` directory. The `fsck` command references the inode number.

To determine the type of file moved to the `lost+found` directory by the `fsck` command, perform the following steps:

1. List the contents of the file system's `lost+found` directory.

   ```
   # ls /export/home/lost+found
      #788
   ```

2. Determine the file type by using the `file` command.

   ```
   # file /export/home/lost+found/#788
   /export/home/lost+found/#788: ascii text
   ```

3. To view the contents of an ASCII text file, use the `more` or `cat` command. To view the contents of a binary file, use the `strings` command. If the file is associated with an application, such as a word processing document, use the application to view the contents of the file.

   ```
   # cat /export/home/lost+found/#788
   ```

4. If the file is intact and you know where it belongs, you can copy the file back to its original location in the file system.

```
# cp /export/home/lost+found/#788 /export/home/user1/report
```

Adjusting a Link Counter

If the fsck program discovers that the value of a directory inode link counter and the actual number of directory links are inconsistent, the command displays a message that asks you if you want to adjust the counter.

```
** Phase 4 - Check Reference Counts
LINK COUNT DIR I=2 OWNER=root MODE=40755
SIZE=512 MTIME=Oct 18 15:59 2004 COUNT 4 SHOULD BE 3
ADJUST? y
```

In the example, a **y** (yes) response causes the fsck command to correct the directory inode link counter from 4 to 3.

During this phase, you might also be asked to clear or remove a link.

```
BAD/DUP type I=200 OWNER=root MODE=40755
SIZE=512 MTIME=Mar 14 08:03 2004
CLEAR? y
```

Salvaging the Free List

If the fsck utility discovers that the unallocated block count and the free block number listed in the superblock are inconsistent, the fsck command displays a message that asks if you want to salvage the free block count by rectifying it with the unallocated block count.

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

In the example, a **y** (yes) response causes the fsck command to update the information in the file system superblock.

Using Backup Superblocks

Superblock corruption can cause a file system to be unmountable. A file system is unusable when the message such as "`Can't mount` *`file_system_name`*" or "*`device_name`* `is not this fstype`" appears. For example:

```
Can't mount /dev/dsk/c0t0d0s7

mount: /dev/dsk/c2d0s0 is not this fstype
```

This message can appear during a system boot or when you are manually mounting the file system.

If the `fsck` command fails because of a corrupted superblock, you see a series of messages that suggest searching for alternate superblocks using different methods. For example:

```
# fsck /dev/rdsk/c2d0s3
** /dev/rdsk/c2d0s3
BAD SUPERBLOCK AT BLOCK 16: MAGIC NUMBER WRONG

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? n


LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? n

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS? n


USE GENERIC SUPERBLOCK FROM NEWFS? n

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED. YOU MUST USE THE -o b OPTION
TO FSCK TO SPECIFY THE LOCATION OF A VALID ALTERNATE SUPERBLOCK TO
SUPPLY NEEDED INFORMATION; SEE fsck(1M).
#
```

In this example the `n` responses are meant to illustrate where `fsck` would look for backup superblocks.

You can try any of these methods to locate a backup superblock. For example:

```
# fsck /dev/rdsk/c2d0s3
** /dev/rdsk/c2d0s3
BAD SUPERBLOCK AT BLOCK 16: MAGIC NUMBER WRONG

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? y

FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS

USE ALTERNATE SUPERBLOCK? y

FOUND ALTERNATE SUPERBLOCK AT 32 USING MKFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? y

Please verify that the indicated block contains a proper
superblock for the filesystem (see fsdb(1M)).
#
```

Note the warning issued by `fsck`. If you manually specified the file system geometry when you created it, this automatic discovery process may not provide a valid superblock, and could damage the file system if used. Use these automatic discover methods only on file systems that were created using default parameters.

To use a backup superblock that you manually select, use the `fsck` command with the `-o` option and with the `b` flag followed by a backup superblock number. Every file system has an alternative backup superblock at block number 32, which can be used with the `fsck` command to repair the primary superblock.

The following command uses a backup superblock.

```
# fsck -o b=32 /dev/rdsk/c2d0s0
Alternate super block location: 32.
** /dev/rdsk/c2d0s0
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups

UPDATE STANDARD SUPERBLOCK? y

2 files, 9 used, 19626536 free (8 frags, 2453316 blocks, 0.0%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
#
```

The `fsck` utility compares the information in the backup superblock with the actual file system and attempts to rebuild the primary superblock. However, if the first backup superblock is part of the file system that was damaged, it might be unusable. If it is unusable, select another backup superblock to continue the `fsck` command.

To list the locations of all the alternative backup superblocks in the file system, run the `newfs -N` command.

Caution – This method works if the underlying file system was built using the `newfs` default parameters. If the file system was not built with these defaults, execute the `newfs -N` command, using the same parameters originally used, to generate identical superblock locations.

Use the `-N` option to view the file system parameters that you could use to create a new file system without actually creating the file system. A portion of the output lists the locations of alternative backup superblocks that can be used with the `fsck -o b=#` *command*.

```
# newfs -N /dev/rdsk/c2d0s0
Warning: 5008 sector(s) in last cylinder unallocated
/dev/rdsk/c2d0s0:       39857264 sectors in 6488 cylinders of 48 tracks,
128 sectors
        19461.6MB in 406 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488,
885920,
Initializing cylinder groups:
.......
super-block backups for last 10 cylinder groups at:
 38929952, 39028384, 39126816, 39225248, 39323680, 39422112, 39520544,
 39618976, 39717408, 39815840
#
```

If you use the `newfs` command with the -T option, the structure created allows the file system to grow to a multi-terabyte file system. You can view the file system parameters using this option without actually creating the file system.

Note – Using the `newfs -T` option on smaller file systems that will contain many small files can cause a shortage of inodes.

```
# newfs -N -T /dev/rdsk/c2d0s0
Warning: 14176 sector(s) in last cylinder unallocated
/dev/rdsk/c2d0s0:       39857264 sectors in 2637 cylinders of 240 tracks,
63 sectors
        19461.6MB in 48 cyl groups (56 c/g, 413.44MB/g, 448 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 846816, 1693600, 2540384, 3387168, 4233952, 5080736, 5927520,
6774304,
 7621088,
 32177824, 33024608, 33871392, 34718176, 35564960, 36411744, 37258528,
 38105312, 38952096, 39798880
#
```

You can use any other alternative superblock number in the list with the `fsck` command.

```
# fsck -o b=590624 /dev/rdsk/c2d0s0
Alternate super block location: 590624.
** /dev/rdsk/c2d0s0
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups

UPDATE STANDARD SUPERBLOCK? y

2 files, 9 used, 19626536 free (8 frags, 2453316 blocks, 0.0%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
#
```

Monitoring File System Use

An important activity of a system administrator is to monitor file system use on a regular basis. There are three useful commands available for this task:

- `df` – Displays the number of free disk blocks
- `du` – Summarizes disk use
- `quot` – Summarizes file system ownership

Using the `df` Command

Use the `df` command to display the amount of disk space used in file systems. This command lists the amount of used and available space and the amount of the file system's total capacity being used.

The format for the `df` command is:

```
df -option resource
```

Table 4-1 lists some of the more common options used with the `df` command.

Table 4-1 Partial Listing of Options for the `df` Command

Option	Description
`-a`	Reports on all file systems, including those with entries in the `/etc/mnttab` file for which the ignore option is set
`-b`	Prints the total number of Kbytes free
`-e`	Prints only the number of files free
`-k`	Displays disk allocation in Kbytes
`-h`	Acts like the `-k` option, except that sizes are in a more readable format, for example, `14K`, `234M`, `2.7G`, or `3.0T`
`-l`	Reports on local file systems only
`-F` FSType	Specifies the file system type on which to operate. This is intended for use on unmounted file systems.

To display the capacity of file systems, perform the command:

```
# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0    5042262  156789 4835051     4%    /
/devices                   0       0       0     0%    /devices
ctfs                       0       0       0     0%    /system/contract
proc                       0       0       0     0%    /proc
mnttab                     0       0       0     0%    /etc/mnttab
swap                 1149584    1224 1148360     1%    /etc/svc/volatile
objfs                      0       0       0     0%    /system/object
/dev/dsk/c0t0d0s6    10085284 5132912 4851520    52%    /usr
/platform/sun4u-us3/lib/libc_psr/libc_psr_hwcap1.so.1
                     5042262  156789 4835051     4%    /platform/sun4u-
us3/lib/libc_psr.so.1
/platform/sun4u-us3/lib/sparcv9/libc_psr/libc_psr_hwcap1.so.1
                     5042262  156789 4835051     4%    /platform/sun4u-
us3/lib/sparcv9/libc_psr.so.1
fd                         0       0       0     0%    /dev/fd
/dev/dsk/c0t0d0s3     493487   81718  362421    19%    /var
swap                 1148392      32 1148360     1%    /tmp
swap                 1148416      56 1148360     1%    /var/run
/dev/dsk/c0t0d0s5     481551    5562  427834     2%    /opt
/dev/dsk/c0t0d0s7    98771928   65750 97718459     1%    /export/home
```

The same file system displayed with the -h option would appear in human-readable format.

```
# df -h
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0t0d0s0      4.8G   153M   4.6G     4%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   1.1G   1.2M   1.1G     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
/dev/dsk/c0t0d0s6      9.6G   4.9G   4.6G    52%    /usr
/platform/sun4u-us3/lib/libc_psr/libc_psr_hwcap1.so.1
                       4.8G   153M   4.6G     4%    /platform/sun4u-
us3/lib/libc_psr.so.1
/platform/sun4u-us3/lib/sparcv9/libc_psr/libc_psr_hwcap1.so.1
                       4.8G   153M   4.6G     4%    /platform/sun4u-
us3/lib/sparcv9/libc_psr.so.1
fd                       0K     0K     0K     0%    /dev/fd
/dev/dsk/c0t0d0s3      482M    80M   354M    19%    /var
swap                   1.1G    32K   1.1G     1%    /tmp
swap                   1.1G    56K   1.1G     1%    /var/run
/dev/dsk/c0t0d0s5      470M   5.4M   418M     2%    /opt
/dev/dsk/c0t0d0s7       94G    64M    93G     1%    /export/home
```

Table 4-2 defines the fields displayed by the `df -k` command.

Table 4-2 Fields for the `df -k` Command

Field	Definition
`Filesystem`	The mounted file system
`kbytes`	The size of the file system in Kbytes (1024 bytes)
`used`	The number of Kbytes used
`avail`	The number of Kbytes available
`capacity`	The percentage of file system capacity used
`Mounted on`	The mount point

The amount of space that is reported as `used` and `avail` is typically less than the amount of total space in the file system. A fraction of space, from 1 to 10 percent, is reserved in each file system as the `minfree` value.

When all of the reported space on the file system is in use, the file system capacity is displayed as 100 percent. Regular users receive the message "`File System Full`" and cannot continue working. The reserved space is available to the `root` user, who can then delete or back up files in the file system.

The **`df -k`** command can be used with the device as the resource to show available space on the device:

```
# df -k /dev/dsk/c0t1d0s6
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t1d0s6    17153338       9 16810225     0%
#
```

Note – This command does not work on a partition without a file system, but does work on a partition with an unmounted file system.

Using the du Command

Use the du command to display the number of disk blocks used by directories and files. Each disk block consists of 512 bytes. The format for the du command is:

```
du -options directory
```

Table 4-3 describes the options for the du command.

Table 4-3 Options for the du Command

Option	**Description**
-k	Displays disk use in Kbytes.
-s	Displays only the summary in 512-byte blocks. Using the s and k options together shows the summary in Kbytes.
-a	Displays the number of blocks used by all files in addition to directories within the specified directory hierarchy.

To display disk usage in kilobytes, perform the command:

```
# cd /opt
# du -k
3       ./SUNWits/Graphics-sw/xil/lib
4       ./SUNWits/Graphics-sw/xil
5       ./SUNWits/Graphics-sw
6       ./SUNWits
15      ./SUNWmlib/lib/sparcv8
15      ./SUNWmlib/lib/sparcv8plus
15      ./SUNWmlib/lib/sparcv8plus+vis
15      ./SUNWmlib/lib/sparcv8plus+vis2
15      ./SUNWmlib/lib/sparcv9
15      ./SUNWmlib/lib/sparcv9+vis
15      ./SUNWmlib/lib/sparcv9+vis2
120     ./SUNWmlib/lib
24      ./SUNWmlib/include
146     ./SUNWmlib
376     ./SUNWrtvc/bin
10      ./SUNWrtvc/examples/rtvc_capture_movie
24      ./SUNWrtvc/examples/rtvc_display
68      ./SUNWrtvc/examples/rtvc_video_conference
25      ./SUNWrtvc/examples/test
128     ./SUNWrtvc/examples
7       ./SUNWrtvc/man/man1
```

```
19      ./SUNWrtvc/man/man3
28      ./SUNWrtvc/man
533     ./SUNWrtvc
686     .
```

To display disk use in human readable form, perform the command:

```
# du -h /opt |more
   3K   /opt/SUNWits/Graphics-sw/xil/lib
   4K   /opt/SUNWits/Graphics-sw/xil
   5K   /opt/SUNWits/Graphics-sw
   6K   /opt/SUNWits
  24K   /opt/SUNWmlib/include
  15K   /opt/SUNWmlib/lib/sparcv8
  15K   /opt/SUNWmlib/lib/sparcv8plus
  15K   /opt/SUNWmlib/lib/sparcv8plus+vis
  15K   /opt/SUNWmlib/lib/sparcv8plus+vis2
  15K   /opt/SUNWmlib/lib/sparcv9
  15K   /opt/SUNWmlib/lib/sparcv9+vis
  15K   /opt/SUNWmlib/lib/sparcv9+vis2
 120K   /opt/SUNWmlib/lib
 146K   /opt/SUNWmlib
 376K   /opt/SUNWrtvc/bin
  10K   /opt/SUNWrtvc/examples/rtvc_capture_movie
  24K   /opt/SUNWrtvc/examples/rtvc_display
  68K   /opt/SUNWrtvc/examples/rtvc_video_conference
  25K   /opt/SUNWrtvc/examples/test
 128K   /opt/SUNWrtvc/examples
   7K   /opt/SUNWrtvc/man/man1
  19K   /opt/SUNWrtvc/man/man3
```

To display disk use including files, perform the command:

```
# du -ak /opt
1       /opt/SUNWits/Graphics-sw/xil/lib/libxil.so
1       /opt/SUNWits/Graphics-sw/xil/lib/libxil.so.1
3       /opt/SUNWits/Graphics-sw/xil/lib
4       /opt/SUNWits/Graphics-sw/xil
(output removed for brevity)
19      /opt/SUNWrtvc/man/man3
1       /opt/SUNWrtvc/man/windex
28      /opt/SUNWrtvc/man
533     /opt/SUNWrtvc
686     /opt
```

To display a disk use summary, perform the command:

```
# du -sk /opt
686     /opt
```

Using the `quot` Command

Use the `quot` command to display how much disk space, in kilobytes, is being used by users. The format for the `quot` command is:

```
quot -options filesystem
```

Table 4-4 describes the options for the `quot` command.

Table 4-4 Options for the `quot` Command

Option	Description
`-a`	Reports on all mounted file systems
`-f`	Includes the number of files

To display disk space being used by users on all mounted file systems, perform the command:

```
# quot -af
/dev/rdsk/c0t0d0s0 (/):
112410   5246   root
   31      12   uucp
   11      11   lp
    1       1   adm
/dev/rdsk/c0t0d0s6 (/usr):
2313692 102415  root
  806      15   uucp
   11       4   bin
    1       1   adm
...
```

The columns represent kilobytes used, number of files, and owner, respectively.

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

```
# quot -f /dev/dsk/c0t0d0s7
/dev/rdsk/c0t0d0s7 (/export/home):
     9       2   root
    15      35   sue
    51      51   paul
    23      25   jeff
```

Exercise: Creating and Maintaining ufs File Systems

In this exercise, you complete the following tasks:

- Create ufs file systems
- Calculate and adjust minfree values
- Destroy the superblock on an unused file system and repair it using an alternative superblock

Preparation

This exercise requires an unused disk, divided into four slices. Slices 0, 1, and 3 must be equal in size, roughly one quarter of the entire disk each, and Slice 4 takes up the remaining space on the disk. If it is necessary to partition this disk, this exercise requires an understanding of how to use the format utility.

This exercise applies to both SPARC and x86/x64 systems. Keep in mind that these different systems typically use different disk device names. For example:

- The spare disk of a Sun Blade 1500 (SPARC) is typically c0t1d0.
- The spare disk of an Ultra 20 (x86/x64) is typically c2d0.

Throughout this exercise, be certain to use the correct device names for your particular system.

Task

Complete the following steps:

1. Log in as the root user and complete the following steps:
   a. Open a terminal window.
   b. Change directory to /dev/rdsk.
2. Complete the following steps to identify a spare disk:
   a. To find a spare disk, use the ls command to display a list of possible disks and the prtvtoc command to display the VTOC for each disk you find.

b. Examine the Partition list, as well as the Mount Directory field that the prtvtoc command displays.

   On systems in the classroom, disks that are not in use have no mount directory listed.

c. Record the name of the unused disk.

   Unused disk:

   ____

Note – This procedure works for the classroom environment. A disk that does not show mounted slices in the Mount Directory field of the prtvtoc output is not necessarily unused.

---

3. If the spare disk is not divided into four slices as described in the Preparation section of this exercise, complete the following steps:

   a. On x86/x64 systems, use the fdisk menu in the format utility to create a single Solaris fdisk partition that uses the entire disk. Make sure the partition is not marked as *active*.

   b. Use the partition menu in the format utility to create the required slices.

   c. Make slices 0, 1, and 3 exactly the same size (approximately 25 percent of the total disk space each), and use slice 4 for the remainder of the available space. Using the All Free Hog method and choosing slice 4 as the free hog slice makes this easy.

      On SPARC systems, you can also use the Solaris Management Console to partition the disk.

   d. Save your changes and quit the format utility when you are finished.

Example of a correct SPARC partition table:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)  56626320
  1       swap    wu   13879 - 27757       27.00GB    (13879/0/0)  56626320
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm   27758 - 41636       27.00GB    (13879/0/0)  56626320
  4 unassigned    wm   41637 - 57458       30.78GB    (15822/0/0)  64553760
  5 unassigned    wm       0                0         (0/0/0)             0
  6        usr    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
```

Example of a correct x86/x64 partition table:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       3 -  2483       19.01GB    (2481/0/0)   39857265
  1       swap    wu    2484 -  4964       19.01GB    (2481/0/0)   39857265
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wm    4965 -  7445       19.01GB    (2481/0/0)   39857265
  4 unassigned    wm    7446 - 10007       19.63GB    (2562/0/0)   41158530
  5 unassigned    wm       0                0         (0/0/0)             0
  6        usr    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130
```

4. Use the `newfs` command without options to create a new file system on Slice 0 on the spare disk and complete the following steps:

   a. Observe how quickly the `newfs` command creates cylinder groups on this slice.

   b. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

      Cylinder groups: ________________________________

      Cylinders per group: ____________________________

      Inodes per group: _______________________________

5. Use the `newfs` command to create a new file system on slice 1 on the spare disk, and complete the following steps:

   a. Use the `-i` option to create one inode per 16,384 bytes of data space.

   b. Observe how quickly the `newfs` command creates cylinder groups on this slice.

c. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

   Cylinder groups: ______________________________

   Cylinders per group: ____________________________

   Inodes per group: ______________________________

According to the statistics you have gathered, how do the file systems on slices 0 and 1 differ?

____

____

6. Complete the following steps:

   a. Use the `df` command to display statistics for the file systems on slices 0 and 1 that you used in the previous steps, for example:

   b. Record the values listed in the `kbytes`, `used`, and `avail` columns.

   Which file system has the larger amount of available data space and why?

   ____

   ____

7. Complete the following steps:

   a. For each file system, add the `used` and `avail` values, and compare the sum to the `kbytes` value.

      Expressed as a percentage, how much larger is the `kbytes` value than the sum of `used` and `avail`? This percentage should approximately match the `minfree` value.

   b. Use the **`fstyp -v /dev/rdsk/`***`c#t#d#s#`* **`| grep minfree`** command to verify your result.

8. Use the `tunefs -m` *`X`* `/dev/rdsk/`*`c#t#d#s0`* command, where *`X`* is the minifree value, to change the `minfree` value for the file system on slice 0 of the spare disk.

   - If the current `minfree` value is greater than 5 percent, reduce it by 3 percent.
   - If it is less than or equal to 5 percent, add 3 percent.

   What message does the `tunefs` command display?

   ____

9. Complete the following steps:

   a. Use the `df -k` command to display the space utilization for the file system on slice 0 of your spare disk. Observe what that the `minfree` value has changed in the output.

   b. Record the values listed in the `kbytes`, `used`, and `avail` columns.

   ____

   Which value has changed from the information you gathered in Step 6?

10. Create new file systems on slices 3 and 4 of your spare disk.

11. Run the `fsck` command interactively to check the new file system that you created on slice 3 of the spare disk.

    Did the `fsck` command report errors?

    ____

12. Use the `dd` command to destroy the main superblock of the file system on slice 3. For example:

    # **dd if=/dev/zero of=/dev/rdsk/*c0t1d0s3* count=32 bs=512**

13. Run the `fsck` command interactively to check the new file system. Respond `n` to the prompts presented by `fsck`.

    Did the `fsck` command report errors? If so, what corrective action does the `fsck` command suggest?

    ____

    ____

    ____

14. Run the `fsck` command, and specify an alternative superblock to use on slice 3.

    Block 32 is always one of the alternatives available.

15. Run the `fsck` command again to verify that the file system was repaired.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Creating and Maintaining ufs File Systems

This section contains solutions to the exercises.

Task

Review the following solutions:

1. Log in as the root user and complete the following steps:
   a. Open a terminal window.
   b. Change directory to /dev/rdsk.

   ```
   # cd /dev/rdsk
   ```
2. Complete the following steps to identify a spare disk:
   a. To find a spare disk, use the ls command to display a list of possible disks and the prtvtoc command to display the VTOC for each disk you find.
   b. Examine the Partition list, as well as the Mount Directory field that the prtvtoc command displays.

   On systems in the classroom, disks that are not in use have no mount directory listed.

   c. Record the name of the unused disk.

   ```
   # ls *s2

   # prtvtoc /dev/rdsk/c0t1d0s0
   ```

   Unused disk:

Note – This procedure works for the classroom environment. A disk that does not show mounted slices in the Mount Directory field of the prtvtoc output is not necessarily unused.

3. If the spare disk is not divided into four slices as described in the Preparation section of this exercise, complete the following steps:
   a. On x86/x64 systems, use the fdisk menu in the format utility to create a single Solaris fdisk partition that uses the entire disk. Make sure the partition is not marked as *active*.

b. Use the partition menu in the format utility to create the required slices.

c. Make slices 0, 1, and 3 exactly the same size (approximately 25 percent of the total disk space each), and use slice 4 for the remainder of the available space. Using the All Free Hog method and choosing slice 4 as the free hog slice makes this easy.

On SPARC systems, you can also use the Solaris Management Console to partition the disk.

d. Save your changes and quit the format utility when you are finished.

Example of a correct SPARC partition table:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)  56626320
  1       swap    wu   13879 - 27757       27.00GB    (13879/0/0)  56626320
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm   27758 - 41636       27.00GB    (13879/0/0)  56626320
  4 unassigned    wm   41637 - 57458       30.78GB    (15822/0/0)  64553760
  5 unassigned    wm       0               0          (0/0/0)             0
  6        usr    wm       0               0          (0/0/0)             0
  7 unassigned    wm       0               0          (0/0/0)             0
```

Example of a correct x86/x64 partition table:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       3 -  2483       19.01GB    (2481/0/0)   39857265
  1       swap    wu    2484 -  4964       19.01GB    (2481/0/0)   39857265
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wm    4965 -  7445       19.01GB    (2481/0/0)   39857265
  4 unassigned    wm    7446 - 10007       19.63GB    (2562/0/0)   41158530
  5 unassigned    wm       0               0          (0/0/0)             0
  6        usr    wm       0               0          (0/0/0)             0
  7 unassigned    wm       0               0          (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130
```

4. Use the `newfs` command without options to create a new file system on Slice 0 on the spare disk and complete the following steps:
   a. Observe how quickly the `newfs` command creates cylinder groups on this slice.
   b. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

   ```
   # newfs /dev/rdsk/c0t1d0s0
   ```

   Cylinder groups:

   Cylinders per group:

   Inodes per group:

5. Use the `newfs` command to create a new file system on slice 1 on the spare disk, and complete the following steps:
   a. Use the `-i` option to create one inode per 16,384 bytes of data space.
   b. Observe how quickly the `newfs` command creates cylinder groups on this slice.
   c. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

   ```
   # newfs -i 16384 /dev/rdsk/c0t1d0s1
   ```

   Cylinder groups:

   Cylinders per group:

   Inodes per group:

   According to the statistics you have gathered, how do the file systems on slices 0 and 1 differ?

   The number of inodes per group is less on file system 1 than on file system 0. The number of cylinder groups and cylinders per group have not changed.

6. Complete the following steps:

   a. Use the `df` command to display statistics for the file systems on slices 0 and 1 that you used in the previous steps, for example:

```
# df -k /dev/dsk/c0t1d0s0
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t1d0s0    27883846       9 27604953     0%
# df -k /dev/dsk/c0t1d0s1
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t1d0s1    28091566       9 27810576     0%
```

   b. Record the values listed in the `kbytes`, `used`, and `avail` columns.

   Which file system has the larger amount of available data space and why?

   File system 1 has the larger amount of available data space because it holds fewer inode records.

7. Complete the following steps:

   a. For each file system, add the `used` and `avail` values, and compare the sum to the `kbytes` value.

      Expressed as a percentage, how much larger is the `kbytes` value than the sum of `used` and `avail`? This percentage should approximately match the `minfree` value.

   b. Use the **`fstyp -v /dev/rdsk/`***`c#t#d#s#`* **`| grep minfree`** command to verify your result.

   To calculate the percentage difference between the sum of `used` *and* `avail` *and the* `kbytes` *value, complete the following steps:*

   a. *Add the values listed as* `used` *and* `avail`*, for example:*

      9 + 27604953 = 27604962

   b. *Divide the sum of* `used` *and* `avail` *by the* `kbytes` *value, for example:*

      27604962 / 27883846 = 0.989998

   c. *Multiply the result of Step b by 100, for example*:

      0.989998 * 100 = 98.9998

   d. *Subtract the result of Step c from 100, for example:*

      100 - 98.9998 = 1.0002

   e. *Round the result of Step d to the nearest whole number, for example:*

      1.0002 = 1percent

8. Use the `tunefs -m X /dev/rdsk/c#t#d#s0` command, where *X* is the minifree value, to change the `minfree` value for the file system on slice 0 of the spare disk.
   - If the current `minfree` value is greater than 5 percent, reduce it by 3 percent.
   - If it is less than or equal to 5 percent, add 3 percent, for example:

   ```
   # tunefs -m 4 /dev/rdsk/c0t1d0s0
   ```

   What message does the `tunefs` command display?

```
minimum percentage of free space changes from (old value) to (new value)
```

9. Complete the following steps:
   a. Use the `df -k` command to display the space utilization for the file system on slice 0 of your spare disk. Observe that the `minfree` value has changed in the output.
   b. Record the values listed in the `kbytes`, `used`, and `avail` columns, for example:

```
# df -k /dev/dsk/c0t1d0s0
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t1d0s0    27883846       9 26768439     0%
```

   Which value has changed from the information you gathered in Step 6?

   The `avail` *column changes, but not the* `kbytes` *or* `used` *columns.*

10. Create new file systems on slices 3 and 4 of your spare disk, for example:

   ```
   # newfs /dev/rdsk/c0t1d0s3
   # newfs /dev/rdsk/c0t1d0s4
   ```

11. Run the `fsck` command interactively to check the new file system that you created on slice 3 of the spare disk.

   ```
   # fsck /dev/rdsk/c0t1d0s3
   ```

   Did the `fsck` command report errors?

   No.

12. Use the `dd` command to destroy the main superblock of the file system on slice 3.

```
# dd if=/dev/zero of=/dev/rdsk/c0t1d0s3 count=32 bs=512
```

13. Run the fsck command interactively to check the new file system. Respond n to the prompts presented by fsck.

    `# fsck /dev/rdsk/c0t1d0s3`

    Did the fsck command report errors? If so, what corrective action does the fsck command suggest?

    The fsck command indicates that the magic number in the superblock is wrong and suggests looking for alternate superblocks in different ways, and repairing it by using one of them, for example:

```
# fsck /dev/rdsk/c0t1d0s3
** /dev/rdsk/c0t1d0s3
BAD SUPERBLOCK AT BLOCK 16: MAGIC NUMBER WRONG

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? n


LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? n

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS? n


USE GENERIC SUPERBLOCK FROM NEWFS? n

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED. YOU MUST USE THE -o b OPTION
TO FSCK TO SPECIFY THE LOCATION OF A VALID ALTERNATE SUPERBLOCK TO
SUPPLY NEEDED INFORMATION; SEE fsck(1M).
#
```

14. Run the `fsck` command, and specify an alternative superblock to use on slice 3.

    Block 32 is always one of the alternatives available.

```
# fsck -o b=32 /dev/rdsk/c0t1d0s3
Alternate super block location: 32.
** /dev/rdsk/c0t1d0s3
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups

UPDATE STANDARD SUPERBLOCK? y

2 files, 9 used, 27883837 free (13 frags, 3485478 blocks, 0.0%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****
#
```

15. Run the `fsck` command again to verify that the file system was repaired.

    `# fsck /dev/rdsk/c0t1d0s3`

    This time the `fsck` *command output does not report that the file system was modified.*

Notes:

Module 5

ZFS File System Introduction

Objectives

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

- Describe the Solaris ZFS file system
- Understand ZFS terminology
- Create new ZFS pools and file systems
- Modify ZFS file system properties
- Destroy ZFS pools and file systems
- Work with ZFS snapshots and clones

What Is ZFS?

ZFS is a file system that:

- Integrates filesystem and volume management to simplify administration
- Supports high storage capacities and avoids arbitrary limits
- Supports unlimited snapshots and clones
- Checksums all data
- Is open source, and available in Solaris 10, OpenSolaris, MacOS X and FreeBSD.

ZFS Pooled Storage

ZFS uses storage pools to manage physical storage. Historically, file systems were constructed on top of a single physical device. To address multiple devices and provide for data redundancy, the concept of a volume manager was introduced to provide the image of a single device so that file systems would not have to be modified to take advantage of multiple devices.

ZFS eliminates the volume management altogether. Instead of forcing you to create virtualized volumes, ZFS aggregates devices into a storage pool. The storage pool describes the physical characteristics of the storage (device layout, data redundancy, and so on) and acts as an arbitrary data store from which file systems can be created.

File systems are no longer constrained to individual devices, allowing them to share space with all file systems in the pool. You don't need to predetermine the file system size, as file systems grow automatically within the space allocated to the storage pool.

Transactional Semantics

ZFS is a transactional file system, which means that the file system state is always consistent on disk. With a transactional file system, data is managed using copy on write semantics.

Data is never overwritten, and any sequence of operations is either entirely committed or entirely ignored. This mechanism means that the file system can never be corrupted through accidental loss of power or a system crash. So, no need for a `fsck` equivalent exists. While the most recently written pieces of data might be lost, the file system itself will always be consistent.

Checksums and Self-Healing Data

With ZFS, all data and metadata is checksummed using a user-selectable algorithm. ZFS checksums are stored in a way such that these failure modes are detected and can be recovered from gracefully. All checksumming and data recovery is done at the file system layer, and is transparent to applications.

In addition, ZFS provides for self-healing data. ZFS supports storage pools with varying levels of data redundancy, including mirroring and a variation on RAID-5. When a bad data block is detected, ZFS fetches the correct data from another replicated copy, and repairs the bad data, replacing it with the good copy.

Scalability

The ZFS file system was designed to be the most scalable file system, ever. The file system is 128-bit, allowing for 256 quadrillion zettabytes of storage. All metadata is allocated dynamically, so you don't need to pre-allocate inodes or otherwise limit file system scalability when you first create it. All the algorithms were written with scalability in mind. Directories can have up to 2^{48} (256 trillion) entries, and no limit exists on the number of file systems or number of files that can be contained within a file system.

Snapshots

A snapshot is a read-only copy of a file system or volume. You can create snapshots. Initially, snapshots consume no additional space within the pool.

Administration

The ZFS file system provides simplified administration. Through the use of hierarchical file system layout, property inheritance, and automanagement of mount points and NFS share semantics, ZFS makes it easy to create and manage file systems without needing multiple commands or editing configuration files. You can set quotas or reservations, turn compression on or off, or manage mount points for numerous file systems with a single command.

Devices can be examined or repaired without having to understand a separate set of volume manager commands. You can take an unlimited number of instantaneous snapshots of file systems. You can backup and restore individual file systems.

ZFS Terminology

Term	Definition
checksum	A 256-bit data hash in a file-system block. A checksum entity can range from the simple and fast `fletcher2` (the default) to cryptographically strong hashes like `SHA256`.
clone	A file system whose initial contents are identical to the contents of a snapshot.
dataset	A generic name for the following ZFS entities: clones, file systems, snapshots, or volumes. Each dataset is identified by a unique name in the ZFS namespace. Identify datasets with the following format: ***pool/path[@snapshot]***
file system	A ZFS dataset of type filesystem that is mounted within the standard system namespace and behaves like other file systems.
mirror	A virtual device that stores identical data copies on two or more disks. If any disk in a mirror fails, any other disk in that mirror can provide the same data.
pool	Identifies the storage pool name that contains the dataset. A logical group of devices describing the layout and physical characteristics of the available storage. Dataset space is allocated from a pool.
RAID-Z	A virtual device that stores data and parity on multiple disks, similar to RAID-5. For more information about RAID-Z, see ''RAID-Z Storage Pool Configuration'' on page 5-12.
resilvering	The process of transferring data from one device to another device. For example, if a mirror component is replaced or taken offline, the data from the up-to-date mirror component is copied to the newly restored mirror component. This process is called *mirror resynchronization* in traditional volume management products.
snapshot	A read-only image of a file system or volume at a given point in time.
virtual device	A logical device in a pool, which can be a physical device, a file, or a collection of devices.
volume	A dataset used to emulate a physical device. For example, you can create a ZFS volume as a swap device.

Creating a Basic ZFS File System

The ZFS file system reduces the commands you need to create a usable file system. When you create a new pool, a new ZFS file system is created and mounted automatically.

The following example illustrates how to create a storage pool named `tank` and a ZFS name `tank` in one command. Assume that the whole disk `/dev/dsk/c1t0d0` is available for use.

```
# zpool create tank c1t0d0
```

The new ZFS file system, `tank`, can use as much of the disk space on `c1t0d0` as needed, and is automatically mounted at `/tank`.

```
# mkfile 100m /tank/foo
# df -h /tank
Filesystem               size   used  avail capacity
Mounted on
tank                     80G   200M    80G     1%
/tank
```

Within a pool, you may want to create additional file systems. File systems provide points of administration that allow you to manage different sets of data within the same pool. In most cases, you will want to create and organize a hierarchy of file systems that matches your organizational needs

ZFS Storage Pool Components

This section provides detailed information about the following storage pool components:

- Disks
- Files
- Virtual devices
- Cache devices

Using Disks in a ZFS Storage Pool

The most basic element of a storage pool is a piece of physical storage. Physical storage can be any block device of at least 128 Mbytes. Typically, this device is a hard drive in the `/dev/dsk` directory.

A storage device can be a whole disk (`c1t0d0`) or an individual slice (`c0t0d0s7`). Sun recommends that you use an entire disk so that you don't have to specially format the disk.he disk ZFS formats the disk using an EFI label to contain a single large slice.

ZFS applies an EFI label when you create a storage pool with whole disks. Disks can be labeled with a traditional Solaris VTOC label when you create a storage pool with a disk slice.

Use a slice only under the following conditions:

- The device name is non-standard.
- A single disk is shared between ZFS and another file system, such as the UFS.
- A disk is used as a swap or a dump device.

Specify a disk by using a full path like `/dev/dsk/c1t0d0` or a shorthand name that consists of the device name within the `/dev/dsk` directory, such as `c1t0d0`. For example, the following are valid disk names:

- `c1t0d0`
- `/dev/dsk/c1t0d0`
- `c0t0d6s2`
- `/dev/foo/disk`

Using whole physical disks is the simplest way to create ZFS storage pools. ZFS configurations become progressively more complex, from management, reliability, and performance perspectives, when you build pools from disk slices, LUNs in hardware RAID arrays, or volumes presented by software-based volume managers. The following considerations might help you determine how to configure ZFS with other hardware or software storage solutions:

- If you construct ZFS configurations on top of LUNs from hardware RAID arrays, you need to understand the relationship between ZFS redundancy features and the redundancy features offered by the array. Certain configurations might provide adequate redundancy and performance, but other configurations might not.
- You can construct logical devices for the ZFS using volumes presented by software-based volume managers, such as Solaris™ Volume Manager (SVM) or Veritas Volume Manager (VxVM). However, these configurations are not recommended. While the ZFS functions properly on such devices, less-than-optimal performance could result.

Using Files in a ZFS Storage Pool

The ZFS file system allows you to use UFS files as virtual devices in your storage pool. This feature enables testing and simple experimentation. It is not for production use. This is because any file use relies on the underlying file system for consistency. If you create a ZFS pool backed by files on a UFS file system, then you are relying on the UFS to guarantee correctness and synchronous semantics.

Virtual Devices in a Storage Pool

Each storage pool consists of one or more virtual devices. A virtual device is an internal representation of the storage pool that describes the layout of physical storage and its fault characteristics. As such, a virtual device represents the disk devices or files that are used to create the storage pool.

Two top-level virtual devices provide data redundancy: mirror and RAID-Z virtual devices. These virtual devices consist of disks, disk slices, or files.

Disks, disk slices, or files that are used in pools outside of mirrors and RAID-Z virtual devices, function as top-level virtual devices.

Storage pools typically contain multiple top-level virtual devices. ZFS file dynamically stripes data among all of the top-level virtual devices in a pool.

Using Cache Devices in a ZFS Storage Pool

In the Solaris 10 10/09 release, you can create a ZFS storage pool and specify cache devices, which are used to cache storage pool data. Cache devices provide an additional layer of caching between main memory and disk memory. Using cache devices provide the greatest performance improvement for random read-workloads of mostly static content. One or more cache devices can be specified when the pool is created. For example:

```
# zpool create pool mirror c0t2d0 c0t4d0 cache c0t0d0
# zpool status pool
   pool: pool
  state: ONLINE
    scrub: none requested
   config:

          NAME         STATE    READ WRITE  CKSUM
          pool         ONLINE      0     0      0
            mirror     ONLINE      0     0      0
              c0t2d0   ONLINE      0     0      0
              c0t4d0   ONLINE      0     0      0
          cache
         c0t0d0        ONLINE      0     0      0

errors: No known data errors
```

Cache devices can be added, or removed from a storage pool after the pool is created.

Note – For information about determining whether using cache devices is appropriate for your environment, see the *Solaris ZFS Administration Guide*.

ZFS Storage Pool Replication Features

Mirrored Storage Pool Configuration

A mirrored storage pool configuration requires at least two disks, preferably on separate controllers. Many disks can be used in a mirrored configuration. In addition, you can create more than one mirror in each pool. Conceptually, a simple mirrored configuration would look like this:

```
mirror c1t0d0 c2t0d0
```

Conceptually, a more complex mirrored configuration would look like this:

```
mirror c1t0d0 c2t0d0 c3t0d0 mirror c4t0d0 c5t0d0 c6t0d0
```

Each `mirror` argument defines a separate top-level virtual device. The first example defines one two-way mirror device. The second example defines two, three-way mirror devices.

RAID-Z Storage Pool Configuration

In addition to a mirrored storage pool configuration, the ZFS file system provides a RAID-Z configuration. RAID-Z is similar to RAID-5.

In RAID-Z, ZFS file uses variable-width RAID stripes so that all writes are full-stripe writes. This design is only possible because ZFS integrates file system and device management in such a way that the file system's metadata has enough information about the underlying data replication model to handle variable-width RAID stripes.

A RAID-Z configuration with N disks of size X with P parity disks can hold approximately (N-P)*X bytes and can withstand one device failing before data integrity is compromised. You need at least two disks for a single-parity RAID-Z configuration and at least three disks for a double-parity RAID-Z configuration.

For example, if you have three disks in a single-parity RAID-Z configuration, parity data occupies space equal to one of the three disks. Otherwise, no special hardware is required to create a RAID-Z configuration.

Conceptually, a RAID-Z configuration with three disks looks like:

```
raidz c1t0d0 c2t0d0 c3t0d0
```

A more complex conceptual RAID-Z configuration looks like:

```
raidz c1t0d0 c2t0d0 c3t0d0 c4t0d0 c5t0d0 c6t0d0 c7t0d0
raidz c8t0d0 c9t0d0 c10t0d0 c11t0d0 c12t0d0 c13t0d0
c14t0d0
```

If you are creating a RAID-Z configuration with many disks, as in above example, a RAID-Z configuration with 14 disks is better split into a two 7-disk groupings. RAID-Z configurations with single-digit groupings of disks should perform better.

Each `raidz` argument defines a separate top-level virtual device. The first example defines a single RAID-Z device that consists of three disks. The second example defines two RAID-Z devices, where each device consists of seven disks.

Self-Healing Data in a Replicated Configuration

ZFS file enables self-healing data in a mirrored or RAID-Z configuration. When a bad data block is detected, ZFS fetches the correct data from another replicated copy, and repairs the bad data by replacing it with the good copy.

Dynamic Striping in a Storage Pool

For each virtual device that is added to a pool, ZFS file dynamically stripes data across all available devices. The decision about where to place data is done at write time, so no fixed width stripes are created at allocation time.

When virtual devices are added to a pool, ZFS gradually allocates data to the new device in order to maintain performance and space allocation policies. Each virtual device can also be a mirror or a RAID-Z device that contains other disk devices or files. This configuration allows for flexibility in controlling the fault characteristics of your pool. For example, you could create the following configurations out of 4 disks:

- Four disks using dynamic striping
- One four-way RAID-Z configuration
- Two two-way mirrors using dynamic striping

The ZFS file system supports combining different virtual devices within the same pool, but this isn't recommended. For example, you can create a pool with a two-way mirror and a three-way RAID-Z configuration. But, your fault tolerance will be as good as your worst virtual device, RAID-Z in this case. Sun recommends that you use top-level virtual devices of the same type with the same replication level in each device.

Creating and Destroying ZFS Storage Pools

Creating and destroying pools is fast and easy. However, be cautious when doing so. Although checks are performed to prevent using devices known to be in use in a new pool, the ZFS file system cannot always know when a device is already in use. Also, use `zpool destroy` with caution. Using this command can have significant consequences.

Creating a ZFS Storage Pool

Use the `zpool create` command to create a storage pool. This command takes a pool name and any number of virtual devices as arguments. The pool name must conform to the ZFS component naming conventions as described in the *Solaris ZFS Administration Guide*.

Caution – Do not add a disk that is currently configured as a Sun Cluster quorum device to a ZFS storage pool. After the disk is added to a storage pool, then the disk can be configured as a quorum device.

Creating a Basic Storage Pool

The following command creates a new pool named tank that consists of the disks `c1t0d0` and `c1t1d0`:

```
# zpool create tank c1t0d0 c1t1d0
```

These whole disks are found in the `/dev/dsk` directory and are labelled appropriately by ZFS to contain a single, large slice. Data is dynamically striped across both disks.

Creating a Mirrored Storage Pool

To create a mirrored pool, use the `mirror` keyword, followed by any number of storage devices that will comprise the mirror. Multiple mirrors can be specified by repeating the `mirror` keyword on the command line. The following command creates a pool with two, two-way mirrors:

```
# zpool create tank mirror c1d0 c2d0 mirror c3d0 c4d0
```

The second `mirror` keyword indicates that a new top-level virtual device is being specified. Data is dynamically striped across both mirrors, with data being replicated between each disk appropriately.

Creating a Single-Parity RAID-Z Storage Pool

Creating a single-parity RAID-Z pool is identical to creating a mirrored pool, except that the `raidz` keyword is used instead of `mirror`. The following example shows how to create a pool with a single RAID-Z device that consists of five disks:

```
# zpool create tank raidz c1t0d0 c2t0d0 c3t0d0 c4t0d0 /dev/dsk/c5t0d0
```

This example demonstrates that disks can be specified by using their full paths. The `/dev/dsk/c5t0d0` device is identical to the `c5t0d0` device.

A similar configuration could be created with disk slices. For example:

```
# zpool create tank raidz c1t0d0s0 c2t0d0s0 c3t0d0s0 c4t0d0s0 c5t0d0s0
```

However, the disks must be preformatted to have an appropriately sized slice zero.

Destroying ZFS Storage Pools

Use the Pools `zpool destroy` command to destroy a pool. This command destroys the pool even if it contains mounted datasets.

```
# zpool destroy tank
```

Caution – Be careful when you destroy a pool. Make sure you are destroying the right pool and you have copies of your data. If you accidentally destroy the wrong pool, you can attempt to recover the pool.

Querying ZFS Storage Pool Status

Use the `zpool` command to request pool status. This section describes how to display basic use information and pool `HEALTH` status.

Getting Basic ZFS Storage Pool Information

Use the `zpool list` command to display basic pool information. With no arguments, the `zpool list` command displays all the fields for all pools on the system. Specify the pool name to gather statistics for that pool. For example:

```
# zpool list
NAME                    SIZE    USED   AVAIL    CAP  HEALTH     ALTROOT
tank                   80.0G   22.3G   47.7G    28%  ONLINE     -
dozer                   1.2T    384G    816G    32%  ONLINE     -
# zpool list tank
NAME                    SIZE    USED   AVAIL    CAP  HEALTH     ALTROOT
tank                   80.0G   22.3G   47.7G    28%  ONLINE
```

This output displays the following information:

`NAME`	Pool name
`SIZE`	Pool size (equal to the sum of the size of all top-level virtual devices)
`USED`	Space allocated by all datasets and internal metadata
`AVAILABLE`	Unallocated space in the pool.
`CAPACITY (CAP)`	Space used (expressed as a percentage of total space)
`HEALTH`	Current pool health status
`ALTROOT`	Alternate root of the pool, if any.

Double Parity RAID-Z

With Solaris 10 11/06, a redundant RAID-Z configuration can have either single- or double-parity, which means that one or two device failures can be sustained respectively, without any data loss. Specify the `raidz2` keyword for a double-parity RAID-Z configuration. Specify the `raidz` or `raidz1` keyword for a single-parity RAID-Z configuration. For more information, see the `zpool(1M)` man page.

ZFS Storage Pool `HEALTH` Status

ZFS provides an integrated method of examining pool and device `HEALTH` status. Use the `zpool status` command to display pool `HEALTH` status. In addition to the `zpool status` command, the fault manager daemon (`fmd`) displays potential pool and device failures on the system console and in the `/var/adm/messages` file. A device can be in one of the following states:

`ONLINE`	Some transient errors may still occur, the virtual device is working normally.
`DEGRADED`	The virtual device experienced a failure but still functions. This state is most common when a mirror or RAID-Z device has lost one or more constituent devices. The fault tolerance of the pool might be compromised, as a subsequent fault in another device might be unrecoverable.
`FAULTED`	The virtual device is completely inaccessible. This status typically indicates total failure of the device, such that the ZFS file system is incapable of sending or receiving data from it. If a top-level virtual device is in this state, then the pool is completely inaccessible.
`OFFLINE`	The virtual device has been explicitly taken offline by the administrator.
`UNAVAILABLE`	The device or virtual device cannot be opened. In some cases, pools with `UNAVAILABLE` devices appear in `DEGRADED` mode. If a top-level virtual device is unavailable, then nothing in the pool can be accessed.

Determine the HEALTH of a pool from the status of all its top-level virtual devices. If all the virtual devices are ONLINE, then the pool is also ONLINE. If any of the virtual devices is DEGRADED or UNAVAILABLE, then the pool is also DEGRADED. If a top-level virtual device is FAULTED or OFFLINE, then the pool is also FAULTED. A FAULTED pool is completely inaccessible.

Getting Basic Storage Pool HEALTH Status

To request a quick overview of pool HEALTH, use the `zpool status` command:

```
# zpool status -x
all pools are healthy
```

Specify a pool name to examine a specific pool. Investigate potential problems in any pool that isn't in the ONLINE state.

Getting Detailed Storage Pool HEALTH Status

Use the `-v` option to request a detailed HEALTH summary. The following output displays a complete description of why the pool is in its current state. It includes a readable description of the problem and a link to a knowledge article for more information. Each knowledge article provides up-to-date information on the best way to recover from your current problem. Using the detailed configuration information, you should be able to determine which device is damaged and how to repair the pool.

```
# zpool status -v tank
  pool: tank
 state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas
exist for the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: none requested
config:

        NAME                STATE     READ WRITE CKSUM
        tank                DEGRADED     0     0     0
          mirror            DEGRADED     0     0     0
            c1t0d0          FAULTED      0     0     0  cannot open
            c1t1d0          ONLINE       0     0     0
errors: No known data errors
```

ZFS Log Device Recovery

In the Solaris 10 10/09 release, ZFS identifies intent-log failures in the `zpool status` command. ZFS describes how to recover from an intent-log failure.

For example, if the system shuts down abruptly before synchronous write operations are committed to a pool with a separate log device, you will see intent-log related error messages in the `zpool status` output.

Note – For information about resolving log device failures, see the *Solaris ZFS Administration Guide*.

Creating and Destroying ZFS File Systems

ZFS is a lightweight POSIX file system that is built on top of a storage pool. You can dynamically create and destroy file systems without allocating or formatting underlying space. Because file systems are lightweight and the central point of administration in ZFS, you will probably create many.

Use the `zfs` commands to administer ZFS. The `zfs` command includes subcommands that perform specific operations on file systems. Use the `zfs create` and `zfs destroy` command to create and destroy file systems.

Note – The term *dataset* is used in this module as a generic term to refer to a filesystem, snapshot, clone, or volume.

Creating a ZFS File System

Use the `zfs create` command to create a ZFS file system. The `create` subcommand takes a single argument: the name of the file system to create. The file system name is specified as a path name starting from the name of the pool:

`pool-name/[filesystem-name/]filesystem-name`

The pool name and initial file system names in the path identify the location in the hierarchy where the new file system will be created. All the intermediate file system names must already exist in the pool. The last name in the path identifies the name of the file system to be created.

In the following example, a file system named `bonwick` is created in the `tank/home` file system.

```
# zfs create tank/home/bonwick
```

The ZFS file system automatically mounts the newly created file system if it is created successfully. By default, file systems are mounted as */`dataset`*, using the path provided for the file system name in the `create` subcommand. In this example, the newly created `bonwick` file system is at `/tank/home/bonwick`.

Destroying a ZFS File System

Use the `zfs destroy` command to destroy a ZFS file system. The destroyed file system is automatically unmounted and unshared.

In the following example, the `tabriz` file system is destroyed.

```
# zfs destroy tank/home/tabriz
```

Caution – No confirmation prompt appears with the `destroy` subcommand. Use it with extreme caution.

If the file system to be destroyed is busy and therefore can't be unmounted, the `zfs destroy` command fails. To destroy an active file system, use the `-f` option. Use this option with caution as it can unmount, unshare, and destroy active file systems, causing unexpected application behavior.

```
# zfs destroy tank/home/ahrens
cannot unmount 'tank/home/ahrens': Device busy
# zfs destroy -f tank/home/ahrens
```

The `zfs destroy` command also fails if a file system has children. To recursively destroy a file system and all its descendants, use the `-r` option. Note that a recursive destroy also destroys snapshots so use this option with caution.

```
# zfs destroy tank/ws
cannot destroy 'tank/ws': filesystem has children
use '-r' to destroy the following datasets:
tank/ws/billm
tank/ws/bonwick
tank/ws/maybee

# zfs destroy -r tank/ws
```

If the file system to be destroyed has indirect dependents, even the recursive destroy command described above fails. To force the destruction of all dependents, including cloned file systems outside the target hierarchy, the `-R` option must be used. Use extreme caution with this option.

```
# zfs destroy -r tank/home/schrock
cannot destroy 'tank/home/schrock': filesystem has
dependent clones
use '-R' to destroy the following datasets:
tank/clones/schrock-clone

# zfs destroy -R tank/home/schrock
```

Renaming a ZFS File System

Use `zfs rename` to rename a ZFS file system. Using the `rename` subcommand, you can perform the following operations:

- Change the name of a file system
- Relocate the file system to a new location within the ZFS hierarchy
- Change the file system name and relocate it with the ZFS hierarchy

The following example uses the `rename` subcommand to rename a file system. This example renames the `kustarz` file system to `kustarz_old`.

```
# zfs rename tank/home/kustarz tank/home/kustarz_old
```

The following example shows how to use `zfs rename` to relocate a file system. In this example, the `maybee` file system is relocated from `tank/home` to `tank/ws`. When you relocate a file system through `rename`, the new location must be within the same pool and it must have enough space to hold this new file system. If the new location does not have enough space, possibly because it has reached its quota, the rename will fail.

```
# zfs rename tank/home/maybee tank/ws/maybee
```

The rename operation attempts an unmount/remount sequence for the file system and any descendant file systems. The rename fails if the operation is unable to unmount an active file system. If this problem occurs, you will need to force unmount the file system.

ZFS Properties

Properties are the main mechanism that you use to control the behavior of file systems, volumes, snapshots, and clones. Unless stated otherwise, the properties defined in the section apply to all the dataset types.

Properties are either read-only statistics or settable properties. Most settable properties are also inheritable. An inheritable property is a property that, when set on a parent, is propagated down to all of its descendants.

All inheritable properties have an associated source. The source indicates how a property was obtained. A property source can have the following values:

`default`	A `default` value means that the property setting was not inherited or set locally. This source is a result of no ancestor having the property as source `local`.
`local`	A `local` source indicates that the property was explicitly set on the dataset by using the `zfs set` command.
`inherited from` *`dataset-name`*	A value of `inherited from` *`dataset-name`* means that the property was inherited from the named ancestor.
`temporary`	A `temporary` value means that the property value was set by using the `zfs mount -o` option and is valid for the mount lifetime.
`- (none)`	Read-only property. Its value is generated by ZFS.

Read-Only ZFS Properties

You can retrieve but not set read-only properties. Read-only properties are not inherited. Some properties are specific to a particular type of dataset. In such cases, the particular dataset type is mentioned in the description in the previous table. ZFS read-only properties are:

- `available`
- `compressratio`
- `creation`

- mounted
- origin
- referenced
- type
- used
- usedbychildren
- usedbydataset
- usedbyrefreservation
- usedbysnapshots

Settable ZFS Properties

You can retrieve and set "settable" properties. Use zfs set to set properties. With the exception of quotas and reservations, settable properties are inherited.

Some settable properties are specific to a dataset type. In such cases, the dataset type is mentioned in the description in the previous table. If not specifically mentioned, a property applies to all dataset types: file systems, volumes, clones, and snapshots.

The following table explains settable properties. For more information see the zfs(1M) man page and the *Solaris ZFS System Administration Guide*.

Table 6–1 ZFS Native Property Descriptions

Property Name	Type	Default Value	Description
aclinherit	String	secure	Controls how ACL entries are inherited when files and directories are created. The values are discard, noallow, secure, and passthrough. For a description of these values, see ACL Property Modes.
aclmode	String	groupmask	Controls how an ACL entry is modified during a chmod operation. The values are discard, groupmask, and passthrough. For a description of these values, see ACL Property Modes.

Property Name	Type	Default Value	Description
`atime`	Boolean	on	Controls whether the access time for files is updated when they are read. Turning this property off avoids producing write traffic when reading files and can result in significant performance gains, though it might confuse mailers and other similar utilities.
`canmount`	Boolean	on	Controls whether the given file system can be mounted with the `zfs mount` command. This property can be set on any file system and the property isn't inheritable. When this property is set to off, a mountpoint can be inherited to descendent file systems, but the file system is never mounted. When the `noauto` option is set, a dataset can only be mounted and unmounted explicitly. The dataset is not mounted automatically when the dataset is created or imported, nor is it mounted by the `zfs mount-a` command or unmounted by the `zfs unmount-a` command.
`checksum`	String	on	Controls the checksum used to verify data integrity. The default value is `on`, which automatically selects an appropriate algorithm, currently `fletcher2`. The values are `on`, `off`, `fletcher2`, `fletcher4`, and `sha256`. A value of `off` disables integrity checking on user data. A value of `off` is not recommended.
`compression`	String	off	Enables or disables compression for this dataset. The values are `on`, `off`, and `lzjb`, `gzip`, or `gzip-N`. Currently, setting this property to `lzjb`, `gzip`, or `gzip-N` has the same effect as setting this property to `on`. The default value is `off`. Enabling compression on a file system with existing data only compresses new data. Existing data remains uncompressed. This property can also be referred to by its shortened column name, `compress`.

Property Name	Type	Default Value	Description
copies	Number	1	Sets the number of copies of user data per file system. Available values are 1, 2 or 3. These copies are in addition to any pool-level redundancy. Space used by multiple copies of user data is charged to the corresponding file and dataset and counts against quotas and reservations. In addition, the used property is updated when multiple copies are enabled. Consider setting this property when the file system is created because changing this property on an existing file system only affects newly written data.
devices	Boolean	on	Controls the ability to open device files in the file system.
exec	Boolean	on	Controls whether programs within this file system are allowed to be executed. Also, when set to off, mmap(2) calls with PROT_EXEC are disallowed.
groupquota	Number	none	Sets a quota on the amount of space consumed by files that are owned by a particular group.
mountpoint	String	NA	Controls the mount point used for this file system. When the mountpoint property is changed for a file system, the file system and any children that inherit the mount point are unmounted. If the new value is legacy, then they remain unmounted. Otherwise, they are automatically remounted in the new location if the property was previously legacy or none, or if they were mounted before the property was changed. In addition, any shared file systems are unshared and shared in the new location.

Property Name	Type	Default Value	Description
primarycache	String	NA	Controls the primary cache (ARC) on an existing ZFS file system or when the file system is created. The default, all caches both user data and metadata. The other values are none where neither user data or metadata is cached, or metadata where only metadata is cached.
quota	Number (or none)	none	Limits the amount of space a dataset and its descendents can consume. This property enforces a hard limit on the amount of space used, including all space consumed by descendents, including file systems and snapshots. Setting a quota on a descendent of a dataset that already has a quota does not override the ancestor's quota, but rather imposes an additional limit. Quotas cannot be set on volumes, as the volsize property acts as an implicit quota.
readonly	Boolean	off	Controls whether this dataset can be modified. When set to on, no modifications can be made to the dataset. This property can also be referred to by its shortened column name, rdonly.
recordsize	Number	128K	Specifies a suggested block size for files in the file system. This property can also be referred to by its shortened column name, recsize.
refquota	Number (or none)	none	Sets the amount of space that a dataset can consume. This property enforces a hard limit on the amount of space used. This hard limit does not include space used by descendents, such as snapshots and clones.

Property Name	Type	Default Value	Description
`refreservation`	Number (or none)	none	Sets the minimum amount of space that is guaranteed to a dataset, not including descendents, such as snapshots and clones. When the amount of space that is used is below this value, the dataset is treated as if it were taking up the amount of space specified by refreservation. The refreservation reservation is accounted for in the parent datasets' space used, and counts against the parent datasets' quotas and reservations. If refreservation is set, a snapshot is only allowed if enough free pool space is available outside of this reservation to accommodate the current number of referenced bytes in the dataset. This property can be referred to by its shortened column name, `refreserv`.
`reservation`	Number (or none)	none	The minimum amount of space guaranteed to a dataset and its descendents. When the amount of space used is below this value, the dataset is treated as if it were using the amount of space specified by its reservation. Reservations are accounted for in the parent datasets' space used, and count against the parent datasets' quotas and reservations. This property can also be referred to by its shortened column name, reserv.
`secondarycache`	String	NA	Controls the secondary cache (L2ARC) on an existing ZFS file system or when the file system is created. The default, `all` caches both user data and metadata. The other values are `none` where neither user data or metadata is cached, or `metadata` where only metadata is cached.
`setuid`	Boolean	on	Controls whether the `setuid` bit is honored in the file system.

Property Name	Type	Default Value	Description
`sharenfs`	String	off	Controls whether the file system is available over NFS, and what options are used. If set to on, the `zfs share` command is invoked with no options. Otherwise, the `zfs share` command is invoked with options equivalent to the contents of this property. If set to `off`, the file system is managed by using the legacy share and unshare commands and the `dfstab` file.
`snapdir`	String	hidden	Controls whether the `.zfs` directory is hidden or visible in the root of the file system.
`userquota`	Number	none	Sets a quota on the amount of space consumed by files that are owned by a particular user.
`volsize`	Number	NA	For volumes, specifies the logical size of the volume.
`volblocksize`	Number	8 Kbytes	For volumes, specifies the block size of the volume. The block size cannot be changed once the volume has been written, so set the block size at volume creation time. The default block size for volumes is 8 Kbytes. Any power of 2 from 512 bytes to 128 Kbytes is valid. This property can also be referred to by its shortened column name, volblock.
`zoned`	Boolean	NA	Indicates whether this dataset has been added to a non-global zone. If this property is set, then the mount point is not honored in the global zone, and ZFS cannot mount such a file system when requested. When a zone is first installed, this property is set for any added file systems.

ZFS Property Enhancements

The Solaris 10 10/09 release provides several new ZFS property enhancements for the read-only ZFS properties and settable properties.

Using Space (Usage) Accounting Properties

In the Solaris 10 10/09 release you can use the space accounting properties to identify space usage for clones, file systems, and volumes, but not snapshots. The space accounting properties are as follows:

- `usedbychildren` – Identifies the amount of space that is used by children of this dataset, which would be freed if all the dataset's children were destroyed. The property abbreviation is `usedchild`.
- `usedbydataset` – Identifies the amount of space that is used by this dataset itself, which would be freed if the dataset was destroyed, after first destroying any snapshots and removing any refreservation. The property abbreviation is `usedds`.
- `usedbyrefreservation` – Identifies the amount of space that is used by a refreservation set on this dataset, which would be freed if the refreservation was removed. The property abbreviation is `usedrefreserv`.
- `usedbysnapshots` – Identifies the amount of space that is consumed by snapshots of this dataset. In particular, it is the amount of space that would be freed if all of this dataset's snapshots were destroyed. Note that this is not simply the sum of the snapshots' used properties, because space can be shared by multiple snapshots. The property abbreviation is `usedsnap`.

These new properties break down the value of the used property into the various elements that consume space. In particular, the value of the used property breaks down as follows:

```
used property = usedbychildren + usedbydataset +
usedbyrefreservation + usedbysnapshots
```

You can view these properties by using the `zfs list -o` space command. For example:

```
# zfs list -o space
NAME            AVAIL   USED  USEDSNAP   USEDDS  USEDREFRESERV  USEDCHILD
pool            33.2G   7 2K         0      21K              0        51K
rpool           27.0G  6.27G     20.5K      97K              0      6.27G
rpool/ROOT      27.0G  4.73G         0      21K              0      4.73G
```

```
rpool/ROOT/zfsBE  27.0G  4.73G     97.5M   4.63G           0         0
rpool/dump        27.0G  1.00G       16K   1.00G           0         0
rpool/export      27.0G    60K       16K     23K           0       21K
rpool/export/home 27.0G    21K         0     21K           0         0
rpool/swap        27.5G   553M         0   41.5M        512M         0
```

Setting ZFS File System Properties at Pool Creation Time

You can set ZFS file system properties at pool creation time. In the following example, compression is enabled on the ZFS file system that is created when the pool is created.

```
# zpool create -O compression=on pool mirror c0t1d0 c0t2d0
```

Setting Cache Properties on a ZFS File System

There are two new ZFS file system properties that control the primary cache and secondary cache.

- `primarycache`
- `secondarycache`

The values for setting cache properties are as follows:

- `all` – default both user data and metadata are cached
- `none` – neither user data or metadata is cached
- `metadata` – only metadata is cached

You can set these properties on an existing ZFS file system, or when the file system is created:

 # zfs set primarycache=metadata tank/datab

 # zfs create -o primarycache=metadata tank/newdatab

Some database environments might benefit from not caching user data. You will have to determine if setting cache properties is appropriate for your environment.

Setting ZFS User and Group Quotas

In previous Solaris OS releases, you could apply quotas and reservations to ZFS file systems to manage and reserve space. In the Solaris 10 10/09 release, you can now set a quota on the amount of space consumed by files that are owned by a particular user or group.

You might consider setting user and group quotas in any environment that has a large number of users or groups.

You can set user or group quotas by using the `zfs userspace` and `zfs groupspace` properties as follows:

```
# zfs set userquota@user1=5G tank/data

# zfs set groupquota@staff=10G tank/staff/admins
```

You can display a user's or group's current quota setting as follows:

```
# zfs get userquota@user1 tank/data
NAME         PROPERTY         VALUE       SOURCE
tank/data    userquota@user1  5G          local

# zfs get groupquota@staff tank/staff/admins
NAME              PROPERTY           VALUE      SOURCE
tank/staff/admins groupquota@staff   10G        local
```

To display general quota information, type the following:

```
# zfs userspace tank/data
TYPE             NAME        USED    QUOTA
POSIX User       root          3K     none
POSIX User       user1          0       5G

# zfs groupspace tank/staff/admins
TYPE             NAME        USED    QUOTA
POSIX Group      root         3K      none
POSIX Group      staff          0      10G
```

Querying ZFS Information

The `zfs list` command provides an extensible mechanism for viewing and querying dataset information. Both basic and complex queries are explained in this section.

Listing Basic ZFS Information

List basic dataset information by using the `zfs list` command with no options. This command displays system dataset names including their `used`, `available`, `referenced`, and `mountpoint` properties. For example:

```
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
pool                     84.0K  33.5G      -  /pool
pool/clone                   0  33.5G  8.50K  /pool/clone
pool/test                   8K  33.5G     8K  /test
pool/home                17.5K  33.5G  9.00K  /pool/home
pool/home/marks          8.50K  33.5G  8.50K  /pool/home/marks
pool/home/marks@snap         0      -  8.50K  /pool/home/marks@snap
```

Use `zfs list` to display specific datasets by providing the dataset name on the command line. Use the `-r` option to display dataset descendants. The following example shows how to display `tank/home/chua` and its descendant datasets. For additional information about the `zfs list` command, see `zfs(1M)`.

```
# zfs list -r tank/home/chua
tank/home/chua                26.0K  4.81G  10.0K  /tank/home/chua
tank/home/chua/projects         16K  4.81G   9.0K  /tank/home/chua/projects
tank/home/chua/projects/fs1     8K  4.81G     8K
/tank/home/chua/projects/fs1
tank/home/chua/projects/fs2     8K  4.81G     8K
/tank/home/chua/projects/fs2
```

ZFS Snapshots Omitted from `zfs list` Output

In the Solaris 10 10/09 release, snapshots are omitted from the `zfs list` output. The `listsnaps` pool property controls whether snapshot information is displayed by the `zfs list` command. If you use the `zfs list -t` snapshots command, snapshot information is displayed. The default value is `off`, which means snapshot information is not displayed by default.

Sending and Receiving ZFS Data

The `zfs send` command creates a stream representation of a snapshot that is written to standard output. By default, a full stream is generated. You can redirect the output to a file or to a different system. The `zfs receive` command creates a snapshot whose contents are specified in the stream that is provided on standard input. If a full stream is received, a new file system is created. You can send ZFS snapshot data and receive ZFS snapshot data and file systems with these commands. For more information on sending and receiving ZFS data, see the *Solaris ZFS System Administration Guide*.

Creating and Destroying ZFS Snapshots

Use the `zfs snapshot` command to create snapshots. Enter the snapshot name for the `zfs snapshot` command argument as follows. You must follow snapshot naming conventions. See the Solaris *ZFS System Administration Guide* to find the naming conventions.

```
filesystem@snapname
volume@snapname
```

In the following, a snapshot of `tank/home/ahrens` named `friday` is created.

```
# zfs snapshot tank/home/ahrens@friday
```

Use the `-r option` to create snapshots for all descendent file systems.

```
# zfs snapshot -r tank/home@now
# zfs list -t snapshot
NAME                     USED  AVAIL  REFER  MOUNTPOINT
tank/home@now               0      -  29.5K  -
tank/home/ahrens@now        0      -  2.15M  -
tank/home/anne@now          0      -  1.89M  -
tank/home/bob@now           0      -  1.89M  -
tank/home/cindys@now        0      -  2.15M  -
```

Snapshots have no modifiable properties. Dataset properties can't be applied to a snapshot.

```
# zfs set compression=on tank/home/ahrens@tuesday
cannot set compression property for 'tank/home/ahrens@tuesday':
snapshotproperties cannot be modified
```

Use `zfs destroy` to destroy snapshots.

```
# zfs destroy tank/home/ahrens@friday
```

A dataset can't be destroyed if snapshots of the dataset exist (see example below). If clones were created from a snapshot, you must destroy them before you can destroy the snapshot.

```
# zfs destroy tank/home/ahrens
cannot destroy'tank/home/ahrens':filesystem has children
use '-r' to destroy the following datasets:
tank/home/ahrens@tuesday
tank/home/ahrens@wednesday
tank/home/ahrens@thursday
```

Exercise: ZFS File System Introduction

In this exercise, you complete the following tasks:

- Create and destroy ZFS storage pools
- Create, rename, and destroy ZFS file systems
- Work with ZFS properties

Preparation

This exercise requires a minimum of one unused spare disk. Tasks 1 and 2 provide steps to create storage pools that consist of 1, 2, or 3 disks, if they are available. You must evaluate the list of disk devices on your system to determine the correct choices to make in tasks 1 and 2. The path names of your disk devices might differ from those described in these exercises. Use the device path names appropriate to your system.

Note – If only one spare disk is available on your system, this exercise asks you to use a single disk to create a storage pool. This is not a recommended configuration. Single-disk pools offer no data redundancy. Failure of the disk supporting the pool causes the pool and its data to become inaccessible.

Exercise Summary

This exercise includes tasks that illustrate the main ZFS features and cover the following concepts:

- Working with ZFS storage pools - creating and destroying a basic ZFS storage pool using spare disks on your system.
- Creating, renaming, and destroying ZFS file systems - creating a basic storage pool, and manipulating a ZFS file within that pool.
- Working with ZFS properties - setting and listing ZFS properties, principles of ZFS property inheritance.
- Working with ZFS snapshots and clones - creating and destroying ZFS snapshots and clones, and dependencies that exist among them.
- Removing the ZFS configuration - removing the objects you create during the exercise.

Task 1 – Creating and Destroying ZFS Storage Pools

Complete the following steps:

1. Use the `zpool` command to display the list of ZFS pools. Verify that no ZFS pool currently exists.
2. Use the `zfs` command to display the list of ZFS file systems. Verify that no ZFS file system currently exists.
3. Use the `df` command to display the list of UFS file systems currently in use on your system. Use this list to determine which disk or disks on your system are currently in use.

Note – The `df` command with the `-F` option allows you to list file systems of a specific type.

---

4. Use the `format` command to list what disks are attached to your system. Compare this list with the output of step 3 to determine what disks are currently not in use on your system. Use `<Ctrl-D>` to exit your `format` session.

Note – You may find that your system uses different sized disks. For this exercise, although it is not the best practice, use the `zpool` command with the `-f` option to override errors that this problem causes.

Note – When you try to create a pool, your system will probably display errors that indicate UFSs exist on your spare disk. If it does, verify that the spare disk is not in use, then use the `zpool` command with the `-f` option to override the errors and create the pool.

---

5. If you have three unused disks available, use the `zpool` command to create a storage pool called `firstpool` that consists of one, three-disk RAID-Z virtual device.
6. If you have only two unused disks available, use the `zpool` command to create a storage pool called `firstpool` that consists of one mirror virtual device.
7. If you have only one unused disk available, use the `zpool` command to create a storage pool called `firstpool` that uses the single disk as a non-redundant, top-level virtual device.
8. Use the `zpool` command to display the list of ZFS pools.

9. Use the `zpool` command to display the basic health status of your pool.
10. Use the `zpool` command to display the detailed health status of your pool. What kind of top-level virtual device does the `zpool` command report, and what devices does it consist of? What is the state of the top-level virtual device, and of the pool itself?
11. Use the `zfs` command to list ZFS file systems, and display the space utilization for `firstpool`. Does the space available match what you would expect for a pool of the type you created?
12. Use the `zpool` command to display the name of your pool, and only its space used and space available statistics.

Note – For a RAID-Z pool, the `zpool list` command reports the sum of all disk space in the `AVAIL` column. This differs from the value reported by `zfs list`. For a RAID-Z pool, the `zfs list` command reduces the value reported in the `AVAIL` column by the amount of parity data that the pool requires.

---

13. Use the `zfs` command to display the list of currently mounted ZFS file systems. Display a long listing for the `/firstpool` directory.
14. Use the `zpool` command to destroy `firstpool`. Verify that `firstpool` no longer exists.
15. Verify that the `firstpool` file system and the `/firstpool` mount point no longer exist.

Task 2 – Create, Rename, and Destroy ZFS File Systems

Complete the following steps:

1. Use the `zpool` command to display the list of ZFS pools. Verify that no ZFS pool currently exists.
2. If you have three unused disks available, use the `zpool` command to create a storage pool called `mypool` that consists of one, three-disk RAID-Z virtual device.
3. If you have only two unused disks available, use the `zpool` command to create a storage pool called `mypool` that consists of one mirror virtual device.

4. If you have only one unused disk available, use the `zpool` command to create a storage pool called `mypool` that uses the single disk as a non-redundant, top-level virtual device.

Note – Example command output throughout the remainder of this exercise assumes that a non-redundant pool is in use. This is not a recommended pool configuration.

---

5. Use the `zfs` command to list ZFS file systems, and display the space use for `/mypool`.
6. Attempt to create a file system named `mypool/fs1/fs2`. What happens and why?
7. Create a directory called `/mypool/fs1`. Again attempt to create a file system named `mypool/fs1/fs2`. What happens and why?

   The attempt fails because the intermediate file system `fs1` does not exist. ZFS only creates new file systems immediately below a parent file system.
8. Create a new empty file named `/mypool/fs1/file1`.
9. Attempt to create a new file system named `mypool/fs1`. Use the `zfs list` and `zfs mount` commands to list the ZFS file systems that currently exist and that are currently mounted. What happens, and why?
10. Destroy the `mypool/fs1` file system.
11. Remove `file1` from the `/mypool/fs1` directory.
12. Again attempt to create a new file system named `mypool/fs1`. What happens?
13. Verify that the new file system exists and that it is mounted.
14. Rename `mypool/fs1` to `mypool/fs2`.
15. Verify that `mypool/fs2` exists and that it is mounted.
16. List `/mypool`. Does the `fs1` directory exist?
17. Destroy the `mypool/fs2` file system.
18. Verify that `mypool/fs2` is not mounted, and that it no longer exists.
19. List `/mypool`. Does the `fs2` directory exist within it?

Task 3 – Working with ZFS Properties

1. Use the `zfs` command to create a new ZFS in `mypool` called `cmp`. List the space use for `/mypool/cmp`.
2. Use the `tar` command to create a partial archive of the `/usr/lib` directory, and save the archive as `/mypool/cmp/archive.tar`. Use the `-k` option to limit the size of this archive to 400 Mbytes. You will use this file to demonstrate how the `compression` property functions in ZFS file systems.

Note – The `tar` command will ask you to press `RETURN` when it has created an archive of the size (in kilobytes) you specified. Enter `Control-C` in response to this prompt. Doing so prevents `tar` from overwriting `archive.tar` with the next set of files that exceed the limit you specified.

---

3. Use the `zfs get` command to display all of the properties for the `mypool/cmp` file system. Do the properties use the same source? If so, which source, and why?
4. Use `ls` with the `-lh` options to list the size of `/mypool/cmp/archive.tar`. Note the size displayed.
5. Use the `zfs get` command to display the `mypool/cmp used`, `compression`, and `compressratio` property settings. Verify the following:
   - That the compression is `off`
   - That the compression ratio is `1.00x`
   - That the space used matches the `archive.tar` size.
6. Set the `compression` property for `mypool/cmp` to `on`.
7. Use the `zfs get` command to display the current settings of the `used`, `compression` and `compressratio` properties for `mypool/cmp`. Verify that compression is `on`. Have the compression ratio or space-used values changed? If not, why? What source is now listed for the `compression` property?
8. Change the directory to `/mypool/cmp`. Copy `archive.tar` to `archive1.tar`.
9. Use `ls -lh` and `du -h` to display the sizes of `archive.tar` and `archive1.tar`. What file sizes do these commands report?

Note – The size of the compressed archive and compression ratio values may vary, depending on the archive content.

10. Use the `zfs get` command to display the current settings of the `used`, `compression` and `compressratio` properties for `mypool/cmp`. How have the `compressratio` and `used` values changed from the last time you checked them, and why?
11. Set the `compression` property for `mypool/cmp` to `off`.
12. Copy `archive.tar` to `archive2.tar`.
13. Use `ls -lh` and `du -h` to display the sizes of `archive.tar` and `archive1.tar`. What file sizes do these commands report?
14. Use the `zfs get` command to display the current settings of the `used`, `compression` and `compressratio` properties for `mypool/cmp`. How have the `compressratio` and `used` values changed from the last time you checked them, and why? What source is listed for the `compression` property?
15. Use the `zfs inherit` command to clear the `compression` property for `mypool/cmp`. Verify that the change was made. What source is now listed for the `compression` property?
16. Create two new file systems named `mypool/cmp/a` and `mypool/cmp/a/b`.
17. Use the `zfs get` command to recursively display the `compression` property for `mypool/cmp`. Do all three file systems list the same value and source?
18. Use the `zfs set` command to set the `compression` property to `on` for `mypool/cmp`.
19. Use the `zfs get` command to recursively display the `compression` property for `mypool/cmp`. Do all three file systems list the same value and source?
20. Use the `zfs set` command to set the `compression` property to `off` for `mypool/cmp/a`.
21. Use the `zfs get` command to recursively display the `compression` property for `mypool/cmp`. What has changed in the reported values and sources?
22. Use the `zfs inherit` command to clear the `compression` property for `mypool/cmp`.

23. Use the `zfs get` command to recursively display the `compression` property for `mypool/cmp`. What has changed in the reported values and sources?
24. Change directory to root (/). Use the `zfs destroy` command to destroy `mypool/cmp` and its dependent file systems. Verify that the file systems no longer exist.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solution: ZFS File System Introduction

In this exercise, you complete the following tasks:

- Create and destroy ZFS storage pools
- Create, rename, and destroy ZFS file system
- Work with ZFS properties

Preparation

This exercise requires a minimum of one unused spare disk. Tasks 1 and 2 provide steps to create storage pools that consist of one, two, or three disks, if they are available. You must evaluate the list of disk devices on your system to determine the correct choices to make in tasks 1 and 2. The path names of your disk devices might differ from those described in these exercise. Use the device path names appropriate to your system.

Note – If only one spare disk is available on your system, this exercise asks you to use a single disk to create a storage pool. This is not a recommended configuration. Single-disk pools offer no data redundancy. Failure of the disk supporting the pool causes the pool and its data to become inaccessible.

Exercise Summary

This exercise includes tasks that illustrate the main features of the ZFS file system. The tasks in this exercise illustrate the following concepts:

- Working with ZFS storage pools - creating and destroying a basic ZFS storage pool using spare disks on your system.
- Creating, renaming, and destroying ZFS file systems - creating a basic storage pool, and manipulating ZFS file systems within that pool.
- Working with ZFS properties - setting and listing ZFS properties, and demonstrates principles of ZFS property inheritance.
- Working with ZFS snapshots and clones - creating and destroying ZFS snapshots and clones, and dependencies.
- Removing the ZFS Configuration - removing all of the objects you created during the exercise.

Task 1 – Creating and Destroying ZFS Storage Pools

Complete the following steps:

1. Use the `zpool` command to display the list of ZFS pools. Verify that no ZFS pool currently exists.

```
# zpool list
no pools available
#
```

2. Use the `zfs` command to display the list of ZFS file systems. Verify that no ZFS exists.

```
# zfs list
no datasets available
#
```

3. Use the `df` command to display the list of UFS file systems currently in use on your system. Use this list to determine which disk or disks on your system are currently in use.

Note – The `df` command with the `-F` option allows you to list file systems of a specific type.

```
# df -F ufs -h
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c1t0d0s0      9.9G   3.8G   6.0G    39%    /
/dev/dsk/c1t0d0s7      9.9G   1.9G   7.9G    20%    /export/home
#
```

4. Use the `format` command to list what disks are attached to your system. Compare this list with the output of step 3 to determine what disks are currently not in use on your system. Use `<Ctrl-D>` to exit your `format` session.

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


AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@1f,0/pci@1/scsi@8/sd@0,0
       1. c1t1d0 <FUJITSU-MAP3367N SUN36G-0401-33.92GB>
          /pci@1f,0/pci@1/scsi@8/sd@1,0
       2. c2t0d0 <SEAGATE-ST373307LSUN72G-0507-68.37GB>
          /pci@1f,0/pci@1/scsi@8,1/sd@0,0
       3. c2t1d0 <SEAGATE-ST373307LSUN72G-0507-68.37GB>
          /pci@1f,0/pci@1/scsi@8,1/sd@1,0
       4. c2t2d0 <SEAGATE-ST373307LSUN72G-0507-68.37GB>
          /pci@1f,0/pci@1/scsi@8,1/sd@2,0
Specify disk (enter its number): ^D
#
```

Note – Your system may use different sized disks. For this exercise, it's not the best practice. Use the `zpool` command with the `-f` option to override errors that this problem causes.

Note – When you try to create a pool, your system will probably display errors that indicate UFS file systems exist on your spare disk. If it does, verify that the spare disk is not in use, then use the `zpool` command with the `-f` option to override the errors and create the pool.

---

5. If you have three unused disks available, use the `zpool` command to create a storage pool called `firstpool` that consists of one, three-disk RAID-Z virtual device.

   `# zpool create firstpool raidz c2t0d0 c2t1d0 c2t2d0`

6. If you have only two unused disks available, use the `zpool` command to create a storage pool called `firstpool` that consists of one mirror virtual device.

   `# zpool create firstpool mirror c2t0d0 c2t1d0`

7. If you have only one unused disk available, use the `zpool` command to create a storage pool called `firstpool` that uses the single disk as a non-redundant, top-level virtual device.

```
# zpool create firstpool c2t0d0
```

8. Use the `zpool` command to display the list of ZFS pools.

RAID-Z pool result:

```
# zpool list
NAME                    SIZE    USED   AVAIL    CAP  HEALTH     ALTROOT
firstpool               204G    141K    204G     0%  ONLINE     -
#
```

Mirrored pool result:

```
# zpool list
NAME                    SIZE    USED   AVAIL    CAP  HEALTH     ALTROOT
firstpool                68G     80K   68.0G     0%  ONLINE     -
#
```

Non-redundant pool result:

```
# zpool list
NAME                    SIZE    USED   AVAIL    CAP  HEALTH     ALTROOT
firstpool                68G     79K   68.0G     0%  ONLINE     -
#
```

9. Use the `zpool` command to display the basic health status of your pool.

```
# zpool status -x
all pools are healthy
#
```

10. Use the `zpool` command to display the detailed health status of your pool. What kind of top-level virtual device does the `zpool` command report, and what devices does it consist of? What is the state of the top-level virtual device, and of the pool itself?

RAID-Z pool result:

```
# zpool status -v
  pool: firstpool
 state: ONLINE
 scrub: none requested
config:

        NAME         STATE     READ WRITE CKSUM
        firstpool    ONLINE       0     0     0
          raidz1     ONLINE       0     0     0
            c2t0d0   ONLINE       0     0     0
            c2t1d0   ONLINE       0     0     0
```

```
            c2t2d0   ONLINE       0     0     0

errors: No known data errors
#
```

Mirrored pool result:

```
# zpool status -v
  pool: firstpool
 state: ONLINE
 scrub: none requested
config:

        NAME         STATE     READ WRITE CKSUM
        firstpool    ONLINE       0     0     0
          mirror     ONLINE       0     0     0
            c2t0d0   ONLINE       0     0     0
            c2t1d0   ONLINE       0     0     0

errors: No known data errors
#
```

Non-redundant pool result:

```
# zpool status -v
  pool: firstpool
 state: ONLINE
 scrub: none requested
config:
NAME         STATE     READ WRITE CKSUM
        firstpool    ONLINE       0     0     0
          c2t0d0     ONLINE       0     0     0

errors: No known data errors
#
```

The pool `firstpool` should contain one top-level virtual device that matches the type you specified when you created the pool; a `raidz1`, a `mirror`, or in the non-redundant case, the single disk you specify serves as the only top-level device. The `raidz1` device should consist of three disks. The `mirror` device should consist of two disks. Both the top-level virtual device and the pool should be in the `ONLINE` state.

11. Use the `zfs` command to list ZFS files, and display the space utilization for `firstpool`. Does the space available match what you would expect for a pool of the type you created?

RAID-Z pool result:

```
# zfs list
NAME                       USED  AVAIL  REFER  MOUNTPOINT
```

```
firstpool              89.9K   134G  32.6K  /firstpool
#
```

Mirrored pool result:

```
# zfs list
NAME                    USED  AVAIL  REFER  MOUNTPOINT
firstpool                77K  66.9G  24.5K  /firstpool
#
```

Non-redundant pool result:

```
# zfs list
NAME                    USED  AVAIL  REFER  MOUNTPOINT
firstpool                76K  66.9G  24.5K  /firstpool
#
```

The space available should be roughly equal to twice the size of the smallest disk, if you created a pool that consists of one, three-disk RAID-Z device. This accounts for the space required to store parity data in this particular pool. The space available should roughly match the size of the smallest disk, if you created a pool with a single, two-disk mirror. The space available should roughly match the size of the disk you specified, if you created a non-redundant pool.

12. Use the `zpool` command to display the name of your pool, and only its space used and space available statistics.

RAID-Z pool result:

```
# zpool list -o name,used,available
NAME                     USED  AVAIL
firstpool                141K  204G
#
```

Mirrored pool result:

```
# zpool list -o name,used,available
NAME                     USED  AVAIL
firstpool                 80K  68.0G
#
```

Non-redundant pool result:

```
# zpool list -o name,used,available
NAME                     USED  AVAIL
firstpool                 79K  68.0G
#
```

Note – For a RAID-Z pool, the `zpool list` command reports the sum of all disk space in the `AVAIL` column. This differs from the value reported by `zfs list`. For a RAID-Z pool, the `zfs list` command reduces the value reported in the `AVAIL` column by the amount of parity data that the pool requires.

---

13. Use the `zfs` command to display the list of currently mounted ZFS file systems. Display a long listing for the `/firstpool` directory.

```
# zfs mount
firstpool                       /firstpool
# ls -ld /firstpool
drwxr-xr-x   2 root     sys            2 Dec 12 19:16
/firstpool
#
```

14. Use the `zpool` command to destroy `firstpool`. Verify that `firstpool` no longer exists.

```
# zpool destroy firstpool
# zpool list
no pools available
#
```

15. Verify that the `firstpool` file system and the `/firstpool` mount point no longer exist.

```
# zfs list
no datasets available
# ls -ld /firstpool
/firstpool: No such file or directory
#
```

Task 2 – Creating, Renaming, and Destroying ZFS File Systems

Complete the following steps:

1. Use the `zpool` command to display the list of ZFS pools. Verify that no ZFS pool currently exists.

```
# zpool list
no pools available
#
```

2. If you have three unused disks available, use the `zpool` command to create a storage pool called `mypool` that consists of one, three-disk RAID-Z virtual device.

```
# zpool create mypool raidz c2t0d0 c2t1d0 c2t2d0
```

3. If you have only two unused disks available, use the `zpool` command to create a storage pool called `mypool` that consists of one mirror virtual device.

```
# zpool create mypool mirror c2t0d0 c2t1d0
```

4. If you have only one unused disk available, use the `zpool` command to create a storage pool called `mypool` that uses the single disk as a non-redundant, top-level virtual device.

```
# zpool create mypool c2t0d0
```

Note – Example command output throughout the remainder of this exercise assumes that a non-redundant pool is in use. This isn't a recommended pool configuration.

---

5. Use the `zfs` command to list ZFS file systems and display `/mypool` space use.

```
# zfs list
NAME                     USED  AVAIL  REFER  MOUNTPOINT
mypool                    76K  66.9G  24.5K  /mypool
#
```

6. Attempt to create a file system named `mypool/fs1/fs2`. What happens and why?

```
# zfs create mypool/fs1/fs2
cannot create 'mypool/fs1/fs2': parent does not exist
#
```

The attempt fails because the intermediate file system `fs1` does not exist.

7. Create a directory called `/mypool/fs1`. Again attempt to create a file system named `mypool/fs1/fs2`. What happens and why?

```
# mkdir /mypool/fs1
# zfs create mypool/fs1/fs2
cannot create 'mypool/fs1/fs2': parent does not exist
#
```

The attempt fails because the intermediate file system `fs1` does not exist. The ZFS only creates new file systems immediately below a parent file system.

8. Create a new empty file named `/mypool/fs1/file1`.

```
# touch /mypool/fs1/file1
#
```

9. Attempt to create a new file system named `mypool/fs1`. Use the `zfs list` and `zfs mount` commands to list the ZFS file systems that currently exist and that are currently mounted. What happens, and why?

```
# zfs create mypool/fs1
cannot mount '/mypool/fs1': directory is not empty
filesystem successfully created, but not mounted
# zfs list
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool                  108K  66.9G  27.5K  /mypool
mypool/fs1             24.5K  66.9G  24.5K  /mypool/fs1
# zfs mount
mypool                            /mypool
#
```

The new file system exists, but isn't mounted. The empty file prevented the `/mypool/fs1` file system from mounting.

10. Destroy the `mypool/fs1` file system.

```
# zfs destroy mypool/fs1
#
```

11. Remove `file1` from the `/mypool/fs1` directory.

```
# rm /mypool/fs1/file1
#
```

12. Again attempt to create a new file system named `mypool/fs1`. What happens?

```
# zfs create mypool/fs1
#
```

The `mypool/fs1` file system is created successfully.

13. Verify that the new file system exists and that it is mounted.

```
# zfs list
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool                  108K  66.9G  26.5K  /mypool
mypool/fs1             24.5K  66.9G  24.5K  /mypool/fs1
# zfs mount
mypool                            /mypool
mypool/fs1                        /mypool/fs1
#
```

14. Rename `mypool/fs1` to `mypool/fs2`.

```
# zfs rename mypool/fs1 mypool/fs2
```

15. Verify that `mypool/fs2` exists and that it is mounted.

```
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
mypool                    108K  66.9G  26.5K  /mypool
mypool/fs2               24.5K  66.9G  24.5K  /mypool/fs2
# zfs mount
mypool                               /mypool
mypool/fs2                           /mypool/fs2
#
```

16. List /mypool. Does the fs1 directory exist?

```
# ls /mypool
fs2
#
```

No, the fs1 directory no longer exists.

17. Destroy the mypool/fs2 file system.

```
# zfs destroy mypool/fs2
#
```

18. Verify that mypool/fs2 isn't mounted and no longer exists.

```
# zfs mount
mypool                               /mypool
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
mypool                     76K  66.9G  24.5K  /mypool
#
```

19. List /mypool. Does the fs2 directory exist?

```
# ls /mypool
#
```

No, the /mypool directory is empty.

Task 3 – Working With ZFS Properties

1. Use the zfs command to create a new ZFS in mypool called cmp. List the space use for /mypool/cmp.

```
# zfs create mypool/cmp
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
mypool                    104K  66.9G  24.5K  /mypool
mypool/cmp               24.5K  66.9G  24.5K  /mypool/cmp
#
```

2. Use the `tar` command to create a partial archive of the `/usr/lib` directory, and save the archive as `/mypool/cmp/archive.tar`. Use the `-k` option to limit the size of this archive to 400 Mbytes. You will use this file to demonstrate how the `compression` property functions in ZFS file systems.

```
# tar cfk /mypool/cmp/archive.tar 409600 /usr/lib
tar: please insert new volume, then press RETURN.(Enter Control-C)
#
```

Note – The `tar` command will ask you to press `RETURN` when it has created an archive of the size (in kilobytes) you specified. Enter `Control-C` in response to this prompt. Doing so prevents `tar` from overwriting `archive.tar` with the next set of files that exceed the limit you specified.

---

3. Use the `zfs get` command to display all of the properties for the `mypool/cmp` file system. Do the properties all use the same source? If so, which source is it, and why?

```
# zfs get all mypool/cmp
NAME         PROPERTY         VALUE                  SOURCE
mypool/cmp   type              filesystem             -
mypool/cmp   creation          Mon Oct  5 12:57 2009  -
mypool/cmp   used              400M                   -
mypool/cmp   available         72.9G                  -
mypool/cmp   referenced        400M                   -
mypool/cmp   compressratio     1.00x                  -
mypool/cmp   mounted           yes                    -
mypool/cmp   quota             none                   default
mypool/cmp   reservation       none                   default
mypool/cmp   recordsize        128K                   default
mypool/cmp   mountpoint        /mypool/cmp            default
mypool/cmp   sharenfs          off                    default
mypool/cmp   checksum          on                     default
mypool/cmp   compression       off                    default
mypool/cmp   atime             on                     default
mypool/cmp   devices           on                     default
mypool/cmp   exec              on                     default
mypool/cmp   setuid            on                     default
mypool/cmp   readonly          off                    default
mypool/cmp   zoned             off                    default
mypool/cmp   snapdir           hidden                 default
mypool/cmp   aclmode           groupmask              default
mypool/cmp   aclinherit        restricted             default
mypool/cmp   canmount          on                     default
mypool/cmp   shareiscsi        off                    default
```

```
mypool/cmp  xattr                  on                     default
mypool/cmp  copies                 1                      default
mypool/cmp  version                4                      -
mypool/cmp  utf8only               off                    -
mypool/cmp  normalization          none                   -
mypool/cmp  casesensitivity        sensitive              -
mypool/cmp  vscan                  off                    default
mypool/cmp  nbmand                 off                    default
mypool/cmp  sharesmb               off                    default
mypool/cmp  refquota               none                   default
mypool/cmp  refreservation         none                   default
mypool/cmp  primarycache           all                    default
mypool/cmp  secondarycache         all                    default
mypool/cmp  usedbysnapshots        0                      -
mypool/cmp  usedbydataset          400M                   -
mypool/cmp  usedbychildren         0                      -
mypool/cmp  usedbyrefreservation   0                      -
#
```

All of the settable properties have the same source: `default`*. This is true because none of the properties have been set locally or inherited from a locally-set property.*

4. Use `ls` with the `-lh` options to list the size of `/mypool/cmp/archive.tar`. Make note of the size displayed.

```
# ls -lh /mypool/cmp/archive.tar
-rw-r--r--   1 root     root        400M Dec 18 17:18
/mypool/cmp/archive.tar
#
```

5. Use the `zfs get` command to display the current settings of the `used`, `compression` and `compressratio` properties for `mypool/cmp`. Verify that compression is `off`, the compression ratio is 1.00x, and that the space used matches the size of `archive.tar`.

```
# zfs get used,compression,compressratio mypool/cmp
NAME              PROPERTY         VALUE                       SOURCE
mypool/cmp        used            400M                        -
mypool/cmp        compression     off                         default
mypool/cmp        compressratio   1.00x                       -
#
```

6. Set the `compression` property for `mypool/cmp` to `on`.

```
# zfs set compression=on mypool/cmp
#
```

7. Use `zfs get` to display the current settings of the `used`, `compression` and `compressratio` properties for `mypool/cmp`. Verify that compression is now `on`. Have the compression ratio or space used values changed? If not, why not? What source is listed for the `compression` property?

```
# zfs get used,compression,compressratio mypool/cmp
NAME              PROPERTY        VALUE                      SOURCE
mypool/cmp        used           400M                      -
mypool/cmp        compression    on                        local
mypool/cmp        compressratio  1.00x                     -
#
```

No, the compression ratio remains 1.00, and the space used remains 400Mbytes. Turning on compression for a file system does not compress files that already exist within the file system. The `compression` property lists the `local` source.

8. Change directory to `/mypool/cmp`. Copy `archive.tar` to `archive1.tar`.

```
# cd /mypool/cmp
# cp archive.tar archive1.tar
```

9. Use `ls -lh` and `du -h` to display the sizes of `archive.tar` and `archive1.tar`. What file sizes do these commands report?

```
# ls -lh
total 1348943
-rw-r--r--   1 root     root        400M Dec 18 17:18 archive.tar
-rw-r--r--   1 root     root        400M Dec 18 17:38 archive1.tar
# du -h *
 400M   archive.tar
 234M   archive1.tar
#
```

The `ls -lh` command reports that both files are 400 Mbytes in size. The `du -h` command reports that `archive.tar` uses 400 Mbytes of space, and `archive1.tar` uses about 234 Mbytes.

Note – The size of the compressed archive and compression ratio values may vary, depending on the archive content.

---

10. Use the `zfs get` command to display the current settings of the `used`, `compression` and `compressratio` properties for `mypool/cmp`. How have the `compressratio` and `used` values changed from the last time you checked them, and why?

```
# zfs get used,compression,compressratio mypool/cmp
```

```
NAME          PROPERTY       VALUE                    SOURCE
mypool/cmp    used           659M                     -
mypool/cmp    compression    on                       local
mypool/cmp    compressratio  1.24x                    -
#
```

The `used` *property now lists about 659 Mbytes, and the* `compressratio` *property now indicates about 1.24x compression. The space used matches the sum of the file sizes, and the compression ratio indicates the compression achieved for all files in the file system.*

11. Set the `compression` property for `mypool/cmp` to `off`.

```
# zfs set compression=off mypool/cmp
#
```

12. Copy `archive.tar` to `archive2.tar`.

```
# cp archive.tar archive2.tar
#
```

13. Use `ls -lh` and `du -h` to display the sizes of `archive.tar` and `archive1.tar`. What file sizes do these commands report?

```
# ls -lh
total 2168760
-rw-r--r--   1 root     root        400M Dec 18 17:18 archive.tar
-rw-r--r--   1 root     root        400M Dec 18 17:38 archive1.tar
-rw-r--r--   1 root     root        400M Dec 18 18:56 archive2.tar
# du -h *
 400M   archive.tar
 234M   archive1.tar
 334M   archive2.tar
```

The `ls -lh` command reports that all three files are 400 Mbytes in size. The `du -h` command reports that `archive.tar` uses 400 Mbytes, `archive1.tar` uses 234 Mbytes of space, and `archive2.tar` uses 334 Mbytes.

14. Use `zfs get` to display the current settings of the `used`, `compression` and `compressratio` properties for `mypool/cmp`. How have the `compressratio` and `used` values changed from the last time you checked them, and why? What source is listed for the `compression` property?

```
# zfs get used,compression,compressratio mypool/cmp
NAME             PROPERTY       VALUE                    SOURCE
mypool/cmp       used           1.01G                    -
mypool/cmp       compression    off                      local
mypool/cmp       compressratio  1.15x                    -
#
```

The `used` property now lists about 1.01 Gbytes, and the `compressratio` property now indicates about 1.15x compression. The space used matches the sum of the file sizes. The compression ratio is smaller because a smaller proportion of the data in the file system is compressed. The `compression` property still lists the `local` source.

15. Use `zfs inherit` to clear the `compression` property for `mypool/cmp`. Verify that the change was made. What source is listed for the `compression` property?

```
# zfs inherit compression mypool/cmp
# zfs get compression mypool/cmp
NAME             PROPERTY       VALUE                    SOURCE
mypool/cmp       compression    off                      default
#
```

16. Create two new file systems named `mypool/cmp/a` and `mypool/cmp/a/b`.

```
# zfs create mypool/cmp/a
# zfs create mypool/cmp/a/b
```

17. Use `zfs get` to recursively display the `compression` property for `mypool/cmp`. Do all three file systems list the same value and source?

```
# zfs get -r compression mypool/cmp
NAME              PROPERTY       VALUE                        SOURCE
mypool/cmp        compression    off                          default
mypool/cmp/a      compression    off                          default
mypool/cmp/a/b    compression    off                          default
#
```

Yes, all three file systems have compression turned off, and they list the `default` *source for the* `compression` *property.*

18. Use the `zfs set` command to set the `compression` property to `on` for `mypool/cmp`.

```
# zfs set compression=on mypool/cmp
#
```

19. Use the `zfs get` command to recursively display the `compression` property for `mypool/cmp`. Do all three file systems list the same value and source?

```
# zfs get -r compression mypool/cmp
NAME              PROPERTY       VALUE                        SOURCE
mypool/cmp        compression    on                           local
mypool/cmp/a      compression    on                             inherited from
mypool/cmp
mypool/cmp/a/b    compression    on                             inherited from
mypool/cmp
```

All three file systems have compression turned on. The `mypool/cmp file` *system lists the local source for the compression property,* `mypool/cmp/a` *and* `mypool/cmp/a/b` *indicate that they inherited their compression values from* `mypool/cmp`.

20. Use the `zfs set` command to set the `compression` property to `off` for `mypool/cmp/a`.

```
# zfs set compression=off mypool/cmp/a
```

21. Use the `zfs get` command to recursively display the `compression` property for `mypool/cmp`. What has changed in the reported values and sources?

```
# zfs get -r compression mypool/cmp
NAME              PROPERTY       VALUE                        SOURCE
mypool/cmp        compression    on                           local
mypool/cmp/a      compression    off                          local
mypool/cmp/a/b    compression    off                            inherited from
mypool/cmp/a
```

The `mypool/cmp/a` and mypool/cmp/a/b file systems now have compression turned off. The `mypool/cmp/a` file system lists the `local` source for the `compression` property, and `mypool/cmp/a/b` now inherits its `compression` value from `mypool/cmp/a`.

22. Use the `zfs inherit` command to clear the `compression` property for `mypool/cmp`.

```
# zfs inherit compression mypool/cmp
```

23. Use the `zfs get` command to recursively display the `compression` property for `mypool/cmp`. What has changed in the reported values and sources?

```
# zfs get -r compression mypool/cmp
NAME            PROPERTY       VALUE                              SOURCE
mypool/cmp      compression    off                               default
mypool/cmp/a    compression    off                               local
mypool/cmp/a/b   compression    off                                 inherited from
mypool/cmp/a
#
```

All three file systems have compression turned off. The `mypool/cmp`, `mypool/cmp/a`, and `mypool/cmp/a/b` file systems list the `default`, `local`, and `inherited from` sources, respectively. The `mypool/cmp/a/b` file system inherits its `compression` value from `mypool/cmp/a`.

24. Change directory to root (/). Use the `zfs destroy` command to destroy `mypool/cmp` and its dependent file systems. Verify that the file systems no longer exist.

```
# cd /
# zfs destroy -r mypool/cmp
# zfs list
NAME                       USED  AVAIL  REFER  MOUNTPOINT
mypool                     103K  66.9G  25.5K  /mypool
#
```

25. Use the `zpool` command to destroy `mypool` file system.

```
# zpool destroy mypool
#
```

26. Verify that the `mypool` file system no longer exist.

```
# zpool list
no pools available
# zfs list
no datasets available
#
```

Module 6

Performing Mounts and Unmounts

Objectives

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

- Identify mounting basics
- Perform mounts
- Mount ZFS file systems
- Perform unmounts
- Repair important files if boot fails
- Access a mounted diskette, CD-ROM, or DVD
- Restrict access to a mounted diskette, CD-ROM, or DVD
- Access a diskette, CD-ROM or DVD without volume management (`vold`)

Working With Mounting Basics

In the Solaris OS, use the mounting process to attach individual file systems to their mount points on the directory hierarchy. This action makes a file system accessible to the system and to the users.

Use the unmounting process to detach a file system from its mount point in the directory hierarchy. This action makes a file system unavailable to the system or users.

After you create a UNIX[R] file system (UFS) with the `newfs` command, you must attach it to the Solaris OS directory hierarchy at a mount point. A mount point is a directory that is the point of connection for a file system. File systems are commonly referred to by the names of their mount points, for example, the `/` (root) file system or the `/usr` file system.

This module focuses on methods used to mount and unmount UFSs.

Figure 6-1 shows how the directory hierarchy spans from one file system to the next.

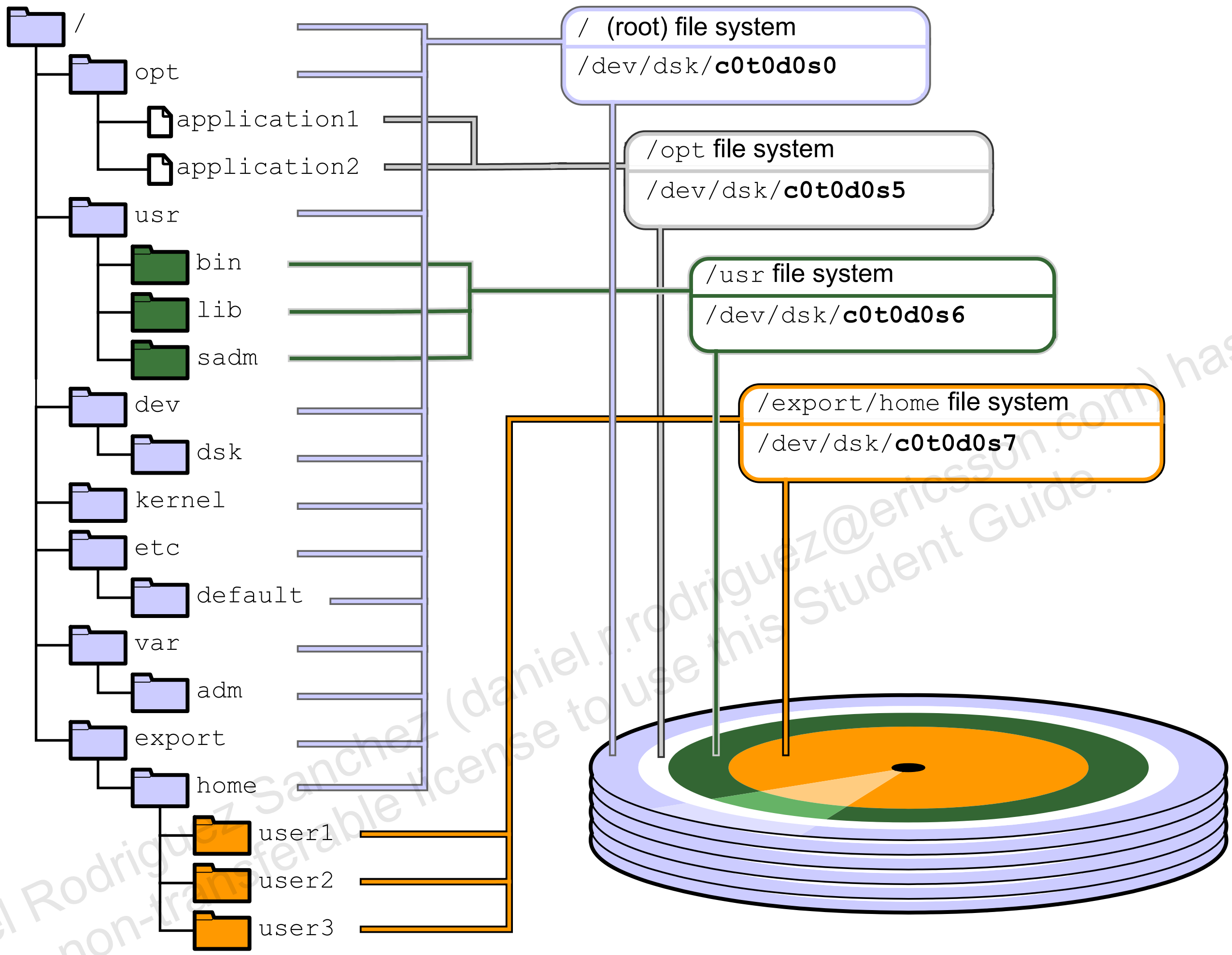


Figure 6-1 File Systems and Mount Points

File systems do not contain their own mount point directories.

Determining Which File Systems Are Currently Mounted

Determine which file systems are currently mounted by using the `mount` or `df` command.

The `df` command displays the amount of disk space occupied by mounted or unmounted file systems and, depending on the options used, displays both locally mounted and virtual file system information.

The `mount` command, which is located in the `/usr/sbin` directory, maintains a table of currently mounted file systems in the `/etc/mnttab` file system table. When the `mount` command is used without arguments, it lists all of the mounted file systems in the `/etc/mnttab` file system table. When used with only a partial argument list, the `mount` command searches the `/etc/vfstab` file for an entry that supplies the missing arguments.

Note – Options to the `mount` command are discussed later in this module.

Note – The symbolic link `/usr/sbin/mount` points to the `/sbin/mount` command.

Mounting a File System Automatically

The Solaris OS provides several methods for automating file system mounts.

The Solaris OS creates a default `/etc/vfstab` file during software installation, based on your selections. You can edit the `/etc/vfstab` file whenever file system entries need to be added or modified.

Note – The automounter can mount network file systems on demand.

Introducing the Virtual File System Table: /etc/vfstab

The /etc/vfstab file lists all the file systems to be automatically mounted at system boot time, with the exception of the /etc/mnttab and /var/run file systems.

The file format includes seven fields per line entry. By default, a tab separates each field, but any white space can be used for separators. The dash (–) character is used as a placeholder for fields when text arguments are not appropriate. Commented lines begin with the (#) symbol.

Note – Because the default is to use tabs to separate the fields in the /etc/vfstab file, the fields often do not line up under their respective headings. This can lead to some confusion when you are viewing this file in a terminal window.

An example /etc/vfstab file from a Sun Blade 1500:

```
# cat /etc/vfstab
#device         device          mount           FS      fsck    mount
mount
#to mount       to fsck         point           type    pass    at boot
options
#
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t0d0s1       -       -       swap    -       no      -
/dev/dsk/c0t0d0s0       /dev/rdsk/c0t0d0s0      /       ufs     1
no      -
/dev/dsk/c0t0d0s6       /dev/rdsk/c0t0d0s6      /usr    ufs     1
no      -
/dev/dsk/c0t0d0s3       /dev/rdsk/c0t0d0s3      /var    ufs     1
no      -
/dev/dsk/c0t0d0s7      /dev/rdsk/c0t0d0s7     /export/home    ufs    2
yes    -
/dev/dsk/c0t0d0s5       /dev/rdsk/c0t0d0s5      /opt    ufs     2
yes     -
/devices        -       /devices        devfs   -       no      -
ctfs    -       /system/contract        ctfs    -       no      -
objfs   -       /system/object  objfs   -       no      -
swap    -       /tmp    tmpfs   -       yes     -
#
```

An example /etc/vfstab file from an Ultra 20 system:

```
# cat /etc/vfstab
#device          device           mount            FS      fsck    mount
mount
#to mount        to fsck          point            type    pass    at boot
options
#
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c1d0s1 -       -       swap    -       no      -
/dev/dsk/c1d0s0 /dev/rdsk/c1d0s0        /       ufs     1       no      -
/dev/dsk/c1d0s6 /dev/rdsk/c1d0s6        /usr    ufs     1       no      -
/dev/dsk/c1d0s3 /dev/rdsk/c1d0s3        /var    ufs     1       no      -
/dev/dsk/c1d0s7 /dev/rdsk/c1d0s7        /export/home    ufs     2
yes     -
/dev/dsk/c1d0s5 /dev/rdsk/c1d0s5        /opt    ufs     2       yes     -
/devices        -       /devices        devfs   -       no      -
ctfs    -       /system/contract        ctfs    -       no      -
objfs   -       /system/object  objfs   -       no      -
swap    -       /tmp    tmpfs   -       yes     -
#
```

To add a new entry to the /etc/vfstab file, you need the following information:

device to mount	The device to be mounted. For example, a local ufs file system /dev/dsk/*c#t#d#s#*, or a pseudo file system /proc.
device to fsck	The raw or *character* device checked by the file system check program (fsck) if applicable. Pseudo and distributed file systems have a dash (-) in this field.
mount point	The name of the directory that serves as the attach mount point in the Solaris OS directory hierarchy.
FS type	The type of file system to be mounted.

`fsck pass`	The pass number used by the `fsck` command to decide whether to check a file system. When the field contains a (-), the file system is not checked. When the field contains a zero, UFS file systems are not checked, however, non-UFS file systems are checked. When the field contains a value greater than zero, the file system is always checked. All file systems with a value of 1 in this field are checked one at a time in the order they appear in the `vfstab` file. When the `fsck` command is run on multiple UFSs that have `fsck` pass values greater than 1 and the `preen` option (`-o p`) is used, the `fsck` command automatically checks the file systems on different disks in parallel to maximize efficiency. Otherwise, the value of the pass number does not have any effect.
`mount at boot`	Enter `yes` to enable the `mountall` command to mount the file systems at boot time. Enter `no` to prevent a file system mount at boot time.

Note – For `/` (root), `/usr`, and `/var` (if it is a separate file system) file systems, the `mount at boot` field value is specified as `no`. The `kernel` mounts these file systems as part of the boot sequence before the `mountall` command is run. SMF uses scripts in the `/lib/svc/method` directory whose names begin with `fs-` to mount all required file systems.

`mount options`	A comma-separated list of options passed to the `mount` command. A dash (-) indicates the use of default mount options.

Introducing the /etc/mnttab File System Table

The file /etc/mnttab is an mntfs file system that provides read-only access to the table of mounted file systems for the current host.

Each time a file system is mounted, the mount command adds an entry to this table. Whenever a file system is unmounted, its entry is removed from this table. These are the fields in the /etc/mnttab file system table:

Device Name	The name of the device that is mounted at the mount point. This block device is where the file system is physically located.
Mount Point	The mount point or directory name where the file system is attached (for example, /usr, /opt).
File System Type	The file system type of the mounted file system.
Mount Options	The list of mount options in effect for the file system.This includes a dev=*number* which reflects the minor device number of the mounted file system in hexidecimal.
Time	The time at which the file system was mounted.

The /var/run file system is a tmpfs mounted file system in the Solaris OS. It is the repository for temporary operating system files that are not needed across system reboots in this Solaris OS release. It is mounted as a pseudo file system rather than a disk-based file system.

The /var/run directory requires no administration. For security reasons, it is owned by the root user.

The /tmp directory continues to be a tmpfs mounted file system in the Solaris OS. It is the repository for temporary user and application files that are not needed across system reboots. It is a pseudo file system rather than a disk-based file system.

The following examples show two ways to display currently mounted file systems.

```
# cat /etc/mnttab
/dev/dsk/c0t0d0s0       /       ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=2200008      1170808975
/devices        /devices        devfs   dev=53c0000     1170808960
ctfs    /system/contract        ctfs    dev=5400001     1170808960
proc    /proc   proc    dev=5440000     1170808960
mnttab  /etc/mnttab     mntfs   dev=5480001     1170808960
swap    /etc/svc/volatile       tmpfs   xattr,dev=54c0001       1170808960
objfs   /system/object  objfs   dev=5500001     1170808960
/dev/dsk/c0t0d0s6       /usr    ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=220000e      1170808975
/platform/sun4u-us3/lib/libc_psr/libc_psr_hwcap1.so.1   /platform/sun4u-
us3/lib/libc_psr.so.1   lofs    dev=2200008     1170808971
/platform/sun4u-us3/lib/sparcv9/libc_psr/libc_psr_hwcap1.so.1   /platform/sun4u-
us3/lib/sparcv9/libc_psr.so.1   lofs    dev=2200008     1170808972
fd      /dev/fd fd      rw,dev=56c0001  1170808975
/dev/dsk/c0t0d0s3       /var    ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=220000b      1170808989
swap    /tmp    tmpfs   xattr,dev=54c0002       1170808989
swap    /var/run        tmpfs   xattr,dev=54c0003       1170808989
/dev/dsk/c0t0d0s5       /opt    ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=220000d      1170808994
/dev/dsk/c0t0d0s7       /export/home    ufs
rw,intr,largefiles,logging,xattr,onerror=panic,dev=220000f      1170808994
-hosts  /net    autofs  nosuid,indirect,ignore,nobrowse,dev=5780001
1170809014
auto_home       /home   autofs  indirect,ignore,nobrowse,dev=5780002
1170809014
sys-05:vold(pid517)     /vol    nfs     ignore,noquota,dev=5740001
1170809022
#
```

```
# mount
/ on /dev/dsk/c0t0d0s0
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000
8 on Tue Feb  6 17:42:55 2007
/devices on /devices read/write/setuid/devices/dev=53c0000 on Tue Feb  6
17:42:40 2007
/system/contract on ctfs read/write/setuid/devices/dev=5400001 on Tue Feb  6
17:42:40 2007
/proc on proc read/write/setuid/devices/dev=5440000 on Tue Feb  6 17:42:40 2007
/etc/mnttab on mnttab read/write/setuid/devices/dev=5480001 on Tue Feb  6
17:42:40 2007
/etc/svc/volatile on swap read/write/setuid/devices/xattr/dev=54c0001 on Tue Feb
6 17:42:40 2007
/system/object on objfs read/write/setuid/devices/dev=5500001 on Tue Feb  6
17:42:40 2007
/usr on /dev/dsk/c0t0d0s6
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000
e on Tue Feb  6 17:42:55 2007
/platform/sun4u-us3/lib/libc_psr.so.1 on /platform/sun4u-
us3/lib/libc_psr/libc_psr_hwcap1.so.1 read/write/setuid/devices/dev=2200008 on
Tue Feb  6 17:42:51 2007
/platform/sun4u-us3/lib/sparcv9/libc_psr.so.1 on /platform/sun4u-
us3/lib/sparcv9/libc_psr/libc_psr_hwcap1.so.1
read/write/setuid/devices/dev=2200008 on Tue Feb  6 17:42:52 2007
/dev/fd on fd read/write/setuid/devices/dev=56c0001 on Tue Feb  6 17:42:55 2007
/var on /dev/dsk/c0t0d0s3
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000
b on Tue Feb  6 17:43:09 2007
/tmp on swap read/write/setuid/devices/xattr/dev=54c0002 on Tue Feb  6 17:43:09
2007
/var/run on swap read/write/setuid/devices/xattr/dev=54c0003 on Tue Feb  6
17:43:09 2007
/opt on /dev/dsk/c0t0d0s5
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000
d on Tue Feb  6 17:43:14 2007
/export/home on /dev/dsk/c0t0d0s7
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=220000
f on Tue Feb  6 17:43:14 2007
#
```

Performing Mounts

Mount file systems manually by running the mount command, or the system can automatically mount file systems at boot time after consulting the /etc/vfstab file.

Mounting a Local File System Manually

The mount command not only lists which file systems are currently mounted, it also provides you with a method for mounting file systems.

Default Behavior of the mount Command

To mount a local file system manually, you need to know the name of the device where the file system resides and its mount point path name. Perform the command:

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

In this example, the default action mounts the file system with the following options: read/write, setuid, intr, logging, largefiles, xattr, and onerror.

The following list explains the default options for the mount command.

read/write	Indicates whether reads and writes are allowed on the file system.
setuid	Permits the execution of setuid programs in the file system.
intr/nointr	Allows and forbids keyboard interrupts to kill a process that is waiting for an operation on a locked file system.
logging	Indicates that logging is enabled for the ufs file system. This is the default for the Solaris 10 OS.
largefiles	Allows for the creation of files larger than 2 Gbytes. A file system mounted with this option can contain files larger than 2 Gbytes.
xattr	Supports extended attributes not found in standard UNIX filesystems. See man fsattr(5).

Note – Due to file system overhead, the largest file that can be created in a ufs file system is approximately 1 Tbyte. The data capacity of a 1 Tbyte file system is approximately 1 Tbyte minus 0.5% overhead and the recommended 1% free space.

`onerror=`*`action`* Specifies the action that the `ufs` file system should take to recover from an internal inconsistency on a file system. An action can be specified as:

`panic` — Causes a forced system shutdown. This is the default.

`lock` — Applies a file system lock to the file system.

`umount` — Forcibly unmounts the file system.

The `/etc/vfstab` file provides you with another important feature. Because the `/etc/vfstab` file contains the mapping between the mount point and the actual device name, the `root` user can manually mount a file system specifying only the mount point on the command line.

```
# mount /export/home
```

Using the `mount` Command Options

When you are using `mount` options on the command line, remember that the options are preceded by the `-o` flag. When you are using multiple options, enter them as a comma-separated list following the `-o` flag.

```
mount -o option,option,... device_name mount_point
```

Note – Mount options are described in detail in the `mount_ufs` man page. This man page describes `mount` command options specific to the ufs file system.

Some options used to mount local file systems include: ro, nosetuid, noatime, nolargefiles, and nologging.

- ro – Mounts the file system as read-only.

  The following is an example using this option on the command line:

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

- nosuid – Prohibits the execution of setuid programs in the file system. This does not restrict the creation of setuid programs.

  The following example shows the use of multiple options on the command line:

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

- noatime – Suppresses the time-last-accessed modification on inodes, which reduces disk activity on a file system where access times are not important. Specifying this option generally improves file access times and boosts overall performance, for example:

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

- nolargefiles – Prevents a file system that contains one or more "large files" from being mounted, for example:

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

  Use of the nolargefiles option fails if the file system to be mounted contains a large file or did contain a large file at one time.

  If the file system currently contains a large file and the root user needs to mount it with this option, then the large file must be located and moved or removed from the file system. Then you must execute the fsck command manually to update the superblock information.

  The mount also fails if the file system at one time contained a large file, even though it was moved or removed. You must execute the fsck command to clear the old information and allow the file system to be mounted.

Mounting All File Systems Manually

The /etc/vfstab file is read by the /usr/sbin/mountall command during the system boot sequence and mounts all file systems that have a yes in the mount at boot field.

The root user can use the mountall command to mount manually every file system in the /etc/vfstab file that has a yes in the mount at boot field, for example:

```
# mountall
```

To mount only the local file systems listed in the /etc/vfstab file, execute:

```
# mountall -l
```

During the boot sequence, the fsck utility checks each local file system in the /etc/vfstab file that has a device to fsck entry and an fsck pass number greater than 0. The utility determines if the file system is in a usable state to be safely mounted.

If the fsck utility determines that the file system is in an unusable state (for example, corrupted), the fsck utility will only perform non-interactive repairs before attempting to mount the file system. If the repairs require interaction, the boot process will take the system to single-user mode to allow you to run fsck manually.

The system attempts to mount any local file systems that have a - (dash) or 0 (zero) entry in the fsck pass field without checking the file system itself

Mounting a New File System

To add a new disk to the system, prepare the disk to hold a file system, and mount the file system, perform these general steps:

1. Set up the disk hardware, which might include setting address switches and connecting cables.
2. Perform a reconfiguration boot or run the `devfsadm` utility to add support for the new disk.
3. Use the `format` utility to define one or more slices. On x86/x64 systems, create a Solaris `fdisk` partition, and then create one or more slices within the Solaris partition.
4. Create a new file system on one slice by using the `newfs` command.
5. Create a mount point for the file system by using the `mkdir` command to create a new directory in the `/` (root) file system.

   ```
   # mkdir /data
   ```
6. Mount the new file system manually by using the `mount` command.

   ```
   # mount /dev/dsk/c1t3d0s7 /data
   ```
7. Use the `mount` command to determine if the file system is mounted.

   ```
   # mount
   ```

```
...
/data on /dev/dsk/c1t3d0s7
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev
=800027 on Sun Oct 24 11:55:34 2004
```

8. Edit the `/etc/vfstab` file to add a line entry for the new file system.

```
# vi /etc/vfstab
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t0d0s1       -       -       swap    -       no      -
/dev/dsk/c0t0d0s0  /dev/rdsk/c0t0d0s0    /       ufs     1     no    -
/dev/dsk/c0t0d0s6  /dev/rdsk/c0t0d0s6    /usr    ufs     1     no    -
/dev/dsk/c0t0d0s3  /dev/rdsk/c0t0d0s3    /var    ufs     1     no    -
/dev/dsk/c0t0d0s7  /dev/rdsk/c0t0d0s7   /export/home   ufs  2   yes  -
/devices        -       /devices        devfs   -       no      -
ctfs    -       /system/contract        ctfs    -       no      -
objfs   -       /system/object  objfs   -       no      -
swap    -       /tmp    tmpfs   -       yes     -
/dev/dsk/c1t3d0s7  /dev/rdsk/c1t3d0s7  /data  ufs  2  yes  -
#
```

The file system automatically mounts whenever the system boots.

Mounting Different Types of File Systems

Different file system types have unique properties that affect how the mount command functions.

By default, the mount command assumes it is mounting a ufs-type file system. However, when you are mounting a different type of file system, you might have to specify its type on the command line.

You use the -F option with the mount command to specify the type of file system mounted. The file system type must be determinable from the /etc/vfstab, /etc/default/fs, or /etc/dfs/fstypes files.

Determining a File System's Type

Because the mount command needs the file system type to function properly, the file system type must be explicitly specified or determined by searching the following files:

- The /etc/vfstab file for the FS type field
- The /etc/default/fs file for a local file system type
- The /etc/dfs/fstypes file for a remote file system type

If the file system's type has not been explicitly specified on the command line using the mount -F *FStype* option, the mount command examines the /etc/vfstab file to determine the file system's type. The mount command makes this determination by using the file system's block device name, raw device name, or mount point directory name.

If the mount command cannot determine the file system's type by searching the /etc/vfstab file, the mount command uses the default file system type specified in either the /etc/default/fs file or the /etc/dfs/fstypes file, depending on whether the file system is local or remote.

The default local file system type is specified in the /etc/default/fs file by the line entry LOCAL=*fstype*.

```
LOCAL=ufs
```

The first line entry in the /etc/dfs/fstypes file determines the default remote file system type.

```
nfs NFS Utilities
autofs AUTOFS Utilities
cachefs CACHEFS Utilities
```

Using the fstyp Command

You can also use the fstyp command with the raw device name of the disk slice to determine a file system's type.

```
# fstyp /dev/rdsk/c0t0d0s7
ufs
```

Specifying a ZFS File System Type

To mount a ZFS file system using the mount command, as the root user, perform the command:

```
# mount -F zfs rzpool/users/jf /home/jf
```

In this example, the file system type is zfs, the file system is named rzpool/users/jf and resides within the ZFS pool called rzpool. The mount point /home/jf is an existing directory in the Solaris OS.

Note – To use the mount command to mount a ZFS file system, you must set the mountpoint property for the ZFS file system to the legacy value. By default, ZFS file systems mount automatically when you create them. See the zfs(1M) command man page for more information.

Specifying an hsfs File System Type

To mount a file system that resides on a CD-ROM when the Volume Management (vold) services are stopped, as the root user, perform the command:

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

In this example, the file system type is hsfs, the file system resides on disk slice /dev/dsk/c0t6d0s0, and the mount point /cdrom is an existing directory in the Solaris OS.

Specifying a pcfs File System Type

To mount a file system that resides on a diskette when the Volume Management (vold) services are stopped, perform the commands:

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

In this example, the file system type is pcfs. This file system resides on the device /dev/diskette, and the mount point is /pcfs.

Managing ZFS Mount Points

By default, all ZFS file systems are mounted by ZFS at boot by using the SMF `svc://system/filesystem/local` service. File systems are mounted under */path*, where *path* is the name of the file system.

You can override the default mount point by setting the `mountpoint` property to a specific path by using the `zfs set` command. ZFS automatically creates this mount point, if needed, and automatically mounts this file system when the `zfs mount -a` command is invoked, without requiring you to edit the `/etc/vfstab` file.

The `mountpoint` property is inherited. For example, if `pool/home` has its `mountpoint` set to `/export/stuff`, then `pool/home/user` inherits `/export/stuff/user` for its `mountpoint` property.

The `mountpoint` property can be set to `none` to prevent the file system from being mounted.

If desired, file systems can also be explicitly managed through legacy mount interfaces by setting the `mountpoint` property to `legacy` by using `zfs set`. Doing so prevents ZFS from automatically mounting and managing this file system. Legacy tools including the `mount` and `umount` commands, and the `/etc/vfstab` file must be used instead.

When changing mount point management strategies, the following behaviors apply:

- Automatic mount point behavior
- Legacy mount point behavior

Automatic Mount Points

- When changing from `legacy` or `none`, ZFS automatically mounts the file system.
- If ZFS is currently managing the file system but it is currently unmounted, and the `mountpoint` property is changed, the file system remains unmounted.

You can also set the default mount point for the root dataset at pool creation time by using `zpool create` with the `-m` option.

Any dataset whose mountpoint property is not legacy is managed by ZFS. In the following example, a dataset is created whose mount point is automatically managed by ZFS.

```
# zfs create pool/filesystem
# zfs get mountpoint pool/filesystem
NAME             PROPERTY      VALUE                         SOURCE
pool/filesystem  mountpoint    /pool/filesystem              default
# zfs get mounted pool/filesystem
NAME             PROPERTY      VALUE                         SOURCE
pool/filesystem  mounted       yes                           -
```

You can also explicitly set the mountpoint property as shown in the following example:

```
# zfs set mountpoint=/mnt pool/filesystem
# zfs get mountpoint pool/filesystem
NAME             PROPERTY      VALUE                         SOURCE
pool/filesystem  mountpoint    /mnt                          local
# zfs get mounted pool/filesystem
NAME             PROPERTY      VALUE                         SOURCE
pool/filesystem  mounted       yes                           -
```

When the mountpoint property is changed, the file system is automatically unmounted from the old mount point and remounted to the new mount point. Mount point directories are created as needed. If ZFS is unable to unmount a file system due to it being active, an error is reported and a forced manual unmount is necessary.

Legacy Mount Points

Manage ZFS file systems with legacy tools by setting the mountpoint property to legacy. Manage the legacy file systems through the mount and umount commands and the /etc/vfstab file. ZFS does not automatically mount legacy file systems on boot, and the zfs mount and zfs umount commands do not operate on datasets of this type. The following examples show how to set up and manage a ZFS dataset in legacy mode:

```
# zfs set mountpoint=legacy tank/home/eschrock
# mount -F zfs tank/home/eschrock /mnt
```

If you set up separate ZFS /usr or /var file systems, you must indicate that they are legacy file systems. You must mount them by creating entries in the /etc/vfstab file. Otherwise, the system/filesystem/local service enters maintenance mode when the system boots.

To automatically mount a legacy file system on boot, you must add an entry to the /etc/vfstab file. The following example shows what the entry in the /etc/vfstab file might look like:

```
#device         device       mount          FS      fsck    mount
mount
#to mount       to fsck      point          type    pass    at boot
options
#

tank/home/eschrock -      /mnt            zfs   -  yes-
```

The device to fsck and fsck pass entries are set to -. This syntax is used because the fsck command is not applicable to ZFS file systems.

Mounting a ZFS File System

ZFS automatically mounts file systems when file systems are created or when the system boots. Use the zfs mount command only when your are changing mount options or explicitly mounting or unmounting file systems.

The zfs mount command with no arguments shows all currently mounted file systems that are managed by ZFS. Legacy managed mount points are not displayed. For example:

```
# zfs mount
tank                                /tank
tank/home                           /tank/home
tank/home/bonwick                   /tank/home/bonwick
tank/ws                             /tank/ws
```

Use the -a option to mount all ZFS managed file systems. Legacy managed file systems are not mounted. For example:

```
# zfs mount -a
```

By default, ZFS doesn't allow mounting on top of a non-empty directory. To force a mount on top of a non-empty directory, use the -O option. For example:

```
# zfs mount tank/home/lalt
cannot mount '/export/home/lalt': directory is not empty
use legacy mountpoint to allow this behavior, or use the -O flag
# zfs mount -O tank/home/lalt
```

Use legacy tools to manage legacy mount points. If you use ZFS tools, you will get errors. For example:

```
# zfs mount pool/home/billm
cannot mount 'pool/home/billm': legacy mountpoint
use mount(1M) to mount this filesystem
# mount -F zfs tank/home/billm
```

Performing Unmounts

A file system is commonly unmounted if it needs to be checked and repaired by the `fsck` command, or if it needs to be backed up completely.

Unmounting a File System

Some file system administration tasks cannot be performed on mounted file systems.

To unmount a file system to prepare it for system maintenance, use the `umount` command. Unmounting a file system by using the `umount` command detaches it from the file system mount point and deletes its entry from the `/etc/mnttab` file system table.

Note – Notify users before unmounting a file system that they are currently accessing.

To unmount a file system manually by using the directory mount point, perform the command:

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

To unmount a file system manually by using the logical disk device name, perform the command:

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

Unmounting All File Systems

The `/etc/mnttab` file system table is read by the `/usr/sbin/umountall` command during the system `shutdown` sequence or when `umountall` is invoked from the command line. The `umountall` unmounts all file systems specified in the `vfstab` file except `/` (root), `/usr`, `/proc`, `/dev/fd`, `/var`, `/var/run`, and `/tmp`.

Run the `umountall` command as the `root` user when you want to unmount manually all the file systems listed in `/etc/mnttab`, for example:

```
# umountall
```

To unmount only the local file systems listed in /etc/mnttab, perform the command:

```
# umountall -l
```

To verify that a file system or a number of file systems have been unmounted, invoke the mount command and check the output.

Unmounting a Busy File System

Any file system that is busy is not available for unmounting. Both the umount and umountall commands display the error message:

umount: *file_system_name* busy

A file system is considered to be busy if one of the following conditions exists:

- A program is accessing a file or directory in the file system
- A user is accessing a directory or file in the file system
- A program has a file open in the file system
- The file system is being shared

Aside from eliminating the conditions above, there are two methods to make a file system available for unmounting if it is busy:

- fuser command – Lists all of the processes that are accessing the file system and kills them if necessary
- umount -f command – Forces the unmount of a file system

Note – The fuser command displays the process IDs of all processes currently using the specified file system. Each process ID is followed by a letter code. These letter codes are described in the man page for this command.

Using the `fuser` Command

To stop all processes that are currently accessing a file system, follow these steps:

1. As the `root` user, list all of the processes that are accessing the file system. Use the following command to identify which processes need to be terminated.

   # fuser -cu *mount_point*

   This command displays the name of the file system and the user login name for each process currently active in the file system.

2. Kill all processes accessing the file system.

   # **fuser -ck *mount_point***

   A `SIGKILL` message is sent to each process that is using the file system.

3. Verify that there are no processes accessing the file system.

   # **fuser -c *mount_point***

4. Unmount the file system.

   # umount *mount_point*

Using the `umount -f` Command

As the `root` user, you can use the `umount` command with the `-f` (force) option to unmount a file system even if it is busy. The following is the format for this command:

umount -f ***mount_point***

This command unmounts the file system even if it contains open files. A forced unmount can result in loss of data and in zombie processes that are left running on the system. However, it is particularly useful for unmounting a shared file system if the remote file server is nonfunctional.

Unmounting a ZFS File System

Unmount ZFS file systems with the `zfs unmount` subcommand. The `unmount` command can take either the mount point or the file system name as arguments.

In the following example, a file system is unmounted by file system name:

```
# zfs unmount tank/home/tabriz
```

In the following example, the file system is unmounted by mount point:

```
# zfs unmount /export/home/tabriz
```

The `unmount` command fails if the file system is active or busy. To forcibly unmount a file system, you can use the `-f` option. Be cautious when forcibly unmounting a file system, if its contents are actively being used. Unpredictable application behavior can result.

```
# zfs unmount tank/home/eschrock
cannot unmount '/export/home/eschrock': Device busy
# zfs unmount -f tank/home/eschrock
```

For backward compatibility, use the legacy `umount` command to unmount a ZFS file system. For example:

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

Repairing Important Files if Boot Fails on a SPARC System

The following procedure describes how to boot from a Solaris OS DVD to edit a misconfigured `/etc/vfstab` file.

1. Insert the Solaris 10 OS Software DVD into the CD-ROM drive.
2. Execute a single-user boot from the DVD.

```
ok boot cdrom -s
Boot device: /pci@1e,600000/ide@d/cdrom@2,0:f File and args -s
<<Output omitted>>
Booting to milestone "milestone/single-user:default"
Configuring devices.
Using RPC Bootparams for network configuration information.
Attempting to configure interface bge0...
Skipped interface bge0
Requesting System Maintenance Mode
SINGLE USER MODE
#
```

Note – Performing a single-user boot operation from this software CD-ROM creates an *in-memory* copy of the `/root` file system, which supports your ability to perform administrative tasks.

3. Use the `fsck` command on the faulty `/` (root) partition to check and repair any potential problems in the file system and make the device writable.

   ```
   # fsck /dev/rdsk/c0t0d0s0
   ```

4. If the `fsck` command is successful, mount the `/` (root) file system on the `/a` directory to gain access to the file system on disk.

   ```
   # mount /dev/dsk/c0t0d0s0 /a
   ```

5. Set and export the `TERM` variable, which enables the `vi` editor to work properly.

   ```
   # TERM=sun
   # export TERM
   ```

6. Edit the `/etc/vfstab` file, and correct any problems. Then exit the file.

   ```
   # vi /a/etc/vfstab
   :wq!
   ```

7. Unmount the file system.

```
# cd /
# umount /a
```

8. Reboot the system.

```
# init 6
```

Repairing Files if Boot Fails on x86/x64 Systems

The following procedure describes how to boot from the Solaris OS DVD to edit a misconfigured `/etc/vfstab` file.

1. Power on the system and insert the Solaris 10 OS Software DVD into the DVD-ROM drive.

Note – Your system needs to be configured to boot from the DVD. This might need to be set in your BIOS settings before booting.

---

2. The system detects the DVD, loads the grand unified boot (GRUB) loader, and presents the GRUB boot menu. Three boot choices are displayed:

```
Solaris
Solaris Serial Console ttya
Solaris Serial Console ttyb (for lx50, v60, and v65x)
```

The first boot choice (`Solaris`) is the default, and it boots the system from a DVD. With the `Solaris` choice the system automatically boots after 60 seconds. Press Return to boot immediately from this source.

3. The system presents the following list of installation methods and possible tasks to perform. Press the space bar to pause at this menu.

```
1.Solaris Interactive (default)
2.Custom Jump-start
3.Solaris Interactive Text (Desktop session)
4.Solaris Interactive Text (Console session)
5.Apply driver updates
6.Single user shell
```

4. Enter `6` to select the single user shell. The system searches for installed Solaris OS instances, and asks if you want to mount the root file system on the `/a` mount point. For example:

```
Searching for installed OS instances...

Solaris 10 10/09 s10x_u8wos_08a X86 was found on /dev/dsk/c1d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?]
```

5. Enter `n` to choose not to mount the file system.

```
Do you wish to have it mounted read-write on /a? [y,n,?] n

Starting shell.
#
```

6. Use the `fsck` command on the faulty `/` (root) partition to check and repair any potential problems in the file system and make the device writable.

   `# fsck /dev/rdsk/c1d0s0`

7. If the `fsck` command is successful, mount the `/` (root) file system on the `/a` directory to gain access to the file system.

   `# mount /dev/dsk/c1d0s0 /a`

8. Set and export the `TERM` variable, which enables the `vi` editor to work properly.

   ```
   # TERM=sun
   # export TERM
   ```

9. Edit the `/etc/vfstab` file, and correct any problems. Then exit the file.

   ```
   # vi /a/etc/vfstab
   :wq!
   ```

10. Unmount the file system.

    ```
    # cd /
    # umount /a
    ```

11. Eject the DVD.

12. Reboot the system.

    `# init 6`

Accessing Mounted Diskettes, CD-ROMs, or DVDs

To provide access to file systems on diskettes and CD-ROMs, the Solaris OS provides the volume management service standard interface.

Note – The Solaris 10 OS includes support for additional removable media such as DVDs, Jaz drives, and Zip drives. For more information on using these devices, see the resources available on the Solaris 10 Documentation CD or visit `http:/docs.sun.com` to access online documentation.

The volume management service provides two major benefits:

- It automatically mounts removable media for both the `root` user and non-`root` users.
- It can give other systems on the network automatic access to any removable media currently inserted in the local system.

The volume management service is controlled by the `/usr/sbin/vold` daemon. With a default install, this service automatically manages diskettes and CD-ROMs for regular users.

The volume management service features automatic DVD and CD-ROM detection. It doesn't detect the presence of a diskette that has been inserted in the drive until the `volcheck` command is run. This command instructs the `vold` daemon to check the diskette drive for any inserted media. The volume management service can mount `ufs`, `pcfs`, `hsfs`, and `udfs` file systems.

Using the Volume Management Service

To make working with diskettes, DVDs, and CD-ROMs simple, each device is easy to mount and mounts at an easy-to-remember location.

If the vold daemon detects that the mounted device contains a file system, then the device is mounted at the directory location.

Table 6-1 lists the directory locations of mounted devices that contain file systems.

Table 6-1 Directory Locations

Media Device	Access File Systems On
First diskette drive	/floppy/floppy0
First CD-ROM or DVD drive	/cdrom/cdrom0
First Jaz drive	/rmdisk/jaz0
First Zip drive	/rmdrive/zip0
First PCMCIA card	/pcmem0

If the vold daemon detects that the mounted device does not contain a file system, the device is accessible through a path.

Table 6-2 lists the paths for mounted devices that don't contain file systems.

Table 6-2 Paths for Accessing Devices

Media Device	Access Raw Device On
First diskette drive	/vol/dev/aliases/floppy0
First CD-ROM or DVD drive	/vol/dev/aliases/cdrom0
First Jaz drive	/vol/dev/aliases/jaz0
First Zip drive	/vol/dev/aliases/zip0
First PCMCIA card	/vol/dev/aliases/pcmem0

When the Volume Management service is running on the system, a regular user can easily access a diskette or CD-ROM by following these basic steps:

1. Insert the media.
2. For diskettes only, enter the `volcheck` command.
3. Use the `cd` command to change to the directory of the mounted volume.
4. Work with files on the media.
5. Use the `cd` command to leave the directory structure of the mounted volume.
6. Eject the media.

Table 6-3 shows the configuration files used by the Volume Management service.

Table 6-3 Volume Management Service Configuration Files

File	Description
`/etc/vold.conf`	The `vold` configuration file. This file defines items, such as what action should be taken when media is inserted or ejected, which devices are managed by vold, and which file system types are unsafe to eject.
`/etc/rmmount.conf`	The `rmmount` command configuration file. The `rmmount` command is a removable media mounter that is executed by the `vold` daemon whenever a CD-ROM or diskette is inserted.

Restricting Access to Mounted Diskettes, CD-ROMs, or DVDs

To restrict regular users from accessing diskettes or CD-ROMs on the system, you can, as the `root` user, disable the Volume Management service.

Stopping Volume Management (`vold`)

To stop Volume Management service from running on a system, as the `root` user, perform the command:

```
# svcadm disable svc:/system/filesystem/volfs:default
```

To restart the Volume Management service, as the `root` user, perform the command:

```
# svcadm enable svc:/system/filesystem/volfs:default
```

Legacy syntax to start and stop the Volume Management service is provided in Solaris 10. The `/etc/init.d/volmgt` script calls the `svcadm` command with the arguments required to stop and start the `vold` daemon. You can use the following legacy commands to stop and start the Volume Management service:

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

Troubleshooting Volume Management Service

If a CD-ROM fails to eject from the drive, as the `root` user, attempt to stop Volume Management service. If this is unsuccessful, stop the `vold` daemon.

```
# svcadm disable svc:/system/filesystem/volfs:default
```

or as a last resort:

```
# pkill -9 vold
```

Push the button on the system to eject the CD-ROM. The CD-ROM tray ejects. Remove the CD-ROM, and leave the tray out. Then enable the Volume Management service.

```
# svcadm enable svc:/system/filesystem/volfs:default
```

Wait a few seconds, and then push the CD-ROM tray back into the drive.

Accessing a Diskette, CD-ROM, or DVD Without VMS

When the `vold` daemon is not running, only the `root` user can mount and access a diskette or CD-ROM. Follow these steps:

1. Insert the media device.
2. Become the `root` user.
3. Create a mount point, if necessary.
4. Determine the file system type.
5. Mount the device by using the mount options listed in the following sections.
6. Work with files on the media device.
7. Unmount the media device.
8. Eject the media device.
9. Exit the `root` session.

Using the `mount` Command

To mount a file system that resides on a DVD or CD-ROM when the Volume Management service is stopped, as the `root` user, perform the command:

```
# mount -F hsfs -o ro /dev/dsk/c0t2d0s2 /cdrom
```

In this example, the file system type is `hsfs`, the file system resides on CD-ROM slice `/dev/dsk/`c0t2d0s2, and the mount point `/cdrom` is an existing directory in the Solaris OS.

Note – Different systems use different names for their DVD or CD-ROM devices. The example above is from a Sun Blade 1500.

To mount a file system that resides on a diskette when the Volume Management service is stopped, as the `root` user, perform the command:

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

In this example, the file system type is `pcfs`. This file system resides on the `/dev/diskette` device, and the mount point used is `/pcfs`.

Exercise: Mounting File Systems

In this exercise, you complete the following tasks:

- Create mount points
- Mount file systems
- Specify mount options

Preparation

This exercise requires a spare disk that contains 4 unmounted UFSs on slices 0, 1, 3, and 4 of the spare disk on your system. If you have to partition this disk for this exercise, you must understand how to use the `format` utility. If you need to make file systems on the slices for this exercise, you must understand how to use the `newfs` command.

This exercise applies to both SPARC and x86/x64 systems. These systems typically have different disk device names. For example:

- The spare disk of a Sun Blade 1500 (SPARC) is typically `c0t1d0`.
- The spare disk of an Ultra 20 (x86/x64) is typically `c2d0`.

Throughout this exercise, use the correct device names for your system.

Task

Complete the following steps:

1. Log in as the `root` user and open a terminal window.
2. Use the `prtvotc` command to determine if the required slices exist on your spare disk.
3. Use the `fstyp` command to determine if a ufs file system exists on the required slices on your spare disk.
4. If the spare disk is not divided into 4 slices as described in the "Preparation" section of this exercise, complete the following steps:
   a. On x86/x64 systems, use the `fdisk` menu in the `format` utility to create define a single Solaris `fdisk` partition that uses the entire disk. Make sure the partition is not marked as *active*.

b. Use the `partition` menu in the `format` utility to create the required slices.

c. Make slices 0, 1, and 3 exactly the same size (approximately 25 percent of the total disk space each), and use slice 4 for the remainder of the available space. Using the `All Free Hog` method and choosing slice 4 as the free hog slice makes this easy.

On SPARC systems, you can also use the Solaris Management Console to partition the disk.

d. Save your changes and quit the `format` utility when you are finished.

Example of a correct SPARC partition table:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)  56626320
  1       swap    wu   13879 - 27757       27.00GB    (13879/0/0)  56626320
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm   27758 - 41636       27.00GB    (13879/0/0)  56626320
  4 unassigned    wm   41637 - 57458       30.78GB    (15822/0/0)  64553760
  5 unassigned    wm       0                0         (0/0/0)             0
  6        usr    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
```

Example of a correct x86/x64 partition table:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       3 -  2483       19.01GB    (2481/0/0)   39857265
  1       swap    wu    2484 -  4964       19.01GB    (2481/0/0)   39857265
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wm    4965 -  7445       19.01GB    (2481/0/0)   39857265
  4 unassigned    wm    7446 - 10007       19.63GB    (2562/0/0)   41158530
  5 unassigned    wm       0                0         (0/0/0)             0
  6        usr    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130
```

5. If the 4 slices on the spare disk do not have UFSs on them as described in the "Preparation" section of this exercise, create the required file systems.

6. Use the `mount` command to list the file systems that are currently mounted on your system.

What are the default mount options applied to the `/` (root) file system?

____

____

7. Create a directory named `/morespace` to use as a mount point.
8. Complete the following steps:
   a. Mount the file system on slice 4 of your spare disk to the `/morespace` directory.
   b. Record the default mount options that were applied to this mount.

      ____

      ____

9. Change to the `/morespace` directory, and then create a new file that has one line of content.
10. Display a long listing for this file, and then record the time value it reports.

    This time value represents when the file was last modified.

    ____

11. Add the `-u` option to the `ls` command to show when the file was last accessed, and then record the time value it reports.

    This time value is updated whenever you read the file.

    ____

12. Wait one minute or more and complete the following steps:
    a. Use the `cat` command to display the file.
    b. Again, check and record the access time.

       The access time should differ from the access time indicated in the previous step.

       ____

13. Change to the `/` (root) directory and complete the following steps:
    a. Unmount the `/morespace` file system.
    b. Mount the same file system to the `/morespace` directory, but add the option that prevents update of access time values.
    c. Verify that the `noatime` option is now set for the `/morespace` mount.

14. Return to the `/morespace` file system and complete the following steps:

    a. Use the `cat` command to display your test file.

    b. Again, check and record the access time.

       The access time should match the access time that existed before you mounted the `/morespace` file system with the `noatime` option.

15. Use the `vi` editor to add a line to the `/etc/vfstab` file to make the mount for the `/morespace` file system happen when you boot the system.

16. Reboot your system, and complete the following steps:

    a. Log in as the `root` user.

    b. Open a terminal window.

    c. Use the `mount` command to verify that the `/morespace` file system is mounted, and the `noatime` option is set.

17. Create a directory called `/dir0` and mount the file system that resides on slice 0 of your spare disk as `/dir0`.

18. Create a directory called `/dir0/dir1` and mount the file system that resides on slice 1 of your spare disk as `/dir0/dir1`.

19. Open a second terminal window and change the directory in this second window to `/dir0/dir1`.

20. In your original terminal window, attempt to unmount the file system mounted below the `/dir0/dir1` directory.

    What message is displayed? Does the file system unmount?

Note – To discover why you could not unmount the file system, use the `fuser -cu /dir0/dir1` command. The `fuser` command should show the process ID of the shell currently using the filesystem.

21. In your original terminal window, add the `-f` option to the `umount` command, and attempt to unmount the file system mounted below the `/dir0/dir1` directory.

    Which message displays? Does the file system unmount?

22. In the second terminal window, complete the following steps:
    a. Attempt to determine your current working directory.
       Which message displays?
    b. Change the directory to / (root), and then verify that the pwd command works.

       ____

Note – Different shells display different messages. Users of the BASH shell see the correct directory.

---

23. Unmount the file systems mounted below /dir0 and /morespace.
24. Use the vi editor to remove the line from /etc/vfstab that mounts a file system below /morespace.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Mounting File Systems

This section contains solutions to the exercise.

Task

Complete the following steps:

1. Log in as the `root` user and open a terminal window.
2. Use the `prtvotc` command to determine if the required slices exist on your spare disk. For example:

   ```
   # prtvtoc /dev/rdsk/c2d0s2
   ```

3. Use the `fstyp` command to determine if a ufs file system exists on the required slices on your spare disk. For example:

   ```
   # fstyp /dev/dsk/c2d0s0
   ufs
   #
   ```

4. If the spare disk is not divided into four slices as described in the Preparation section of this exercise, complete the following steps:

   a. On x86/x64 systems, use the `fdisk` menu in the `format` utility to define a single Solaris `fdisk` partition that uses the entire disk. Make sure the partition is not marked as *active*.

   b. Use the `partition` menu in the `format` utility to create the required slices.

   c. Make slices 0, 1, and 3 exactly the same size (approximately 25 percent of the total disk space each), and use slice 4 for the remainder of the available space. Using the `All Free Hog` method and choosing slice 4 as the free hog slice makes this easy.

   On SPARC systems, you can also use the Solaris Management Console to partition the disk.

d. Save your changes and quit the `format` utility when you are finished.

Example of a correct SPARC partition table:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 13878       27.00GB    (13879/0/0)  56626320
  1       swap    wu   13879 - 27757       27.00GB    (13879/0/0)  56626320
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm   27758 - 41636       27.00GB    (13879/0/0)  56626320
  4 unassigned    wm   41637 - 57458       30.78GB    (15822/0/0)  64553760
  5 unassigned    wm       0                0         (0/0/0)             0
  6        usr    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
```

Example of a correct x86/x64 partition table:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       3 -  2483       19.01GB    (2481/0/0)   39857265
  1       swap    wu    2484 -  4964       19.01GB    (2481/0/0)   39857265
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wm    4965 -  7445       19.01GB    (2481/0/0)   39857265
  4 unassigned    wm    7446 - 10007       19.63GB    (2562/0/0)   41158530
  5 unassigned    wm       0                0         (0/0/0)             0
  6        usr    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130
```

5. If the four slices on the spare disk do not have `ufs` file systems on them as described in the Preparation section of this exercise, create the required file systems.

```
# newfs /dev/rdsk/c2d0s0
```

6. Use the `mount` command to list the file systems that are currently mounted on your system.

   What are the default mount options applied to the `/` (root) file system?

```
# mount
/ on /dev/dsk/c1d0s0
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev
=1980000 on Tue Feb  6 16:54:32 2007
(output omitted)
```

The mount options should match those in this example, and your `dev=` *number depends on the devices in use on your system.*

7. Create a directory named `/morespace` to use as a mount point.

```
# mkdir /morespace
```

8. Complete the following steps:
   a. Mount the file system on slice 4 of your spare disk to the `/morespace` directory.
   b. Record the default mount options that were applied to this mount.

```
# mount /dev/dsk/c2d0s4 /morespace
# mount
(output omitted)
/morespace on /dev/dsk/c2d0s4
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev
=1980044 on Sun Feb 11 17:20:24 2007
```

9. Change to the `/morespace` directory, and then create a new file that has one line of content.

```
# cd /morespace
# echo "Some Text" > testfile
#
```

10. Display a long listing for this file, and then record the time value it reports.

    This time value represents when the file was last modified.

```
# ls -l
total 18
drwx------   2 root     root        8192 Feb  8 17:42 lost+found
-rw-r--r--   1 root     root          10 Feb 11 17:21 testfile
#
```

11. Add the `-u` option to the `ls` command to show when the file was last accessed, and then record the time value it reports.

    This time value is updated whenever you read the file.

```
# ls -lu testfile
-rw-r--r--   1 root     root          10 Feb 11 17:21 testfile
#
```

12. Wait one minute or more and complete the following steps:
    a. Use the `cat` command to display the file.
    b. Again, check and record the access time.

    The access time should differ from the access time indicated in the previous step.

```
# cat testfile
Some Text
# ls -lu testfile
-rw-r--r--   1 root     root           10 Feb 11 17:23 testfile
#
```

13. Change to the `/` (root) directory and complete the following steps:
    a. Unmount the `/morespace` file system.
    b. Mount the same file system to the `/morespace` directory, but add the option that prevents update of access time values.
    c. Verify that the `noatime` option is now set for the `/morespace` mount.

```
# cd /
# umount /morespace
# mount -o noatime /dev/dsk/c2d0s4 /morespace
# mount
(output omitted)
/morespace on /dev/dsk/c2d0s4
read/write/setuid/devices/intr/largefiles/logging/xattr/noatime/onerror=p
anic/dev=1980044 on Sun Feb 11 17:24:50 2007
#
```

14. Return to the `/morespace` file system and complete the following steps:
    a. Use the `cat` command to display your test file.
    b. Again, check and record the access time.

    The access time should match the access time that existed before you mounted the `/morespace` file system with the `noatime` option.

```
# cd /morespace
# cat testfile
Some Text
# ls -lu testfile
-rw-r--r--   1 root     root           10 Feb 11 17:23 testfile
#
```

15. Use the vi editor to add a line to the /etc/vfstab file to make the mount for the /morespace file system happen when you boot the system.

```
/dev/dsk/c2d0s4   /dev/rdsk/c2d0s4  /morespace  ufs  1  yes  noatime
```

16. Reboot your system, and complete the following steps:
    a. Log in as the root user.
    b. Open a terminal window.
    c. Use the mount command to verify that the /morespace file system is mounted, and the noatime option is set.

```
# init 6
(reboot messages & login prompts display)
# mount
(output omitted)
/morespace on /dev/dsk/c2d0s4
read/write/setuid/devices/intr/largefiles/logging/xattr/noatime/onerror=p
anic/dev=1980044 on Sun Feb 11 17:32:12 2007
```

17. Create a directory called /dir0 and mount the file system that resides on slice 0 of your spare disk as /dir0.

```
# mkdir /dir0
# mount /dev/dsk/c2d0s0 /dir0
#
```

18. Create a directory called /dir0/dir1 and mount the file system that resides on slice 1 of your spare disk as /dir0/dir1.

```
# mkdir /dir0/dir1
# mount /dev/dsk/c2d0s1 /dir0/dir1
```

19. Open a second terminal window and change the directory in this second window to /dir0/dir1.

```
# cd /dir0/dir1
```

20. In your original terminal window, attempt to unmount the file system mounted below the /dir0/dir1 directory.

    What message is displayed? Does the file system unmount?

```
# umount /dev/dsk/c2d0s1
umount: /dir0/dir1 busy
# mount
(output omitted)
/dir0/dir1 on /dev/dsk/c2d0s1
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev
=1980041 on Sun Feb 11 17:46:38 2007
```

The file system does not unmount.

Note – To discover why you could not unmount the file system, use the `fuser -cu /dir0/dir1` command. The `fuser` command should show the process ID of the shell currently using the filesystem.

---

21. In your original terminal window, add the `-f` option to the `umount` command, and attempt to unmount the file system mounted below the `/dir0/dir1` directory.

    Which message displays? Does the file system unmount?

    ```
    # umount -f /dir0/dir1
    ```

    No message displays. The file system unmounts.

22. In the second terminal window, complete the following steps:

    a. Attempt to determine your current working directory.

       Which message displays?

    b. Change the directory to `/` (root), and then verify that the `pwd` command works.

    ```
    # pwd
    cannot determine current directory
    #
    ```

Note – Different shells display different messages. Users of the `BASH` shell see the correct directory.

    ```
    # cd /
    # pwd
    /
    ```

23. Unmount the file systems mounted below `/dir0` and `/morespace`.

    ```
    # umount /dir0
    # umount /morespace
    ```

24. Use the `vi` editor to remove the line from `/etc/vfstab` that mounts a file system below `/morespace`.

Notes:

Module 7

Performing Solaris 10 OS Package Administration

Objectives

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

- Describe the fundamentals of package administration
- Administer packages using the command-line interface

Introducing the Fundamentals of Package Administration

Software package administration adds software to systems and removes software from systems. Sun and its third-party vendors deliver software products to users in software packages.

Software Packages

The term package refers to the method of distributing software products and installing them in systems. In its simplest form, a package is a collection of files and directories.

Note – The Solaris OS software installation process installs all the required software packages automatically, based on the software group configuration choice.

Software packages contain:

- Files that describe the package and the amount of disk space required for installation
- Compressed software files to be installed on the system
- Optional scripts that run when the package is added or removed

The `/var/sadm/install/contents` File

The `/var/sadm/install/contents` file is a complete record of all the software packages installed on the local system disk. It references every file and directory belonging to every software package and shows the configuration of each product installed. To list the contents of the `/var/sadm/install/contents` file, perform the command:

```
# more /var/sadm/install/contents
/bin=./usr/bin s none SUNWcsr
/dev d none 0755 root sys SUNWcsr SUNWcsd
/dev/allkmem=../devices/pseudo/mm@0:allkmem s none SUNWcsd
/dev/arp=../devices/pseudo/arp@0:arp s none SUNWcsd
/dev/conslog=../devices/pseudo/log@0:conslog s none SUNWcsd
/dev/console=../devices/pseudo/cn@0:console s none SUNWcsd
/dev/dsk d none 0755 root sys SUNWcsd
(output omitted)
```

The pkgadd command updates the /var/sadm/install/contents file each time new packages are installed.

The pkgrm command uses the /var/sadm/install/contents file to determine where the files for a software package are located on the system. When a package is removed from the system, the pkgrm command updates the /var/sadm/install/contents file.

To determine if a particular file was installed on the system disk and to find the directory in which it is located, use the pkgchk command with either the full or partial path name of the command you want to report on. For example, to verify that the showrev command is installed on the system disk, perform the command:

```
# pkgchk -l -P showrev
Pathname: /usr/bin/showrev
Type: regular file
Expected mode: 0755
Expected owner: root
Expected group: sys
Expected file size (bytes): 29992
Expected sum(1) of contents: 59728
Expected last modification: Nov 08 23:02:51 2006
Referenced by the following packages:
        SUNWadmc
Current status: installed

Pathname: /usr/share/man/man1m/showrev.1m
Type: regular file
Expected mode: 0644
Expected owner: root
Expected group: root
Expected file size (bytes): 3835
Expected sum(1) of contents: 64380
Expected last modification: Sep 15 18:42:55 2006
Referenced by the following packages:
        SUNWman
Current status: installed
```

Package Formats

Solaris OS packages can be in one of two formats:

- File system (or Directory) format
- Data stream format

Packages delivered in file system format consist of multiple files and directories. Packages delivered in data stream format consist of a single file only.

File System Format

An example of a package (SUNWzsh) in file system format:

```
# ls -ld SUNWzsh
drwxr-xr-x   5 root     staff       2048 Nov 14 20:54 SUNWzsh
# cd SUNWzsh
# ls -l
total 3280
total 127
drwxr-xr-x   2 root     staff       2048 Nov 14 20:54 archive
drwxr-xr-x   2 root     staff       2048 Nov 14 20:54 install
-rw-r--r--   1 root     staff        455 Nov 14 15:56 pkginfo
-rw-r--r--   1 root     staff      58248 Nov 14 15:56 pkgmap
drwxr-xr-x   3 root     staff       2048 Nov 14 20:54 reloc
#
```

The package consists of a directory that matches the package name, and other files and directories including the pkginfo and pkgmap files.

Data Stream Format

An example of a package in data stream format:

```
# ls -l SUNWzsh.pkg
-rw-r--r--   1 root     root     1092608 Feb 15 14:56 SUNWzsh.pkg
# file SUNWzsh.pkg
SUNWzsh.pkg:    package datastream
# head SUNWzsh.pkg
# PaCkAgE DaTaStReAm
SUNWzsh 1 2110
# end of header
NAME=Z shell (zsh)
ARCH=sparc
VERSION=11.10.0,REV=2005.01.08.05.16
SUNW_PRODNAME=SunOS
SUNW_PRODVERS=5.10/SunOS Development
SUNW_PKGTYPE=usr
MAXINST=1000
#
```

Packages downloaded from the Internet are often in data stream format.

Administering Packages From the Command Line

From the command line, you can translate, add, remove, check the state of, and display information about packages.

The command-line tools are used for translating packages, viewing software, adding software, and removing software from a workstation after the Solaris OS software is installed on a system include:

pkgtrans	Translates packages from one format to another
pkgadd	Installs software packages to the system
pkgrm	Removes a package from the system
pkginfo	Displays software package information
pkgchk	Checks package installation state

Translating Package Formats

Use the `pkgtrans` command to translate a package from file system format to data stream format, or from data stream format to file system format. The command syntax for the `pkgtrans` command is:

```
# pkgtrans source_device destination_device [ package_name ...]
```

For example, to translate a package from file system format in `/var/tmp` to data stream format, use:

```
# pkgtrans /var/tmp /tmp/SUNWrsc.pkg SUNWrsc
Transferring <SUNWrsc> package instance
```

The first argument above specifies the source directory where the file system format package is stored. The second argument specifies the destination package data stream file. The third argument is the package to translate that exists in the source directory specified by the first argument.

If a package name is not given, the `pkgtrans` command provides a list of all packages in the source directory, and prompts the user for the packages to translate.

Note – Students need to insert the Solaris 10 OS Software DVD to demonstrate the steps described in this module.

Displaying Installed Software Packages Information

Use the pkginfo command to display information about the software packages installed on a local system disk. The /var/sadm/pkg directory maintains a record of installed packages. For example, to display information about software packages installed on a local system disk:

```
# pkginfo | more
(output omitted)

system      SUNWcti2     Netra ct I2C and System Drivers
ALE         SUNWctltk    Simplified Chinese ToolTalk Runtime Package
system      SUNWctlu     Print utilities for CTL locales
CTL         SUNWctpls    Portable layout services for Complex Text Layout support
ALE         SUNWcttf     Simplified Chinese (EUC) True Type Fonts
ALE         SUNWcttfe    Simplified Chinese (EUC) True Type Fonts Extension
application SUNWcudc     Simplified Chinese User Defined Character tool for Solaris CDE
ALE         SUNWcufnt    Simplified Chinese (UTF-8) X Windows Platform required Fonts
Package
system      SUNWcupdatemgru  Simplified Chinese Update Manager GUI
system      SUNWcvc       Network Console
system      SUNWcvcr      Network Console daemon
(output omitted)
```

The column on the left displays the package category, such as application, system, complex text layout (CTL), or alternate language environment (ALE). A CTL language is any language that stores text differently than it is displayed. An ALE is an alternate language, different from the basic Solaris OS languages.

The center column displays the software package name. If it begins with SUNW, it is a Sun Microsystems product. Otherwise, it represents a third-party package.

The column on the right displays a brief description of the software product.

Displaying Information for All Packages

To display all the available information about the software packages, use the pkginfo command with the -l option.

For example, to view additional information about each software package installed on the local systems hard drive, perform the command:

```
# pkginfo -l | more
(output omitted)
```

Displaying Information for a Specific Package

To display the information for a specific software package, specify its name on the command line. For example, to view the information for the SUNWman software package, enter:

```
# pkginfo -l SUNWman

   PKGINST:  SUNWman
      NAME:  On-Line Manual Pages
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  43.0,REV=75.0
   BASEDIR:  /usr
    VENDOR:  Sun Microsystems, Inc.
      DESC:  System Reference Manual Pages
    PSTAMP:  2004.12.21.15.30
  INSTDATE:  Jan 11 2007 13:07
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:    12402 installed pathnames
                  21 shared pathnames
                 190 directories
               67327 blocks used (approx)
#
```

The last line identifies the size of the package. The number of blocks used defines how much space is needed on the disk to install the package.

Note – A block is a 512-byte disk block.

To determine how many packages are currently installed on disk, perform the command:

```
# pkginfo | wc -l
     1720
```

Displaying Information for Software Packages

To view information about packages that are located on a SPARC platform-based Solaris 10 OS Software DVD, perform the command:

```
pkginfo -d /cdrom/cdrom0/Solaris_10/Product | more
```

To view information about packages that are located on an x86/x64 Solaris 10 OS Software DVD, perform the command:

```
pkginfo -d /cdrom/cdrom0/Solaris_10/Product | more
```

To view information about packages that are located on an x86/x64 Solaris 10 OS Software 1 CD-ROM, perform the command:

```
pkginfo -d /cdrom/cdrom0/Solaris_10/Product | more
```

The software groups located on a Solaris 10 OS Software 1 CD-ROM are Reduced Networking Core System Support and Core System Support.

To view information about packages that are located on any of the Solaris 10 Software CD-ROMs other than the Software 1 CD-ROM, perform the command:

```
pkginfo -d /cdrom/cdrom0/Solaris_10/Product | more
```

The software groups located on Solaris 10 OS Software 2, 3, and 4 CD-ROMs are the End User System Support, Developer System Support, Entire Distribution, and Entire Distribution Plus OEM Support software groups.

Adding a Software Package

When you add a software package, the pkgadd command copies the files from the installation media to the local system's disk and executes scripts to uncompress files. By default, the pkgadd command requires confirmation during the package add process.

For example, on a SPARC system, to transfer the SUNWvts software package from a DVD and install it on the system, perform the commands:

```
# cd /cdrom/cdrom0/Solaris_10/ExtraValue/CoBundled/SunVTS_6.3/Packages
# pkgadd -d . SUNWvts

Processing package instance <SUNWvts> from
</cdrom/sol_10_1106_sparc/s0/Solaris_10/ExtraValue/CoBundled/SunVTS_6.3/P
ackages>

SunVTS Framework(sparc) 6.3,REV=2006.10.19.18.34
Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Using </opt> as the package base directory.
## Processing package information.
```

```
## Processing system information.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

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

Do you want to continue with the installation of <SUNWvts> [y,n,?] y

Installing SunVTS Framework as <SUNWvts>

## Installing part 1 of 1.
8094 blocks

Installation of <SUNWvts> was successful.
#
```

On an x86/x64 system, you would find the SUNWvts package in the following directory on the Solaris OS DVD:

```
/cdrom/cdrom0/Solaris_10/ExtraValue/CoBundled/SunVTS_6.3/Packages
```

To install all packages in a data stream format package, perform the command:

```
# pkgadd -d /tmp/SUNWzsh.pkg all

Processing package instance <SUNWzsh> from </tmp/SUNWzsh.pkg>

Z shell (zsh)(sparc) 11.10.0,REV=2005.01.08.05.16

(output omitted)

Using </> as the package base directory.
## Processing package information.
## Processing system information.
   6 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 <SUNWzsh> [y,n,?] y

Installing Z shell (zsh) as <SUNWzsh>

## Installing part 1 of 1.
5481 blocks

Installation of <SUNWzsh> was successful.
#
```

Packages in data stream format can also be added from a web server. The following example commands require a web server running on the instructor system in the classroom, and the SPARC and x86/x64 versions of the SUNWzsh package stored in the specified locations.

Note – Before you try these examples, use the `pkgrm SUNWzsh` command to remove the `SUNWzsh` package that is installed on your system.

```
# pkgadd -d http://instructor/packages/sparc/SUNWzsh.pkg all

# pkgadd -d http://instructor/packages/x86/SUNWzsh.pkg all
```

SPARC and x86/x64 systems display installation messages similar to the following:

```
## Downloading...
..............25%..............50%..............75%..............100%
## Download Complete


Processing package instance <SUNWzsh> from
<http://instructor/packages/SUNWzsh.pkg>

Z shell (zsh)(sparc) 11.10.0,REV=2005.01.08.05.16
(output omitted)

Using </> as the package base directory.
## Processing package information.
## Processing system information.
   3 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
```

```
The following files are already installed on the system and are being
used by another package:
  /usr <attribute change only>
  /usr/sfw <attribute change only>
  /usr/sfw/lib <attribute change only>

Do you want to install these conflicting files [y,n,?,q] y
## 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 <SUNWzsh> [y,n,?] y

Installing Z shell (zsh) as <SUNWzsh>

## Installing part 1 of 1.
5481 blocks

Installation of <SUNWzsh> was successful.

#
```

Checking a Package Installation

The pkgchk command checks to determine if a package has been completely installed on the system. The pkgchk command also checks the path name, the file size and checksum, and the file attributes of a package. If the pkgchk command does not display a message, it indicates the package was installed successfully and that no changes have been made to any files or directories in the package.

The following example checks the contents and attributes of the SUNWladm software package currently installed on the system.

```
# pkgchk SUNWladm
#
```

To list the files contained in a software package, use the -v option.

For example, to list the files in the SUNWladm software package, perform the command:

```
# pkgchk -v SUNWladm
/usr
/usr/sadm
/usr/sadm/lib
/usr/sadm/lib/localeadm
/usr/sadm/lib/localeadm/Locale_config_S10.txt
/usr/sadm/lib/localeadm/admin
/usr/sadm/lib/localeadm/core_packages.txt
/usr/sadm/lib/localeadm/gen_configfile.pl
/usr/sadm/lib/localeadm/retrievecd.pl
/usr/sadm/lib/localeadm/retrievecombined.pl
/usr/sbin
/usr/sbin/localeadm
#
```

To determine if the contents or attributes of a file have changed from when it was installed, use the -p option.

For example, to check the /etc/shadow file, perform the command:

```
# pkgchk -p /etc/shadow
ERROR: /etc/shadow
    modtime <11/09/06 09:51:50 AM> expected <12/27/06 05:44:53 PM> actual
    file size <296> expected <344> actual
    file cksum <20180> expected <23909> actual
```

The differences in modtime, file size, and checksum indicate that the original /etc/shadow file has changed in size. For this particular file, this happened because of the initial Solaris OS software installation.

The -l option with the pkgchk command lists information about selected files that make up a package.

For example, to list information about the /usr/bin/showrev file, perform the command:

```
# pkgchk -l -p /usr/bin/showrev
Pathname: /usr/bin/showrev
Type: regular file
Expected mode: 0755
Expected owner: root
Expected group: sys
Expected file size (bytes): 29992
Expected sum(1) of contents: 59728
Expected last modification: Nov 08 23:02:51 2006
Referenced by the following packages:
        SUNWadmc
Current status: installed
```

If the -p option is used, the full path must be typed for the pkgchk command to return information about the file. If the -P option is used, a partial path name can be supplied.

For example, the pkgchk command returns no information if the /usr/bin/ path is omitted from the previous example.

```
# pkgchk -l -p showrev
#
```

Removing a Software Package

The pkgrm command removes a software package from the system and deletes all of the files associated with that package, unless other packages share those files.

By default, the pkgrm command requires confirmation to continue removing a package and issues a message to warn about possible package dependencies. If package dependencies do exist, the command again requires confirmation to continue with the package removal process.

The following command removes the `SUNWapchr` software package from the system.

Caution – Be cautious of the dependency warnings you receive when removing a package. The system allows you to remove these packages even though they may be required by a different package.

```
# pkgrm SUNWapchr

The following package is currently installed:
   SUNWapchr  Apache Web Server (root)
              (sparc) 11.10.0,REV=2005.01.08.05.16

Do you want to remove this package? [y,n,?,q] y

## Removing installed package instance <SUNWapchr>
## Verifying package <SUNWapchr> dependencies in global zone
WARNING:
    The <SUNWipplr> package depends on the package
    currently being removed.
WARNING:
    The <SUNWapchu> package depends on the package
    currently being removed.
WARNING:
    The <SUNWapchd> package depends on the package
    currently being removed.
WARNING:
    The <SUNWserweb> package depends on the package
    currently being removed.
Dependency checking failed.

Do you want to continue with the removal of this package [y,n,?,q] y
## Processing package information.
## Removing pathnames in class <initd>
/etc/rcS.d/K16apache
/etc/rc3.d/S50apache

(output omitted)

/etc/apache <shared pathname not removed>
/etc <shared pathname not removed>
## Updating system information.

Removal of <SUNWapchr> was successful.
```

Note – A file shared by two or more packages displays the message *filename* <shared pathname not removed>.

Adding Packages by Using a Spool Directory

For convenience, you can copy frequently installed software packages from the Solaris 10 Software DVD or Solaris 10 Software CD-ROMs to a spool directory on the system.

The default installation directory for packages that have been spooled, but not installed, is /var/spool/pkg. The pkgadd command, by default, looks in the /var/spool/pkg directory for any packages specified on the command line.

Copying packages from the CD-ROM or DVD into a spool directory is not the same as installing the packages on disk.

On a SPARC system, to copy a package from the Solaris 10 OS Software DVD into the /var/spool/pkg directory, perform the command:

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product -s spool SUNWauda
Transferring <SUNWauda> package instance
```

On an x86/x64 system, to copy a package from the Solaris 10 OS Software DVD into the /var/spool/pkg directory, perform the command:

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product -s spool SUNWauda
Transferring <SUNWauda> package instance
```

The -s option with the keyword spool copies the package into the /var/spool/pkg directory by default.

To verify that the package exists in the spool directory, perform the command:

```
# ls -al /var/spool/pkg
total 6
drwxrwxrwt   3 root     bin            512 Feb 15 22:19 .
drwxr-xr-x  12 root     bin            512 Feb  6 17:26 ..
drwxr-xr-x   5 root     root           512 Feb 15 22:19 SUNWauda
```

To add the package from the spool area, perform the following:

```
# pkgadd SUNWauda
(output omitted)
```

To remove software packages from a spool directory, use the pkgrm command with the -s option.

```
# pkgrm -s spool SUNWauda

The following package is currently spooled:
   SUNWauda  Audio Applications
             (sparc) 11.10.0,REV=2005.01.21.15.53

Do you want to remove this package? [y,n,?,q] y

Removing spooled package instance <SUNWauda>
```

If alternative spooling directories exist, specify the directory to use by substituting the directory path for the spool keyword.

For example, to select the /export/pkg directory, perform the commands:

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product -s /export/pkg SUNWauda

# pkgrm -s /export/pkg SUNWauda
```


Streaming One or More Packages

Packages can be individually or collectively packaged into a data stream file format. The data stream file can then be made available as a shared network file or from a web page.

Worked Example

To create a data streamed package, perform the following commands:

On a SPARC system:

```
# cd /cdrom/cdrom0/s0/Solaris_10
```

On an x86/x64 system:

```
# cd /cdrom/cdrom0/Solaris_10
```

On both SPARC and x86/x64 systems:

```
# pkgtrans -s  Product  /var/tmp/stream.pkg SUNWzlib SUNWftpr SUNWftpu
Transferring <SUNWzlib> package instance
Transferring <SUNWftpr> package instance
Transferring <SUNWftpu> package instance

# file /var/tmp/stream.pkg
/var/tmp/stream.pkg:    package datastream

# head -5 /var/tmp/stream.pkg
# PaCkAgE DaTaStReAm
SUNWzlib 1 186
SUNWftpr 1 60
SUNWftpu 1 300
# end of header

# pkgadd -d /var/tmp/stream.pkg

The following packages are available:
  1  SUNWftpr     FTP Server, (Root)
                  (sparc) 11.10.0,REV=2005.01.21.15.53
  2  SUNWftpu     FTP Server, (Usr)
                  (sparc) 11.10.0,REV=2005.01.21.15.53
  3  SUNWzlib     The Zip compression library
                  (sparc) 11.10.0,REV=2005.01.08.05.16

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: q
```

Reviewing Package Administration

This section details the package administration tasks.

Table 7-1 summarizes the commands used for package administration.

Table 7-1 Package Administration Commands

Command Name	Description
pkginfo	Lists packages installed on the system or available on distribution media
pkgadd	Installs packages
pkgrm	Removes packages
pkgchk	Verifies the attributes of the path names that belong to packages
pkgtrans	Translates a package from one format to another, such as a file system format to a datastream, or one file system format to another file system format.

Table 7-2 summarizes the files and directories used in package administration.

Table 7-2 Package Administration Files and Directories

File or Directory	Description
/var/sadm/install/contents	A software package map of the entire system
/opt/*pkgname*	The preferred location for the installation of unbundled packages
/opt/*pkgname*/bin or /opt/bin	The preferred location for the executable files of unbundled packages
/var/opt/*pkgname* or /etc/opt/*pkgname*	The preferred location for log files of unbundled packages

Exercise 1: Adding a Software Package

In this exercise, you use the package-related commands to add software packages to the Solaris 10 OS.

Preparation

This exercise requires the packages found in the /opt/ses/lab/packages directory on student systems. If the packages are located on a server, your instructor will provide its location and information about how to retrieve them.

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

Note – Use the packages in the /opt/ses/lab/packages directory to understand how to use package administration commands. The packages used for this exercise are for demonstration only.

Task

Complete the following steps:

1. List the packages found in the /opt/ses/lab/packages directory.
2. Display a long-format listing of the information for the packages.
3. Add the packages from the /opt/ses/lab/packages directory. Respond y to questions asked by the pkgadd command.
4. Use pkginfo to list information about the installed packages.
5. Use pkgchk to verify that the packages were installed successfully.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise 1 Solutions: Adding a Software Package

In this exercise, you use the package-related commands to add a software packages to the Solaris 10 OS.

Preparation

This exercise requires the packages found in the `/opt/ses/lab/packages` directory on student systems. If the packages are located on a server, your instructor will provide its location and information about how to retrieve them.

Note – Use the packages in the `/opt/ses/lab/packages` directory to understand how to use package administration commands. The packages used for this exercise are for demonstration only.

Task

Complete the following steps:

1. List the packages found in the `/opt/ses/lab/packages` directory.

   ```
   # ls /opt/ses/lab/packages
   SUNWoptdir SUNWusrdir
   ```

2. Display a long-format listing of the information for the packages.

   ```
   # pkginfo -d /opt/ses/lab/packages -l SUNWoptdir
   # pkginfo -d /opt/ses/lab/packages -l SUNWusrdir
   ```

3. Add the packages from the `/opt/ses/lab/packages` directory. Respond `y` to questions asked by the `pkgadd` command.

   ```
   # pkgadd -d /opt/ses/lab/packages SUNWoptdir
   # pkgadd -d /opt/ses/lab/packages SUNWusrdir
   ```

4. Use `pkginfo` to list information about the installed packages.

   ```
   # pkginfo SUNWoptdir
   # pkginfo SUNWusrdir
   ```

5. Use `pkgchk` to verify that the packages were installed successfully.

   ```
   # pkgchk -v SUNWoptdir
   # pkgchk -v SUNWusrdir
   ```

Exercise 2: Manipulating Software Packages

In this exercise, you use package-related commands to remove, install, and spool packages.

Preparation

This exercise requires the Solaris 10 OS Software DVD appropriate for your system architecture. Your instructor will provide these items.

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

Note – If you are using CD-ROM media, the `SUNWman` and `SUNWdoc` packages are located on the CD 4 of 4 and CD 2 of 4 Solaris 10 OS Software CD-ROMs, respectively.

Task

Complete the following steps:

1. Insert the Solaris 10 Software SPARC or x86/x64 DVD into the drive.
2. Use the `pkginfo` command to search for packages currently on your system that are related to manuals.

   Which packages were listed?

   ____

   ____

   ____

   ____

Note – The list that displays depends on selections that were made during the installation process.

---

3. Display a long-format listing of the information for the `SUNWman` package installed on your system.

   What is listed for the following:

   - The status of this package?

     ____

   - The install date of this package?

     ____

   - The number of files used by this package?

     ____

   - The number of blocks used by this package?

     ____

Note – These values may differ among different releases of Solaris 10. Values will be the same for SPARC and x86/x64 systems for any given release.

---

4. Display a long-format listing of the information for the `SUNWman` package on the Solaris 10 OS Software DVD.

   Obtain the same information as in the previous step.

   - The status of this package?

     ____

   - The install date of this package?

     ____

   - The number of files used by this package?

     ____

   - The number of blocks used by this package?

     ____

5. Remove the `SUNWman` package from your system and complete the following steps:

   a. Respond `y` to questions asked by the `pkgrm` command.

   b. Verify that `SUNWman` has been removed by using the `pkginfo` command, and trying to access the manual pages.

6. Reinstall the SUNWman package from the Solaris 10 OS Software DVD and complete the following steps:
   a. Respond y to questions asked by the pkgadd command.
   b. Use pkginfo to list information about the SUNWman package.
   c. Verify that the manual pages work.
7. Check if the SUNWdoc package is installed on your system, and complete the following steps:
   a. Remove the SUNWdoc package from your system.
   b. Respond y to questions asked by the pkgrm command.

   Are there any package dependencies related to removing this package?

   ____
8. Use the pkgadd command to spool the SUNWdoc package into the default spool area.
9. Use the pkginfo command (with the appropriate options) to verify the presence of the SUNWdoc package in the default spool area.
10. Install the SUNWdoc package and complete the following steps:
    a. Observe the messages.
    b. Respond y to questions asked by the pkgadd command.
    c. Verify that the package was installed from the /var/spool/pkg directory.
11. Remove the SUNWdoc package from the default spool area.
12. Verify that the SUNWdoc package no longer exists in the spool area and that it is installed on your system.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise 2 Solutions: Manipulating Software Packages

In this exercise, you use package-related commands to remove, install, and spool packages.

Preparation

This exercise requires the Solaris 10 OS Software DVD appropriate for your system architecture. Your instructor will provide these items.

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

Note – If you are using CD-ROM media, the `SUNWman` and `SUNWdoc` packages are located on the CD 4 of 4 and CD 2 of 4 Solaris 10 OS Software CD-ROMs, respectively.

Task

Complete the following steps:

1. Insert the Solaris 10 Software SPARC or x86/x64 DVD into the drive.
2. Use the `pkginfo` command to search for packages currently on your system that are related to manuals.

```
# pkginfo | grep -i manual
Which packages were listed?
system      SUNWcrman                           Encryption Kit On-Line
Manual Pages
system      SUNWman                             On-Line Manual Pages
system      SUNWmfman                           CDE Motif Manuals
system      SUNWopenssl-man                     OpenSSL Manual Pages
system      SUNWperl584man                      Perl 5.8.4 Reference Manual
Pages
system      SUNWpl5m                            Perl 5.6.1 Reference Manual
Pages
system      SUNWtltkm                           ToolTalk manual pages
```

Note – The list that displays depends on selections that were made during the installation process.

3. Display a long-format listing of the information for the `SUNWman` package installed on your system.

```
# pkginfo -l SUNWman
```

What is listed for the following:

- The status of this package?
- The install date of this package?
- The number of files used by this package?
- The number of blocks used by this package?

Status:	*Completely installed*
Install date:	*Should match the date and time when you installed Solaris OS on your system*
Number of files:	*12402 installed path names, 21 shared directories, 190 directories*
Number of blocks:	*67327*

Note – These values may differ among different releases of Solaris 10. Values will be the same for SPARC and x86/x64 systems for any given release. The values listed are examples only.

4. Display a long-format listing of the information for the `SUNWman` package on the Solaris 10 OS Software DVD.

   Obtain the same information as in the previous step.

   For example:

```
# pkginfo -d /cdrom/cdrom0/Solaris_10/Product -l SUNWman
```

Status:	*Spooled*
Install date:	*No install date indicated*
Number of files:	*12406 spooled path names, 190 directories, 4 package information files*
Number of blocks:	*67327*

5. Remove the `SUNWman` package from your system and complete the following steps:

   a. Respond `y` to questions asked by the `pkgrm` command.

   b. Verify that `SUNWman` has been removed by using the `pkginfo` command, and trying to access the manual pages.

```
# pkgrm SUNWman
<output omitted>
# pkginfo SUNWman
ERROR: information for "SUNWman" was not found
# man ls
No manual entry for ls.
#
```

6. Reinstall the `SUNWman` package from the Solaris 10 OS Software DVD and complete the following steps:
   a. Respond `y` to questions asked by the `pkgadd` command.
   b. Use `pkginfo` to list information about the SUNWman package.
   c. Verify that the manual pages work.

   For example:

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product SUNWman
(output omitted)
# pkginfo SUNWman
system      SUNWman On-Line Manual Pages
# man ls
<output omitted>
```

The manual page for `ls` *displays.*

7. Check if the `SUNWdoc` package is installed on your system, and complete the following steps:
   a. Remove the `SUNWdoc` package from your system.
   b. Respond `y` to questions asked by the `pkgrm` command.

```
# pkginfo SUNWdoc
system      SUNWdoc Documentation Tools
# pkgrm SUNWdoc

The following package is currently installed:
   SUNWdoc  Documentation Tools
            (sparc) 11.10.0,REV=2005.01.21.15.53

Do you want to remove this package? [y,n,?,q] y

## Removing installed package instance <SUNWdoc>

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

Do you want to continue with the removal of this package [y,n,?,q] y
## Verifying package <SUNWdoc> dependencies in global zone
WARNING:
    The <SUNWuium> package depends on the package currently
    being removed.
WARNING:
    The <SUNWman> package depends on the package currently
    being removed.
WARNING:
    The <SUNWpmowm> package depends on the package
```

```
    currently being removed.
WARNING:
    The <SUNWopenssl-man> package depends on the package
    currently being removed.
WARNING:
    The <SUNWperl584man> package depends on the package
    currently being removed.
WARNING:
    The <SUNWpl5m> package depends on the package currently
    being removed.
WARNING:
    The <SUNWvtsmn> package depends on the package
    currently being removed.
Dependency checking failed.

Do you want to continue with the removal of this package [y,n,?,q] y
## Processing package information.
## Executing preremove script.
## Removing pathnames in class <none>
/usr/share/man <shared pathname not removed>
/usr/share/lib/tmac/vgrind

(output omitted)

/usr/bin/apropos
/usr/bin/addbib
/usr/bin <shared pathname not removed>
/usr <shared pathname not removed>
## Updating system information.

Removal of <SUNWdoc> was successful.
```

Are there any package dependencies related to removing this package?

Yes, there are. There are seven other packages dependent on the SUNWdoc *package.*

8. Use the pkgadd command to spool the SUNWdoc package into the default spool area.

   For example:

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product -s spool SUNWdoc
```

9. Use the pkginfo command (with the appropriate options) to verify the presence of the SUNWdoc package in the default spool area.

```
# pkginfo -d spool SUNWdoc
system  SUNWdoc   Documentation Tools
# pkginfo -d /var/spool/pkg -l SUNWdoc
PKGINST:  SUNWdoc
(output omitted)
```

10. Install the SUNWdoc package and complete the following steps:

    a. Observe the messages.

    b. Respond y to questions asked by the pkgadd command.

    c. Verify that the package was installed from the /var/spool/pkg directory.

```
# pkgadd SUNWdoc

Processing package instance <SUNWdoc> from </var/spool/pkg>
(output omitted)
```

11. Remove the SUNWdoc package from the default spool area.

```
# pkgrm -s spool SUNWdoc

The following package is currently spooled:
   SUNWdoc  Documentation Tools
            (sparc) 11.10.0,REV=2005.01.21.15.53

Do you want to remove this package? [y,n,?,q] y

Removing spooled package instance <SUNWdoc>
#
```

12. Verify that the SUNWdoc package no longer exists in the spool area and that it is installed on your system.

```
# pkginfo -d spool SUNWdoc
ERROR: information for "SUNWdoc" was not found
# pkginfo -l SUNWdoc
PKGINST:  SUNWdoc
(output omitted)
```

13. Eject the Solaris 10 OS Software DVD.

Notes:

Module 8

Managing Software Patches on the Solaris 10 OS

Objectives

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

- Describe patch administration fundamentals
- Describe patch administration best practices
- Obtain patches using SunSolve
- Install and remove patches
- Install recommended patch clusters
- Implement patch management using the `smpatch` command

Patch Administration Preparation

Patch administration involves installing or removing Solaris OS patches from a running Solaris OS. Please read the patch README file before installing any patch.

For information on patching with Live Upgrade, see the "System Installation" module.

Introducing Solaris OS Patches

A patch contains a collection of files and directories. This collection replaces existing files and directories that prevent proper software execution. Some patches contain product enhancements. Patches are also described as *software updates*.

Solaris OS patch types include:

- Standard patches – Patches that fix specific problems with the Solaris OS and other Sun hardware and software products.
- Recommended patches – Sun Customer Service designates operating system patches that are of universal interest or reflect security concerns to be "recommended" and "security" patches, respectively.
- Firmware and PROM patches.
- Patch clusters – A group of standard, recommended, security, or Y2K patches that have been bundled into a single archive for easy downloading and installation.

A patch is distributed as a directory that is named using a unique number. The number assigned to a patch includes the patch base code first, a hyphen, and a number that represents the patch revision number.

For example, a patch directory named `105050-01`, indicates that `105050` is the patch number and `01` is the revision number.

The Solaris 10 OS unsigned patches are delivered in `zip` format, for example, `105050-01.zip`.

Signed patches are delivered as Java Archive format files, for example, `105050-01.jar`.

A signed patch is a patch that is signed with a valid digital signature. A signed patch offers greater security than an unsigned patch. The digital signature of the patch can be verified before the patch is applied to your system. A valid digital signature ensures that the signed patch has not been modified since the signature was applied.

To add a patch with a digital signature to your system, you must set-up a keystore. A keystore is a repository of certificates and keys that is queried when you apply a signed patch.

For more information about signed patches and setting up keystores, see the Managing Solaris Patches and Updates (Overview) section of the *Solaris 10 System Administration Guide: Basic Administration* found on `http://docs.sun.com/app/docs/doc/817-1985/6mhm8o61l`.

Note – Not all patches available from Sun Microsystems must be installed. Review the README documents for each patch, and then decide if you should install the patch.

Best Practices

Following are some of the patching related best practices:

- Always install the latest patch and package utility patches first.
- Upgrade to the latest Solaris 10 Update release during your next major maintenance effort. Each Solaris Update is intensely tested and provides a quality baseline.
- Between major maintenance efforts, stay current with the Sun Alert patch cluster contents.
- Use Solaris Live Upgrade to patch or upgrade an inactive boot environment.
  - Live Upgrade enables you to avoid much of the risk and downtime associated with patching the live boot environment.
  - Live Upgrade provides a simple roll-back mechanism.

Accessing Patches and Patch Information

Use SunSolve to access software updates (patches). SunSolve provides direct access to a variety of support resources, including patches and updates, the Sun Knowledge Database, support forums, the Sun System Handbook, support cases, Sun Alerts, and Sun Security Resources.

Discontinued Patch Access Methods

Formerly, Sun customers were able to gain access to patches through the `sunsolve.sun.com` anonymous ftp server. In September of 2006, Sun ceased providing access to this anonymous ftp server for the purpose of downloading patches. These changes to the methods used to access patches are described in two infodoc articles, 82023 and 83061, available using these two URLs:

```
http://sunsolve.sun.com/search/document.do?assetkey=1-9-82023-1
http://sunsolve.sun.com/search/document.do?assetkey=1-9-83061-1
```

Access to patches through PatchPro Interactive has also been discontinued.

Requirements for Patch Access

Currently, customers are required to have a Sun Online account to obtain patches. This requires creating a Sun Online Account, and accepting the new click through Software Agreement.

You already have a Sun Online Account if you have an account on MySun, Sun Store, SunSolve, the Online Support Center, or Sun Download Center. Use the same user name and password during registration and to log in to SunSolve web site.

A Sun Service Plan is not required to access security patches and device drivers; however, you do need to have a Sun Online account and accept the Software License Agreement applicable to all Sun patches to have access to such patches.

Restricted patches require a Sun Service Plan or Solaris Subscription to gain access. Restricted patches include recommended patch clusters. Service plan descriptions are available using this URL:

`http://www.sun.com/service/serviceplans/index.jsp`

Access to Solaris 8 and 9 patches (except for security patches and device drivers) will also require a Solaris Subscription or Sun System Service Plan to gain access.

Accessing Patches Using SunSolve

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

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

The comprehensive set of patches and patch information is available to contract customers when they log in through these sites.

You can register to create a Sun Online Account by clicking the Register link found on any of these sites, reading and accepting the Software License Agreement, and providing the information requested on the SunSolve Online Registration page. Registering to create a Sun Online Account is all that is required to obtain access to security patches and driver updates.

You can provide your Sun Service Plan number at the time that you register to create a Sun Online Account. If you have a Sun Online account without a registered Service Plan, after you log in, you can use the Edit Account link to submit a request to update your account with your Service Plan contract number.

Accessing Patch Documents

Before you install patches on your system, you should review the related patch documents available on the SunSolve site. When reviewing patch documentation, start with the Patch Report document. This report is divided into categories that include information about all patches for a specific Solaris OS release. The Patch Report helps you identify patches to choose to install.

Individual patches and patch clusters have associated `README` files that provide specific information about patches and patch dependencies. Review these `README` files to learn more about what patches and patch clusters require, and what they provide. These documents are available to you through SunSolve whether or not you log in to the site.

Accessing a Patch Report

On the SunSolve site, you can access Patch Reports from the Patches and Updates page, under the section titled Patch Documents and Articles.

Click on `--Select to view report--` located under Patch Reports, and make your selection for the Solaris OS release and system architecture that you wish to view.

```
-- Select to view report --
Solaris 10
Solaris 10_x86
Solaris 9
Solaris 9_x86
Solaris 8
Solaris 7
Solaris 2.6
Solaris 2.5.1
Solaris 8_x86
Solaris 7_x86
Solaris 2.6_x86
Solaris 2.5.1_x86
----------------
Archived Patch Reports
Archived Patch Reports
2.5
2.5_x86
2.4
2.4_x86
1.1.2
```

The Patch Report describe different categories of patches that are available. For example, the Solaris 10 Patch Report Update from October 1, 2009, lists the following patches in the Quick Reference Section:

```
New Patches Released Since Last Report:
-------------------------------------------------------------------------
124581-01   commcli 7.0-1.00: core patch
139620-01   CDE 1.6: Dthelp patch
142288-01   SunOS 5.10: su patch
142371-01   Sun Studio 12.1 Update 1: Patch for dbx
142702-01   Hardware/PROM: Sun Fire V480 / V490 Flash PROM Update
142703-01   Hardware/PROM: Sun Ultra 45 Flash PROM Update
142704-01   Hardware/PROM: Sun Ultra 25 Flash PROM Update
142705-01   Hardware/PROM: Sun Fire V125 Flash PROM Update
142706-01   Hardware/PROM: Sun Fire V880 / V890 Flash PROM Update
142840-01   Logical Domains (LDoms) Manager 1.2 ldmd patch

Updated Revs Released Since Last Report:
-------------------------------------------------------------------------


118245-21   comm_dssetup 6.4-5.05: core patch
119169-32   Sun Java System App Server Enterprise Ed 8.1 2005Q1 Solaris
Patch28   : File based patch
119173-32   Sun Java System App Server Platform Ed 8.1 2005Q1 Solaris
Patch28   : File based patch
119963-18   SunOS 5.10: Shared library patch for C++
120094-27   X11 6.6.2: xscreensaver patch
120460-16   GNOME 2.6.0: Gnome libs Patch
121657-40   Calendar Server SunOS 5.9 5.10: Core patch
(output omitted)
```

Note – Patch Reports are published at the beginning and middle of each month.

Accessing Information about Specific Patches

The patch numbers listed in the online Patch Report are links to the pages that describe their respective patches. For example, to view information for the `su` patch listed above, you would click the `142288-01` link.

Access the same information by using the PatchFinder link located at the top of the Patches and Updates page. You click on the PatchFinder link to open the Software Update Finder page, enter the number of the patch you wish to find, and click on the Search button.

Note – Omitting the patch revision number causes the Software Update Finder to display information for the latest revision of the patch you specify. Including the revision number causes the Software Update Finder to display information for that specific patch and revision.

Latest Patch Utility Patches

You may also find patches using the Latest Patch Utility Patches pulldown menu located directly under the PatchFinder link on the Patches and Updates page to access information about patches.

Click on `--Select SunOS and Patch Number--` from the pulldown menu and you make your selection for the Solaris OS release, system architecture, and patch you wish to view.

```
-- Select SunOS and Patch Number --
SunOS 5.10: 138253
SunOS 5.10: 128342
SunOS 5.10: 127884
SunOS 5.10: 125555
SunOS 5.10: 122034
SunOS 5.10: 121296
SunOS 5.10: 119317
SunOS 5.10: 119254

SunOS 5.10x86: 138254
SunOS 5.10x86: 128343
SunOS 5.10x86: 127885
SunOS 5.10x86: 125556
SunOS 5.10x86: 122035
SunOS 5.10x86: 121297
SunOS 5.10x86: 119318
SunOS 5.10x86: 119255
(output omitted)
```

Selection of the latest patch utility patches is dependent on the Solaris OS release and system architecture that you wish to view. For example if you have a Solaris 10 SPARC system, you will want to select patch 119254. If you have a Sailors 10 x86 system, you will want to choose patch 119255.

Note – You will need to download and apply any other patches listed for Solaris 10. All the patches shown are required to avoid issues that can impact correct patch application.

For example if you have a Solaris 10 SPARC system you can click on patch `119254`. The README file for the patch utilities patch is displayed. Read through this file and give special attention to any installation requirements, required patches or dependencies, patch installation instructions, and special install instructions.

After you've read through the README file, you are ready to download the patch.

To download the patch, simply click on the `HTTP` link to the right of Download Patch.

```
Download Patch (1663600 bytes): HTTP
```

Notice that you have the option of downloading a signed version of the patch.

```
Download Signed Patch (1664612 bytes): HTTP
```

A signed patch is one that has a digital signature applied to it. A patch that has its digital signature verified has not been modified since the signature was applied. The digital signature of a signed patch is verified after the patch is downloaded to your system.

After you click on the `HTTP` link, the download process begins.

You'll need to repeat this process for each of the patches listed for Solaris 10 in the Latest Patch Utility Patches area.

Note – Just so you are aware, the Patch Utilities Patch is also contained in the Recommended and Sun Alert patch clusters; however, you have to have a valid support contract to download either of these patch clusters, located on the Patch Cluster and Patch Bundle Downloads Page.

Downloading Patches Using SunSolve

To download any patch from SunSolve you must login using a Sun Online account. Figure 8-1 shows the Sun Online Account login screen.

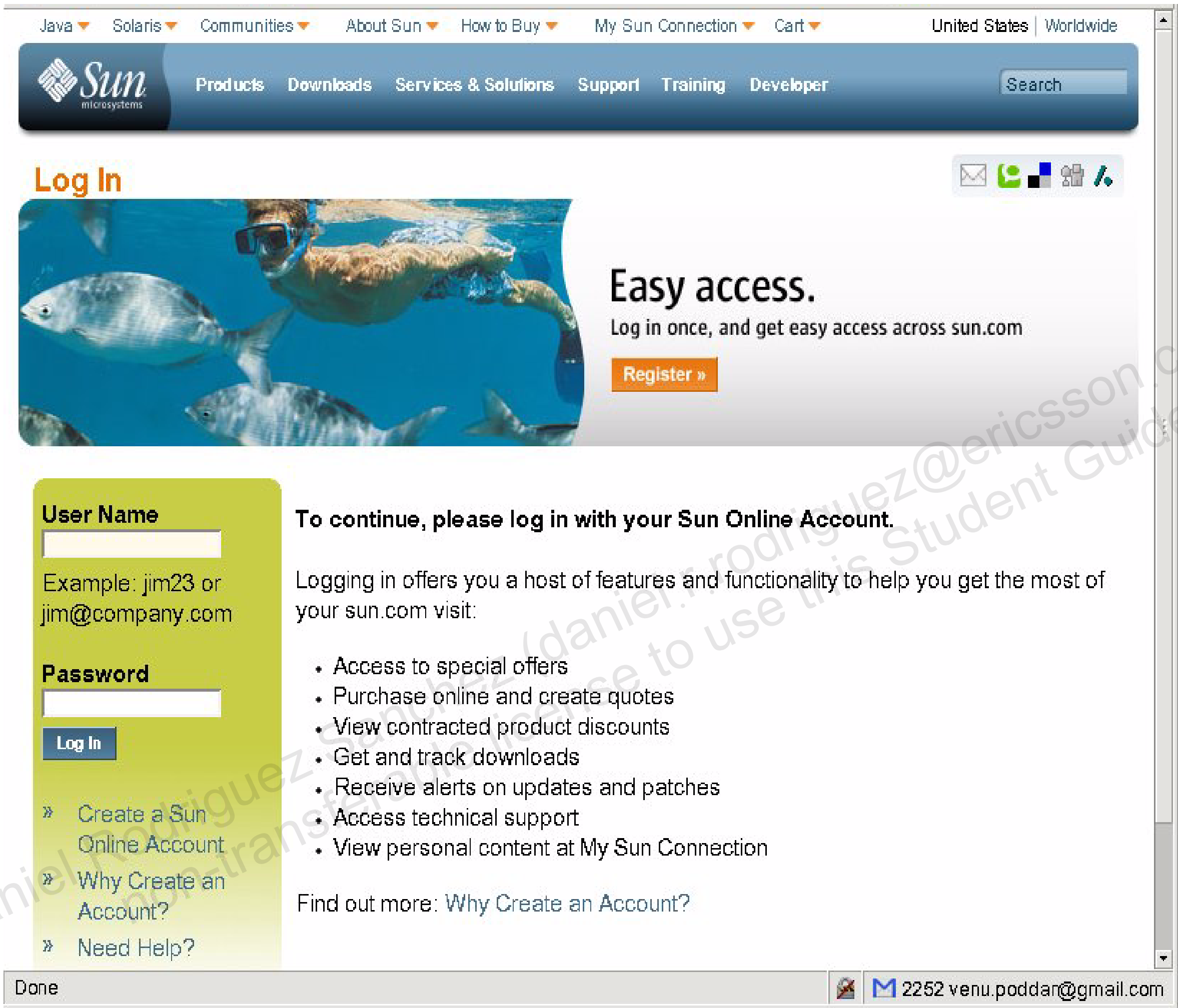


Figure 8-1 The Sun Online Account Login Screen

Patch Cluster and Patch Bundle Downloads

Patch Clusters and Patch Bundles are popular collections of patches providing specific functionality. They contain an install script for ease of installation. A Sun support contract which covers Solaris is required to access or download a Patch Cluster or Patch Bundle containing Solaris OS patches, and they may only be installed and used on systems covered by an active Sun support contract for Solaris.

Note – The Patch Clusters and Patch Bundle Downloads page replaces both the Recommended Patch Cluster and Recommended and Security Patches pages.

Once you have logged into SunSolve and the Patches and Updates page appears, scroll down to the Downloads section and locate the Patch Cluster and Patch Bundle Downloads link. Click on this link to access this page.

The Patch Cluster and Patch Bundle Downloads page contains all the patch clusters.

The Solaris Patch Clusters include:

- Recommended Patch Clusters

  The Solaris Recommended Patch Clusters contain Solaris Operating System patches that include the latest revision of the patch and package utility patches to help ensure correct patching operations. The latest revision of any Solaris OS patch which address Sun Alert issues. These are patches which fix Security, Data Corruption, or System Availability issues. And any patch that is required to correctly install the above patches.

- Sun Alert Patch Clusters

  The Solaris Sun Alert Patch Clusters contain Solaris Operating System patches that include the latest revision of the patch and package utility patches to help ensure correct patching operations. The minimum revision of any Solaris Operating System patch which address Sun Alert issues. These are patches which fix Security, Data Corruption, or System Availability issues. And any patch that is required to correctly install the above patches. The clusters are updated whenever a patch meeting the above inclusion criteria is released.

- Solaris Update Patch Bundles

These Solaris Patch Bundles contain the equivalent set of patches that are pre-applied to the corresponding Solaris Update image. They are provided for customers whose Change Control procedures make it difficult to upgrade but OK to patch. Applying these Patch Bundles is not equivalent to installing or upgrading to the corresponding Solaris Update as these patch bundles don't typically contain any new packages contained in the corresponding Solaris Update.

Note – After applying one of these Patch Bundles, you must apply the Sun Alert or Recommended Patch Cluster to obtain all available Solaris OS patches which address Security, Data Corruption, and System Availability issues released since these Patch Bundles were created.

- Solaris Live Upgrade Bundles

  The Live Upgrade Zones Starter Patch Bundles are provided specifically for use on systems with non-global zones that are running Solaris 10 8/07 (Update 4) or an earlier Solaris 10 Update.

Accessing Sun Alert Patch Clusters

On the Patch Cluster and Patch Bundle Download page, scroll down to the Solaris Patch Clusters and click on the Sun Alert Patch Clusters link to view the available patch clusters.

Patch Cluster Name	Cluster Size	View README	Download Cluster
Solaris 10 SPARC Sun Alert Patch Cluster	1103 MB	README	Download
Solaris 10 x86 Sun Alert Patch Cluster	838 MB	README	Download
Solaris 9 SPARC Sun Alert Patch Cluster	507 MB	README	Download
Solaris 9 x86 Sun Alert Patch Cluster	357 MB	README	Download
Solaris 8 Sun Alert Patch Cluster	277 MB	README	Download
Solaris 8 x86 Sun Alert Patch Cluster	158 MB	README	Download

Note – The **`Release Date`** field for each of the patch clusters listed above has been omitted from this example.

If you have a SPARC system, select the `Solaris 10 SPARC Sun Alert Patch Cluster`. After you select the cluster, you have the option to view the README file or to download the patch cluster. Click on README and review the file before you download the patch cluster.

The patch cluster README files have a different look from the individual patch README files. You will want to take notice of what patches are included in the cluster, the Important Notes and Warnings, and the install instructions.

When you have finished reading the patch cluster's README file, you will return to the Patch Cluster and Patch Bundle Download Page. At this point you can download the `Solaris 10 SPARC Sun Alert Patch Cluster`.

Note – In applying Sun Alert Patch Clusters be aware that any `reconfigimmediate` and `rebootimmediate` patches require a reboot of the system before normal operations can be resumed. If you use Solaris Live Upgrade to install a patch cluster, you will not need to reboot the system.

Checking Patch Levels

Before installing operating system patches, determine which patches are already installed on your system.

The `showrev` command and the `patchadd` command provide useful information about currently installed patches. The `patchadd -p` command can only be run as root. A normal user can use `showrev -p`. The command output is the same for `patchadd -p` and `showrev -p`. The `patchadd` command sometimes takes longer to display information.

```
# showrev -p
Patch: 118344-14 Obsoletes: 122397-01 Requires:  Incompatibles:
Packages: SUNWcsu, SUNWcsl, SUNWckr, SUNWhea, SUNWarc, SUNWfmd
Patch: 118368-04 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
. . .
# patchadd -p
Patch: 118344-14 Obsoletes: 122397-01 Requires: Incompatibles: Packages:
SUNWcsu SUNWcsl SUNWckr SUNWhea SUNWarc SUNWfmd
Patch: 118368-04 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
. . .
```

The `/var/sadm/patch` directory includes the following information:

- Historical information about patches currently installed on a system
- A listing of patches you can remove using the `patchrm` command

```
# ls /var/sadm/patch
107558-05  107594-04  107630-01  107663-01  107683-01 107696-01
107817-01  107582-01  107612-06  107640-03
```

Caution – Deleting files from the `/var/sadm` directory to make more space is a Solution Center call generator. The only way to correct the problems that occur is to restore the deleted files from backup tapes or to reload the Solaris OS.

Note – Allocated enough space for the `/var` file system. There must be sufficient space for the `/var/sadm` directory to grow as new software packages and patches are installed on the system.

Preparing Patches for Installation

When patches are downloaded to the local system, you must place the patches in a temporary directory to prepare them for installation. Use the /var/tmp directory.

One of the common reasons for patch installation failure is directory permission or ownership problems. The /var/tmp directory is open to all and eliminates any of these types of problems.

The Solaris 7, Solaris 8, Solaris 9, and Solaris 10 OS unsigned patches are in zip format, for example, 105050-01.zip.

Use the unzip command to unpack the patch files.

```
# /usr/bin/unzip 124998-01.zip
   creating: 124998-01/
  inflating: 124998-01/.diPatch
  inflating: 124998-01/patchinfo
   creating: 124998-01/SUNWcsu/
  inflating: 124998-01/SUNWcsu/pkgmap
  inflating: 124998-01/SUNWcsu/pkginfo
   creating: 124998-01/SUNWcsu/install/
  inflating: 124998-01/SUNWcsu/install/checkinstall
  inflating: 124998-01/SUNWcsu/install/copyright
  inflating: 124998-01/SUNWcsu/install/i.none
  inflating: 124998-01/SUNWcsu/install/patch_checkinstall
  inflating: 124998-01/SUNWcsu/install/patch_postinstall
  inflating: 124998-01/SUNWcsu/install/postinstall
  inflating: 124998-01/SUNWcsu/install/preinstall
   creating: 124998-01/SUNWcsu/reloc/
   creating: 124998-01/SUNWcsu/reloc/usr/
   creating: 124998-01/SUNWcsu/reloc/usr/bin/
  inflating: 124998-01/SUNWcsu/reloc/usr/bin/tip
  inflating: 124998-01/README.124998-01
  inflating: 124998-01/LEGAL_LICENSE.TXT
#
```

Earlier versions of the Solaris OS used patches in patches in compressed tar archive format, for example, 101010-01.tar.Z. For these patches, use the zcat command to uncompress the tar archive, and the tar command to extract the patch files from the tar archive.

```
# /usr/bin/zcat 101010-01.tar.Z | tar xvf -
```

Patch Contents

Figure 8-2 shows the contents of a patch directory after it is extracted from the zip file.

Figure 8-2 An Extracted Patch Directory

Installing and Removing Patches

The commands used to manage patches are:

- `patchadd` – Installs uncompressed patches to the Solaris OS
- `patchrm` – Removes patches installed on the Solaris OS
- `install_cluster` – Installs the patches delivered in patch clusters
- `smpatch` – Manages the update process on a single system

Installing a Patch

When you install a patch, the `patchadd` command calls the `pkgadd` command to install the package files that the patch delivers.

The following example shows installing a patch using the `patchadd` command. This example assumes that the patch to be installed exists in the `/var/tmp` directory and has been unzipped for installation.

Note – The patches are delivered as compressed tar files.

```
# cd /var/tmp
# patchadd 124998-01
Validating patches...

Loading patches installed on the system...

Done!

Loading patches requested to install.

Done!

Checking patches that you specified for installation.

Done!

Approved patches will be installed in this order:

124998-01

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
```

```
Installing patch packages...

Patch 124998-01 has been successfully installed.
See /var/sadm/patch/124998-01/log for details

Patch packages installed:
  SUNWcsu
#
```

Caution – Patches can be added using the -d (nosave) option to save space. When this option is used, patchadd does not save copies of the files being updated or replaced. Patches added in this way cannot be backed out.

Removing a Patch

When you remove a patch, the patchrm command restores all files that were modified or replaced by that patch, unless:

- The patch was installed with the patchadd -d option (which instructs the patchadd command not to save copies of files being updated or replaced)
- The patch is required by another patch
- The patch has been obsoleted by a later patch

The patchrm command calls the pkgadd utility to restore packages that were saved during the initial patch installation.

The following example shows how to remove a patch using the patchrm command.

```
# patchrm 124998-01
Validating patches...

Loading patches installed on the system...

Done!

Checking patches that you specified for removal.

Done!

Approved patches will be removed in this order:
```

```
124998-01

Checking installed patches...

Backing out patch 124998-01...

Patch 124998-01 has been backed out.

#
```

Installing Patch Clusters

Patch clusters provide a selected set of patches for a designated Solaris OS level, conveniently wrapped for one-step installation. Patch clusters usually consist of a set of recommended and security patches.

You should not install cluster patches on systems with limited disk space. Patch clusters often consume substantial amounts of disk space. For example, the Solaris 10_x86 Recommended Patch Cluster, made available February 13, 2007, is approximately 176 Mbytes in its zipped form, and 464 Mbytes in its unzipped form.

By default, the cluster installation procedure saves the base objects being patched. Prior to installing the patches, the cluster installation script first determines if enough system disk space is available in the `/var/sadm/pkg` directory to save the base packages and terminates if not enough space is available.

Caution – You can override the save feature by using the `-nosave` option when you are executing the cluster installation script. If you use the `-nosave` option, you cannot back out these patches if the need arises.

You can remove individual patches that were installed by the patch cluster using the `patchrm` command. `README` files for the patches installed by a patch cluster are located in the patch directories that the installation process creates below the `/var/sadm/patch` directory.

It is generally recommended to install patch clusters with the system operating in single user mode.

To install a patch cluster, perform the following steps:

1. Be sure the patch cluster has been unzipped, or uncompressed and extracted if the cluster was received as a `tar.Z` file. For example, on a Sparc system:

   ```
   # unzip 10_Recommended.zip
   ```

   On an x86/x64 system:

   ```
   # unzip 10_x86_Recommended.zip
   ```

2. Decide which method to use to install the cluster—using the recommended default `save` option, or the `-nosave` option.
3. Change to the directory that contains the patch cluster. Read the `CLUSTER_README` file, which contains information about the bundled set of patches, including:
   - Cluster description
   - Patches included
   - Important notes and warnings
   - Save and backout options
   - Special install instructions
   - Special patch circumstances
   - Any notices and other recommendations

Caution – Recent Solaris 10 patch clusters exceed the capability of the unzip utility shipped with Solaris 10 versions prior to Solaris 10 10/08. Install the latest unzip patch before you extract files.

For example:

```
# cd 10_x86_Recommended
# more CLUSTER_README
# CLUSTER_README

NAME: Solaris 10_x86 Recommended Patch Cluster
DATE: Feb/13/07

########################################################################

This patch cluster is intended to provide a selected set of patches for
the designated Solaris release level.  This is a bundled set of patches
conveniently wrapped for one-step installation.  Only install this
cluster on the appropriate Solaris system.  Carefully read all important
notes and install instructions provided in this README file before
```

```
installing the cluster.  A cluster grouping does not necessarily imply
that additional compatibility testing has occurred since the individual
patches were released.

WARNING!! IT IS HIGHLY RECOMMENDED that the installation of this patch
cluster be performed in single-user mode (Run Level S).

########################################################################

CLUSTER DESCRIPTION
-------------------

These Solaris Recommended patches are considered the most important and
highly recommended patches that avoid the most critical system, user, or
security related bugs which have been reported and fixed to date.  In
```

4. Run the `install_cluster` script.

```
# ./install_cluster

Patch cluster install script for Solaris 10_x86 Recommended Patch Cluster


*WARNING* SYSTEMS WITH LIMITED DISK SPACE SHOULD *NOT* INSTALL PATCHES:
With or without using the save option, the patch installation process
will still require some amount of disk space for installation and
administrative tasks in the /, /usr, /var, or /opt partitions where
patches are typically installed.  The exact amount of space will
depend on the machine's architecture, software packages already
installed, and the difference in the patched objects size.  To be
safe, it is not recommended that a patch cluster be installed on a
system with less than 4 MBytes of available space in each of these
partitions.  Running out of disk space during installation may result
in only partially loaded patches.  Check and be sure adequate disk space
is available before continuing.

Are you ready to continue with install? [y/n]: y
```

(output omitted)

```
The following patches were able to be installed:
        120720-02
        122035-05
        119255-34
        123840-04
```

(output omitted)

```
ERROR:  The following patches were not able to be installed:
        121297-01
        120901-03
        118873-04
        121334-04
```

(output omitted)

```
For more installation messages refer to the installation logfile:
  /var/sadm/install_data/Solaris_10_x86_Recommended_Patch_Cluster_log

Use '/usr/bin/showrev -p' to verify installed patch-ids.
Refer to individual patch README files for more patch detail.
Rebooting the system is usually necessary after installation.
#
```

5. Read individual patch README files to determine if any additional steps are required to fully install any individual patch.
6. Check the patch cluster log file if more detail is needed.

   Reviewing the log provides information about why the patches listed above were not able to be installed:

```
# more
/var/sadm/install_data/Solaris_10_x86_Recommended_Patch_Cluster_log
*** Install Solaris 10_x86 Recommended Patch Cluster begins Sun Feb 25
22:01:19 MST 2007 ***
*** PATCHDIR = /patches/10_x86_Recommended ***

Installing 120720-02...
Validating patches...

Loading patches installed on the system...

Done!

Loading patches requested to install.

Done!

The following requested patches have packages not installed on the system
Package SUNWgzipS from directory SUNWgzipS in patch 120720-02 is not
installed on the system. Changes for package SUNWgzipS will not
 be applied to the system.

Checking patches that you specified for installation.
```

```
Done!


Approved patches will be installed in this order:

120720-02
(output omitted)
#
```

7. Reboot the system for all patches to take effect.

Note – For recent Solaris 10 patch clusters, you must use an option with the `install_cluster` command. Refer to the README to find out which option you must use.

Patch Administration From the Command Line (CLI)

An update (also known as a patch) contains a collection of files and directories. This collection replaces existing files and directories that prevent proper execution of the software. Some updates contain product enhancements.

A Solaris OS update types include:

- Standard updates – Updates that fix specific problems with the Solaris OS and other Sun hardware and software products.
- Recommended patches – Solaris OS updates that fix problems that might occur on a large percentage of systems.
- Update clusters – A group of standard, recommended, or security updates that have been bundled into a single archive for easy downloading and installation.

Note – In previous versions of the Solaris OS, maintenance updates were also available. These were sets of patches that had been tested together and packaged for one-step installation. Maintenance updates were available to service contract customers. Maintenance updates are now replaced by the Solaris OS distributions of the Solaris Express Program. Such updates to the Solaris OS are free for download and are available on a monthly schedule. See the following URL for details about the Solaris Express program:

`http://www.sun.com/software/solaris/solaris-express/`

An update is distributed as a directory that is identified by a unique number. The number assigned to an update includes the update base code first, a hyphen, and a number that represents the update revision number. For example, an update directory named `105050-01`, indicates that `105050` is the base code and `01` is the revision number.

Note – Not all updates from Sun Microsystems must be installed. Only install the recommended and security updates, and those required to fix problems specific to your site.

Using the `smpatch` Command

The `smpatch` command (and its sub-commands) are the preferred commands to use for update/patch management using the CLI.

Starting with the Solaris 9 OS, the `smpatch` command is available in two modes, local mode and remote mode:

- Local mode can only be run on the local system in single-user or multiuser mode.
- Remote mode can be used to perform tasks on remote systems, using the `-n` *`system_name`* option with `smpatch` commands.

By default, `smpatch` runs in local mode. In local mode none of the authentication options or options that refer to remote systems are available.

In releases earlier than Solaris 10, if you specify any of the remote or authentication options (except for `-L`), remote mode is used.

Remote mode is currently not supported in the Solaris 10 OS.

In Solaris 8 only local mode `smpatch` is available.

Phases for Applying Updates

The full sequence for applying an update involves these phases or steps:

1. Analyzing your system.
2. Downloading the necessary updates.
3. Applying the updates.

You can exercise as much control of the phases as needed:

- The `smpatch update` command will perform all three functions using one command.

  This command requires multiuser mode and will not apply an update that has interactive as one of its property install types. The application of updates will be governed by the update policy.
- The `smpatch analyze` and `smpatch update` commands will perform all three functions using two commands.

If you want to first analyze your system and then download and apply them in a single subsequent step, first use the `smpatch analyze` command followed by the `smpatch update` command.

The `smpatch analyze` command requires multiuser mode.

The `smpatch update` command will also download any prerequisite patches.

- The `smpatch analyze`, `smpatch download`, and `smpatch add` commands will perform all three functions using three commands.

  If you want to analyze your system, download the updates and add them to your system in three separate steps, first use the `smpatch analyze` command followed by the `smpatch download` command followed by the `smpatch add` command.

  The `smpatch add` command can be used in single user mode or multiuser mode. The `smpatch add` command will not consult the update policy.

Applying an Update In Three Steps

Using three commands gives more control and flexibility when you apply a patch.

1. Assume that you want to have the latest update(s) for the North America locales. The following command will analyze your local system and determine the appropriate available updates for it. It will not download or apply them. The command will write the list to the file `plist`. Look in the `plist` file for updates involving locales.

```
# smpatch analyze > plist
# vi plist
...
119397-06 SunOS 5.10: patch for North America region locales issues
...
```

The patchadd -p command shows what updates were applied to the system. Use it to verify that the locales update you found in the plist file is not already on the system:

```
# patchadd -p | grep 119397
#
```

Note – You can still use showrev -p to accomplish the same thing, and it executes more quickly.

2. The following command will download (but not apply) the new update for the locales issue:

```
# smpatch download -i 119397-06
119379-06 has been validated.
```

The update was downloaded to the download area and validated. By default, this directory is /var/sadm/spool. If it was changed from the default, query the system with smpatch get to find the new location. The default is still in effect:

```
# smpatch get | grep download
patchpro.download.directory     -       /var/sadm/spool
```

The following commands show the update has been downloaded as a *.jar file:

```
# cd /var/sadm/spool; ls
119397-06.jar
...
```

3. Apply or install this update using the following smpatch add command:

```
# smpatch add -i 119397-06
add patch 119397-06
Patch 119397-06 has been successfully installed.
```

Verify that the patch is installed on your system using this command:

```
# patchadd -p | grep 119397-06
Patch: 119397-06 Obsoletes: Requires: 121734-01 Incompatibles: Packages:
SUNWnameos SUNWnamdt SUNWnamow
```

A subsequent analysis of this system will no longer show this update as appropriate:

```
# smpatch analyze | grep 119397-06
#
```

An update is easily removed (backed out). The following command does this for the update just applied:

```
# smpatch remove -i 119397-06
remove patch 119397-06
Transition old-style patching.
Patch 119397-06 has been backed out.
```

An analysis now shows that this update is once again appropriate and available for this system:

```
# smpatch analyze | grep 119397-06
119397-06 SunOS 5.10: patch for North America region locales issues
```

Applying an Update In One Step

Use the `smpatch update` to analyze your system, download and apply the update in one step. For example this `awk`/`nawk` patch can be applied as follows:

```
# smpatch update -i 118815-05
118815-05 has been validated.
Installing patches from /var/sadm/spool...
118815-05 has been applied.
/var/sadm/spool/patchpro_dnld_2007.03.16@12:36:36:MST.txt has been moved
to
/var/sadm/spool/patchproSequester/patchpro_dnld_2007.03.16@12:36:36:MST.t
xt
```

Configuring the Patch Management Environment

The `smpatch get`, `smpatch set` and `smpatch unset` commands are used to configure the patch management environment:

- `smpatch get` displays the current settings for environment parameters.
- `smpatch set` changes values for environment parameters. The new values are not validated in anyway so verify the intended changes.
- `smpatch unset` enables the default values for environment parameters.

Examples for Configuring Patch Management Environment

Use the following command to display the current environment parameter values:

```
# smpatch get
patchpro.backout.directory          -               ""
patchpro.baseline.directory         -               /var/sadm/spool
patchpro.download.directory         -               /var/sadm/spool
patchpro.install.types              -
rebootafter:reconfigafter:standard
patchpro.patch.source    http://192.168.201.1:3816/solaris/
https://getupdates1.sun.com/solaris/
patchpro.patchset                   -               current
patchpro.proxy.host                 -               ""
patchpro.proxy.passwd               ****            ****
patchpro.proxy.port                 -               8080
patchpro.proxy.user                 -               ""
```

Note – The `smpatch(1M)` man page contains a detailed description of the environment parameters.

The following `smpatch set` and `get` commands will set a new value for the update source. (This is typically what you would do to direct your local client to a new update proxy server.)

```
# smpatch set
patchpro.patch.source=http://newproxy.apex.com:3816/solaris/
# smpatch get
```

```
patchpro.backout.directory      -               ""
patchpro.baseline.directory     -               /var/sadm/spool
patchpro.download.directory     -               /var/sadm/spool
patchpro.install.types          -
rebootafter:reconfigafter:standard
patchpro.patch.source http://newproxy.apex.com:3816/solaris/
https://getupdates1.sun.com/solaris/
patchpro.patchset               -               current
patchpro.proxy.host             -               ""
patchpro.proxy.passwd           ****            ****
patchpro.proxy.port             -               8080
patchpro.proxy.user             -               ""
```

You can also set the source of updates to a local or remote directory as the following examples illustrate:

```
# smpatch set patchpro.patch.source=file:/net/sys-04/export/updates

# smpatch set patchpro.patch.source=file:/local/updates

# smpatch set patchpro.patch.source=file:/cdrom/cdrom0
```

The following command sets the `patchpro.patch.source` parameter back to the default value:

```
# smpatch unset patchpro.patch.source
# smpatch get
patchpro.backout.directory      -       ""
patchpro.baseline.directory     -       /var/sadm/spool
patchpro.download.directory     -       /var/sadm/spool
patchpro.install.types          -     rebootafter:reconfigafter:standard
patchpro.patch.source           -
https://getupdates1.sun.com/solaris/
patchpro.patchset               -       current
patchpro.proxy.host             -       ""
patchpro.proxy.passwd           ****    ****
patchpro.proxy.port             -       8080
patchpro.proxy.user             -       ""
```

You can configure an update set which defines a subset of updates that commands will work with. For example, the following commands will result in an analysis only on recommended updates:

```
# smpatch set patchpro.patchset=recommended
# smpatch analyze
```

Using the Update Policy for Applying Updates

The `patchpro.install.types` property defines the update policy in effect for the update management environment.

When you apply patches using the `smpatch update` command the update policy is consulted before an update is actually applied.

The following are the types of updates that are applied to the system:

- Standard updates that are applied immediately and require no system restart
- Updates that require a system restart

If you use the `smpatch update` command to update your system, you get the benefit of the guidelines established by update/patch developers in how best to apply the update. However, you can customize the policy for applying updates using the `patchpro.install.types` parameter.

Table 8-1 shows the `patchpro.install.types` parameter values and also describes the value and if it is part of the default update policy.

Table 8-1 Install Type Parameter Values

patchpro. install. types value	**Description**
standard	A default. Can be applied in multiuser mode and visible immediately unless the application being updated is running while the update is applied. In this case, the effects of the update are visible after the affected application is restarted.
rebootafter	A default. Effects not visible until system reboot.
reconfigafter	A default. Effects not visible until a reconfiguration reboot (`boot -r`). See the `boot`(1M) man page.
rebootimmediate	System becomes unstable (unpredictable behavior or possible data loss) until system reboot.
reconfigimmediate	System becomes unstable (unpredictable behavior or possible data loss) until system reconfiguration reboot (`boot -r`).

Table 8-1 Install Type Parameter Values (Continued)

patchpro. install. types value	**Description**
singleuser	Do not apply this update in multiuser mode. You must apply this update on a quiet system with no network traffic and with extremely restricted I/O activity.
interactive	Only downloaded to your system and must be applied manually according to the instructions in the update's README file.
clientroot | clientuser	Only applies to diskless clients and indicates which portion of the client's filesystem is affected by the update (clientroot indicates `/` and clientuser indicates `/usr`)

The default value for this parameter is shown with this `smpatch get` command:

```
# smpatch get patchpro.install.types
patchpro.install.types       -      rebootafter:reconfigafter:standard
```

Per Update Policy Value

Each update has properties associated with it. The PATCH_PROPERTIES values are the install types for the update. You can learn these values with the following command sequence if the update is downloaded in the spool area of your system:

```
# cd /var/sadm/spool
# jar xvf 119578-15.jar 119578-15/patchinfo
 inflated: 119578-15/patchinfo
# cat 119578-15/patchinfo
PATCHINFOVERSION="1.0"
PATCHID=119578-15
PATCH_CORRECTS='BaseOS.SolarisCore-5.10 BaseOS.SolarisFaultMgmt-5.10'
PATCH_ARCH='sparc'
PATCH_OS='SunOS'
PATCH_OSRELEASE='5.10'
PATCH_PROPERTIES='rebootafter'
PATCH_OBSOLETES="119330-01 119331-01 119559-01 119576-01 120635-02"
```

The patchinfo file is included in the collection of files in an update. It contains more metadata than just the PATCH_PROPERTIES value.

Example of Using the Update Policy

It is good practice to always use the update policy when adding patches, otherwise you might miss an important patch property such as immediate reboot or a prerequisite patch.

Not Using the smpatch update Command

Analyze your system and learn if any updates involving wanboot are appropriate and available:

```
# smpatch analyze | grep wanboot
119681-06 SunOS 5.10: wanboot patch
```

Determine if any prior versions of the wanboot update are already on the system:

```
# patchadd -p | grep 119681
Patch: 119681-05 Obsoletes: Requires: Incompatibles: Packages: SUNWcakr
```

Only the earlier 05 version of this update is already installed.

Download, but do not apply, the newer wanboot update:

```
# smpatch download -i 119681-06
119681-06 has been validated.
```

Apply the patch using the smpatch add command:

```
# smpatch add -i 119681-06
add patch 119681-06
...
Validating patches...
Loading patches installed on the system...
Done!
Loading patches requested to install.
Done!
Checking patches that you specified for installation.
Done!
Approved patches will be installed in this order:
119681-06
```

```
Patch 119681-06 has been successfully installed.
```

Verify that the patch is installed on your system:

```
# patchadd -p | grep 119681
Patch: 119681-05 Obsoletes: Requires: Incompatibles: Packages: SUNWcakr
Patch: 119681-06 Obsoletes: Requires: Incompatibles: Packages: SUNWcakr
```

A subsequent analysis of this system will no longer show this update as appropriate:

```
# smpatch analyze | grep 119681-06
#
```

Especially when you use the `smpatch add` command it is always a good practice to read information about the update. Go to the download spool area and see what information there is about this update:

```
# cd /var/sadm/spool; ls
119681-06.jar
cache
patchpro_dnld_2006.02.13@10:10:29:MST.txt
# cat *.txt
This patch bundle was generated by PatchPro.

Please refer to the README file within each patch for installation
instructions. To properly patch your system, the following patches
should be installed in the listed order:


1) 119681-06!!! IMMEDIATE REBOOT!!!
```

The `*.txt` and other README files often contain important information. In this case the warning to immediately reboot implies that the PATCH_PROPERTIES value for install type is either `reconfigimmediate` or `rebootimmediate`.

The following command sequence will display the install type value for this update:

```
# cd /var/sadm/spool
# jar xvf 119681-06.jar 119681-06/patchinfo
 inflated: 119681-06/patchinfo
# grep PROP 119681-06/patchinfo
PATCH_PROPERTIES='reconfigimmediate'
```

A reconfiguration reboot (`boot -r`) should be done on this system to render it stable again.

Note – The `/var/adm/messages` file identifies problems that are found when applying a patch to a system.

Using the `smpatch update` Command

The `smpatch update` command will analyze your system, download the update and apply it in one step. It also provides safeguards that are not available with `smpatch add` because it consults the update policy.

The `smpatch update` command also is knowledgeable about update dependencies and applies any dependencies for the updates you specify.

```
# smpatch update -i 119681-06
119681-06 has been validated.
Installing patches from /var/sadm/spool...
NOTICE: Patch 119681-06 cannot be installed until the next system
shutdown.
/var/sadm/spool/patchpro_dnld_2006.02.15@06:02:43:MST.txt has been moved
to
/var/sadm/spool/patchproSequester/patchpro_dnld_2006.02.15@06:02:43:MST.t
xt
/var/sadm/spool/patchpro_dnld_2006.02.15@06:09:14:MST.txt has been moved
to
/var/sadm/spool/patchproSequester/patchpro_dnld_2006.02.15@06:09:14:MST.t
xt

ID's of the updates that are disallowed by installation policy have been
written to file
        /var/sadm/spool/disallowed_patch_list

One or more updates that you installed requires a system shutdown to
activate it. To initiate the system shutdown, you must use one of the
following commands:
o Power down the system - init 0 or shutdown -i 0
o Drop to the firmware prompt - init 5 or shutdown -i 5
o Restart the system - init 6 or shutdown -i 6
```

Recall that `smpatch add` command informs you about the required reboot in the `*.txt` in the download spool area. `smpatch update`, on the other hand, displayed this to standard out, creates a `disallowed_patch_list` and gave instructions about the reboot.

```
# cat /var/sadm/spool/disallowed_patch_list
119681-06
```

Part of the `smpatch update` command applies the updates. Updates that cannot be applied for some reason are listed in the `disallowed_patch_list`. Typically you attend to updates listed in this file manually.

Verify that the only version of this update installed on the system is the prior version (05):

```
# patchadd -p | grep 119681
Patch: 119681-05 Obsoletes: Requires: Incompatibles: Packages: SUNWcakr
```

A subsequent analysis of the system still shows that this patch is available and still appropriate for this system. It is in the spooled area awaiting installation and a system reboot.

```
# smpatch analyze | grep wanboot
119681-06 SunOS 5.10: wanboot patch
```

Manually add the patch:

```
# smpatch add -i 119681-06
Validating patches...
Loading patches installed on the system...
Done!
Loading patches requested to install.
Architecture for package SUNWcakr from directory SUNWcakr.v in patch
119681-06 differs from the package installed on the system.
Architecture for package SUNWcakr from directory SUNWcakr.us in patch
119681-06 differs from the package installed on the system.
Done!
Checking patches that you specified for installation.
Done!
Approved patches will be installed in this order:
119681-06
Patch 119681-06 has been successfully installed.
```

Verify that it is installed:

```
# patchadd -p | grep 119681
Patch: 119681-05 Obsoletes: Requires: Incompatibles: Packages: SUNWcakr
Patch: 119681-06 Obsoletes: Requires: Incompatibles: Packages: SUNWcakr
```

Analyze the system to show that it is no longer appropriate and available:

```
# smpatch analyze | grep wanboot
#
```

Since this update is marked as `reconfigimmediate`, reboot the system with the -r option:

```
ok boot -r
...
```

Remove this update's entry in the `disallowed_patch_list` file so you know you are finished administering this update:

```
# cat /var/sadm/spool/disallowed_patch_list
#
```

Working With Multiple Updates

Many of the `smpatch` sub-commands can be applied to multiple updates. Following are some examples:

Multiple instances of the `-i` option are permitted if you just have a few updates to apply:

```
# smpatch update -i 118927-02 -i 118822-15 -i 119681-06
```

A list of update IDs can be listing in a file, one per line, and referenced using the `-x idlist=` option:

```
# smpatch update -x idlist=/var/sadm/spool/disallowed_patch_list
```

Note – If you specify particular patches by using the `-i` or `-x idlist=` options, the list is augmented with patches on which they depend before the update occurs.

The following example shows how to create a list of patches that you actually want to apply from the larger list available and appropriate. It also resolves the dependencies for the updates you want to apply.

Generate the full list of updates available and appropriate for your system:

```
# smpatch analyze > my.list
```

Edit `my.list` and remove the ones you are not interested in:

```
# vi my.list
...
```

Analyze just the ones that are left and resolve dependencies:

```
# smpatch analyze -x idlist=my.list > /tmp/justdothese.list
```

Apply the updates:

```
# smpatch update -x -idlist=/tmp/justdothese.list
```

Authorization and Authentication

The root user certainly can manage updates. You can designate non root users for these tasks by having them assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.

The following command shows a `smpatch get` command requiring assumption of the role (`-r`) called `patchman`:

```
sys-01> smpatch get -r patchman
```

Exercise: Maintaining Patches

In this exercise, you use commands to prepare patches for installation, evaluate patch descriptions, and install and remove patches.

Preparation

This exercise requires the set of patches found in `/opt/ses/lab/patches` directory on student systems. If the patches are located on a server, your instructor will provide its location and information about how to retrieve them.

Note – Use the patch archives in the `/opt/ses/lab/patches` directory to understand how to use patch administration commands.

Task

Complete the following steps

1. Copy the patch archives found in `/opt/ses/lab/patches` directory to the `/var/tmp` directory.

Note – If the patch archives are located on a server, follow the instructions your instructor provides to copy them to the `/var/tmp` directory on your system.

---

2. Change directory to `/var/tmp`.
3. Choose two patches that are named using consecutive numbers, and use the `unzip` command to extract the patches from their `.zip` archives.
4. Use the `uname -r -p` command to display your system's OS release level, and architecture (processor) type.
5. Display the `README` files for the two patches you extracted, and determine which of them is the correct patch for your system architecture. Also verify that the patch you choose is for the Solaris release level in use on your system.

6. Use the `patchadd` command to determine if the patch you identified in the previous step is currently installed on your system.
7. Verify that the `/var/sadm/patch` directory is empty.
8. Add the patch.
9. Verify that the patch is now installed.

   ____

   ____

10. Examine the patch installation log file.
11. Remove the patch you just installed, and verify that the patch is no longer installed.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Maintaining Patches

This section provides task steps and their solutions.

Task

Complete the following steps:

1. Copy the patch archives found in /opt/ses/lab/patches to the /var/tmp directory.

Note – Use the patch archives in the /opt/ses/lab/patches directory to understand how to use patch administration commands. Use the patch that is appropriate for your system architecture. The patches for this exercise for demonstration only.

```
# cd /opt/ses/lab/patches
# ls
123456-01.zip 654321-01.zip
# cp *.zip /var/tmp
#
```

Note – If the patch archives are located on a server, follow the instructions your instructor provides to copy them to the/var/tmp directory on your system.

---

2. Change directory to /var/tmp.

```
# cd /var/tmp
```

3. Extract the patches from their archives. For example:

```
# unzip 123456-01.zip
Archive: 123456-01.zip
creating: 123456-01/
...
(output omitted)
# unzip 654321-01.zip
Archive: 654321-01.zip
creating: 654321-01/
...
(output omitted)
```

4. Use the `uname -r -p` command to display your system OS release level, and architecture (processor) type.

```
# uname -r -p
5.10 sparc
#
```

5. Display the `README` files for the two patches you extracted, and determine which of them is the correct patch for your system architecture. Also verify that the patch you choose is for the Solaris release level in use on your system. For example:

```
# more 123456-01/README*
# more 654321-01/README*
```

6. Use the `patchadd` command to determine if the patch you identified in the previous step is currently installed on your system. For example:

```
# patchadd -p | grep 123456
# patchadd -p | grep 654321
```

7. Verify that the `/var/sadm/patch` directory is empty.

```
# ls /var/sadm/patch
#
```

8. Add the patch. For example:

```
# patchadd 123456-01
# patchadd 654321-01
```

9. Verify that the patch is now installed.

   What package does the patch affect?

   For example:

```
# patchadd -p | grep 123456-01
# patchadd -p | grep 654321-01
```

10. Examine the patch installation log file. This file is located in the `/var/sadm/patch` directory. For example:

```
# cd /var/sadm/patch/123456-01
# more log
# cd /var/sadm/patch/654321-01
# more log
```

11. Remove the patch you just installed, and verify that the patch is no longer installed.

```
# cd
# patchrm 123456-01
# patchrm 654321-01
```

Notes:

Module 9

Using Boot PROM Commands

Objectives

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

- Identify boot programmable read-only memory (PROM) fundamentals
- Use basic boot PROM commands
- Identify the system boot device
- Create and remove custom device aliases
- View and change non-volatile random access memory (NVRAM) parameters from the shell
- Interrupt an unresponsive system

Introducing Boot PROM Fundamentals

All Sun systems have resident boot PROM firmware that provides basic hardware testing and initialization prior to booting. The boot PROM also enables you to boot from a wide range of devices. In addition, there is a user interface that provides several important functions.

The Sun boot PROM has access to a standard set of generic device drivers. The system needs these drivers to access and control the buses and the boot device to boot the system properly.

All versions of the OpenBoot™ architecture allow a third-party board to identify itself and load its own plug-in device driver. Each device identifies its type and furnishes its plug-in device driver when requested by the OpenBoot PROM during the system hardware configuration phase of the boot process.

Figure 9-1 shows the identification process.

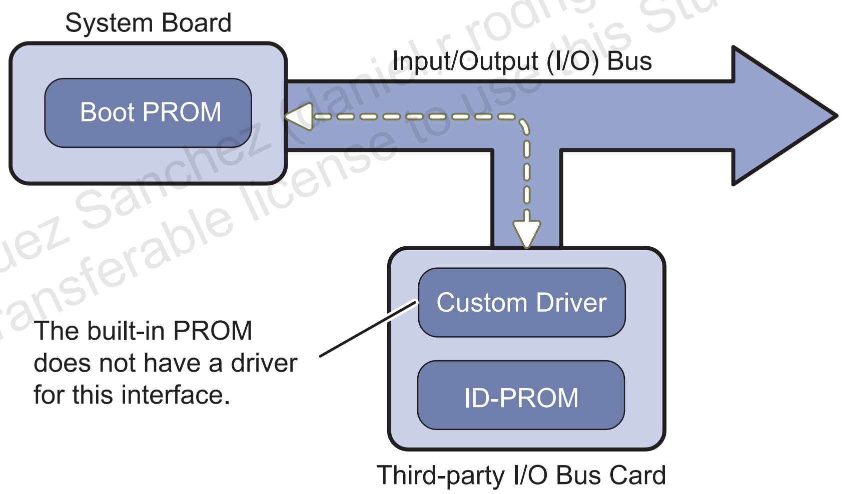


Figure 9-1 Third-Party Device Identification Process

Goal of the OpenBoot™ Architecture Standard

The overall goal of the Institute of Electrical and Electronics Engineers (IEEE) standard #1275 for the OpenBoot architecture is to provide the capabilities to do the following:

- Test and initialize system hardware
- Determine the system hardware configuration
- Boot the operating system (OS)
- Provide an interactive interface for configuration, testing, and debugging
- Enable the use of third-party devices

Boot PROM

Each Sun SPARC system has a boot PROM chip. This 1-Mbyte chip is typically located on the same board as the central processing unit (CPU). Boot PROM chips are usually found in a pluggable socket on older systems. As of the 3.*x* PROM, they are permanently soldered to the main system board.

The Ultra workstations use a reprogrammable boot PROM called a flash PROM (FPROM). The FPROM allows you to load new boot program data into the PROM by using software, instead of having to replace the chip.

Desktop systems have a write-protect jumper that must be moved before you can write to the PROM. You have to move the jumper because the default position is write-protect. Refer to the *Sun Flash PROM Guide for Workstations and Workgroup Servers - Standalone Version* part number 802-3233-27, for the jumper location on your system.

Caution – Many systems have the jumper under an installed frame buffer or other removable card. Be careful when removing or replacing this card.

The main functions of the boot PROM are to test the system hardware and to boot the operating system. The boot PROM firmware is referred to as the *monitor* program.

The boot PROM firmware controls the operation of the system before the operating system has been booted and the `kernel` is available. The boot PROM also provides the user with a user interface and firmware utility commands, known as the FORTH command set. Commands include the boot commands, diagnostics commands, and commands to modify the default configuration.

Note – The boot PROM does not work with the Solaris OS file systems or files. It handles mainly hardware devices. The OS works with and is dependent on firmware, but firmware is independent of the OS.

To determine which revision of OpenBoot PROM is running on the system, you can use the `prtdiag` or `prtconf` command. For example:

```
# /usr/platform/`uname -m`/sbin/prtdiag -v

(output omitted)
System PROM revisions:
----------------------
OBP 4.16.4 2004/12/18 05:21 Sun Blade 1500 (Silver)
OBDIAG 4.16.4 2004/12/18 05:21
#
or
# prtconf -V
OBP 4.16.4 2004/12/18 05:21
#
```

System Configuration Information

Another important element in each Sun system is the system configuration information. The system configuration information includes the following:

- The Ethernet or MAC address, such as 8:0:20:5d:6f:9e
- The system host ID, such as 805d6f9e
- User-configurable parameters which have been modified from the default settings

The user-configurable parameters are known as NVRAM variables, or EEPROM parameters. They allow an administrator to control things such as the default boot device, the level of Power-on self-test (POST), and so on.

Depending on the system, one of three different components store the system configuration information:

- NVRAM chip
- Serial Electronically Erasable Programmable Read Only Memory (SEEPROM) chip
- System Configuration Card (SCC)

NVRAM Chip

Older systems contain a removable NVRAM chip, normally located on the main system board. In addition to the system configuration information, the NVRAM chip contains an integrated lithium battery which provides battery backup for the configuration information and also provides the system's time-of-day (TOD) function.

SEEPROM Chip

Most newer systems contain a non-removable SEEPROM chip, normally located on the main system board. SEEPROM chips do not require a battery to maintain the system configuration information.

System Configuration Card

Some newer systems contain a removable System Configuration Card which holds the system configuration information. It is inserted into the System Configuration Card Reader.

Figure 9-2 shows the basic elements of the Boot PROM and NVRAM

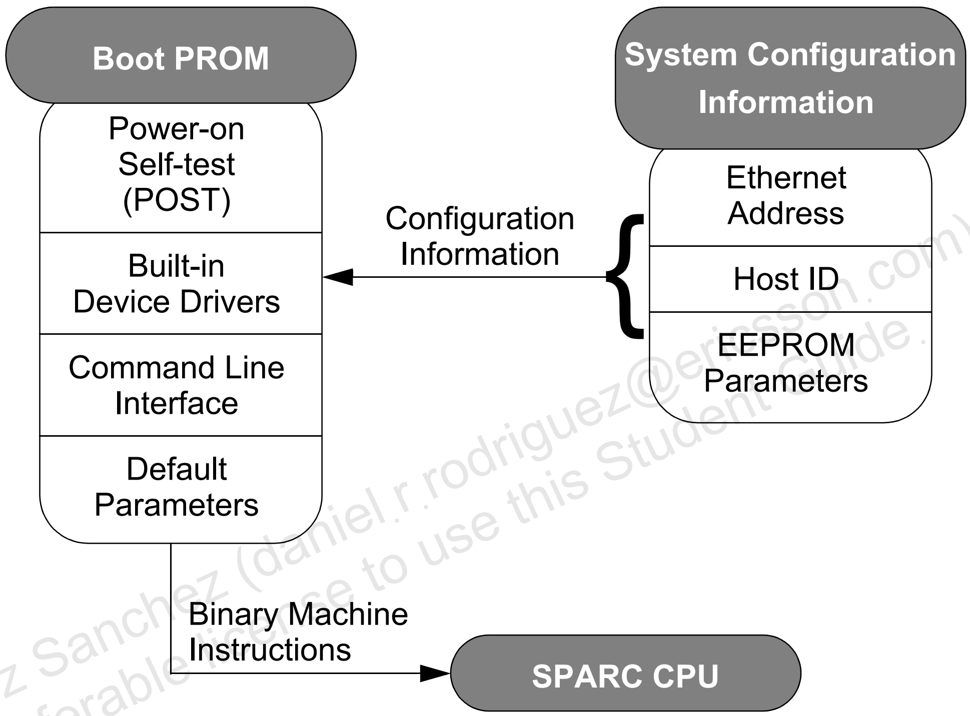


Figure 9-2 Basic Elements of the Boot PROM and NVRAM POST

When a system's power is turned on, a low-level POST is initiated. This low-level POST code is stored in the boot PROM and is designed to test the most basic functions of the system hardware.

At the successful completion of the low-level POST phase, the boot PROM firmware takes control and performs the following initialization sequence:

- Probes the memory and then the CPU
- Probes bus devices, interprets their drivers, and builds a device tree
- Installs the console

After the boot PROM initializes the system, the banner displays on the console. The system checks parameters stored in the boot PROM and NVRAM to determine if and how to boot the operating system.

One of the first tests that POST runs is to check to determine if a keyboard is connected to the system and if a Stop-key option is present.

The Stop-key is located on the left side of the keyboard. To enable various diagnostic modes, hold down the Stop-key simultaneously with another key. The Stop-key sequences have an effect on the OpenBoot PROM and define how POST runs when the system power is on. The following is a list of the Stop-key sequences:

- Stop-D key sequence – Hold down the Stop and D keys simultaneously while system power is turned on, and the firmware automatically switches to diagnostic mode. This mode runs more extensive POST diagnostics on the system hardware. The OpenBoot PROM variable `diag-switch?` is set to `true`.

See Figure 9-3 to show the effect of the variable `diag-switch?`.

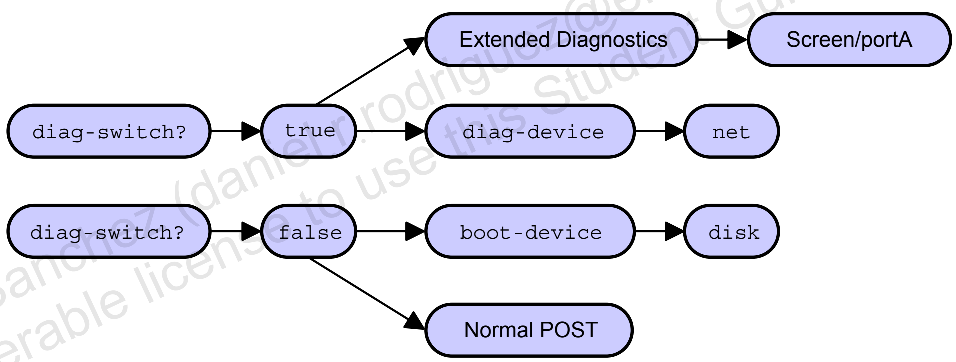


Figure 9-3 Post Diagnostics

Note – The Stop-D key sequence is not available on a serial port terminal.

- Stop-N key sequence – Hold down the Stop and N keys simultaneously while the system power is turned on to set the NVRAM parameters to the default values. When you see the light emitting diodes (LEDs) on the keyboard begin to flash, you can release the keys, and the system should continue to boot.

  Incorrect NVRAM settings can cause system boot failure. For example, during a flash PROM download, if a power failure occurs, some of the contents of the NVRAM can become unusable.

If the system does not boot and you suspect that the NVRAM parameters are set incorrectly, the parameters can easily be changed to the default values.

Caution – Where possible, capture non-default NVRAM values before using the Stop-N key sequence.

- Stop-A key sequence – Hold down the Stop and A keys simultaneously to interrupt any program that is running at the time these keys are pressed and to put the system into the command entry mode for the OpenBoot PROM. The system presents an `ok` prompt for the user, which signifies it is ready to accept OpenBoot PROM commands.

Caution – The Stop-A key sequence, as a method for getting to the `ok` prompt, is not recommended unless there is no alternative. The Stop-A key sequence can cause Solaris OS file system corruption which can be difficult to repair.

Disabling the Abort Sequence

You may want to disable the abort key sequence on a system to prevent possible corruption of a file system or to provide tighter security. To disable the abort key sequence do the following. When you complete these steps, the system allows the Stop-A key sequence only during the boot process.

1. Edit the `/etc/default/kbd` file.

   Inside the file, the statement `KEYBOARD_ABORT=disable` is commented out.
2. Remove the comment character from the beginning of the line.
3. Save the file
4. Execute the command `kbd -i`.

You can also configure the system to change the keyboard abort sequence to an alternate keystroke. Review the `kbd` command man page for more information.

Displaying POST to the Serial Port

Attach a terminal to the serial port of a system to capture more information from the POST output. When the power is on, POST looks for a keyboard. If there is no keyboard present, POST diverts system output to serial port A. POST runs more extensive tests when the system is in diagnostic mode with the PROM parameter `diag-switch?` set to `true`. Attach the correct type of null modem cable for your system type to serial port A.

Some systems require a serial port cable, as shown in Figure 9-4.

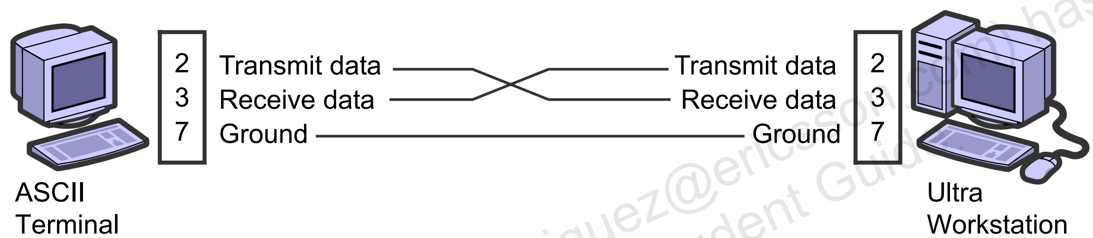


Figure 9-4 Serial Port Connection to a Sun Workstation

The following example is the POST output from a Sun Blade™ 1000:

```
@(#)  4.0 Version 29 created 2000/07/12 16:46
Clearing TLBs Done
Reset: 0000.0000.0000.0010 SPOR
Loading Configuration
Membase: 0000.0000.0000.0000
MemSize: 0000.0000.2000.0000
Init CPU arrays Done
Init E$ tags Done
Setup TLB Done
MMUs ON
Block Scrubbing Done
Copy Done
PC = 0000.07ff.f000.3138
Decompressing Done
Size = 0000.0000.0006.e3b0
ttya initialized
Start Reason: Soft Reset
System Reset: (SPOR)
Probing gptwo at 0,0 SUNW,UltraSPARC-III (750 MHz @ 5:1, 8 MB)
   memory-controller
Probing gptwo at 1,0 Nothing there
Probing gptwo at 8,0 pci pci upa ppm
```

```
Loading Support Packages: kbd-translator
Loading onboard drivers: ebus flashprom bbc ppm i2c dimm-fru dimm-fru
   dimm-fru dimm-fru dimm-fru dimm-fru dimm-fru dimm-fru nvram idprom
   i2c cpu-fru temperature fan-control card-reader motherboard-fru
Memory Configuration:
Segment @ Base:        0  Size:   512 MB (2-Way)
Probing /upa@8,480000 Device 0,0 Nothing there
Probing /upa@8,480000 Device 1,0 Nothing there
Probing /pci@8,600000 Device 4  SUNW,qlc fp disk
Probing /pci@8,600000 Device 1  Nothing there
Probing /pci@8,700000 Device 5  network firewire usb
dev-descrip
next-add
node made
Probing /pci@8,700000 Device 6  scsi disk tape scsi disk tape
Probing /pci@8,700000 Device 1  Nothing there
Probing /pci@8,700000 Device 2  Nothing there

 (UltraSPARC-III) , Keyboard Present
OpenBoot 4.0, 512 MB memory installed, Serial #12134217.
Ethernet address 8:0:20:b9:27:49, Host ID: 80b92749.
```

Using Basic Boot PROM Commands

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

Caution – Modifying OPB contents can seriously damage your system. Use caution when modifying settings.

Note – The `ok` prompt indicates that the Solaris OS is currently not running.

Table 9-1 shows typical commands entered at the `ok` prompt.

Table 9-1 Typical Commands Used at the `ok` Prompt

Command	Description
`banner`	Displays the power-on banner
`boot`	Boots the system
`help`	Lists the main help categories
`printenv`	Displays all parameter current and default values
`setenv`	Sets the specified NVRAM parameter to some value
`reset-all`	Resets the entire system; similar to a power cycle
`set-defaults`	Resets all parameter values to the factory defaults
sifting *`text`*	Displays the FORTH commands containing *`text`*
`.registers`	Displays the contents of the registers
`probe-scsi`	Identifies the devices on the internal Small Computer System Interface (SCSI) bus
`probe-scsi-all`	Identifies the devices on all SCSI buses
`probe-ide`	Identifies devices on the internal integrated device electronics (IDE) bus
`probe-fcal-all`	Identifies devices on all Fibre Channel loops

Table 9-1 Typical Commands Used at the `ok` Prompt (Continued)

`show-devs`	Displays the entire device tree
`devalias`	Identifies the current boot device alias for the system
`nvalias`	Creates a new device alias name
`nvunalias`	Removes a device alias name
`show-disks`	Displays and allows a selection of device paths for the disks to be used for `nvalias`
sync	Manually attempts to flush memory and synchronize file systems
`test`	Runs self-tests on specified devices

Identifying the System Boot PROM Version

The `banner` command lists system information, for example:

- model name
- boot PROM version number (for example, 1.*x*, 2.*x*, 3.*x*, 4.*x*, or 5.*x*)
- memory
- Ethernet address
- host ID

The following example shows output from the `banner` command.

```
ok banner

Sun Blade 1500 (Silver), Keyboard Present
Copyright 1998-2004 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.16.4, 1024 MB memory installed, Serial #64512991.
Ethernet address 0:3:ba:d8:63:df, Host ID: 83d863df.
```

Booting the System

Use the `boot` command to boot the Solaris OS from the `ok` prompt. This command has several options available for booting the system in different situations. The format for the boot command is:

```
boot device_name -options
```

Enter the `boot` command at the `ok` prompt to boot the system to multiuser mode automatically.

```
ok boot
```

The following list describes some of the options for the `boot` command:

- `-s` – Boots the system to a single-user mode and asks the user for the `root` password.

  ```
  ok boot -s
  ```

- `cdrom -s` – Boots the system to single user mode from a CD-ROM or a DVD.

  ```
  ok boot cdrom -s
  ```

- `-a` – Boots the system interactively. Use this option if an alternative file must be executed during boot. The boot program asks for the following information. Enter the Return key to accept the values presented.

```
ok boot -a
(output omitted)
Rebooting with command: boot -a
Boot device: /pci@1e,600000/ide@d/disk@0,0  File and args: -a
Enter filename [kernel/sparcv9/unix]: <Return>
Enter default directory for modules [/platform/SUNW,Sun-Blade-1500/kernel
/platform/sun4u/kernel /kernel /usr/kernel]:<Return>
Name of system file [etc/system]:<Return>
SunOS Release 5.10 Version Generic_118833-33 64-bit
Copyright 1983-2006 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
root filesystem type [ufs]:<Return>
Enter physical name of root device
[/pci@1e,600000/ide@d/disk@0,0:a]:<Return>
(output omitted)
```

- `-r` – Performs a reconfiguration boot. Use this option to find a newly attached device and to create new device entries in the `/devices` and `/dev` directories. It also updates the `/etc/path_to_inst` file.

  ```
  ok boot -r
  ```

- `-v` – Boots the system while displaying more detailed device information to the console. Use this option to troubleshoot problems during the boot process. You can use this option with other options.

  ```
  ok boot -v

  ok boot -rv

  ok boot -sv
  ```

Accessing More Detailed Information

Use the `help` command to obtain help on the main categories in the OpenBoot firmware. The following is an example of the `help` output from a Sun Blade 1500 system that is running OpenBoot PROM version 4.16.4:

```
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:
Breakpoints (debugging)
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
eject devices
Power on reset
Diag (diagnostic routines)
Resume execution
File download and boot
nvramrc (making new commands permanent)
ok
```

The `help` command listing provides keywords you can use to view further details. For example, to view information for one of the main categories listed in the preceding example, perform one of the following commands:

```
ok help boot
```

```
ok help nvramrc
```

```
ok help diag
```

```
ok help line
```

Listing NVRAM Parameters

Use the `printenv` command to list all the NVRAM parameters. If the parameter can be modified, the `printenv` command displays its default setting and current setting. The following example shows output from the `printenv` command.

```
ok printenv
Variable Name          Value                          Default Value

asr-policy             normal                         normal
test-args
diag-passes            1                              1
local-mac-address?     true                           true
fcode-debug?           false                          false
scsi-initiator-id      7                              7
oem-logo                                              No default
oem-logo?              false                          false
oem-banner                                            No default
oem-banner?            false                          false
ansi-terminal?         true                           true
screen-#columns        80                             80
screen-#rows           34                             34
ttyb-rts-dtr-off       false                          false
ttyb-ignore-cd         true                           true
ttya-rts-dtr-off       false                          false
ttya-ignore-cd         true                           true
ttyb-mode              9600,8,n,1,-                   9600,8,n,1,-
ttya-mode              9600,8,n,1,-                   9600,8,n,1,-
output-device          ttya                           screen
input-device           ttya                           keyboard
auto-boot-on-error?    false                          false
error-reset-recovery   sync                           sync
load-base              16384                          16384
auto-boot?             true                           true
boot-command           boot                           boot
diag-file
diag-device            net                            net
boot-file
boot-device            disk net                       disk net
use-nvramrc?           false                          false
nvramrc
security-mode          none                           No default
security-password                                     No default
security-#badlogins    0                              No default
verbosity              min                            min
```

```
diag-trigger         none                            none
service-mode?        false                           false
diag-script          normal                          normal
diag-level           max                             max
diag-switch?         false                           false
ok
```

Use the `printenv` command to display a single parameter and its values. For example, to display only the `boot-device` parameter, perform the following command:

```
ok printenv boot-device
boot-device = disk net
```

The possible values of the `boot-device` parameter include `disk`, `net`, and `cdrom`.

Note – Some OpenBoot PROM parameters, such as `auto-boot?`, end in a question mark. The question mark is part of the parameter name. If an OpenBoot PROM parameter ends in a question mark, the parameter value is typically either `true` or `false`.

Changing NVRAM Parameters

Use the `setenv` command to change the current values assigned to NVRAM parameters.

If the `auto-boot?` parameter is set to `true,` the system boots automatically. If it is set to false, the system stops at the `ok` prompt.

The following example changes the `auto-boot?` parameter from its default setting of `true` to the value of `false`.

```
ok printenv auto-boot?
auto-boot? = true
ok
ok setenv auto-boot? false
auto-boot? = false
```

The `reset-all` command halts the system, clears all buffers and registers, and performs a software simulated power-off/power-on of the system.

```
ok reset-all
Resetting ...
```

Note – The `reset-all` command, combined with the `auto-boot? = false` setting clears system registers, which is required on a system with a PROM 3.*x* or higher, before you can use the `probe` command or perform other tests.

Restoring Default NVRAM Parameters

Use the `set-defaults` command to reset all NVRAM parameters to their default values. It affects only parameters that have assigned default values.

```
ok set-defaults
Setting NVRAM parameters to default values.
```

To reset a specific parameter to its default value, use the `set-default` command followed by the parameter name.

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

For example, to reset the `diag-level` parameter, perform the command:

```
ok set-default diag-level
```

Displaying Devices Connected to the System

Use the `probe` command to identify peripheral devices, such as disks, tape devices, or CD-ROMs, currently connected to a system.

Use the `sifting` command to identify available `probe` commands (see the following example) and to find OpenBoot PROM commands when you don't know the command syntax. Commonly-used `probe` commands include `probe-scsi`, `probe-scsi-all`, and `probe-ide`.

```
ok sifting probe

        In vocabulary  forth
(f0060930) probe-all    (f005fe98) probe-ide    (f005fc88) probe-scsi-all
(f005023c) probe-all    (f00501cc) probe-io     (f004a7c4) probe-root-
slot
(f00342e0) probe        (f00341a0) probe-virtual      (f002fc10) xprobe
(f002fbe8) lprobe       (f002fbc0) wprobe       (f002fb98) cprobe

(output omitted)
```

Caution – The following warning message might be displayed if you invoke the `probe` commands on Sun systems that contain a boot PROM that is version 3.*x* and above.

```
This command may hang the system if a Stop-A or halt command has been
executed. Please type reset-all to reset the system before executing this
command.
Do you wish to continue? (y/n) n
```

If any portion of the Solaris OS was loaded into memory when the system was aborted or shut down, the probe commands can hang the system. Look at OBP registers to determine if a reset-all command is necessary. Non-zero values in the registers listed indicate the possibility that the system may hang if you use a probe command. For example:

```
ok .registers
        Normal          Alternate        MMU               Vector
0:               0               0               0               0
1:         1047cf0        fff686c0        f00447d4             798
2:               8        f0000000               0               4
3:               8               3               0         180c660
4:    30005c8ac00               3             3ff         180c000
5:               9    fe000000000f        fff6c000              10
6:              10         100769c 800000003ff640b6         180c9f0
7:    300022f09a0            4058               2              20
%PC  f00475c8 %nPC f00475cc
%TBA f0000000 %CCR 0 XCC:nzvc   ICC:nzvc
ok
```

To avoid hanging your system, perform the commands:

```
ok setenv auto-boot? false
ok reset-all
```

Use the .registers command again, and verify that all values are set to zero. For example:

```
ok .registers
        Normal          Alternate        MMU               Vector
0:               0               0               0               0
1:               0               0               0               0
2:               0               0               0               0
3:               0               0               0               0
4:               0               0               0               0
5:               0               0               0               0
6:               0               0               0               0
7:               0               0               0               0
%PC  0 %nPC 0
%TBA 0 %CCR 0 XCC:nzvc   ICC:nzvc
okc
```

With registers in this state, probe commands will function properly.

The `probe-scsi` Command

The `probe-scsi` command identifies peripheral devices attached to the primary on-board SCSI controller. The `probe-scsi` command identifies such peripheral devices as disks, tape drives, or CD-ROMs by their target addresses. This example shows the two internal SCSI disks of a Sun Fire V120 system:

```
ok probe-scsi
Target 0
  Unit 0   Disk     FUJITSU MAP3367N SUN36G 0401
Target 1
  Unit 0   Disk     FUJITSU MAP3367N SUN36G 0401
ok
```

Not all systems support the `probe-scsi` command. For example, the `probe-scsi` command is absent from SunBlade 1500 systems:

```
ok probe-scsi
probe-scsi ?
ok
```

The `probe-scsi-all` Command

The `probe-scsi-all` command identifies the peripheral devices that are attached to the on-board SCSI controller and all peripheral devices attached to additional SBus or PCI SCSI controllers.

The following example from a Sun Fire V120 system lists three disks in an array attached to the external SCSI initiator identified by the `/pci@1f,0/pci@1/scsi@8,1` path, and the two internal SCSI disks attached to the internal SCSI initiator identified by the `/pci@1f,0/pci@1/scsi@8 path`. Both of these SCSI initiators are found on the system board of the V120, but the `probe-scsi-all` command is required to show the initiator used to connect external devices.

```
ok probe-scsi-all
/pci@1f,0/pci@1/scsi@8,1
Target 0
  Unit 0   Disk     SEAGATE ST373307LSUN72G 0507
Target 1
  Unit 0   Disk     SEAGATE ST373307LSUN72G 0507
Target 2
  Unit 0   Disk     SEAGATE ST373307LSUN72G 0507

/pci@1f,0/pci@1/scsi@8
```

```
Target 0
  Unit 0   Disk     FUJITSU MAP3367N SUN36G 0401
Target 1
  Unit 0   Disk     FUJITSU MAP3367N SUN36G 0401

ok
```

The `Target` and `Unit` values identify the *target* and *disk* (logical unit number, or LUN) values used in the physical device paths for these disks. Each disk attached to the same SCSI initiator will use a unique target value. The disk or LUN value used in the physical device paths for these disks is 0. This is true for disks that use embedded target controllers.

The `probe-ide` Command

The `probe-ide` command identifies disks and CD-ROMs that are attached to the on-board IDE controller. This command displays the device numbers of internal devices. This output from a SunBlade 1500 system lists two internal disks and one DVD-ROM drive.

```
ok probe-ide
  Device 0  ( Primary Master )
          ATA Model: ST3120026A

  Device 1  ( Primary Slave )
          ATA Model: ST3120026A

  Device 2  ( Secondary Master )
         Removable ATAPI Model: TOSHIBA ODD-DVD SD-
R1512

  Device 3  ( Secondary Slave )
         Not Present

ok
```

On the SunBlade 1500, the `Device` values of the disks, in this case, 0 and 1, reflect the *target* values used in the physical device paths for these disks. The *disk* (logical unit number, or LUN) value used in the physical device paths for these disks is 0.

Identifying the System Boot Device

Sun hardware uses the concept of a device tree to organize devices that are attached to the system.

Figure 9-5 shows the organizational structure of a device tree for an Ultra 5 or an Ultra 10 workstation.

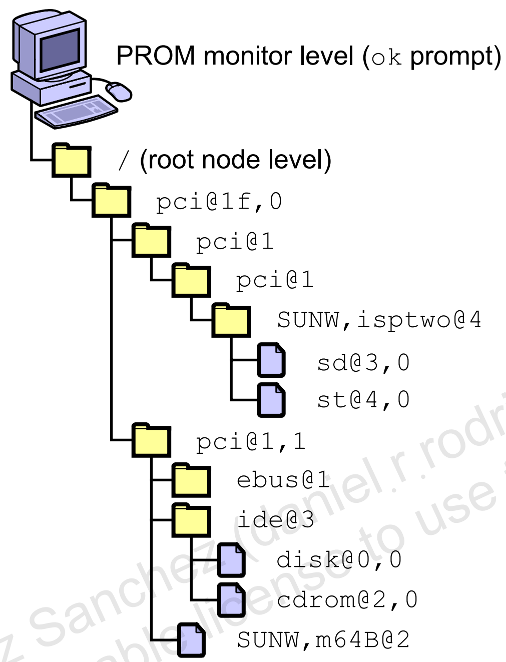


Figure 9-5 Partial Device Tree for an Ultra 5 or Ultra 10 Workstation

Note – In Figure 9-5, some license was taken in naming these directories to simplify the illustration.

The OpenBoot firmware builds the device tree from information gathered at the POST. This device tree is loaded into memory and is used by the `kernel` during the boot process to identify all configured devices. The top of the device tree is the root device node. Following the root device node is a bus nexus node. Connected to a bus nexus node is a leaf node, typically a controller for an attached device.

In Figure 9-5, the disk@0,0 device is the IDE device for the hard disk drive, and the cdrom@2,0 device is the IDE device for the CD-ROM drive. Both are attached to the IDE controller ide@3. Similarly, the sd@3,0 device is the SCSI disk device and the st@4,0 device is the SCSI tape device. Both are attached to the PCI-based SCSI controller SUNW,isptwo@4.

The paths built in the device tree by the OpenBoot firmware vary depending on the type of system and its device configuration.

Figure 9-6 shows a sample disk device path on an Ultra workstation with a PCI bus.

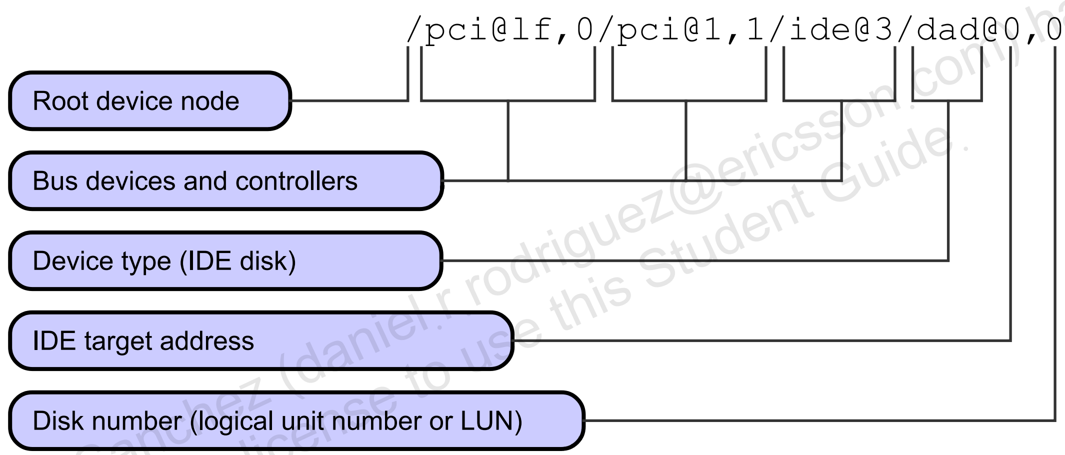


Figure 9-6 Disk Device Path – Ultra Workstation With a PCI IDE Bus

Figure 9-7 shows a sample disk device path on an Ultra workstation with a PCI-SCSI bus.

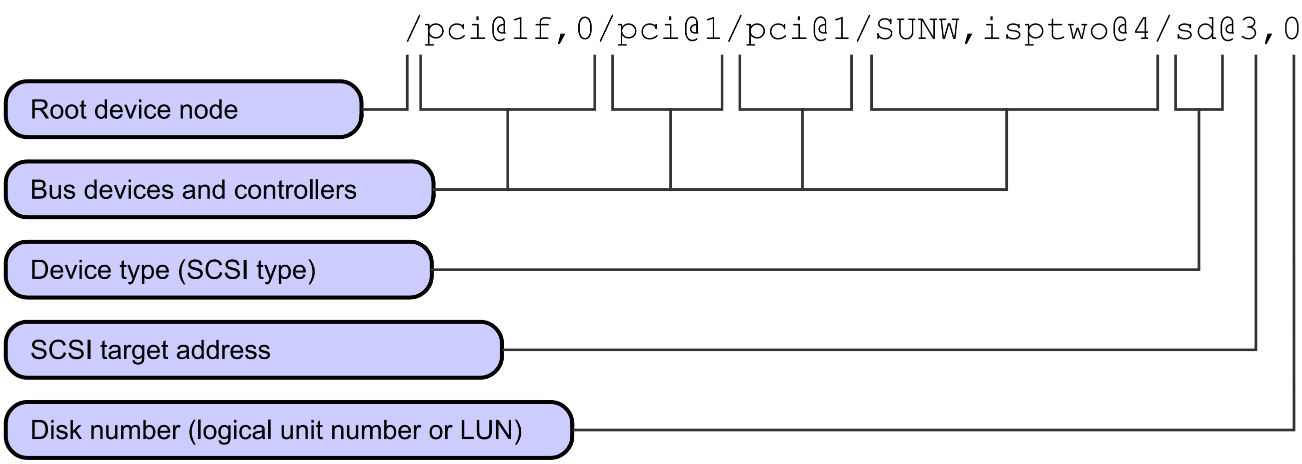


Figure 9-7 Disk Device Path – Ultra Workstation With a PCI-SCSI Bus

The show-devs Command

Use the show-devs command to view the entire device tree. The following example shows partial output from the show-devs command on an Ultra 5 or Ultra 10 system:

```
ok show-devs
/SUNW,UltraSPARC-IIi@0,0
/pci@1f,0
/virtual-memory
/memory@0,10000000
/pci@1f,0/pci@1
/pci@1f,0/pci@1,1
/pci@1f,0/pci@1,1/ide@3
/pci@1f,0/pci@1,1/SUNW,m64B@2
/pci@1f,0/pci@1,1/network@1,1
/pci@1f,0/pci@1,1/ebus@1
/pci@1f,0/pci@1,1/ide@3/cdrom
/pci@1f,0/pci@1,1/ide@3/disk
/pci@1f,0/pci@1,1/ebus@1/SUNW,CS4231@14,200000
/pci@1f,0/pci@1,1/ebus@1/flashprom@10,0
/pci@1f,0/pci@1,1/ebus@1/eeprom@14,0
/pci@1f,0/pci@1/pci@1
/pci@1f,0/pci@1/pci@1/SUNW,isptwo@4
(output omitted)
ok
```

The following example shows partial output from the show-devs command on a SunBlade 1500 system:

```
ok show-devs
/i2c@1f,464000
/pci@1f,700000
/ppm@1e,0
/pci@1e,600000
/memory-controller@0,0
/SUNW,UltraSPARC-IIIi@0,0
/virtual-memory
/memory@m0,0
/aliases
/options
/openprom
/chosen
/packages
/i2c@1f,464000/idprom@0,ae
/i2c@1f,464000/nvram@0,ae
```

```
/pci@1f,700000/SUNW,XVR-100@3
/pci@1f,700000/network@2
/pci@1e,600000/pci@3
/pci@1e,600000/pci@2
/pci@1e,600000/ide@d
/pci@1e,600000/usb@b
/pci@1e,600000/usb@a
/pci@1e,600000/sound@8
/pci@1e,600000/pmu@6
/pci@1e,600000/isa@7
/pci@1e,600000/pci@3/SUNW,isptwo@4
/pci@1e,600000/pci@3/SUNW,hme@0,1
/pci@1e,600000/pci@3/pci108e,1000@0
/pci@1e,600000/pci@3/SUNW,isptwo@4/st
/pci@1e,600000/pci@3/SUNW,isptwo@4/sd
(output omitted)
ok
```

Note – In addition to the `show-devs` command, use the following additional OpenBoot PROM commands to view specific device information: `show-ttys`, `show-displays`, `show-nets`, `show-disks`, and `show-tapes`.

The devalias Command

Use the `devalias` command to display the list of defined device aliases on your system. Device aliases provide short names used to identify longer physical device paths. This example shows output from the `devalias` command on an Ultra 5 or Ultra 10 system:

```
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
```

This example shows output from the `devalias` command on a SunBlade 1500 system:

```
ok devalias
screen                   /pci@1f,700000/SUNW,XVR-100@3
mouse                    /pci@1e,600000/usb@b/mouse@2
keyboard                 /pci@1e,600000/usb@b/keyboard@1
net                      /pci@1f,700000/network@2
disk                     /pci@1e,600000/ide@d/disk@0,0
cdrom                    /pci@1e,600000/ide@d/cdrom@2,0:f
ide                      /pci@1e,600000/ide@d
ttyb                     /pci@1e,600000/isa@7/serial@0,2e8
ttya                     /pci@1e,600000/isa@7/serial@0,3f8
name                     aliases
ok
```

The left side of the `devalias` output lists the device alias names, and the right side of the output lists the physical address of each device.

Predefined device aliases are built into the OpenBoot PROM firmware, and they are easier to remember and use than the physical device addresses. The disk device alias typically identifies the default boot device for the system. Some systems list more predefined device aliases than others.

The boot-device Parameter

The value of the boot-device parameter specifies the device from which the system will boot by default. The default value of the boot-device parameter is disk net. Both disk and net are device aliases.

Use the printenv command to view the current setting of the boot-device parameter. For example:

```
ok printenv boot-device
boot-device =         disk net
ok
```

To determine what devices the disk and net device aliases represent, use the devalias command. For example:

```
ok devalias disk
disk                        /pci@1e,600000/ide@d/disk@0,0
ok devalias net
net                         /pci@1f,700000/network@2
ok
```

To boot the system from the default device, use the boot command:

```
ok boot
```

With the boot-device parameter set to these two values, the system attempts to boot from the disk device first, and attempts to boot from the net device only if it cannot boot from disk.

Knowing what physical device it is that the disk device alias represents is useful when you use the show-disks command to list disk devices that are attached to your system.

Creating and Removing Custom Device Aliases

A portion of the NVRAM called NVRAMRC contains registers to hold custom parameters and is also reserved for storing new device alias names. External devices do not, by default, have built-in device aliases associated with them.

The NVRAMRC is affected by the commands `nvalias` and `nvunalias`, and the `use-nvramrc?` parameter.

The `nvalias` Command

You use the `nvalias` command to create a new device alias name to access a newly attached external device. The command format is:

```
nvalias aliasname device_path
```

The effect of the `nvalias` command is to store the following command line in the NVRAMRC:

```
devalias aliasname device_path
```

The following example procedure shows how to use the `nvunalias` command to create a user-defined device alias, and how to boot the system using the new device alias.

Note – A shortcut that the `show-disks` command provides enables you to select a device path and use the Control-Y keys to copy the device path onto the command line.

The following example uses the devalias and show-disks commands to select the device path for the disk. It then uses the nvalias command to create a new device alias called mydisk.

1. Verify that the boot-device parameter is set to disk net.

```
ok printenv boot-device
boot-device =          disk net
ok
```

2. Use the devalias command to display the device path that the disk device alias represents:

```
ok devalias disk
disk                    /pci@1e,600000/ide@d/disk@0,0
ok
```

   The disk@0,0 device identifies target 0, and disk or LUN 0, attached to the /pci@1e,600000/ide@d controller.

3. Use the show-disks command to display the list of disk types available on your system.

```
ok show-disks
a) /pci@1e,600000/pci@3/SUNW,isptwo@4/sd
b) /pci@1e,600000/ide@d/cdrom
c) /pci@1e,600000/ide@d/disk
q) NO SELECTION
Enter Selection, q to quit:
```

4. Enter the letter that selects the disk type that matches the current boot disk.

```
Enter Selection, q to quit: c
/pci@1e,600000/ide@d/disk has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y
         for creating devalias mydev for /pci@1e,600000/ide@d/disk
ok
```

5. Use the nvalias command to define the new mydisk alias, and enter Control-Y to paste the device path you selected from the show-disks command into the command line.

```
ok nvalias mydisk ^y
```

   The device path inserts into the command line:

```
ok nvalias mydisk /pci@1e,600000/ide@d/disk
```

6. Complete the disk device path by adding the target and disk (LUN) numbers so the device path matches the device that the `disk` device alias represents, and press Return.

```
ok nvalias mydisk /pci@1e,600000/ide@d/disk@0,0
ok
```

7. Use the `devalias` command to verify that the new device alias is defined.

```
ok devalias mydisk
mydisk                   /pci@1e,600000/ide@d/disk@0,0
ok
```

8. Use the `printenv` command to display the contents of the NVRAMRC registers. The `devalias` command used at boot time to establish the mydisk device alias should display.

```
ok printenv nvramrc
nvramrc =            devalias mydisk /pci@1e,600000/ide@d/disk@0,0

ok
```

9. Set the `boot-device` parameter to the new value, in this case `mydisk`, and boot the system.

```
ok setenv boot-device mydisk
boot-device =       mydisk
ok boot
```

When you create a device alias and use the partial path that the `show-disks` command provides, you must enter the correct target and disk (LUN) values to complete the device path.

A device path that ends with the disk (LUN) value implies that slice 0 on that disk is the boot slice.

To specify that a slice other than slice 0 is the boot slice, you must enter a letter that identifies the correct slice. For this purpose, slices 0 through 7 are represented by the letters `a` through `h`.

For example, the device path that ends with `disk@0,0` is interpreted to mean `disk@0,0:a`, where `:a` specifies slice 0.

If slice 3 was the boot slice on this disk, you would enter `disk@0,0:d` to specify the correct device path. The complete `nvalias` command would look like this:

```
ok nvalias mydisk /pci@1e,600000/ide@d/disk@0,0:d
```

The `nvunalias` Command

Use the `nvunalias` command to remove an alias name.

To remove a custom device alias name, use the following command format:

```
ok nvunalias aliasname
```

Note – The `nvunalias` command is the single exception to the rule that changes to NVRAM occur immediately and do not require a `reset-all` command.

In this example procedure shows how to use the `nvunalias` command to remove a user-defined device alias.

1. Use the `devalias` command to verify that the `mydisk` device alias is defined.

```
ok devalias mydisk
mydisk                    /pci@1e,600000/ide@d/disk@0,0
ok
```

2. Use the `printenv` command to verify that the NVRAMRC registers contain command used at boot time to establish the `mydisk` device alias.

```
ok printenv nvramrc
nvramrc =              devalias mydisk /pci@1e,600000/ide@d/disk@0,0

ok
```

3. Use the `nvunalias` command to remove the devalias command for `mydisk` from the NVRAMRC registers.

```
ok nvunalias mydisk
ok
```

4. Use the `printenv` command to verify that the NVRAMRC registers have been cleared.

```
ok printenv nvramrc
nvramrc =
ok
```

5. Use the `devalias` command to verify that the `mydisk` device alias is still defined. The `nvunalias` command removes the `devalias` command for `mydisk` from the NVRAMRC registers, but does not remove the `mydisk` alias that was defined when the system booted.

```
ok devalias mydisk
mydisk                        /pci@1e,600000/ide@d/disk@0,0
ok
```

6. Use the `set-default` command to set the `boot-device` parameter back to its default value. Verify the change.

```
ok set-default boot-device
ok printenv boot-device
boot-device =          disk net
ok
```

7. Run the `reset-all` command, and then verify that the `mydisk` alias is no longer defined. This procedure assumes that the `auto-boot?` parameter is set to `false`.

```
ok reset-all


Sun Blade 1500 (Silver), Keyboard Present
Copyright 1998-2004 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.16.4, 1024 MB memory installed, Serial #64512991.
Ethernet address 0:3:ba:d8:63:df, Host ID: 83d863df.


ok devalias mydisk
mydisk : no such alias
ok
```

Viewing and Changing NVRAM Parameters From the OS

Use the /usr/sbin/eeprom command to view and to change the NVRAM parameters while the Solaris OS is running.

Using the eeprom Command

Be aware of the following guidelines when using the eeprom command:

- Only the root user can change the value of a parameter.
- You must enclose parameters with a trailing question mark in single quotation marks (single quotes) when the command is executed in the C shell.

The following examples use the eeprom command to view and change NVRAM parameters.

- To list all of the parameters with their current values, perform the command:

```
# eeprom
asr-policy=normal
test-args: data not available.
diag-passes=1
local-mac-address?=true
fcode-debug?=false
scsi-initiator-id=7
oem-logo: data not available.
oem-logo?=false
(output omitted)
```

- To list a single parameter and its value, in this case, the boot-device parameter, perform the command:

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

- To change the value of the default boot device to disk2, perform the command:

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

Note – The predefined device alias `disk2` is present on Ultra 5 and Ultra 10 systems, and not present on SunBlade 1500 systems.

- To change the value of the `auto-boot?` parameter, perform the command:

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

Interrupting an Unresponsive System

When a system freezes or stops responding to the keyboard, you might have to interrupt it. When you interrupt the system, all active processes stop immediately, and the processor services the OpenBoot PROM exclusively. It does not allow you to flush memory or to synchronize file systems.

Aborting an Unresponsive System

To abort or interrupt an unresponsive system:

1. Attempt a remote login on the unresponsive system to locate and kill the offending process.
2. Attempt to reboot the unresponsive system gracefully.
3. Hold down the Stop-A key sequence on the keyboard of the unresponsive system. The system is placed at the `ok` prompt.

Note – If an ASCII terminal is being used as the system console, use the Break sequence keys.

---

4. Manually synchronize the file systems by using the OpenBoot PROM `sync` command.

```
ok sync
```

This command causes the system to panic, synchronize the file systems, perform a a crash dump of memory, and then reboot the system.

Exercise: Using Boot PROM Commands

In this exercise, you use the OpenBoot^{tm} PROM and Solaris OS commands to set and view OBP parameters, create a new device alias, and boot the system using the new alias.

Preparation

Refer to the lecture notes, as necessary, to complete the following steps and answer the questions listed.

This exercise lists examples from two systems, a V120 server and a Sun Blade 1500. These provide examples of systems that use SCSI and IDE internal disks.

Task

Complete the following steps:

1. If the Solaris OS is currently running, log in as the `root` user and halt your system.
2. When the `ok` prompt displays, use the `help` command to display the list of help topics.
3. Use the `help` command to display information about the `boot` command.

   What does the `help` command list for `boot`?

   ____

   ____

4. Use the `banner` command to obtain the following information:

   OpenBoot PROM revision: ____________________

   Megabytes of installed memory:____________________

   System type:____________________

   NVRAM serial number:____________________

   Ethernet address:____________________

   Host ID:____________________

5. Use the `printenv` command to display the list of OpenBoot PROM parameters and record the current values for the following parameters:

   ```
   output-device
   input-device
   auto-boot?
   boot-device
   ```

6. Prevent the system from booting automatically after using the `reset-all` command by setting the `auto-boot?` parameter to `false`.

7. Use the `reset-all` command to verify that the new `auto-boot?` value is in effect.

   The system should remain at the `ok` prompt after the `reset-all` command completes.

8. Use the `probe-scsi`, `probe-scsi-all` and `probe-ide` commands to display the list of devices attached to your system's SCSI and IDE busses.

   Not all of these commands are present on all systems.

   What are the main differences that you see in the information that these commands display?

   ____

   ____

   ____

9. Record the target number and device type (disk, tape, or CD-ROM) of all the devices shown by the `probe-scsi`, `probe-scsi-all` and `probe-ide` commands.

10. Verify that your default `boot-device` is set to `disk net`.

11. Use the `devalias` command to display the full device path for the `disk` alias and record the path name reported:

    ____

    ____

12. Use the `show-disks` command to select the device path that relates to the disk recorded in Step 11 and complete the following steps:

    a. Use the `nvalias` command to create a new device alias called `mydisk`.

    b. Set the `mydisk` alias to the path and disk name you recorded in Step 11.

Remember to use the Control-Y key sequence to paste the disk path into your nvalias command. You must manually complete the path to specify the disk you want to use.

13. Verify that the new alias is correctly set.
14. Use the printenv command to display the contents of the nvramrc file.

    What command does the nvramrc file contain that creates the mydisk alias?

    ____

    ____

15. Use the printenv command to display the setting of the use-nvramrc? parameter.

    What is the current setting of the use-nvramrc? parameter?

    ____

    ____

16. Boot your system using the mydisk alias.
17. Log in as the root user on your system and open a new terminal window.
18. Use the eeprom command to list all NVRAM parameters.
19. Use the eeprom command to list the setting of the boot-device parameter.
20. Use the eeprom command to set the boot-device parameter to the alias mydisk.
21. Bring your system to run level 0.
22. Verify that the eeprom command set the boot-device parameter to the alias mydisk.
23. Set the boot-device parameter to its default value and verify the setting.
24. Use the nvunalias command to remove the alias mydisk.
25. Verify that the mydisk alias is no longer in the nvramrc file.
26. Use the devalias command to see if the mydisk alias has been removed from the list of device aliases.

    Has it?

    ____

27. Run the `reset-all` command, and then check again if the `mydisk` alias has been removed from the list of device aliases.

    (If your system reboots, interrupt the reboot with a Stop-A key sequence.)

    Has it?

    ____

28. Set the OpenBoot PROM parameters back to their default values and boot the system from the default device.
29. Log in as the `root` user.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Using Boot PROM Commands

This section contains solutions to the exercise.

Task

Complete the following steps:

1. If the Solaris OS is currently running, log in as the `root` user and halt your system.

   `# init 0`

2. When the `ok` prompt displays, use the `help` command to display the list of help topics.

   V120 example:

```
ok help
Enter 'help command-name' or 'help category-name' for more help
(Use ONLY the first word of a category description)
Examples:  help select   -or-   help line
    Main categories are:
Breakpoints (debugging)
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
nvramrc (making new commands permanent)
ok
```

   Sun Blade 1500 example:

```
ok help
Enter 'help command-name' or 'help category-name' for more help
(Use ONLY the first word of a category description)
Examples:  help select   -or-   help line
    Main categories are:
```

```
Breakpoints (debugging)
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
eject devices
Power on reset
Diag (diagnostic routines)
Resume execution
File download and boot
nvramrc (making new commands permanent)
ok
```

3. Use the `help` command to display information about the `boot` command.

   Sun Blade 1500 example:

```
ok help boot
boot <specifier>  ( -- )    boot kernel ( default ) or other file
  Examples:
    boot                    - boot kernel from default device.
                              Factory default is to boot
                              from DISK if present, otherwise from NET.
    boot net                - boot kernel from network
    boot cdrom              - boot kernel from CD-ROM
    boot disk1:h            - boot from disk1 partition h
    boot tape               - boot default file from tape
    boot disk myunix -as    - boot myunix from disk with flags "-as"
dload <filename>  ( addr -- )     debug load of file over network at
address
  Examples:
     4000 dload /export/root/foo/test
     ?go        - if executable program, execute it
                  or if Forth program, compile it
ok
```

   What does the `help` command list for `boot`?

   The help command lists examples of boot command syntax used to boot the system from various devices.

4. Use the `banner` command to obtain the following information:

   OpenBoot PROM revision:

Megabytes of installed memory:

System type:

NVRAM serial number:

Ethernet address:

Host ID:

Each system presents its own unique information.

V120 example:

```
ok banner
Sun Fire V120 (UltraSPARC-IIe 648MHz), No Keyboard
OpenBoot 4.0, 2048 MB memory installed, Serial #57165033.
Ethernet address 0:3:ba:68:44:e9, Host ID: 836844e9.

ok
```

Sun Blade 1500 example:

```
ok banner

Sun Blade 1500 (Silver), Keyboard Present
Copyright 1998-2004 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.16.4, 1024 MB memory installed, Serial #64512991.
Ethernet address 0:3:ba:d8:63:df, Host ID: 83d863df.

ok
```

5. Use the `printenv` command to display the list of OpenBoot PROM parameters and record the current values for the following parameters:

   ```
   output-device  - screen
   input-device - keyboard
   auto-boot?  - true
   boot-device - disk net
   ```

```
ok printenv
Variable Name        Value                         Default Value

ras-shutdown-enabled?  false                       false
shutdown-temp        75                            75
warning-temp         70                            70
env-monitor          disabled                      disabled
diag-passes          1                             1
(output omitted)
```

6. Prevent the system from booting automatically after using the `reset-all` command by setting the `auto-boot?` parameter to `false`.

```
ok setenv auto-boot? false
auto-boot? =          false
ok
```

7. Use the `reset-all` command to verify that the new `auto-boot?` value is in effect.

   The system should remain at the `ok` prompt after the `reset-all` command completes.

```
ok reset-all
Resetting
LOM event: +14d+1h39m5s host reset
g ...

?
Sun Fire V120 (UltraSPARC-IIe 648MHz), No Keyboard
OpenBoot 4.0, 2048 MB memory installed, Serial #57165033.
Ethernet address 0:3:ba:68:44:e9, Host ID: 836844e9.


ok
```

8. Use the `probe-scsi`, `probe-scsi-all` and `probe-ide` commands to display the list of devices attached to your system's SCSI and IDE busses. Not all of these commands are present on all systems.

   V120 example:

```
ok probe-scsi
Target 0
  Unit 0   Disk     FUJITSU MAP3367N SUN36G 0401
Target 1
  Unit 0   Disk     FUJITSU MAP3367N SUN36G 0401
ok probe-scsi-all
/pci@1f,0/pci@1/scsi@8,1

/pci@1f,0/pci@1/scsi@8
Target 0
  Unit 0   Disk     FUJITSU MAP3367N SUN36G 0401
Target 1
  Unit 0   Disk     FUJITSU MAP3367N SUN36G 0401

ok probe-ide
  Device 0  ( Primary Master )
         Removable ATAPI Model: DV-28E-C
```

```
  Device 1  ( Primary Slave )
         Not Present

  Device 2  ( Secondary Master )
         Not Present

  Device 3  ( Secondary Slave )
         Not Present

ok
```

Sun Blade 1500 example:

```
ok probe-scsi
probe-scsi ?
ok probe-scsi-all
/pci@1e,600000/pci@3/SUNW,isptwo@4
Target 4
  Unit 0   Removable Tape     HP      C1537A          L706

ok probe-ide
  Device 0  ( Primary Master )
          ATA Model: ST3120026A

  Device 1  ( Primary Slave )
          ATA Model: ST3120026A

  Device 2  ( Secondary Master )
         Removable ATAPI Model: TOSHIBA ODD-DVD SD-R1512

  Device 3  ( Secondary Slave )
         Not Present

ok
```

What are the main differences that you see in the information that these commands display?

The `probe-scsi` *command only lists devices on the built-in SCSI chain and does not list the full device paths. The* `probe-scsi-all` *command lists all devices on all SCSI chains and their full device paths. The* `probe-ide` *command reports the list of IDE devices attached to the system.*

9. Record the target number and device type (disk, tape, or CD-ROM) of all the devices shown by the `probe-scsi`, `probe-scsi-all` and `probe-ide` commands.

Each system presents its own unique information.

10. Verify that your default `boot-device` is set to `disk net`.

```
ok printenv boot-device
boot-device =        disk net
ok
```

11. Use the `devalias` command to display the full device path for the `disk` alias.

```
ok devalias disk
```

Record the path name reported:

This differs from system to system. On an Ultra™ 5 workstation, the alias is defined as follows:

`/pci@1f,0/pci@1,1/ide@3/disk@0,0`

On a Sun Fire™ V120 server, the alias is defined as follows:

/pci@1f,0/pci@1/scsi@8/disk@0,0

On a Sun Blade™ 1500 system, the alias is defined as follows:

/pci@1e,600000/ide@d/disk@0,0

12. Use the `show-disks` command to select the device path that relates to the disk recorded in Step 11 and complete the following steps:

a. Use the `nvalias` command to create a new device alias called `mydisk`.

b. Set the `mydisk` alias to the path and disk name you recorded in Step 11.

Remember to use the Control-Y key sequence to paste the disk path into your `nvalias` command. You must manually complete the path to specify the disk you want to use.

V120 example:

```
ok show-disks
a) /pci@1f,0/pci@1/scsi@8,1/disk
b) /pci@1f,0/pci@1/scsi@8/disk
c) /pci@1f,0/pci@1,1/ide@d/cdrom
d) /pci@1f,0/pci@1,1/ide@d/disk
e) /pci@1f,0/pci@1,1/ebus@c/SUNW,lomh@14,200000
q) NO SELECTION
Enter Selection, q to quit: b
/pci@1f,0/pci@1/scsi@8/disk has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y
         for creating devalias mydev for
```

```
/pci@1f,0/pci@1/scsi@8/disk
ok nvalias mydisk /pci@1f,0/pci@1/scsi@8/disk@0,0
ok
```

Sun Blade 1500 example:

```
ok show-disks
a) /pci@1e,600000/pci@3/SUNW,isptwo@4/sd
b) /pci@1e,600000/ide@d/cdrom
c) /pci@1e,600000/ide@d/disk
q) NO SELECTION
Enter Selection, q to quit: c
/pci@1e,600000/ide@d/disk has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y
         for creating devalias mydev for
/pci@1e,600000/ide@d/disk
ok nvalias mydisk /pci@1e,600000/ide@d/disk@0,0
ok
```

13. Verify that the new alias is correctly set.

V120 example:

```
ok devalias mydisk
mydisk
/pci@1f,0/pci@1/scsi@8/disk@0,0
ok
```

Sun Blade 1500 example:

```
ok devalias mydisk
mydisk                    /pci@1e,600000/ide@d/disk@0,0
ok
```

14. Use the `printenv` command to display the contents of the `nvramrc` file.

```
ok printenv nvramrc
```

What command does the `nvramrc` file contain that creates the `mydisk` alias?

Systems differ according to the disk devices they use. An Ultra 5 workstation would report the following:

```
nvramrc = devalias mydisk /pci@1f,0/pci@1,1/ide@3/disk@0,0
```

A Sun Fire V120 server would report the following:

```
nvramrc = devalias mydisk /pci@1f,0/pci@1/scsi@8/disk@0,0
```

A SunBlade 1500 system would report the following:

```
nvramrc = devalias mydisk /pci@1e,600000/ide@d/disk@0,0
```

15. Use the printenv command to display the setting of the use-nvramrc? parameter.

```
ok printenv use-nvramrc?
use-nvramrc? =          true
ok
```

What is the current setting of the use-nvramrc? parameter?

true

16. Boot your system using the mydisk alias.

```
ok boot mydisk
(output omitted)
```

17. Log in as the root user on your system and open a new terminal window.

18. Use the eeprom command to list all NVRAM parameters.

```
# eeprom
ras-shutdown-enabled?=false
shutdown-temp=75
warning-temp=70
env-monitor=disabled
diag-passes=1
diag-continue?=0
diag-targets=0
(output omitted)
```

19. Use the eeprom command to list the setting of the boot-device parameter.

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

20. Use the eeprom command to set the boot-device parameter to the alias mydisk.

```
# eeprom boot-device=mydisk
```

21. Bring your system to run level 0.

```
# init 0
```

22. Verify that the eeprom command set the boot-device parameter to the alias mydisk.

```
ok printenv boot-device
boot-device =           mydisk
ok
```

23. Set the boot-device parameter to its default value and verify the setting.

```
ok set-default boot-device
ok printenv boot-device
boot-device =          disk net
ok
```

24. Use the `nvunalias` command to remove the alias `mydisk`.

```
ok nvunalias mydisk
```

25. Verify that the `mydisk` alias is no longer in the `nvramrc` file.

```
ok printenv nvramrc
nvramrc =
ok
```

26. Use the `devalias` command to see if the `mydisk` alias has been removed from the list of device aliases.

```
ok devalias mydisk
mydisk
/pci@1f,0/pci@1/scsi@8/disk@0,0

ok
```

Has it?

No

27. Run the `reset-all` command, and then check again if the `mydisk` alias has been removed from the list of device aliases.

```
ok reset-all
```

(If your system reboots, interrupt the reboot with a Stop-A key sequence.)

```
ok devalias mydisk
mydisk : no such alias
ok
```

Has it?

Yes

28. Set the OpenBoot PROM parameters back to their default values and boot the system from the default device.

```
ok set-defaults
ok printenv
Variable Name         Value                          Default Value

ras-shutdown-enabled?  false                         false
shutdown-temp         75                             75
warning-temp          70                             70
env-monitor           disabled                       disabled
diag-passes           1                              1
(output omitted)
ok reset-all
```

29. Log in as the `root` user.

Notes:

Module 10

Using the Grand Unified Bootloader

Objectives

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

- Discuss the purpose of the Grand Unified Bootloader (GRUB)
- Describe GRUB terminology and basic functions
- Modify x86 system boot behavior in the Solaris OS
- Use the `eeprom` command
- Use the `kernel` command
- Use the `findroot` command
- Describe and manage GRUB boot archives
- Boot a system in the GRUB-based boot environment
- Interrupt an unresponsive system

Introduction to GRUB

Starting with the Solaris 10 1/06 release, the open source GNU GRUB is available on x86-based systems that are running the Solaris OS. GRUB is the boot loader that is responsible for loading a boot archive into a system's memory. The boot archive contains the kernel modules and configuration files that are required to boot the system.

To support GRUB, x86-based systems must have at least 256 Mbytes of main memory.

GRUB Terminology

The following GRUB terms are used throughout this module, and are helpful in understanding GRUB concepts.

- Boot archive

  A boot archive is a collection of critical files that is used to boot the Solaris OS. These files are needed during system startup before the root (/) file system is mounted. Two boot archives are maintained on a system:

  - The boot archive that is used to boot the Solaris OS on a system. This boot archive is sometimes called the primary boot archive.
  - The boot archive that is used for recovery when the primary boot archive is damaged. This boot archive starts the system without mounting the root (/) file system. On the GRUB menu, this boot archive is called `Solaris failsafe`. The failsafe archive's essential purpose is to boot the system and regenerate the primary boot archive, which is usually used to boot the system.

- Boot loader

  The boot loader is the first software program that runs after you turn on a system. This program begins the booting process.

- GRUB edit menu

  A boot menu that is a submenu of the GRUB main menu. GRUB commands are displayed on this menu. These commands can be edited to change boot behavior.

- GRUB main menu

  A boot menu that lists the operating systems that are installed on a system. From this menu, you can easily boot an operating system without modifying the BIOS or `fdisk` partition settings.

- `menu.lst` file

  A file that lists all the operating systems that are installed on a system. The contents of this file dictate the list of operating systems that is displayed on the GRUB menu. From the GRUB menu, you can easily boot an operating system without modifying the BIOS or `fdisk` partition settings.

- Miniroot

  A minimal, bootable root (/) file system that resides on the Solaris installation media. A miniroot consists of the Solaris software that is required to install and upgrade systems. On x86-based systems, the miniroot is copied to the system to be used as the failsafe boot archive.

GRUB-Based Booting

The boot loader is the first software program that runs after you turn on an x86-based system. This program begins the boot process. After an x86-based system is powered on, the Basic Input/Output System (BIOS) initializes the CPU, the memory, and the platform hardware. When the initialization phase has completed, the BIOS loads the boot loader from the configured boot device, and then gives control of the system to the boot loader.

GRUB implements a simple menu interface that includes boot options that are predefined in a configuration file. GRUB also has a command-line interface that is accessible from the menu interface for performing various boot commands. The GRUB implementation on x86-based systems in the Solaris OS is compliant with the Multiboot Specification, which is described in detail at `http://www.gnu.org/software/grub/grub.html`.

Because the Solaris kernel is fully compliant with the Multiboot Specification, you can boot x86-based systems using GRUB. With GRUB, you can more easily boot the various operating systems that are installed on your system.

For example, on one x86-based system, you could individually boot the following operating systems:

- Solaris OS
- Linux
- Microsoft Windows

When you boot an x86-based system, the GRUB menu displays. This menu provides a list of boot entries from which to choose. A boot entry is an OS instance that is installed on your system.

The GRUB main menu is based on a configuration file that you can modify.

- If you install or upgrade the Solaris OS, the GRUB menu is automatically updated. The Solaris OS is displayed as a new boot entry.
- If you install an operating system other than the Solaris OS, you must modify the `/boot/grub/menu.lst` configuration file to include the new OS instance. Adding the new OS instance enables the new boot entry to appear in the GRUB menu the next time that you boot the system.

How GRUB-Based Booting Works

After GRUB gains control of the system, the GRUB menu displays on the console.

In the GRUB main menu, you can do the following:

- Select a boot entry.
- Modify a boot entry using the GRUB edit menu.
- Load an OS kernel from the command line manually.

The following is a sample of a typical GRUB main menu. This menu lists the boot entries to choose from. If you do not choose an entry, the system boots the default OS. The default OS is usually the first entry that is displayed in the GRUB main menu.

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-------------------------------------------------------------------+
| Solaris 10 10/09 s10x_u8wos_08a X86
| Solaris failsafe
| Diagnostic Partition
|
+-------------------------------------------------------------------+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

Use the arrow keys to highlight a boot entry, then press Enter to boot the system using the selected entry.

If you want to edit a boot entry using the GRUB edit menu, use the arrow keys to highlight a boot entry in the GRUB main menu. Then, type `e` to display the GRUB edit menu. If you want to use the command-line interface, use the arrow keys to select a boot entry in the GRUB main menu, then type `c`.

When the system boots, GRUB loads the primary boot archive and the `multiboot` program. The primary boot archive, named `/platform/i86pc/boot_archive`, is a ram image of the file system that contains the Solaris kernel modules and data.

GRUB puts this information in memory, without any interpretations. After the operation has completed, GRUB gives control of the system to the multiboot program. GRUB itself then becomes inactive, and system memory is reclaimed.

By reading the boot archive and passing boot-related information to the kernel, the multiboot program is responsible for assembling core kernel modules into memory. The Multiboot Specification specifies the boot-related information.

GRUB Device Naming Conventions

GRUB uses device naming conventions that are slightly different from previous Solaris OS versions. Understanding the GRUB device naming conventions can assist you in correctly specifying drive and partition information when you configure GRUB on your system.

The following table describes the GRUB device naming conventions.

Table 10-1 Naming Conventions for GRUB Devices

Device Name	**Description**
`(fd0), (fd1)`	First diskette, second diskette
`(nd)`	Network device
`(hd0,0), (hd0,1)`	First and second `fdisk` partition of first bios disk
`(hd0,0,a), (hd0,0,b)`	Solaris/BSD slice 0 and 1 (`a` and `b`) on the first `fdisk` partition on the first bios disk

Note – All GRUB device names must be enclosed in parentheses. Partition numbers are counted from 0 (zero), not from 1.

GRUB Functional Components

GRUB consists of the following functional components:

- `stage1` – Is installed on the first sector of the Solaris `fdisk` partition.
- `stage2` – Is installed in a reserved area in the Solaris `fdisk` partition. `stage2` is the core image of GRUB.
- `menu.lst` – Is a file that is typically located in the `/boot/grub` directory. The GRUB `stage2` file reads this file.

You cannot use the `dd` command to write `stage1` and `stage2` to disk. `stage1` must be able to receive information about the location of `stage2` on the disk. Use the `installgrub` command, which is the supported method for installing GRUB boot blocks.

The GRUB Menu

By default, the Solaris menu file resides in `/boot/grub/menu.lst`. The contents of this file dictate what is displayed in the GRUB menu when you boot the system. The GRUB menu file contains entries for all the OS instances that are installed on your system, as well as other important boot directives.

Note – In some situations, the GRUB `menu.lst` file resides elsewhere. For example, on a system that uses Solaris Live Upgrade, the GRUB `menu.lst` file might exist on a boot environment that is not the currently running boot environment. Or, if you have upgraded a system with an x86-based boot partition, the `menu.lst` file might reside in the `/stubboot` directory.

Only the active GRUB `menu.lst` file is used to boot the system. In order to modify the GRUB menu that is displayed when you boot the system, the active GRUB `menu.lst` file must be modified. Changing any other GRUB `menu.lst` file has no effect on the menu that is displayed when you boot the system.

To determine the location of the active GRUB `menu.lst` file, use the `bootadm` command. The `list-menu` subcommand displays the location of the active GRUB menu. For example:

```
# bootadm list-menu
The location for the active GRUB menu is: /boot/grub/menu.lst
default 0
timeout 10
0 Solaris 10 11/06 s10x_u3wos_10 X86
1 Solaris failsafe
#
```

You might need to customize the `menu.lst` file for one of the following reasons:

- To add new OS entries
- To add GRUB console redirection information
- To change an aspect of system boot behavior that is not controlled using the `eeprom` or `kernel` commands

The instructions for the syntax to use to edit the file can be found in the /boot/grub/menu.lst file comments. Because these changes are not directly related to the Solaris OS, you cannot make them using the eeprom command.

When you install the Solaris OS, two GRUB menu entries are installed on the system by default. The first entry is the Solaris OS entry. The second entry is the miniroot (failsafe) archive, which is intended to be used for system recovery.

The GRUB menu entries are installed and then updated automatically as part of the Solaris software installation and upgrade process. The OS directly manages these entries and they should not be manually edited.

Modifying menu.lst to Prevent Automatic System Boot

The timeout value in the menu.lst file determines if the system will boot automatically, and if so, how long the GRUB main menu displays before the boot process proceeds.

To prevent the system from booting automatically, and cause the GRUB main menu to remain displayed until you provide a response, edit the menu.lst file and set the timeout value to -1. For example:

```
timeout -1
```

A positive timeout value specifies the timeout in seconds before the system boots from the default source. The comments in the /boot/grub/menu.lst file describe this parameter.

Although you can use the eeprom command to set the auto-boot? parameter to false in the /boot/solaris/bootenv.rc file, this has no effect on the x86 system boot process.

Modifying x86 System Boot Behavior in the Solaris OS

This section describes the methods you use to modify boot behavior in the Solaris OS on x86 systems. The primary ways you can modify boot behavior are:

- Use the `eeprom` command to assign a different value to a standard set of properties. These values, which are the equivalent to the SPARC OpenBoot PROM NVRAM variables, are stored in the `/boot/solaris/bootenv.rc` file. Changes that are made to the Solaris boot behavior using the `eeprom` command persist over each system reboot.
- Use the `kernel` command, which is available in the GRUB menu when the system boots. Boot entries listed in the GRUB main menu provide access to a list of commands used to boot the system, including the `kernel` command. Editing `kernel` command in the GRUB menu allows you change the system boot behavior for a single boot cycle.

Note – Changes that you make with the `kernel` command override options that you set using the `eeprom` command. These changes remain in effect until the next time you boot the system.

You can also modify the Solaris boot behavior in the GRUB `menu.lst` file. Modifying the Solaris boot behavior this way is similar to using the `eeprom` command, but isn't recommended because the Solaris OS entries in the `menu.lst` file could be modified during a software upgrade. In this instance, any changes that you made to the file are lost. In contrast, changes you make using the `eeprom` command are added to the `/boot/solaris/bootenv.rc` file. This file is handled properly during an upgrade.

Setting Solaris Boot Parameters Using `eeprom`

Display or set Solaris boot parameters on x86 systems using the `eeprom` command. The parameters that you set with the `eeprom` command are stored in the `/boot/solaris/bootenv.rc` file. Changes that you make using the `eeprom` command persist across system reboots. However, you can override `eeprom` settings if you boot the system by specifying options and arguments to the `kernel` command in the GRUB menu.

Use the eeprom command to display the current values stored in the bootenv.rc file. For example:

```
kbd-type=US-English
ata-dma-enabled=1
atapi-cd-dma-enabled=0
ttyb-rts-dtr-off=false
ttyb-ignore-cd=true
ttya-rts-dtr-off=false
ttya-ignore-cd=true
ttyb-mode=9600,8,n,1,-
ttya-mode=9600,8,n,1,-
lba-access-ok=1
prealloc-chunk-size=0x2000
bootpath=/pci@0,0/pci-ide@5/ide@0/cmdk@0,0:a
console=text
bootpath=/pci@0,0/pci-ide@5/ide@0/cmdk@0,0:a
```

For more information about the eeprom command and parameters, see the eeprom(1M) man page.

Note – Not all systems support all eeprom parameters. Defaults vary depending on the system configuration. Use the following procedure to display and set boot parameters using the eeprom command.

---

1. Become superuser or assume an equivalent role.
2. To check if a parameter is listed in the /boot/solaris/bootenv.rc file and view its current value, use the eeprom command and specify the parameter name. For example, if the parameter is not listed in the bootenv.rc file, the eeprom command reports that data is not available for it:

   ```
   # eeprom boot-file
   boot-file: data not available.
   ```

   If the parameter is listed in the bootenv.rc file, but is set to a null value, the eeprom command reports the parameter name, and no value:

   ```
   # eeprom boot-file
   boot-file=
   ```

3. To set or change a parameter, enter the `eeprom` command with the appropriate arguments. For example, to cause the system to boot using the 64-bit UNIX kernel found in the `/platform/i86pc/kernel/amd64 directory`, you would use the following command:

```
# eeprom boot-file=kernel/amd64/unix
```

4. Use the `eeprom` command to verify that the new parameter has been set.

```
# eeprom boot-file
boot-file=kernel/amd64/unix
```

5. You can use the `grep` command to view the entry that the `eeprom` command placed in the `bootenv.rc` file.

```
# grep boot-file /boot/solaris/bootenv.rc
setprop boot-file 'kernel/amd64/unix'
```

6. To remove a setting for a parameter, use double quotes to specify a null parameter value. For example, setting the `boot-file` parameter to a null value on an x86 system causes the system to use its default autodetect boot behavior:

```
# eeprom boot-file=""
# eeprom boot-file
boot-file=
```

Identifying the Boot Device

Use the following `eeprom` command to identify the boot device the system uses:

```
# eeprom bootpath
bootpath=/pci@0,0/pci-ide@5/ide@0/cmdk@0,0:a
#
```

Modifying Boot Behavior Using the `kernel` Command

You can modify the Solaris boot behavior by editing `kernel` commands in the GRUB menu when the system first boots.

When you first power on an x86 system, the GRUB main menu displays. For example:

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+----------------------------------------------------------------------+
| Solaris 10 10/09 s10x_u8wos_08a X86
| Solaris failsafe
| Diagnostic Partition
|
+----------------------------------------------------------------------+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

Pressing the space bar or any key other than `e` interrupts the boot process an causes the GRUB main menu to remain displayed. To display the command list you want to edit, highlight the desired boot entry in the GRUB main menu, and enter `e`.

Typing `e` when the GRUB main menu displays interrupts the boot process, and displays the command list for the default boot entry. For example:

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+----------------------------------------------------------------------+
| kernel /platform/i86pc/multiboot
| module /platform/i86pc/boot_archive
|
+----------------------------------------------------------------------+
```

Each line in this menu is an individual command for the boot entry that you chose in the GRUB main menu. To modify the boot behavior for the boot entry you have selected, edit the `kernel` command for that boot entry.

To edit a kernel command, highlight the command in the command list, and enter e. The GRUB edit menu displays the kernel command you selected. For example:

```
[ Minimal BASH-like line editing is supported. FOr the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits. ]

grub edit> kernel /platform/i86pc/multiboot
```

Typically you would edit a kernel command to add an option that affects the boot process. For example, to cause the system to boot to single-user mode, you would add the -s option. For example:

```
grub edit> kernel /platform/i86pc/multiboot -s
```

Enter Return when the kernel command reflects the options you want to use. The command list displays. For example:

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-------------------------------------------------------------------------+
| kernel /platform/i86pc/multiboot -s
| module /platform/i86pc/boot_archive
|
+-------------------------------------------------------------------------+
```

To boot the system using the new kernel command, enter b.

Using kernel Command Options and Arguments

Table 10-2 describes some of the kernel command options and arguments you can use to modify the boot behavior of x86 systems. See the kernel(1M) man page for more information.

Table 10-2 kernel Command Options and Arguments

Option or Argument	Description
-a	Asks the user for configuration information, such as where to find the system file, where to mount root, and even override the name of the kernel itself. Default responses will be contained in square brackets.

Table 10-2 kernel Command Options and Arguments (Continued)

Option or Argument	Description
-m *smf_options*	The -m option accepts the quiet and verbose options to control the type and amount of messages that smf(5) displays during boot, and the debug and milestone=*milestone_name* options to determine the services which are used to boot the system. For the milestone=*milestone_name* option, valid milestones are none, single-user, multi-user, multi-user-server, and all.
-r	Reconfiguration boot. The system will probe all attached hardware devices and configure the logical namespace in /dev.
-s	Boots only to init level 's'. See init(1M).
kernel_name	You can specify an alternate kernel to boot by supplying its name as a kernel command argument. The kernel/unix argument specifies the 32-bit kernel. The kernel/amd64/unix argument specifies the 64-bit kernel. Both of these kernels reside below the /platform/i86pc directory structure.

kernel Command Examples

Adding the -m option to the kernel command and specifying the single-user milestone boots the system to single-user mode. For example:

```
grub edit> kernel /platform/i86pc/multiboot -m milestone=single-user
```

Adding the -m option to the kernel command and specifying the verbose argument boots the system to the default milestone and displays verbose messages from smf. For example:

```
grub edit> kernel /platform/i86pc/multiboot -m verbose
```

Adding the kernel/unix argument to the kernel command boots the system to the default milestone using the 32-bit kernel. For example:

```
grub edit> kernel /platform/i86pc/multiboot kernel/unix
```

Adding the `kernel/amd64/unix` argument and the -s option to the `kernel` command boots the system using the 64-bit kernel to the `single-user` milestone. For example:

```
grub edit> kernel /platform/i86pc/multiboot kernel/amd64/unix -s
```

Note – Specify the `kernel/amd64/unix` argument only on systems capable of booting a 64-bit kernel.

Verifying Which Kernel the System is Using

If you used an `eeprom` or `kernel` command to specify a kernel for the system to boot, you may want to check if the system is using the kernel you specified.

You can use the `prtconf` command to print the system configuration, and the `grep` command to search for the path in which the kernel you specified resides. For example:

```
# prtconf -v | grep /platform/i86pc/kernel
            value='/platform/i86pc/kernel/amd64/unix'
#
```

The value displayed comes from the system property that lists the name of the kernel. For example:

```
name='whoami' type=string items=1
    value='/platform/i86pc/kernel/amd64/unix'
```

The `findroot` Command for x86

With the Solaris 10 5/09 release, all Solaris installation methods, including Solaris Live Upgrade, use the `findroot` command for specifying which disk slice to boot on an x86 based system. Previously, the root command, `root (hd0.0.a)`, was used to explicitly specify which disk slice to boot. This information is located in the `menu.lst` file that is used by GRUB. The most common form of the entry in the `menu.lst` file is:

```
findroot (rootfs0,0,a)
kernel$ /platform/i86pc/kernel/$ISADIR/unix
module$ /platform/i86pc/$ISADIR/boot_archive
```

The `findroot` command discovers the targeted disk, irrespective of the boot device. In addition, the `findroot` command provides enhanced support for booting systems with ZFS and UFS roots. In addition to the `findroot` command, the name of a signature file on the slice, (`<mysign>, 0, a`) is provided. The boot signature is located in the `/boot/grub/bootsign` directory on the system. The name of the signature file varies, depending on the installation method that is used.

For more information, see "Implementation of the `findroot` Command" in the *System Administration Guide: Basic Administration* guide.

GRUB Boot Archives

GRUB uses two kinds of boot archives:

- Failsafe boot archive
- Primary boot archive

A failsafe boot archive has the following benefits and characteristics:

- Is self-sufficient
- Can boot on its own
- Is created by default during installation of the OS
- Requires no maintenance

By default, the failsafe boot archive is `/boot/x86.miniroot-safe`.

A primary boot archive shadows a root (`/`) file system. This boot archive contains all of the kernel modules, `driver.conf` files, plus a few configuration files. These files are located in the `/etc` directory. The kernel reads the files in the boot archive before the root (`/`) file system is mounted. After the root (`/`) file system is mounted, the kernel discards the boot archive from memory. Then, file I/O is performed against the root device.

By default, the primary boot archive is `/platform/i86pc/boot_archive`. In the list of commands represented by a GRUB boot entry, the boot archive is listed as the argument to the `module` command. For example the `module` command for the `Solaris 10 11/06 s10x_u3wos_10 X86` GRUB boot entry lists the primary boot archive:

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-------------------------------------------------------------------------+
| root (hd0,1,a)
| kernel /platform/i86pc/multiboot
| module /platform/i86pc/boot_archive
|
+-------------------------------------------------------------------------+
```

The module command for the Solaris Failsafe boot entry in the GRUB main menu lists the failsafe boot archive. For example:

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-------------------------------------------------------------------------+
| root (hd0,1,a)
| kernel /boot/multiboot
| module /boot/x86.miniroot-safe
|
+-------------------------------------------------------------------------+
```

You can use the bootadm list-archive command to list the contents of the primary boot archive. For example:

```
# bootadm list-archive
etc/rtc_config
etc/system
etc/name_to_major
etc/driver_aliases
etc/name_to_sysnum
etc/dacf.conf
etc/driver_classes
etc/path_to_inst
etc/mach
etc/devices/devid_cache
etc/devices/mdi_scsi_vhci_cache
etc/devices/mdi_ib_cache
kernel
platform/i86pc/biosint
platform/i86pc/kernel
boot/solaris.xpm
boot/solaris/bootenv.rc
boot/solaris/devicedb/master
boot/acpi/tables
#
```

Managing the Primary Boot Archive

If any of the files in the operating system that are listed in the archive change, the boot archive must be rebuilt. For modifications to take effect, the rebuild of the archive must take place before the next system reboot.

When the system shuts down, the `svc:/system/boot-archive:default` service rebuilds the primary boot archive if necessary.

To rebuild the boot archive manually, use the `bootadm` command. The `update-archive` argument causes `bootadm` to rebuild the primary boot archive, and the `-v` option lists the stale files found. For example:

```
# bootadm update-archive -v
    changed /etc/path_to_inst
updating /platform/i86pc/boot_archive...this may take a minute
#
```

You can also rebuild the boot archive by booting the system using the failsafe archive which is an available option in the GRUB main menu. During the failsafe boot procedure, when prompted by the system, type **y** to rebuild the primary boot archive.

The `bootadm` command accepts these options:

- `-f` – Forces an update of the boot archive
- `-R` – Allows you to specify an alternate root where the boot archive is located
- `-n` – In an `update-archive` operation, archive content is checked, but not updated

See the `bootadm(1M)` man page for more information.

Booting a System in the GRUB-Based Boot Environment

This section describes performing a GRUB based disk boot on an x86-based system.

Some of the procedures in this section ask you to use the Reset button to restart the system. If your system does not have a Reset button, use the power switch to restart the system. You might be able to press Ctrl-Alt-Del to interrupt system operation, depending upon the state of the system.

Booting a System to Default Run Level (Multiuser)

Use this procedure to boot a system that is currently at run level 0 to the default run level.

1. If the system displays `Press any key to reboot prompt`, press any key to reboot the system.

   You can also use the Reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   When the boot sequence begins, the GRUB menu displays.
2. To boot the system to the default run level, press Enter to boot the default OS instance.

   If you do not choose an entry within 10 seconds, the system automatically boots to the default run level.
3. Login as `root` and verify that the system has booted to run level 3.

```
# who -r
   .       run-level 3  Mar  1 11:32     3      0  S
#
```

Note – The `who` command displays run level 3 because it is not milestone aware.

How to Boot a System to Run Level S (Single-User Level)

Use this procedure to boot a system that is at run level 0 to run level S. Single-user level is used for performing system maintenance.

1. If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.

   You can also use the Reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   When the boot sequence begins, the GRUB menu displays.

2. To boot the system to run level S, type `e` when the GRUB main menu displays.

3. Use the arrow keys to choose the `kernel /platform/i86pc/multiboot` boot entry.

Note – If you cannot use the arrow keys, use the ^ key to scroll up and the v key to scroll down.

---

4. Type `e` to edit the `kernel /platform/i86pc/multiboot` boot entry.

   The GRUB edit menu is displayed, enabling you to add options and arguments to the `kernel` command.

5. Enter a space and `-s` at the end of the `kernel /platform/i86pc/multiboot` line. Then, press Enter. For example:

```
grub edit> kernel /platform/i86pc/multiboot -s <Enter>
```

   After you press Enter, you are returned to the GRUB main menu. For example:

```
root (hd0,1,a)
kernel /platform/i86pc/multiboot -s
module /platform/i86pc/boot_archive
```

6. To boot the system to the single-user level, type `b`. The system boots to the single-user level.

7. Type the root password, if prompted. For example:

```
Root password for system maintenance (control-d to bypass): password
```

8. The superuser prompt displays.

```
Mar 11 16:00:00 su: 'su root' succeeded for root on /dev/console
Sun Microsystems Inc. SunOS 5.10     Generic January 2005
```

```
#
```

9. Verify that the system is at run level S.

```
# who -r
   .        run-level S  Mar  11 16:08     S      0  0
#
```

10. Perform the system maintenance task that required the run level change to S.

11. After you complete the system maintenance task, type Control-D to bring the system to the multiuser state.

How to Boot a System Interactively

Use this procedure to boot a system when you need to specify alternate kernel module directories or an alternate `/etc/system` file.

1. If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.

   You can also use the Reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   When the boot sequence begins, the GRUB menu displays.

2. Type `e` to access the GRUB edit menu.

3. Use the arrow keys to select the `kernel /platform/i86pc/multiboot` line.

4. Type `e` to edit the line.

   The GRUB edit menu is displayed, enabling you to add options to the `kernel` command.

5. Enter a space and `-a` at the end of the `kernel /platform/i86pc/multiboot` line. Then, press Enter. For example:

   ```
   grub edit> kernel /platform/i86pc/multiboot -a <Enter>
   ```

   The GRUB main menu displays. For example:

   ```
   root (hd0,1,a)
   kernel /platform/i86pc/multiboot -a
   module /platform/i86pc/boot_archive
   ```

6. To boot the system interactively, type `b`. A series of system prompts are displayed. If you want to accept the default, press Enter.

7. Provide the names of alternate directories for kernel modules, or accept the default.

```
Enter default directory for modules [/platform/i86pc/kernel /kernel
/usr/kernel]: Return
```

8. Provide the name of an alternate `/etc/system` file or accept the default. To help repair a damaged `/etc/system` file, enter `/dev/null` as the alternate for `/etc/system`. For more information see the `system(4)` man page.

   ```
   Name of system file [etc/system]: Return
   ```

   The system boots to the default milestone.

Stopping an x86 System for Recovery

Stop the system using one of the following commands, if possible:

- If the keyboard and mouse are functional, become superuser. Then, type `init 0` to stop the system. After the `Press any key to reboot` prompt appears, press any key to reboot the system.
- If the keyboard and mouse are functional, become superuser. then, type `init 6` to reboot the system.

If the system does not respond to any input from the mouse or the keyboard, press the Reset key, if it exists, to reboot the system. Or, you can use the power switch to reboot the system.

Booting the Failsafe Archive for Recovery

Follow this procedure to boot the Solaris failsafe archive, so that you can update the primary boot archive or repair a critical system resource.

Note – Starting with the Solaris 10 6/06 release, the GRUB failsafe interaction changed. Previously, when you booted the failsafe archive, you were prompted to update the boot archives whether the boot archives were out-of-date or not. Now, you are prompted to update the boot archives only if the system detects any out-of-date boot archives. For more information, see the `grub(5)` man page.

---

1. Use the procedures described the previous section to stop your system and boot it again, or if the system displays the `Press any key to reboot` prompt, press any key to reboot the system.

   When the boot sequence begins, the GRUB main menu displays. For example:

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-------------------------------------------------------------------+
| Solaris 10 11/06 s10x_u3wos_10 X86
| Solaris failsafe
| Diagnostic Partition
|
+-------------------------------------------------------------------+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

2. When the GRUB main menu displays, press the Space bar to interrupt the boot process.
3. Use the arrow keys to select the `Solaris failsafe` entry.

4. Type b or press Return to boot the failsafe archive.

   The system searches for installed OS instances. If the system detects any out-of-date boot archives, a message similar to the following displays:

```
Searching for installed OS instances...

    An out of sync boot archive was detected on /dev/dsk/c1d0s0.
    The boot archive is a cache of files used during boot and
    should be kept in sync to ensure proper system operation.

    Do you wish to automatically update this boot archive? [y,n,?]
```

5. Type **y** to automatically update the out-of-date boot archive. If multiple out-of-date boot archives are found, the system prompts you to update each boot archive, one at a time. Type **y** and press Return to update each boot archive.
6. After the boot archive has been updated successfully, the system searches again for all installed OS instances. If only one OS instance is found, a message similar to the following displays:

```
Searching for installed OS instances...

Solaris 10 11/06 s10x_u3wos_10 X86 was found on /dev/dsk/c1d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?]
```

   If the system finds multiple OS instances, it presents prompts that allow you to select the OS instance to mount on /a.

7. Type **y** to mount the OS instance /a.
8. Perform the tasks required to correct the problem encountered. For example, to forcibly update a corrupt boot archive, type:

```
# bootadm update-archive -f -R /a
Creating ram disk on /a
updating /a/platform/i86pc/boot_archive...this may take a minute
#
```

   The -f option forces the update, and the -R option allows you to specify an alternate root, in this case /a.

9. When you have finished correcting the problem, change directory to root (/) and unmount the device mounted on /a.

```
# cd /
# umount /a
```

10. Reboot the system.

```
# init 6
```

Interrupting an Unresponsive System

When a system freezes or stops responding to the keyboard, you might have to interrupt it. When you interrupt the system, all active processes stop immediately. It does not allow you to flush memory or to synchronize file systems.

Aborting an Unresponsive System

To abort or 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 Ctrl+Alt+Del key sequence on the keyboard of the unresponsive system. The system reboots.
- Press the reset button.
- Power off the system and power it back on.
- Some Sun systems allow the system to be reset remotely using the LOM software.

Exercise: Using GRUB

In this exercise, you change entries in the GRUB menu to boot the system in different ways, and use the eeprom command to set boot parameters.

Preparation

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

This exercise requires a system that is capable of using a 64-bit kernel in order to complete all lab steps. If you are using a system that is only 32-bit capable, you will only be able to run a subset of the lab steps.

Task

Complete the following steps:

1. Login as root and open a terminal window.
2. Use the init command to shut down the system to run level 0.
3. When the Press any key to reboot prompt displays, press any key to start the boot process.
4. When the GRUB main menu displays, press the Space bar to interrupt the boot process. The GRUB main menu remains displayed.
5. Select the Solaris 10 10/09 s10x_u8wos_08a X86 entry, and press e to display the list of commands it represents.
   - What is the pathname of the boot archive this boot entry uses?

     ____

   - Is a specific kernel listed in the kernel command line?

     ____

6. Press the Esc key to return to the GRUB main menu.
7. Select the Solaris failsafe entry, and press e to display the list of commands it represents. What is the pathname of the boot archive this boot entry uses?

   ____

8. Press the Esc key to return to the GRUB main menu.

9. Select the `Solaris 10 10/09 s10x_u8wos_08a X86` entry, and press `Return` to boot the system to the default milestone.
10. Login as `root` and open a terminal window.
11. Use the `eeprom` command to display the list of boot parameters defined in the `bootenv.rc` file. Is the `boot-file` parameter currently defined?

    ____

12. Use the `prtconf` command to display the name of the kernel that the system is currently using. If the kernel listed is a 64-bit kernel, for example `/platform/i86pc/kernel/amd64/unix`, proceed to the next step. If the kernel listed is a 32-bit kernel, for example, `/platform/i86pc/kernel/unix`, skip to step 20.

Note – You cannot use a 64-bit kernel to boot a 32-bit system.

---

13. Use the `eeprom` command to set the `boot-file` parameter so the system will boot using the 32-bit kernel. Verify that the parameter is now set.
14. Use the `init` command to reboot your system. When the boot process completes, login as `root` and open a terminal window.
15. Use the `prtconf` command to display the name of the kernel that the system is now using.

    ____

16. Use the `init` command to reboot your system. When the GRUB main menu displays, press `e` to interrupt the boot process and edit the default boot entry. Does the kernel command list a specific kernel to boot?

    ____

17. Select the `kernel` command from the GRUB menu, and `e` to enter the GRUB editor.
18. Edit the `kernel` command to add the name of the 64-bit kernel to the end of the command line.
19. Verify the kernel command specifies the 64-bit kernel, and enter `b` to boot the system.
20. Login as `root` and open a terminal window.

21. Use the prtconf command to display the name of the kernel that the system is currently using. Is the system using the 32-bit kernel you specified with the eeprom command, or the 64-bit kernel you specified using the kernel command?

    ____

22. Use the eeprom command to set the boot-file parameter to a null value, and verify the change.
23. Use the init command to reboot your system. When the GRUB main menu displays, press the Space bar to interrupt the boot process.
24. Select the Solaris failsafe boot option and boot the system. Allow the Solaris instance that the system finds to mount on /a.
25. Change directory to /a/platform/i86pc, and create a directory named backup.
26. Move the boot_archive file into the backup directory and verify that it no longer exists in /a/platform/i86pc.
27. Use the bootadm command to create a new boot archive in the /a/platform/i86pc directory. Verify that it exists.
28. Change directory to root (/), unmount /a, and reboot the system.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Using GRUB

This section contains solutions to the exercise.

Task

Complete the following steps:

1. Login as `root` and open a terminal window.
2. Use the `init` command to shut down the system to run level 0.

   ```
   # init 0
   ```
3. When the `Press any key to reboot` prompt displays, press any key to start the boot process.
4. When the GRUB main menu displays, press the Space bar to interrupt the boot process. The GRUB main menu remains displayed. For example:

```
GNU GRUB version 0.95 (572K lower / 3380808K upper memory)
+-------------------------------------------------------------------------+
| Solaris 10 10/09 s10x_u8wos_08a X86
| Solaris failsafe
| Diagnostic Partition
|
+-------------------------------------------------------------------------+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

Note – The `Diagnostic Partition` item in the GRUB menu is not present unless the system had an unrecoverable startup failure.

---

5. Select the `Solaris 10 10/09 s10x_u8wos_08a X86` entry, and press `e` to display the list of commands it represents.
   - What is the pathname of the boot archive this boot entry uses?
   - Is a specific kernel listed in the `kernel` command line?

```
GNU GRUB version 0.95 (572K lower / 3380808K upper memory)
+-------------------------------------------------------------------+
| findroot (rootfs0,0,a)
| kernel /platform/i86pc/multiboot
| module /platform/i86pc/boot_archive
|
+-------------------------------------------------------------------+
```

   The `Solaris 10 10/09 s10x_u8wos_08a X86` boot entry uses the `/platform/i86pc/boot_archive` boot archive. The `kernel` command does not list a specific kernel to load.

6. Press the Esc key to return to the GRUB main menu.
7. Select the `Solaris failsafe` entry, and press `e` to display the list of commands it represents. What is the pathname of the boot archive this boot entry uses?

```
GNU GRUB version 0.95 (572K lower / 3380808K upper memory)
+-------------------------------------------------------------------+
| findroot (rootfs0,0,a)
| kernel /boot/multiboot kernel/unix -s
| module /boot/x86.miniroot-safe
|
+-------------------------------------------------------------------+
```

   The `Solaris failsafe` boot entry uses the `/boot/x86.miniroot-safe` boot archive.

8. Press the Esc key to return to the GRUB main menu.
9. Select the `Solaris 10 10/09 s10x_u8wos_08a X86` entry, and press `Return` to boot the system to the default milestone.
10. Login as `root` and open a terminal window.
11. Use the `eeprom` command to display the list of boot parameters defined in the `bootenv.rc` file. Is the `boot-file` parameter currently defined?

```
# eeprom
kbd-type=US-English
ata-dma-enabled=1
atapi-cd-dma-enabled=0
ttyb-rts-dtr-off=false
ttyb-ignore-cd=true
```

```
ttya-rts-dtr-off=false
ttya-ignore-cd=true
ttyb-mode=9600,8,n,1,-
ttya-mode=9600,8,n,1,-
lba-access-ok=1
prealloc-chunk-size=0x2000
bootpath=/pci@0,0/pci-ide@5/ide@0/cmdk@0,0:a
console=text
#
```

No, the boot-file parameter is not defined.

12. Use the `prtconf` command to display the name of the kernel that the system is currently using. If the kernel listed is a 64-bit kernel, for example `/platform/i86pc/kernel/amd64/unix`, proceed to the next step. If the kernel listed is a 32-bit kernel, for example, `/platform/i86pc/kernel/unix`, skip to step 20.

```
# prtconf -v | grep /platform/i86pc/kernel
            value='/platform/i86pc/kernel/amd64/unix'
#
```

Note – You cannot use a 64-bit kernel to boot a 32-bit system.

13. Use the `eeprom` command to set the `boot-file` parameter so the system will boot using the 32-bit kernel. Verify that the parameter is now set.

```
# eeprom boot-file=kernel/unix
# eeprom boot-file
boot-file=kernel/unix
#
```

14. Use the `init` command to reboot your system. When the boot process completes, login as `root` and open a terminal window.

```
# init 6
```

15. Use the `prtconf` command to display the name of the kernel that the system is now using.

```
# prtconf -v | grep /platform/i86pc/kernel
            value='/platform/i86pc/kernel/unix'
#
```

16. Use the `init` command to reboot your system. When the GRUB main menu displays, press `e` to interrupt the boot process and edit the default boot entry. Does the kernel command list a specific kernel to boot?

```
# init 6

(output omitted)

GNU GRUB version 0.95 (572K lower / 3380808K upper memory)
+-----------------------------------------------------------------------+
| findroot (rootfs0,0,a)
| kernel /platform/i86pc/multiboot
| module /platform/i86pc/boot_archive
|
+-----------------------------------------------------------------------+
```

No, the `kernel` command still lists no specific kernel.

17. Select the `kernel` command from the GRUB menu, and press `e` to enter the GRUB editor.

```
grub edit> kernel /platform/i86pc/multiboot
```

18. Edit the `kernel` command to add the name of the 64-bit kernel to the end of the command line.

```
grub edit> kernel /platform/i86pc/multiboot kernel/amd64/unix
```

19. Verify the kernel command specifies the 64-bit kernel, and enter `b` to boot the system.

20. Login as `root` and open a terminal window.

21. Use the `prtconf` command to display the name of the kernel that the system is currently using. Is the system using the 32-bit kernel you specified with the `eeprom` command, or the 64-bit kernel you specified using the `kernel` command?

```
# prtconf -v | grep /platform/i86pc/kernel
            value='/platform/i86pc/kernel/amd64/unix'
```

The system is using the 64-bit kernel you specified using the `kernel` command, overriding the entry in `bootenv.rc` file created by the `eeprom` command.

22. Use the `eeprom` command to set the `boot-file` parameter to a null value, and verify the change.

```
# eeprom boot-file=""
# eeprom boot-file
boot-file=
#
```

23. Use the `init` command to reboot your system. When the GRUB main menu displays, press the Space bar to interrupt the boot process.

```
# init 6

(output omitted)

GNU GRUB version 0.95 (572K lower / 3380808K upper memory)
+-------------------------------------------------------------------------+
| Solaris 10 5/09 s10x_u7wos_08 X86
| Solaris failsafe
| Diagnostic Partition
|
+-------------------------------------------------------------------------+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

24. Select the `Solaris failsafe` boot option and boot the system. Allow the Solaris instance that the system finds to mount on `/a`.

```
Solaris 10 5/09 s10x_u7wos_08 X86 was found on /dev/dsk/c1d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?] y
```

25. Change directory to `/a/platform/i86pc`, and create a directory named `backup`.

```
# cd /a/platform/i86pc
# mkdir backup
#
```

26. Move the `boot_archive` file into the `backup` directory and verify that it no longer exists in `/a/platform/i86pc`.

```
# mv boot_archive backup
# ls
backup biosint boot kernel multiboot ucode
#
```

27. Use the `bootadm` command to create a new boot archive in the `/a/platform/i86pc` directory. Verify that it exists.

```
# bootadm update-archive -f -R /a
Creating boot_archive for /a
updating /a/platform/i86pc/boot_archive
# ls boot_archive
boot_archive
#
```

28. Change directory to root (`/`), unmount `/a`, and reboot the system.

```
# cd /
# umount /a
# init 6
```

Module 11

Performing Legacy Boot and Shutdown Procedures

Objectives

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

- Describe the Solaris SPARC boot design
- Identify boot process phases
- Identify run level fundamentals
- Control boot processes
- Perform system shutdown procedures

Solaris SPARC Boot Design Overview

The Solaris x86 boot design feature was introduced in the Solaris 10 1/06 release. With the Solaris 10 10/08 release, the Solaris SPARC bootstrap was redesigned to be in line with the Solaris x86 boot design. The improved Solaris boot design provides support for Solaris boot archives, which was earlier available only on x86 platforms, on the SPARC platform. With this simplified boot design, other non-UFS file systems, such as ZFS are supported as root file systems. Although the implementation of the SPARC boot has changed, boot tasks remain the same.

New Design

The Solaris boot design architecture for SPARC is similar to the x86 architecture. The difference is how the boot device and fields are selected at boot time. The SPARC platform continues to use the open boot PROM (OBP) and its commands for selections, while the x86 platform uses the BIOS and the GRUB menu. On the SPARC and x86 platforms, there is one primary boot archive and one fail-safe boot archive. The following are the four phases of the new boot process:

1. Boot loader phase: Loads the Solaris root file system archive from the media to memory.
2. Booter phase: Reads the ramdisk directly and executes it. This is the only portion of the boot process that requires knowledge of the root file system format.
3. Ramdisk phase: Extracts the kernel image from the boot archive and executes it. Nether the booter or the kernel need to know about the format of the boot archive.
4. Kernel phase: Extracts the rest of the primary modules from the boot archive, initializes itself, mounts the real root file system, and discards the boot archive.

For example, when a system powers on, the firmware loads the boot block, which in turn, retrieves a boot loader. The boot loader gathers a root file system and places it in a ramdisk. The ramdisk loads and executes a secondary loader, which assembles core kernel module from the ramdisk and transfers control to the Solaris kernel. The kernel probes input and output (I/O devices and then mounts the root file system.

Features

The SPARC platform bootadm command functions the same as on the x86 platform. It handles the boot archive update and verification. Use it to boot archives:

```
# bootadm update-archive
```

The boot archive service is controlled with the `svcadm` command and managed by the SMF:

```
svc:/system/boot-archive:default
```

Use the `-p` option to specify the platform or machine hardware class on a client system where the client platform differs from the server platform. You must use the `-p` option with the `-R` option to obtain an alternate root path. For example:

```
# boot -p platform -R [altroot]
```

where the platform specified can be `i86pc`, `sun4u`, or `sun4v`.

The boot loader and the OS kernel are separated in the new boot design. The loader can evolve independently from the Solaris kernel, making it a replaceable component. The boot loader is disassociated with the root file system and does not need to comprehend it. The boot loader accepts any file system to be the Solaris root without changing the boot code.

Disk boot and network boot from disk are unified, making the LAN and WAN identical from the Solaris perspective. The kernel always initializes from in-memory text and data regardless from where they are loaded.

System Requirements

The following are the SPARC and x86 boot design minimum system requirements, dependencies, and limitations:

- Memory
  - Text-based installation requires a minimum of 384 Mbytes.
  - GUI based installation requires a minimum of 768 Mbytes.
  - SPARC and x86 platform memory requirements are the same.
- Use the `root_archive` command to pack and unpack the miniroot. Only SPARC based systems that support the new boot architecture have the ability to pack and unpack a compressed miniroot.
- The Solaris 10 `root_archive` tool isn't compatible with Solaris release tool versions. Perform ramdisk manipulation only on systems running the same Solaris and archive release.

Legacy Boot and Shutdown

Prior to Solaris 10, Sun systems used a series of scripts to start and stop processes typically associated with run levels. Each run level had an associated script located in the `/sbin` directory, with some scripts hard-linked to each other. The `init` daemon was responsible for starting and stopping system services. These structures are still present and are referred to through out this module as legacy boot processes. With Solaris 10, you can use the Service Management Facility (SMF) to start and stop processes typically associated with run levels. The SMF is covered in the next module of this course. This module discusses the features and structures of the legacy boot architecture.

Identifying Boot Process Phases

In general, when a system is powered on, the PROM monitor on SPARC systems, and the BIOS monitor on x86/x64 systems, runs a POST procedure that checks the hardware and memory on the system. If no errors are found, and the `auto-boot?` parameter is set to `true` on SPARC systems, or the `timeout` parameter is set to a positive value in the `/boot/grub/menu.lst` file on x86/x64 systems, the system automatically begins the boot process.

The entire boot process is described by five distinct phases:

- The boot PROM phase
- The boot programs phase
- The `kernel` initialization phase
- The `init` phase
- The `svc.startd` phase

The boot PROM and boot programs phases differ significantly between SPARC and x86/x64 systems, but subsequent phases on these platforms are generally the same and use similar sets of files.

Figure 11-1 shows the phases of the boot process.

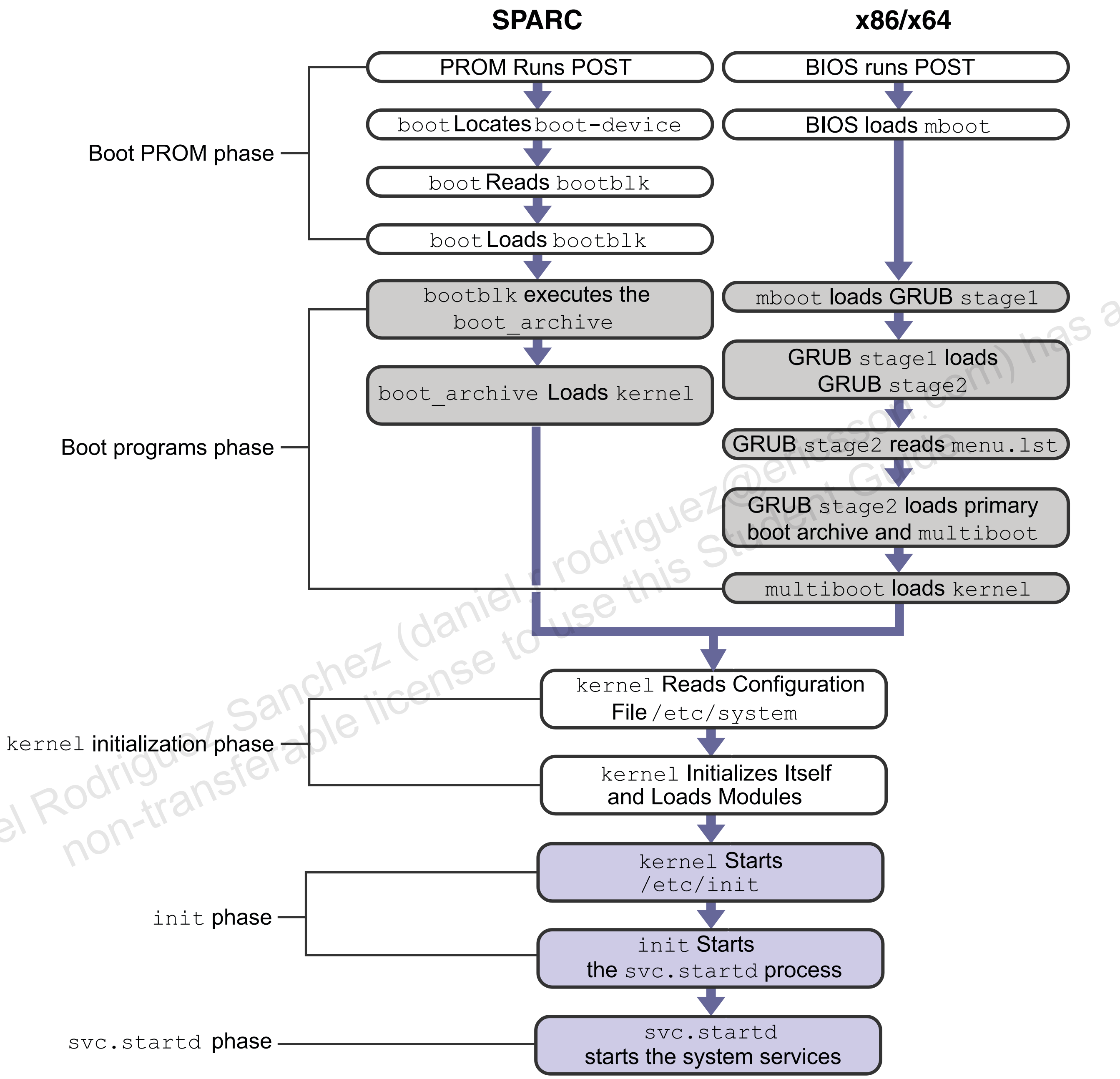


Figure 11-1 Phases of the Boot Process

SPARC Boot PROM Phase

Note – This section describes the first part of the boot sequence on systems with a single board only. Multi-board systems run a sequence that identifies and sets up their hardware before the boot PROM is involved.

The boot PROM performs the following steps during the first part of the boot sequence on a SPARC-based system:

- The PROM runs the POST.

  The boot PROM firmware runs the POST to verify the system hardware and memory. After successful completion of the diagnostics, the boot sequence begins.
- The PROM displays the system identification banner.

  The model type, processor type and speed, keyboard status, PROM revision number, amount of installed random access memory (RAM), NVRAM serial number, Ethernet address, and host ID are displayed.
- The boot PROM determines the boot device by reading the PROM parameter `boot-device`.
- The boot PROM reads the disk label located at Sector 0 on the default boot device.
- The boot PROM finds the `boot` program from the default boot device programmed into the PROM.

  The boot PROM program reads the system primary boot program called `bootblk` (located at sectors 1 through 15) that contains a UNIX file system (`UFS`) reader. (The `bootblk` program is placed on the disk by the `installboot` command during system installation.)

  The `boot` command loads the `bootblk` program from its location on the boot device into memory.

SPARC Boot Programs Phase

Note – Solaris releases that support the SPARC Newboot feature might require SPARC systems with at least 512 Mbytes of memory.

The following describes the boot programs phase on a SPARC-based system:

- The `bootblk` program loads the secondary boot program, `ufsboot`, from the boot device into memory.

  The path to `ufsboot` is recorded in the `bootblk` program, which is installed by the `installboot` command.

- The `ufsboot` program locates and loads the appropriate two-part `kernel`.

  The core of the `kernel` is two pieces of static code called `genunix` and `unix`, where `genunix` is the platform-independent generic `kernel` file and `unix` is the platform-specific `kernel` file.

  When `ufsboot` loads these two files into memory, they are combined to form the running `kernel`.

  On a SPARC system running in 64-bit mode, the two-part `kernel` is located in the directory:

  ```
  /platform/`uname -m`/kernel/sparcv9
  ```

Note – Solaris 10 for SPARC only runs on 64-bit systems.

Note – To determine the platform name (for example, the system hardware class), type the `uname -m` command. For example, when you type this command on an Ultra 10 workstation, the console displays `sun4u`.

---

x86/x64 Boot PROM Phase

The BIOS ROM performs the following steps during the first part of the boot sequence on an x86/x64-based system:

- After a PC-compatible machine is turned on, the system firmware in the BIOS ROM executes a power-on self test (POST), runs BIOS extensions in peripheral board ROMs, and invokes software interrupt INT 19h, Bootstrap.
- The INT 19h handler typically performs the standard PC-compatible boot, which consists of trying to read the first physical sector from the first diskette drive, or, if that fails, from the first hard disk. The processor then jumps to the first byte of the sector image in memory.
- The first sector on a hard disk on an x86/x64 system contains the master boot block, which contains the master boot (`mboot`) program and the FDISK table, named for the PC program that maintains it.

x86/x64 Boot Programs Phase

The following describes the boot programs phase on an x86/x64-based system:

- The master boot finds the active partition in the FDISK table, loads its first sector (GRUB `stage1`), and jumps to its first byte in memory.
- If GRUB `stage1` is installed on the master boot block (see the `-m` option of `installgrub(1M)`), then `stage2` is loaded directly from the Solaris FDISK partition regardless of the active partition.
- An x86 FDISK partition for the Solaris software begins with a one-cylinder boot slice, which contains GRUB `stage1` in the first sector, the standard Solaris disk label and volume table of contents (VTOC) in the second and third sectors, and GRUB `stage2` in the fiftieth and subsequent sectors. The area from sector 4 to 49 might contain boot blocks for older versions of Solaris.
- When the FDISK partition for the Solaris software is the active partition, the master boot program (`mboot`) reads the partition boot program in the first sector of the active partition (GRUB `stage1)` into memory and jumps to it.
- GRUB `stage1` in turn reads the GRUB `stage2` program into memory and jumps to it. The GRUB `stage2` program contains code that allows it to navigate the UFS structure on the root file system.

- The GRUB `stage2` program locates the GRUB menu configuration file `/boot/grub/menu.lst` and displays the GRUB main menu.
- When the GRUB menu displays, the user can choose to boot an operating system on a different partition, a different disk, or possibly from the network.
- Either selected by the user or by default timeout, GRUB executes commands from `/boot/grub/menu.lst` to load a pre-constructed boot archive containing a kernel program and data. The default primary boot archive for Solaris 10 Update 3 is `/platform/i86pc/boot_archive`.
- GRUB loads a program called `multiboot`, which implements the kernel side of the Multiboot Specification.
- The `multiboot` program assembles the core kernel modules from the `boot_archive` and starts the operating system, links in the necessary modules from the boot archive and mounts the root filesystem on the real root device.
- At this point, the kernel regains storage I/O, mounts additional file systems (see `vfstab(4)`), and starts various operating system services `(see smf(5)`).

The `kernel` Initialization Phase

The following describes the `kernel` initialization phase:

- The kernel reads its configuration file, called `/etc/system`.
- The kernel initializes itself and begins loading modules.

  The kernel uses the `ufsboot` command to load the files. When it has loaded enough modules to mount the `/` (root) file system, it unmaps the `ufsboot` program and continues.
- The `kernel` starts the `/etc/init` daemon.

Note – The `/etc/init` file is a symbolic link to `/sbin/init`.

The SunOS™ kernel is a small static core, consisting of `genunix` and `unix` and many dynamically loadable kernel modules.

Modules can consist of device drivers, binary files to support file systems, and streams, as well as other module types used for specific tasks within the system.

The modules that make up the kernel typically reside in the directories /kernel and /usr/kernel. Platform-dependent modules reside in the /platform/‘uname -m‘/kernel and /platform/‘uname -i‘/kernel directories.

On x86/x64 systems, 32-bit and 64-bit modules share parent directory structures in which subdirectories named amd64 store 64-bit modules. For example, 32-bit device drivers are found in /kernel/drv, and 64-bit device drivers are found in /kernel/drv/amd64:

```
# file /kernel/drv/cmdk /kernel/drv/amd64/cmdk
/kernel/drv/cmdk:       ELF 32-bit LSB relocatable 80386 Version 1
/kernel/drv/amd64/cmdk: ELF 64-bit LSB relocatable AMD64 Version 1
#
```

The following describes the types of module subdirectories contained in the /kernel, /usr/kernel, /platform/‘uname -m‘/kernel, or /platform/‘uname -i‘/kernel directories:

- drv/sparcv9, drv, drv/amd64 – Device drivers
- exec/sparcv9, exec, exec/amd64 – Executable file formats
- fs/sparcv9, fs, fs/amd64 – File system types, for example, ufs, nfs, and proc
- misc/sparcv9, misc, misc/amd64 – Miscellaneous modules, for example, usb
- sched/sparcv9, sched, sched/amd64 – Scheduling classes (process execution scheduling)
- strmod/sparcv9, strmod, strmod/amd64 – Streams modules (generalized connection between users and device drivers)
- sys/sparcv9, sys, sys/amd64 – System calls (defined interfaces for applications to use)

On SPARC systems, the /kernel/drv/sparcv9 directory contains all of the device drivers that are used for system boot. The /usr/kernel/drv/sparcv9 directory is used for all other device drivers.

Note – The sparcv9 CPU is the type of SPARC CPU that supports 64-bit processing.

Modules are loaded automatically as needed either at boot time or on demand, if requested by an application. When a module is no longer in use, it might be unloaded on the basis that the memory it uses is needed for another task.

After the boot process is complete, device drivers are loaded when devices, such as tape devices, are accessed. This process is called autoconfiguration because some kernel driver modules are loaded automatically when needed.

Upon initial or reconfiguration boot, the system does a self-test and checks for all devices that are attached.

The advantage of this dynamic `kernel` arrangement is that the overall size of the `kernel` is smaller, which makes more efficient use of memory and allows for simpler modification and tuning. Figure 11-2 shows this arrangement on a SPARC-based system.

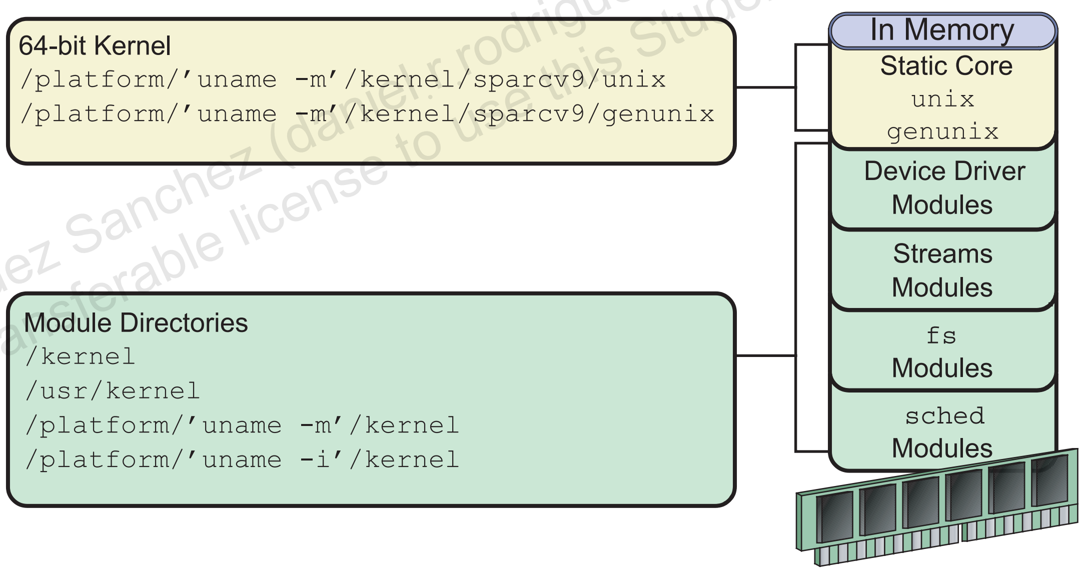


Figure 11-2 Dynamic `kernel` Arrangement on a SPARC system

On x86/x64 systems the 32-bit kernel static core is composed of these files:

- `/platform/i86pc/kernel/unix`
- `/kernel/genunix`

On x86/x64 systems the 64-bit kernel static core is composed of these files:

- `/platform/i86pc/kernel/amd64/unix`
- `/kernel/amd64/genunix`

The /etc/system File and Kernel Configuration

Caution – The Solaris OS builds the kernel based upon the system size (memory, CPUs, etc). The performance of the default kernel that is built can handle most system activities. Make modifications with extreme caution.

The /etc/system file is the control file for modifying which modules and parameters are to be loaded by the kernel at boot time. By default, all lines in this file are commented out.

Modifying the kernel's behavior (or configuration) requires editing the /etc/system file. Altering this file allows you to modify the kernel's treatment of loadable modules as well as to modify kernel parameters for some performance tuning.

The ufsboot program contains a list of default loadable kernel modules that are loaded at boot time. However, you can override this list by modifying the /etc/system file to control which modules, as well as which parameters, are loaded.

All changes to this file take effect after a reboot.

The /etc/system file can explicitly control:

- The search path for default kernel modules to be loaded at boot time
- The root file system type and device
- The modules that are excluded from loading automatically at boot time
- The modules to be forcibly loaded at boot time, rather that at first access
- The new values to override the default kernel parameter values

Note – Command lines must be 80 characters or less in length, and comment lines must begin with an asterisk (*) and end with a newline character.

The /etc/system file is divided into five distinct sections:

- moddir:

  Sets the search path for default loadable kernel modules. You can list together multiple directories to search, delimited either by blank spaces or colons. If the module is not found in the first directory, the second directory is searched, and so on.

- root device and root file system configuration:

  Sets the root file system type to the listed value. The default is rootfs:ufs.

  Sets the root device. The default is the physical path name of the device on which the boot program resides. The physical path name is platform dependent and configuration dependent. The following is an example path:

  ```
  rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a
  ```

- exclude:

  Does not allow the loadable kernel modules to be loaded during kernel initialization, for example:

  ```
  exclude: sys/shmsys
  ```

- forceload:

  Forces the kernel modules to be loaded during kernel initialization, for example:

  ```
  forceload: drv/vx
  ```

  The default action is to load a kernel module automatically when its services are first accessed during runtime by a user or an application.

- set:

  Changes kernel parameters to modify the operation of the system, for example:

  ```
  set maxusers=40
  ```

Editing the /etc/system File

Caution – Before you edit the /etc/system file, you should make a backup copy. If you enter incorrect values in this file, the system might not be able to boot.

The following example shows how to copy the original /etc/system file to a backup file and then edit the /etc/system file.

```
# cp /etc/system /etc/system.orig
# vi /etc/system
```

If a boot process fails because of an unusable /etc/system file, on a SPARC system, issue the interactive boot command: boot -a. When the system prompts you to enter the name of the system file, type in the name of your backup system file, or, alternatively, enter /dev/null for a null configuration file.

```
ok boot -a
Enter filename [kernel/sparcv9/unix]: <Return>
Enter default directory for modules [/platform...]: <Return>
Name of system file [etc/system]: etc/system.orig - or - /dev/null
root filesystem type [ufs]: <Return>
Enter physical name of root device [/...]: <Return>
(further boot messages omitted)
```

Solaris x86/x64-based systems also provide an interactive boot procedure. In the GRUB menu, adding the -a option to the kernel command used to boot the system initiates an interactive boot procedure similar to that found on SPARC systems.

In the command list for the boot option you want to use, add the -a option as in the following example, then boot the system using the modified command.

```
grub edit> kernel /platform/i86pc/multiboot -a
```

Note – Some x86/x64 systems may require the -ar options to initiate an interactive boot procedure.

You can also boot x86/x64 systems using the failsafe boot archive. Doing so allows you to mount the root file system, and make corrections to the /etc/system file as necessary.

To boot the system using the failsafe archive, select the `Solaris failsafe` entry from the GRUB main menu, and boot the system. The root file system mounts automatically below `/a`, so you would edit `/a/etc/system` to correct the file.

The `init` Phase

The next to the last phase of the boot process is the `init` phase. During this phase, the `init` daemon starts the `svc.startd` daemon that is responsible for starting and stopping services as requested. The `/sbin/init` phase uses information stored in the `/etc/inittab` file.

The `/etc/inittab` File

Each line in the `/etc/inittab` file contains the following four fields:

id:*rstate*:*action*:*process*

Table 11-1 describes the fields in an `inittab` entry.

Table 11-1 Fields in the `inittab` File

Field	**Description**
id	Two character identifier for the entry
rstate	Run levels to which this entry applies
action	Defines how the process listed should be run For a description of the action keywords see `man inittab`
process	Defines the command to execute

The following example shows the default `inittab` file installed with the Solaris 10 OS. The lines of output are described after the example:

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
sp::sysinit:/sbin/soconfig -f /etc/sock2path
smf::sysinit:/lib/svc/bin/svc.startd>/dev/msglog 2<>/dev/msglog
</dev/console
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog
2<>/dev/msglog
```

Note – Message output from `rc` scripts is directed to the logfile for the appropriate milestone. Prior to the Solaris 8 OS, all of these messages were written to the `/dev/console` file. The `/dev/msglog` file is used for message output collection from system startup or background applications.

Table 11-2 shows an explanation for each action keyword.

Table 11-2 The `action` Field Keywords

Keyword	Explanation
`sysinit`	Executes the process before the `init` process tries to access the console, for example, the console login prompt. The `init` process waits for completion of the process before it continues to read the `inittab` file.
`powerfail`	Executes the process only if the `init` process receives a power fail signal.

Information about additional `action` keywords is available in the `inittab` man page.The following describes each of the lines in the `/etc/inittab` file in order:

1. Initializes the STREAMS modules used for communication services.
2. Configures the socket transport providers for network connections.
3. Initializes the `svc.startd` daemon for SMF.
4. Describes a power fail shutdown.

Legacy Run Level Fundamentals

A run level is a system state, represented by a digit or letter, that defines the services and resources that are currently available to users.

Table 11-3 shows the eight run levels found in the Solaris OS.

Table 11-3 Solaris OS Run Levels

Run Level	Milestone	Function
0		System is running the PROM monitor.
s or S	single-user	Solaris OS single-user mode with critical file systems mounted and accessible.
1		The system is running in a single-user administrative state with access to all available file systems.
2	multi-user	The system is supporting multiuser operations. Multiple users can access the system. All system daemons are running except for the Network File System (NFS) server and some other network resource server related daemons.
3	multi-user-server	The system is supporting multiuser operations and has NFS resource sharing and other network resource servers available.
4		This level is currently not implemented.
5		A transitional run level in which the Solaris OS is shut down and the system is powered off.
6		A transitional run level in which the Solaris OS is shut down and the system reboots to the default run level.

Determining the Current Run Level of a System

To determine the current run level of a system, use the `who -r` command.

Figure 11-3 shows output from the command.

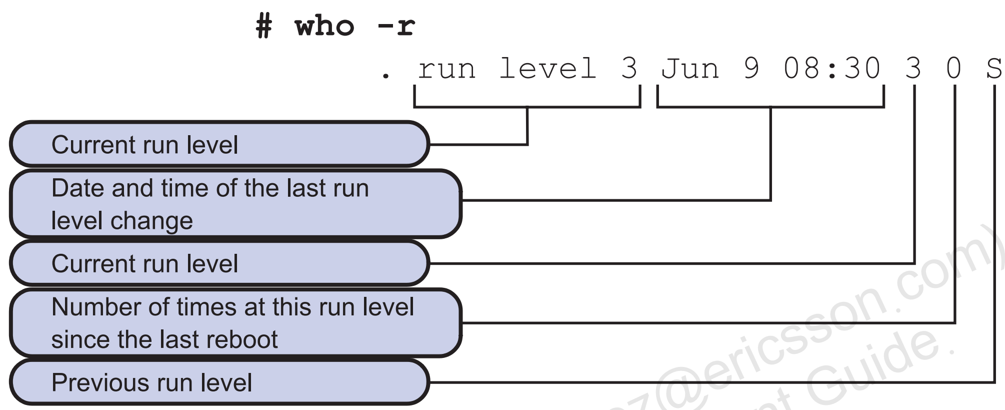


Figure 11-3 The System's Current Run Level

Changing Run Levels

Run levels are sometimes referred to as `init` states because you can use the `init` command to transition between run levels. The `init` command passes the required run level to `svc.startd`.

Use the `init` command to manually initiate run-level transitions. Change run levels with the `shutdown`, `halt`, `reboot`, and `poweroff` commands. Use the `svcadm` command to change system milestones.

The svcadm command doesn't change run levels. It changes system milestones.

Note – Prior to Solaris 10, the `init` daemon started and stopped system services. With Solaris 10, the `svc.startd` daemon starts and stops system services. The `init` daemon initializes stream modules, configures socket transport providers, sets up the system for a correct response to a power fail shutdown, and starts the `svc.startd` daemon.

Controlling Legacy Boot Processes

The Solaris OS provides a series of legacy scripts to stop and start processes typically associated with run levels or milestones.

The /sbin Directory

Each run level has an associated script located in the /sbin directory, with some scripts hard-linked to each other.

The scripts are executed by the svc.startd daemon to set up variables, test conditions, and make calls to other scripts that start and stop processes for that run level.

The rc0, rc5, and rc6 scripts are hard-linked to each other. Notice that each script is assigned the same inode number. The following is an example of the hard links:

```
# ls -li /sbin/rc*
     2317 -rwxr--r--   3 root     sys          1983 Dec 22 18:06 rc0
     2318 -rwxr--r--   1 root     sys          2242 Dec 22 18:06 rc1
     2319 -rwxr--r--   1 root     sys          2536 Dec 22 18:06 rc2
     2320 -rwxr--r--   1 root     sys          2567 Dec 22 18:06 rc3
     2317 -rwxr--r--   3 root     sys          1983 Dec 22 18:06 rc5
     2317 -rwxr--r--   3 root     sys          1983 Dec 22 18:06 rc6
     2321 -rwxr--r--   1 root     sys          5125 Dec 22 18:06 rcS
```

The Solaris OS provides the same series of rc scripts in the /etc directory for backward compatibility. These scripts are symbolic link files to the rc scripts in the /sbin directory. The following example shows this connection:

```
# ls -l /etc/rc?
ls -l rc?
lrwxrwxrwx   1 root     root          11 Oct 12 17:15 rc0 -> ../sbin/rc0
lrwxrwxrwx   1 root     root          11 Oct 12 17:15 rc1 -> ../sbin/rc1
lrwxrwxrwx   1 root     root          11 Oct 12 17:15 rc2 -> ../sbin/rc2
lrwxrwxrwx   1 root     root          11 Oct 12 17:15 rc3 -> ../sbin/rc3
lrwxrwxrwx   1 root     root          11 Oct 12 17:15 rc5 -> ../sbin/rc5
lrwxrwxrwx   1 root     root          11 Oct 12 17:15 rc6 -> ../sbin/rc6
lrwxrwxrwx   1 root     root          11 Oct 12 17:15 rcS -> ../sbin/rcS
rcm:
total 2
drwxr-xr-x   2 root     sys          512 Oct 12 17:18 scripts
```

Table 11-4 summarizes the function performed by each /sbin scripts. These scripts execute legacy rc scripts.

Table 11-4 Legacy Run Control (rc) Scripts and Functions

Script	**Function**
/sbin/rc0	Runs the /etc/rc0.d/K* scripts and the /etc/rc0.d/S* scripts to stop system services and daemons. Start scripts should only perform fast system cleanup functions.
/sbin/rc1	Runs the /etc/rc1.d/S* scripts to perform the following tasks: • Stops system services and daemons • Terminates certain running application processes • Unmounts all remote file systems
/sbin/rc2	Runs the /etc/rc2.d/K* scripts and the /etc/rc2.d/S* scripts to start certain application daemons.
/sbin/rc3	Runs the /etc/rc3.d/K* scripts and the /etc/rc3.d/S* scripts to start certain application daemons. The K scripts are not normally present in the /etc/rc3.d directory, although if they were present, they would be run.
/sbin/rc5 /sbin/rc6	Runs the /etc/rc0.d/K* scripts and then the /etc/rc0.d/S* scripts to perform the following tasks: • Stops system services and daemons • Starts scripts that should only perform fast system cleanup functions
/sbin/rcS	Runs the /etc/rcS.d scripts to bring up the system to run level S and establishes a minimal network.

The `/etc/rc#.d` Directories

The `/etc/rc#.d` directories contain scripts that start and stop system processes for that run level.

Figure 11-4 shows an example of `/etc/rc#.d` directories.

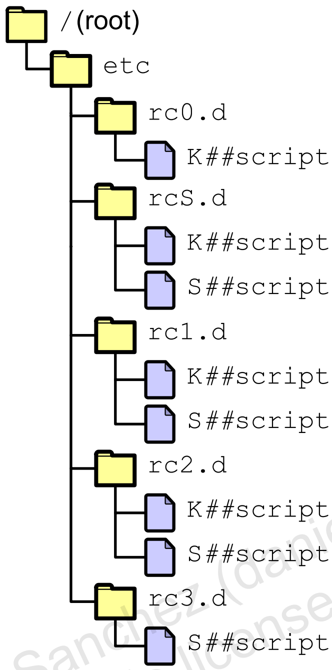


Figure 11-4 `/etc/rc#.d` directories

The `/etc/rc#.d` directories, for example `/etc/rc2.d`, contain scripts to start and stop processes for run level 2. The following output shows a partial script list.

```
# ls -l /etc/rc2.d
total 130
-rwxr--r--   6 root     sys         324 Dec 10 11:26 K03samba
-rwxr--r--   5 root     sys         824 May 27  2004 K05appserv
(some output omitted)
-rwxr--r--   5 root     sys        2329 Dec 11 08:52 S94ncalogd
-rwxr--r--   2 root     sys         733 Dec 11 08:54 S98deallocate
-rwxr--r--   5 root     sys        1023 Dec 11 08:53 S99audit
-rwxr--r--   5 root     sys        2804 Dec  7 20:52 S99dtlogin
```

Start Run Control Scripts

The `/etc/rc#.d` start scripts are run in the sort order shown by `ls`. The files beginning with `S` are run to start a system process. These scripts are called by the appropriate `/sbin/`*`rc#`* and this script passes the argument `start` to them if their names do not end in `.sh`. There are no arguments passed to `.sh` scripts. These files have names in the form of `S##`*`name-of-script`*. For example, the `S99dtlogin` script starts the graphical login process.

Stop Run Control Scripts

The `/etc/rc#.d` stop scripts (also referred to as the kill scripts) are always run in the sort order shown by the `ls` command. The files that begin with `K` are run to stop or kill a system process. These scripts are called by the appropriate `/sbin/`*`rc#`*, and this script passes the argument "`stop`" to them if their names do not end in `.sh`.

These files have names in the form of:

`K##`*`name-of-script`*

For example, the `K03samba` script stops the Samba server.

Note – File names that begin with a lowercase `k` or `s` are ignored, and are not executed. To disable a script, rename it with the appropriate lowercase letter.

The /etc/init.d Directory

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

The run control script /etc/init.d/samba is hard-linked to the corresponding run control script /etc/rc3.d/S90samba, as shown by the ls commands:

```
# cd /etc/init.d
# ls -i samba
4715 samba

# cd /etc/rc3.d
# ls -i S90samba
4715 samba
```

You can stop a process or start a process without changing the system run level.

For example, to stop and restart the samba file and print sharing service, run the following command with a start or stop argument:

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

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

Adding Scripts to Start and Stop Legacy Services

To add run control scripts to start and stop a service not managed by SMF, create the script in the /etc/init.d directory and create links in the appropriate /etc/rc#.d directory for the run level in which the service is to be started and stopped.

The following procedure describes how to add a run control script:

1. Create the script in the /etc/init.d directory.

```
# vi /etc/init.d/filename
# chmod 744 /etc/init.d/filename
# chgrp sys /etc/init.d/filename
```

2. Create links to the appropriate /etc/init.d directory.

```
# cd /etc/init.d
# ln filename /etc/rc#.d/S##filename
# ln filename /etc/rc#.d/K##filename
```

For instance, you might link a file in `/etc/init.d` called `myapplication` to a file called `/etc/rc3.d/S99myapplication`, with a corresponding stop script called `/etc/rc3.d/K01myapplication`.

3. Use the `ls` command to verify that the script has links in the appropriate directories.

```
# ls -li /etc/init.d/filename
# ls -li /etc/rc#.d/S##filename
# ls -li /etc/rc#.d/K##filename
```

4. Test the *`filename`* by performing the following commands:

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

Performing System Shutdown Procedures

Shut down the Solaris OS as follows:

- To perform administration or maintenance
- If you are anticipating a power outage
- If you need to move the system to a new location.

The Solaris OS requires a clean and orderly shutdown. A “clean and orderly” shutdown does the following:

- Stops processes
- Writes data in memory to disks
- Unmounts file systems.

The work you need to do while the system is shut down will determine how you shut the system down and which command you use to do so. The following describes the system shut-down types:

- Shut down the system to single-user mode
- Shut down the system to stop the Solaris OS, and display the `ok` or `Press any key to reboot` prompt
- Shut down the system and turn off power
- Shut down the system and automatically `reboot` to multiuser mode

The commands available to you, as the `root` user, for performing system shutdowns include:

- `/sbin/init` (using run levels S, 0, 1, 5, or 6)
  The `init` command accepts more arguments than listed here. Those arguments are not listed here because they are not part of system shutdown procedures.
- `/usr/sbin/shutdown` (using run levels S, 0, 1, 5, or 6)

Figure 11-5 shows the run-level transitions that occur during system boot or shut down on a SPARC-based system.

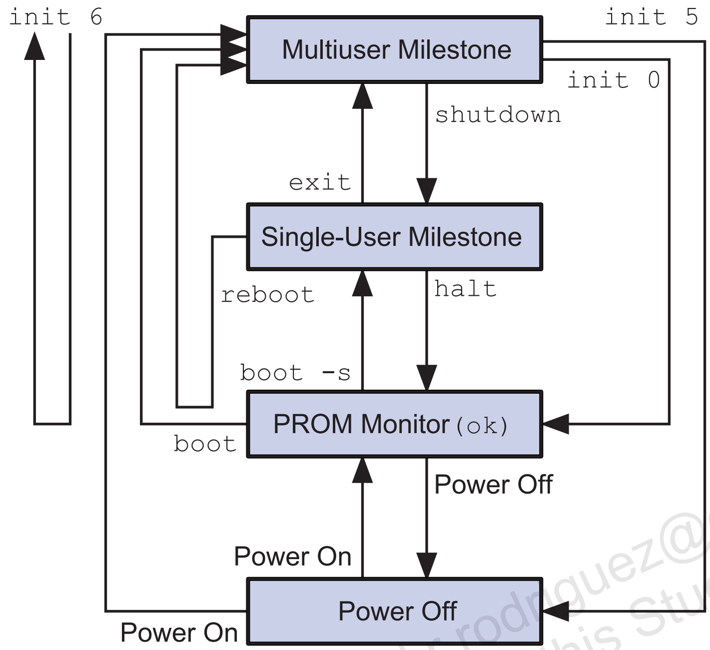


Figure 11-5 Run-Level Transitions

On x86/x64-based systems, `init 0`, `halt`, and the `shutdown` command with the `-i 0` option, shut down the Solaris OS, and leave the system at the `Press any key to reboot` prompt, rather than the `ok` prompt. Pressing any key causes a reset of the x86/x64 system. Without user intervention, x86/x64 systems boot to the default milestone when the boot process is initiated in this manner.

The `init 6`, `init 5`, and `reboot` commands on x86/x64 systems function as they do on SPARC-based systems.

Powering-on x86/x64 systems initiates a boot process to the default run level from the default source, and as such is equivalent to the `boot` command on SPARC systems. Similarly, selecting a boot option from the GRUB main menu and typing `b` boots the system using the source you selected.

Editing the `kernel` command in the command list for a GRUB boot option allows you to specify the `-s` option and boot an x86/x64 system to the `single-user` milestone.

The /usr/sbin/init Command

Use the init command to shut down, power off, or reboot a system in a clean and orderly manner. The init command informs the svc.startd daemon of the runlevel change. The init command doesn't warn logged-in users that the system is being shut down, and there is no grace period.

Command	Execution
# **init S**	Shuts down the system to single-user mode. Use run level S or 1
# **init 0**	Shuts down the system, stops the Solaris OS, and displays the ok or Press any key to reboot prompt.
# **init 5**	Shuts down the system and turns the system power off
# **init 6**	Shut down the system and reboots to multiuser mode

The /usr/sbin/shutdown Command

The shutdown command is a script that invokes the init daemon to shut down, power off, or reboot the system. It executes the rc0 kill scripts to shut down processes and applications gracefully. But unlike the init command, the shutdown command does the following:

- Notifies all logged-in users that the system is being shut down
- Delays the shutdown for 60 seconds by default
- Enables you to include an optional descriptive message to inform your users of what will transpire

The command format for the shutdown command is:

```
shutdown -y  -g grace-period -i init-state
optional message
```

The -y option pre-answers the final shutdown confirmation question so that the command runs without your intervention.

The `-g` *`grace-period`* allows you to change the number of seconds from the 60-second default.

The `-i` *`init-state`* specifies the run level that the `system` is to attain. By default, system state S is used.

Note – If the `shutdown` command displays the error message: "`shutdown:'i' - unknown flag`," it indicates that the shell has located and executed the `/usr/ucb/shutdown` command. Re-issue the command using its full path (for example, `/usr/sbin/shutdown`), or set the `PATH` variable to ensure `/usr/sbin` comes before `/usr/ucb`.

To shut down the system to single-user mode, enter the `shutdown` command without options.

```
# shutdown
```

To shut down the system to stop the Solaris OS, and display the `ok` or `Press any key to reboot` prompt, enter:

```
# shutdown -i0
```

To shut down the system and turn off its power automatically, enter:

```
# shutdown -i5
```

To shut down the system and reboot to multiuser mode, enter:

```
# shutdown -i6
```

You can use the `-i` option with other command options. For example, to shut down the system and reboot to multiuser mode, answer yes to the questions presented, provide a grace period of two minutes, and provide a message to the users by entering:

```
# shutdown -y -g120 -i6 "The system is being rebooted"
```

"Ungraceful" Shutdown Commands

The following commands perform an immediate system shutdown. They do not execute the `rc0` kill scripts. They do not notify logged-in users, and there is no grace period.

```
# halt
# poweroff
# reboot
```

Caution – These commands should be used with extreme caution and only when there is no alternative.

Setting the Default Boot-Time Milestone

Use the `svcadm` command to control the milestone the `svc.startd` daemon meets on boot. The default milestone, if one is not specified, is "all" which is an abstract milestone where all system services are started.

To ensure that the `svc.startd` daemon meets the requirements of the multi-user-server milestone on the next reboot, use the following command before rebooting:

```
# svcadm -v milestone -d multi-user-server:default
```

Valid options for default boot level using the `svcadm` command include the following:

- `all`
- `none`
- `svc:/milestone/single-user:default`
- `svc:/milestone/multi-user:default`
- `svc:/milestone/multi-user-server:default`

Notes:

Module 12

Service Management Facility

Objectives

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

- Describe Service Management Facility (SMF) features
- Compare run levels and SMF milestones
- Use SMF administrative commands

Introduction to SMF

Prior to Solaris 10, Sun systems used a series of scripts to stop and start processes associated with run levels. Each run level had an associated script located in the /sbin directory, with some scripts hard-linked to each other. The init daemon was responsible for starting and stopping system services. These structures are still present. See the "Performing Legacy Boot and Shutdown Procedures" module for more information.

The SMF was introduced in Solaris 10. It delivers a unified Solaris service configuration infrastructure that accurately models Solaris services and their interaction with Solaris and other services. SMF starts services in parallel, according to dependencies, which allows the system to boot faster, and reduces dependency conflicts.

The SMF provides a centralized configuration structure for managing system services and the interaction of a service with other services. The SMF includes the following:

- A mechanism to establish and formalize dependency relationships between services.
- Information on procedures to start, stop, and restart services.
- A centralized repository for information on startup behavior and service status.
- A structured mechanism for Fault Management of system services.
- Detailed information about misconfigured services such as an explanation of why a service is not running.
- Individual log files for each service.

Features

The SMF infrastructure includes the following:

- Service configuration repository
- Process re-starter
- Administrative CLI utilities
- Supporting kernel functionality

This infrastructure allows Solaris services to express:

- Restart requirements for the presence of prerequisite services and system facilities (such as networking)
- Requirements for identity and privileges for various tasks
- Configuration setting per service instance

Solaris services are modeled by describing them in terms of an SMF schema and associated service methods. For existing services converted to SMF services, compatibility or conversion of legacy configuration files is handled on a service-by-service basis. Once service descriptions are bootstrapped into SMF, instances of such services can be created, started, stopped, and status collected by the infrastructure. This saves time and system administration effort.

Service Definition

A *service* provides a resource or list of capabilities to applications and other services, both local and remote. A service isn't a running process, such as a web server. A service can be the software state of a device, such as a configured network device, or a mounted file system.

A system can have more than one occurrence of a service running. For example, a system can have more than one configured network interface, or more than one mounted file system.

Service Identifiers

In the SMF, each service instance has a *service identifier*. The service identifier is in the form of a fault management resource identifier (FMRI). The FMRI indicates the service type or category, name, and instance. The service categories include the following:

- `application`
- `device`
- `legacy`
- `milestone`
- `network`
- `platform`
- `site`
- `system`

An example of a service instance FMRI is:

```
svc:/system/filesystem/root:default
```

Where:

- The prefix `svc` indicates that this service is managed by SMF
- The category of the service is `system`
- The service itself is a `filesystem`
- The instance of the service is the `root` file system
- The word `default` identifies the first, in this case only, instance of the service

Another example of an FMRI for a service is:

```
lrc:/etc/rc3_d/S90samba
```

Where:

- The prefix, legacy run control (`lrc`), indicates that this service isn't currently managed by the SMF
- The pathname `/etc/rc3_d` refers to the directory `/etc/rc3.d` where there is a script used to manage this service
- The name of the script is `S90samba`

Listing Service Information

List service instance names and the service state of with the svcs command.

```
# svcs
STATE          STIME    FMRI
legacy_run     Feb_10   lrc:/etc/rc2_d/S10lu
legacy_run     Feb_10   lrc:/etc/rc2_d/S20sysetup
legacy_run     Feb_10   lrc:/etc/rc2_d/S90wbem
legacy_run     Feb_10   lrc:/etc/rc2_d/S99dtlogin
legacy_run     Feb_10   lrc:/etc/rc3_d/S81volmgt
(output omitted)
online         Feb_10   svc:/system/system-log:default
online         Feb_10   svc:/system/fmd:default
online         Feb_10   svc:/system/console-login:default
online         Feb_10   svc:/network/smtp:sendmail
online         Feb_10   svc:/milestone/multi-user:default
online         Feb_10   svc:/milestone/multi-user-server:default
online         Feb_10   svc:/system/zones:default
offline        Feb_10   svc:/application/print/ipp-listener:default
offline        Feb_10   svc:/application/print/rfc1179:default
maintenance   10:24:15  svc:/network/rpc/spray:default
```

Service States

Use the svcs command to list service identifiers and the service instance state. You can enable or disable a service. Service states include the following:

- online

  The service instance is enabled and has successfully started.

- offline

  The service instance is enabled, but the service is not yet running or available to run.

- disabled

  The service instance is not enabled and is not running.

- legacy_run

  The legacy service is not managed by SMF, but the service can be observed. This state is only used by legacy services.

- `uninitialized`

  This state is the initial state for all services before their configuration has been read.

- `maintenance`

  The service instance has encountered an error that must be resolved by the administrator.

- `degraded`

  The service instance is enabled, but is running at a limited capacity.

Milestones

A milestone service includes a defined set of other services. Regard a milestone as a system state to reach. A *milestone system state* requires a defined set of services to be running. These services depend on other services being available. Hence, there is a hierarchy of dependency relationships. The SMF manages this core feature. There are seven milestones.

- single-user
- multi-user
- multi-user-server
- network
- name-services
- sysconfig
- devices

Figure 12-1 shows the relationship between a milestone and services.

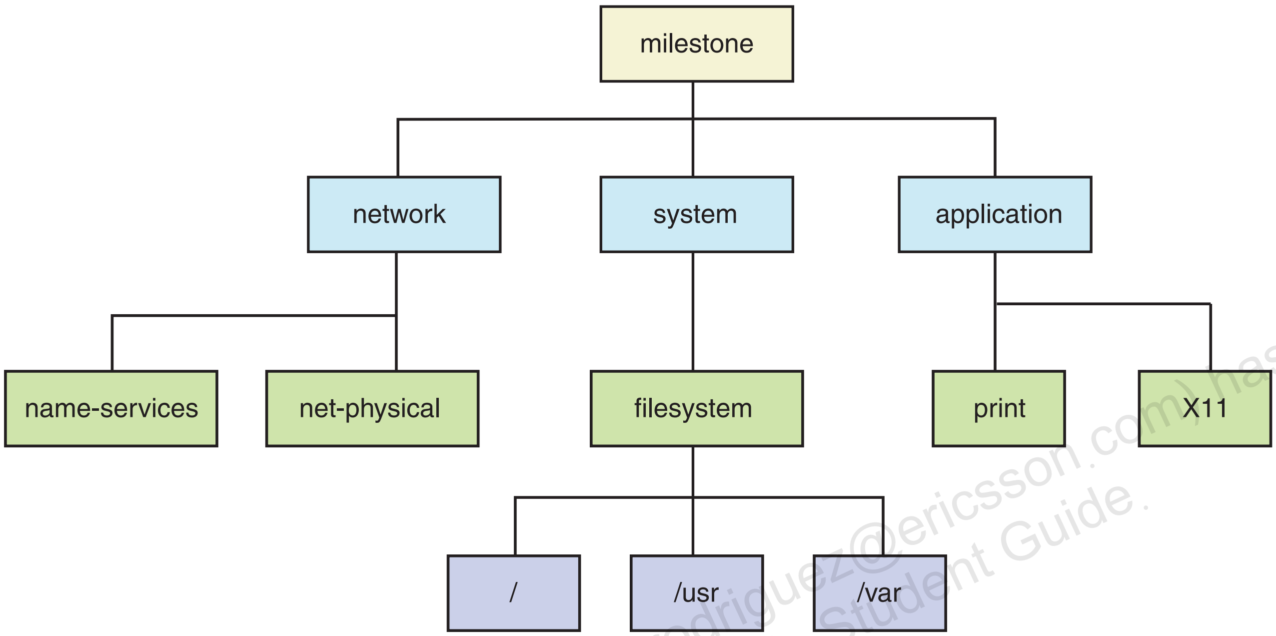


Figure 12-1 SMF Milestone and Services

Figure 12-2 shows an example of the dependency relationships.

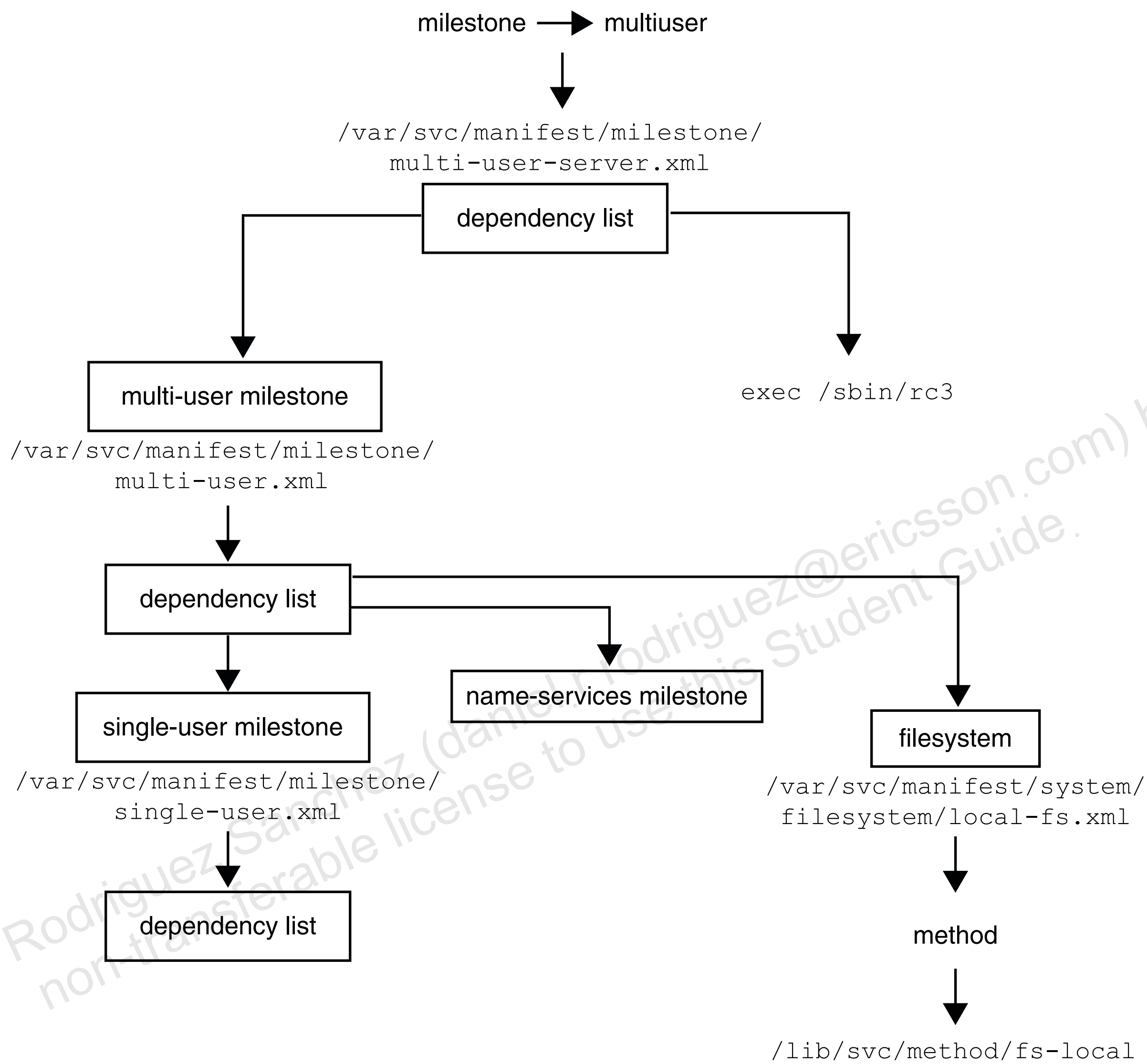


Figure 12-2 SMF Dependency Relationships

The `svc.startd` Daemon

The `svc.startd` daemon maintains system services and ensures that the system boots to the appropriate milestone. If you don't specify a milestone at boot up, `svc.startd` boots to the built-in milestone "all" which includes all system services. Use the following milestones at boot time:

- none
- single-user
- multi-user
- multi-user-server
- all

To boot a system to a specific milestone, the `-m` option is passed to the boot command from OBP.

```
ok> boot -m milestone=single-user
```

The `svc.startd` daemon could also be called the master restarter daemon because it is responsible for ensuring the correct running, starting, and restarting of system services. The `svc.startd` daemon can obtain information about services from the repository. This `init` process did this before.

The `svc.startd` daemon delegates responsibility for services to other delegated restarter daemons, for example, the `inetd` daemon.

The `svc.startd` daemon starts the appropriate processes for the achieved run level. The `svc.startd` daemon uses information in the repository to determine the required milestone and then starts to process the manifests located in the `/var/svc/manifest` directory.

`/var/svc/manifest/milestone` contains XML files which describe the dependencies for this milestone. A milestone includes multiple SMF services.

The following files are in the /var/svc/manifest/milestone directory:

- single-user.xml
- multi-user.xml
- multi-user-server.xml
- network.xml
- name-services.xml
- sysconfig.xml

These.xml files may refer to other directories that contain commands to run, such as:

- /sbin/rc2
- /lib/svc/method/fs-local

These .xml files may refer to other .xml files in subdirectories below /var/svc/manifest that contain commands to run, such as:

- /sbin/rc2
- /lib/svc/method/fs-local

The inetd Restarter

inetd is the delegated restarter for SMF. It manages service states in response to administrative requests, system failures, and service failures. When appropriate, inetd listens for network service requests.

With SMF, services are no longer managed by editing the inetd.conf configuration file. Instead, you use inetconv to convert the configuration file content into SMF format services, then you manage these services using inetadm and svcadm.

Setting inetd Managed Services Backlog Queue Size

Server-side resource demands can fill up a backlog queue. In Solaris 10 5/08, the SMF connection_backlog turnable property was introduced to enable you to manage an inetd managed services backlog queue size. The connection_backlog queue size default is 10, which can easily accommodate client requests. Modify the connection_backlog property by using the inetadm command.

To list the properties, type:

```
#inetadm -l fmri/pattern
```

To change the value for a specific service, type:

```
#inetadm -m fmri/pattern conection_backlog=new value
```

To change the value globally, type:

```
#inetadm -M connection_backlog=newvalue
```

Boot Archive Service

The boot archive service is controlled with the `svcadm` command and managed by SMF:

`svc:/system/boot-archive:default`

The Service Configuration Repository

The repository database stores information about the state of each service instance. It also stores configuration information about the services and system. The repository is distributed among local memory and local disk-based files. The disk-based database is `/etc/svc/repository.db`.

The repository is managed by the `svc.configd` daemon. The `svc.configd` daemon backs up the repository before applying any changes issued by the SMF commands and utilities. These backup copies of the repository ensure that fallback is possible.

A corrupt repository prevents the system from booting. A corrupt repository can be repaired by booting the system to single-user, and running the `/lib/svc/bin/restore_repository` command.

The Service Repository Database

A database is saved in the `/etc/svc` directory that contains details of the available services and their settings. Use the `/lib/svc/bin/restore_repository` utility to repair or restore a corrupt repository.

To see how the repository database is used, perform the following steps:

1. **`cd /lib/svc/bin`**
2. **`./restore_repository`**

Note – This example of the `restore_repository` script output is for the FCS Solaris 10 release, newer releases of Solaris 10 have different output from this script.

```
Repository Restore utility

See http://sun.com/msg/SMF-8000-MY for more information on the use of
this script to restore backup copies of the smf(5) repository.

If there are any problems which need human intervention, this script will
give instructions and then exit back to your shell.

Note that upon full completion of this script, the system will be
rebooted
using reboot(1M), which will interrupt any active services.

The following backups of /etc/svc/repository.db exist, from
oldest to newest:

manifest_import-20050221_112255
manifest_import-20050221_144358
boot-20050223_100423
boot-20050223_211258
boot-20050224_095929
boot-20050225_134532

The backups are named based on their type and the time what they were
taken.
Backups beginning with "boot" are made before the first change is made to
the repository after system boot.  Backups beginning with
"manifest_import"
```

```
are made after svc:/system/manifest-import:default finishes its
processing.
The time of backup is given in YYYYMMDD_HHMMSS format.

Please enter one of:
        1) boot, for the most recent post-boot backup
        2) manifest_import, for the most recent manifest_import backup.
        3) a specific backup repository from the above list
        4) -seed-, the initial starting repository.  (All customizations
           will be lost.)
        5) -quit-, to cancel.

Enter response [boot]: manifest_import-20050221_144358

After confirmation, the following steps will be taken:

svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
    -- renamed --> /etc/svc/repository.db_old_20050225_163816
/etc/svc/repository-manifest_import-20050221_144358
    -- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).

Proceed [yes/no]? no

Exiting...
#
```

In the above example, the alternate `repository.db` file naming convention is YYYYMMDD_number, therefore the `repository.db_old_20050225_163816` file was saved on February 25, 2005.

Stopping and Starting Services Using SMF Commands

Listing Services With the `svcs` Command

The `svcs` command is used to monitor SMF services. It is useful for examining the status of services, and for following the dependency relationship between services.

```
# svcs
STATE          STIME    FMRI
legacy_run     13:45:11 lrc:/etc/rcS_d/S29wrsmcfg
legacy_run     13:45:37 lrc:/etc/rc2_d/S10lu
legacy_run     13:45:38 lrc:/etc/rc2_d/S20sysetup
legacy_run     13:45:38 lrc:/etc/rc2_d/S40llc2
legacy_run     13:45:38 lrc:/etc/rc2_d/S42ncakmod
legacy_run     13:45:39 lrc:/etc/rc2_d/S47pppd
(output omitted)
online         13:45:36 svc:/network/smtp:sendmail
online         13:45:38 svc:/network/ssh:default
online         13:45:38 svc:/system/fmd:default
online         13:45:38 svc:/application/print/server:default
online         13:45:39 svc:/application/print/rfc1179:default
online         13:45:41 svc:/application/print/ipp-listener:default
online         13:45:45 svc:/milestone/multi-user:default
online         13:45:53 svc:/milestone/multi-user-server:default
online         13:45:54 svc:/system/zones:default
online          8:46:25 svc:/system/filesystem/local:default
online          8:46:26 svc:/network/inetd:default
online          8:46:32 svc:/network/rpc/meta:tcp
online          8:46:32 svc:/system/mdmonitor:default
online          8:46:38 svc:/milestone/multi-user:default
online         13:14:35 svc:/network/telnet:default
maintenance     8:46:21 svc:/network/rpc/keyserv:default
```

The `svcs` command can also be used to examine the status of a specific service instance. For example:

```
# svcs svc:/system/console-login:default
STATE          STIME    FMRI
online         14:38:27 svc:/system/console-login:default
#
```

Use the `svcs` command with the `-d` and the `-D` options to examine the dependency relationships of services. The `-d` option shows what other services the named service depends on. The `-D` option shows what other services depend on the named service.

The following example shows what the service dependencies are for the `filesystem/local:default` service instance:

```
# svcs -d svc:/system/filesystem/local:default
STATE          STIME    FMRI
online         Mar_16   svc:/system/filesystem/minimal:default
online         Mar_16   svc:/milestone/single-user:default
#
```

The following command is a example of changing the default milestone from milestone all to milestone multi-user-server:

```
# svcs -d milestone/multi-user:default
STATE          STIME    FMRI
disabled       Mar_16   svc:/application/print/server:default
disabled       Mar_16   svc:/network/ntp:default
disabled       Mar_16   svc:/system/auditd:default
disabled       Mar_16   svc:/system/mdmonitor:default
disabled       Mar_16   svc:/system/rcap:default
online         Mar_16   svc:/milestone/name-services:default
online         Mar_16   svc:/system/rmtmpfiles:default
online         Mar_16   svc:/system/name-service-cache:default
online         Mar_16   svc:/system/power:default
online         Mar_16   svc:/milestone/single-user:default
online         Mar_16   svc:/system/filesystem/local:default
online         Mar_16   svc:/system/cron:default
online         Mar_16   svc:/network/rpc/bind:default
online         Mar_16   svc:/platform/i86pc/kdmconfig:default
online         Mar_16   svc:/milestone/sysconfig:default
online         Mar_16   svc:/system/utmp:default
online         Mar_16   svc:/network/inetd:default
online         Mar_16   svc:/network/nfs/client:default
online         Mar_16   svc:/system/filesystem/autofs:default
online         Mar_16   svc:/system/system-log:default
online         Mar_16   svc:/system/system-log:default
online         Mar_16   svc:/network/smtp:sendmail
#
```

The following example shows what other services depend on the `system/filesystem/local` service:

```
# svcs -D svc:/system/filesystem/local
STATE          STIME    FMRI
disabled       Mar_16   svc:/application/management/webmin:default
disabled       Mar_16   svc:/application/print/server:default
disabled       Mar_16   svc:/application/gdm2-login:default
disabled       Mar_16   svc:/network/apocd/udp:default
disabled       Mar_16   svc:/system/auditd:default
disabled       Mar_16   svc:/system/pools/dynamic:default
disabled       Mar_16   svc:/network/inetd-upgrade:default
disabled       Mar_16   svc:/network/nfs/server:default
online         Mar_16   svc:/system/cron:default
online         Mar_16   svc:/application/opengl/ogl-select:default
online         Mar_16   svc:/network/nfs/status:default
online         Mar_16   svc:/system/sac:default
online         Mar_16   svc:/network/inetd:default
online         Mar_16   svc:/network/rpc/cde-ttdbserver:tcp
online         Mar_16   svc:/network/rpc/cde-calendar-manager:default
online         Mar_16   svc:/system/filesystem/volfs:default
online         Mar_16   svc:/system/filesystem/autofs:default
online         Mar_16   svc:/system/system-log:default
online         Mar_16   svc:/system/dumpadm:default
online         Mar_16   svc:/network/ssh:default
online         Mar_16   svc:/application/font/fc-cache:default
online         Mar_16   svc:/network/smtp:sendmail
online         Mar_16   svc:/milestone/multi-user:default
online         Mar_16   svc:/application/management/common-agent-
container-1:default
#
```

Changing Service States Using the `svcadm` Command

Use the `svcadm` command to change the service state. For example, to verify the status of the `cron` service:

```
# pgrep -fl cron
  180 /usr/sbin/cron
# svcs cron
STATE          STIME    FMRI
online         Mar_16   svc:/system/cron:default
# svcadm -v disable system/cron:default
svc:/system/cron:default disabled.
# svcs cron
STATE          STIME    FMRI
disabled       16:21:52 svc:/system/cron:default
# pgrep -fl cron
# svcadm -v enable system/cron:default
svc:/system/cron:default enabled.
# svcs cron
STATE          STIME    FMRI
online         16:22:28 svc:/system/cron:default
# pgrep -fl cron
 5502 /usr/sbin/cron
#
```

Use the `svcadm -v disable` command to disable the service. With the `svcadm -v disable` command, the service will remain disabled until you enable it from the command line. Disable the service temporarily, until the next reboot, by using the `-t` option.

```
# svcadm -v disable -t system/cron:default
svc:/system/cron:default temporarily disabled.
```

Using svcs to Determine Why Services are Not Running

Use the svcs command to troubleshoot why services are not running. In the following example, the cron service was temporarily disabled by the administrator.

```
# svcs -x cron
svc:/system/cron:default (clock daemon (cron))
 State: disabled since Fri Feb 25 15:05:47 2005
Reason: Temporarily disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-1S
   See: cron(1M)
   See: crontab(1)
   See: /var/svc/log/system-cron:default.log
Impact: This service is not running.
```

To find out more about a service, see the /var/svc/log/system-cron:default.log log file, and http://sun.com/msg/SMF-8000-1S.

Creating New Service Scripts

You can create new scripts to start and stop additional processes or services to customize a system. For example, to eliminate the requirement for a data base server manual start, you can create a script to start the database server automatically after the appropriate network services start. You could then create another script to terminate this service and shut down the database server before the network services are stopped.

You must incorporate the new service into the SMF. Doing this can be complex. The required steps are as follows:

- Determine the service start and stop process.
- Establish a service name and category.
- Determine if the service runs multiple instances.
- Identify any dependency relationships between this service and any other services.
- If a script is required to start and stop the process, create the script and place it in a local directory such as /usr/local/svc/method.

- Create a service manifest file for your service. This file describes the service and any dependency relationships. Service manifests are pulled into the repository either by using the `svccfg` command or at boot time.
- Incorporate the script into the SMF using the `svccfg` utility.

Here's an example:

```
# vi /usr/local/svc/method/newservice
#!/sbin/sh
#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "@(#)newservice 1.14    04/08/30 SMI"

case "$1" in
'start')
        /usr/bin/newservice &
;;

'stop')
    /usr/bin/pkill -x -u 0 newservice
        ;;
*)
        echo "Usage: $0 { start | stop }"
        ;;
esac
exit 0

# chmod 544 /usr/local/svc/method/newservice

# cd /var/svc/manifest/site
# vi newservice.xml
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM
"/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
        Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
        Use is subject to license terms.

        ident   "@(#)newservice.xml     1.2     04/09/13 SMI"
-->

<service_bundle type='manifest' name='OPTnew:newservice'>
```

```
<service
        name='site/newservice'
        type='service'
        version='1'>

        <single_instance/>
       <dependency
                name='usr'
                type='service'
                grouping='require_all'
                restart_on='none'>
                <service_fmri value='svc:/system/filesystem/local' />
        </dependency>

       <dependent
                name='newservice'
                grouping='require_all'
                restart_on='none'>
                <service_fmri value='svc:/milestone/multi-user' />
        </dependent>


        <exec_method
                type='method'
                name='start'
                exec='/usr/local/svc/method/newservice start'
                timeout_seconds='30' />

        <exec_method
                type='method'
                name='stop'
                exec='/usr/local/svc/method/newservice stop'
                timeout_seconds='30' />

        <property_group name='startd' type='framework'>
               <propval name='duration' type='astring' value='transient'
/>
        </property_group>

        <instance name='default' enabled='true' />

        <stability value='Unstable' />

        <template>
                <common_name>
                        <loctext xml:lang='C'>
```

```
                              New service
                        </loctext>
                  </common_name>
            </template>
</service>

</service_bundle>
```

The following describes the entries in the file:

- Standard header.

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM
"/usr/share/lib/xml/dtd/service_bundle.dtd.1">
```

- Comment section.

```
<!--
        Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
        Use is subject to license terms.

        ident   "@(#)newservice.xml     1.2     04/09/13 SMI"
-->
```

- The name of the service. The type (`manifest`) indicates a simple service rather than a milestone, the package providing the service, and the service name.

```
<service_bundle type='manifest' name='OPTnew:newservice'>
```

- Service category, type, name, and version.

```
<service
        name='site/newservice'
        type='service'
        version='1'>
```

- Whether multiple instances of the service will run.

```
<single_instance/>
```

- The service model to use. The entry shows that the service will be started by `svc.startd`. Transient services are started once and not restarted.

```
<property_group name='startd' type='framework'>
                <propval name='duration' type='astring' value='transient'
/>
        </property_group>
```

- How the service is started and stopped.

```
<exec_method
                type='method'
                name='start'
                exec='/usr/local/svc/method/newservice start'
                timeout_seconds='30' />

        <exec_method
                type='method'
                name='stop'
                exec='/usr/local/svc/method/newservice stop'
                timeout_seconds='30' />
```

- Define any dependencies for this service. The first entry states that the `newservice` requires the `filesystem/local` service.

```
    <dependency
                name='usr'
                type='service'
                grouping='require_all'
                restart_on='none'>
                <service_fmri value='svc:/system/filesystem/local' />
        </dependency>
```

- The second entry makes sure that your service is associated with the multi-user milestone and that the multi-user milestone requires this service.

```
<dependent
                name='newservice'
                grouping='require_all'
                restart_on='none'>
                <service_fmri value='svc:/milestone/multi-user' />
        </dependent>
```

- Creating the instance.

```
<instance name='default' enabled='true' />

        <stability value='Unstable' />
```

- Creating information to describe the service.

```
<template>
                <common_name>
                        <loctext xml:lang='C'>
                                New service
                        </loctext>
                </common_name>
        </template>
```

The new service (newservice) now needs to be imported into SMF.

This is done by running the `svccfg` utility:

```
# svccfg import /var/svc/manifest/site/newservice.xml
```

After the service has been imported into SMF it should be visible using the `svcs` command.

```
# svcs newservice
STATE          STIME    FMRI
online          8:43:45 svc:/site/newservice:default
#
```

It should also be possible to manipulate the service using `svcadm`.

```
# svcadm -v disable site/newservice
site/newservice disabled.
# svcs newservice
STATE          STIME    FMRI
disabled        9:11:38 svc:/site/newservice:default
# svcadm -v enable site/newservice
site/newservice enabled.
# svcs newservice
STATE          STIME    FMRI
online          9:11:54 svc:/site/newservice:default
#
```

The multiuser milestone has a dependency on the newservice. This dependency was created by the dependent entry in the newservice xml file. You can observe that the newservice has started before the multiuser milestone completed.

```
# svcs -d milestone/multi-user:default
STATE          STIME    FMRI
disabled        8:43:16 svc:/platform/sun4u/sf880drd:default
online          8:43:16 svc:/milestone/name-services:default
online          8:43:33 svc:/system/rmtmpfiles:default
online          8:43:42 svc:/network/rpc/bind:default
online          8:43:46 svc:/milestone/single-user:default
online          8:43:46 svc:/system/utmp:default
online          8:43:47 svc:/system/system-log:default
online          8:43:47 svc:/system/system-log:default
online          8:43:49 svc:/system/filesystem/local:default
online          8:44:01 svc:/system/mdmonitor:default
online          9:11:54 svc:/site/newservice:default
#
```


Manipulating Legacy Services Not Managed by SMF

If the FMRI prefix for a service is `lrc`, then that service is not currently managed by SMF. To start and stop a service without changing run levels, you must manually run the script associated with that service. The script to run a legacy service (non-SMF managed) is in `/etc/init.d`.

To identify and start or stop a legacy run service, perform the following steps:

1. Use the `svcs` command to display the list of legacy run services and identify the service you want to start or stop.

```
# svcs | grep legacy
legacy_run     Apr_05   lrc:/etc/rcS_d/S50sk98sol
legacy_run     Apr_05   lrc:/etc/rc2_d/S10lu
legacy_run     Apr_05   lrc:/etc/rc2_d/S20sysetup
legacy_run     Apr_05   lrc:/etc/rc2_d/S40llc2
legacy_run     Apr_05   lrc:/etc/rc2_d/S42ncakmod
legacy_run     Apr_05   lrc:/etc/rc2_d/S47pppd
legacy_run     Apr_05   lrc:/etc/rc2_d/S70uucp
legacy_run     Apr_05   lrc:/etc/rc2_d/S72autoinstall
legacy_run     Apr_05
lrc:/etc/rc2_d/S73cachefs_daemon
legacy_run     Apr_05
lrc:/etc/rc2_d/S81dodatadm_udaplt
legacy_run     Apr_05   lrc:/etc/rc2_d/S89PRESERVE
```

```
legacy_run     Apr_05   lrc:/etc/rc2_d/S94ncalogd
legacy_run     Apr_05   lrc:/etc/rc2_d/S98deallocate
legacy_run     Apr_05   lrc:/etc/rc3_d/S16boot_server
legacy_run     Apr_05   lrc:/etc/rc3_d/S50apache
legacy_run     Apr_05   lrc:/etc/rc3_d/S52imq
legacy_run     Apr_05   lrc:/etc/rc3_d/S80mipagent
legacy_run     Apr_05   lrc:/etc/rc3_d/S84appserv
legacy_run     Apr_05   lrc:/etc/rc3_d/S90samba
#
```

2. Verify that the script associated with the service you want to start or stop exists in the `/etc/init.d` directory.

```
# ls /etc/init.d/apache
/etc/init.d/apache
#
```

3. To start the service, run the script in `/etc/init.d` and specify the `start` argument.

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

Note – Some legacy services require configuration steps before you run these scripts and specify the `start` argument. Without a complete configuration, the processes that provide these services may not start.

4. To stop the service, run the script in `/etc/init.d` and specify the `stop` argument.

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

Secure By Default

Traditionally, Solaris systems provided many network services by default. This open approach was convenient, but it also made it easy for remote attackers to exploit any vulnerabilities that existed in the network services software. The Secure by Default (SBD) enhancement reduces vulnerabilities by disabling as many network services as possible while still leaving a useful system. The only network-listening service left enabled in the hardened configuration is SSH.

Installation and Secure by Default

Solaris Express (since 7/06) installs hardened by default. Initial installations of Solaris 10 OS update releases present a Secure by Default option during the installation process.

Custom Configurations

SBD uses the SMF to control network services. (Some legacy services were converted to SMF control in support of SBD.) Some services are disabled completely, while others are configured using SMF properties to accept input only from clients on the local system. Starting from this hardened baseline, you use SMF to enable services for network listening using the `svcadm`(1M) and `svccfg`(1M) commands.

Use the `netservices`(1M) command to disable network services. For example, use the following command to re-establish the hardened state after enabling individual services:

```
# netservices limited
```

Note – The existing `/var/svc/profile/generic_limited_net.xml` SMF profile was extended to implement this hardened state. The `netservices` script may become obsolete once SMF supports setting service properties in its profiles.

You can enable default services, as they were in previous Solaris releases, by running the following command:

```
# netservices open
```

Limited Networking Profile

Table 12-1 includes SMF service settings implemented for SBD plus the settings for existing services. It contains more than networking services (for completeness). The SSH service is enabled. New service properties were added to the first five listed services to implement the `local_only` configuration.

Table 12-1 Limited Networking Profile Details

Service	FMRI	Property	Action Taken
rpcbind	svc:/network/rpc/bind	config/local_only	limit to local connections
syslogd	svc:/system/system-log	config/ log_from_remote	limit to local connections
sendmail	svc:/network/smtp:sendmail	config/local_only	limit to local connections
smcwebserver	svc:/system/ webconsole:console	options/tcp_listen	limit to local connections
wbem	svc:/application/management/ wbem	options/tcp_listen	limit to local connections
X server	`svc:/application/x11/ x11-server`	options/tcp_listen	limit to local connections
X font server	svc:/application/x11/ xfs		disabled
dtlogin	svc:/application/graphical-login/cde-login	dtlogin/args	limit to local connections
ToolTalk	svc:/network/rpc/ cde-ttdbserver:tcp	proto=ticotsord	limit to local connections
dtcm	svc:/network/rpc/ cde-calendar-manager	proto=ticlts	limit to local connections
BSD print	svc:/application/print/ rfc1179:default	bind_addr=localhost	limit to local connections
Internet print protocol	svc:/application/print/ ipp-listener:default		disabled

Table 12-1 Limited Networking Profile Details (Continued)

Service	FMRI	Property	Action Taken
SVM remote metaset	svc:/network/rpc/meta		disabled
SVM remote mediator	svc:/network/rpc/metamed		disabled
SVM remote multihost disk	svc:/network/rpc/metamh		disabled
SVM communication	svc:/network/rpc/mdcomm		disabled
rstatd	svc:/network/rpc/rstat:default		disabled
rusersd	svc:/network/rpc/rusers:default		disabled
telnetd	svc:/network/telnet:default		disabled
statd	svc:/network/nfs/status		disabled
lockd	svc:/network/nfs/nlockmgr		disabled
NFS client	svc:/network/nfs/client		disabled
NFS server	svc:/network/nfs/server		disabled
rquotad	svc:/network/nfs/rquota		disabled
NFS v4 callback daemon	svc:/network/nfs/cbd		disabled
NFS id mapping	svc:/network/nfs/mapid		disabled
ftpd	svc:/network/ftp:default		disabled
fingerd	svc:/network/finger:default		disabled
rlogind	svc:/network/login:rlogin		disabled
rshd	svc:/network/shell:default		disabled
Secure Shell	svc:/network/ssh:default		enabled

Opening Selected Services

To establish a secure or hardened system, use the netservices limited command to lock down the system, and then selectively open the services you need.

Examine the symbolic link for /var/svc/profile to determine the current system security configuration. For example, the following command indicates that the system is open:

```
sys11# cd /var/svc/profile
sys11# ls -l generic.xml
lrwxrwxrwx   1 root     root          18 Jan  3 10:48 generic.xml ->
./generic_open.xml
```

The following netservices command sets the system in secure mode again:

```
sys11# netservices limited
restarting syslogd
Jan  4 14:54:56 sys11 syslogd: going down on signal 15
restarting sendmail
restarting wbem
dtlogin needs to be restarted. Restart now? [Y] Y
restarting dtlogin
```

The symlink verifies the restricted configuration:

```
sys11# ls -l generic.xml
lrwxrwxrwx   1 root     root          18 Jan  3 10:48 generic.xml ->
./generic_limited_net.xml
```

Services like telnet are now disabled as this captured session indicates:

```
instructor# telnet sys11ext
Trying 192.168.30.31...
telnet: Unable to connect to remote host: Connection refused
```

The following svcs command verifies that telnet is disabled:

```
sys11# svcs telnet
STATE          STIME    FMRI
disabled       14:54:53 svc:/network/telnet:default
```

Even for a limited system, SSH is still enabled as this session shows:

```
instructor# ssh root@sys11ext
The authenticity of host 'sys11ext (192.168.30.31)' can't be established.
RSA key fingerprint is c8:8b:35:00:65:0d:fd:94:96:eb:f3:95:27:87:7b:cf.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'sys11ext,192.168.30.31' (RSA) to the list of
known hosts.
Password: password
Last login: Wed Jan  3 16:12:52 2007
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
Welcome to SA300-S10_B on sys11
sys11#
```

The svcs command also verifies that sshd is accepting connections:

```
sys11# svcs svc:/network/ssh
STATE          STIME    FMRI
online         Jan_03   svc:/network/ssh:default
```

Removing Restrictions

After you verify that a system is closed, use the svcadm(1M) command to completely enable selected services. For example, the following command enables telnet in an otherwise hardened system. You can also enable services that support local mode.

```
# svcadm enable telnet
# svcs telnet
STATE          STIME    FMRI
online         10:09:58 svc:/network/telnet:default
```

As indicates, some services are enabled in local_only mode as a result of executing the netservices limited command. For services that support this restrictive configuration, use the svccfg(1M) command to remove the local_only restriction and completely open the service. For example, the following commands enable logging from remote systems for syslogd and make the change immediate:

```
# svccfg -s system-log setprop config/log_from_remote =
true
# svcadm refresh system-log
# svcadm restart system-log
```

Use the svcprop command to query service property settings. The following example shows how to verify that syslogd will not allow logging from remote hosts:

```
sys11# svcprop system-log | grep config
...
config/log_from_remote boolean true
...
```

In the following example, the sendmail service is configured in local_only mode after the netservices limited command is executed.

```
sys11# svcprop sendmail | grep config
...
config/local_only boolean true
...
```

You can use the svccfg and svcadm commands to remove the restriction and make the change immediate:

```
# svccfg -s sendmail setprop config/local_only = false
# svcadm refresh sendmail
# svcadm restart sendmail
```

Use the svcprop command to verify the change:

```
# svcprop sendmail | grep config
...
config/local_only boolean false
...
```

There are two configuration options:

1. Enabling or disabling a service
2. Setting a local_only property

For example, if the system-log service was configured local_only but disabled. Use svcadm enable to enable it for local use only. Similarly, if you set the local_only property using the svccfg command on a disabled service, nothing will happen until you enable it with the svcadm command.

This example shows how you use svccfg to open the Xserver service:

```
# svccfg -s x11-server setprop options/tcp_listen = true
```

In some cases, you may see status information that conflicts with the service state. For example, the results from entering the following `netstat` command indicate that `rpcbind` will accept network connections when it will not. This is because the SMF is controlling the service with a `local_only` property value of `true`.

```
# netstat -f inet -P tcp -a

TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q
State
-------------------- -------------------- ----- ------ ----- ------ -----
------
...
      *.sunrpc             *.*                 0      0 49152      0
LISTEN
...
      *.ssh                *.*                 0      0 49152      0
LISTEN
...
```

Most services implement the `local_only` mode by not binding to system IP addresses. The `rpcbind` command binds to a system IP address and discards any packets that are received from remote systems. This is necessary because RPC clients on a local system expect to be able to contact `rpcbind` using the system IP address.

Use the `svcprop` command to verify the correct security configuration.

The following example illustrates that while `rpcbind` listens for incoming connections, it refuses connection requests coming from remote systems.

Check the status of the `rpcbind` service on the local host:

```
sys-05# svcprop bind |grep config/local
config/local_only boolean false
sys-05# netstat -a |grep rpc
      *.sunrpc                            Idle
      *.sunrpc             *.*                 0      0 49152      0
LISTEN
sys-05#
```

From a remote host, attempt a connection using an RPC service, in this example, `rusers`:

```
sys-06# rusers -l sys-05
root         sys-05:pts/1              Apr  8 19:35       1
(192.168.201.1)
sys-06#
```

The connection succeeds. Set the `local_only` property for `rpcbind` on the local system to `true`, and verify that the status of `rpcbind` has not changed:

```
sys-05# svccfg -s bind setprop config/local_only = true
sys-05# svcadm refresh bind
sys-05# svcprop bind |grep config/local
config/local_only boolean true
sys-05# netstat -a |grep rpc
      *.sunrpc                            Idle
      *.sunrpc            *.*                 0      0 49152      0
LISTEN
sys-05#
```

Then test the remote connection with the same RPC service. The connection fails.

```
sys-06# rusers -l sys-05
sys-05: RPC: Rpcbind failure
sys-06#
```

Exercise: Managing Services With SMF

In this exercise, you practice adding, using, and removing services in the Solaris Management Facility. You also demonstrate the function of the `/etc/system` file.

Preparation

This exercise requires the `banner-rc`, `banner-smf`, and `banner-smf.xml` files, found in the `/opt/ses/lab/smf` directory. Your instructor will indicate if the files are located elsewhere. Refer to the lecture notes, as necessary, to perform the tasks listed.

Task

Complete the following steps:

1. Change to the directory `/opt/ses/lab/smf`, and check that the files `banner-rc`, `banner-smf`, and `banner-smf.xml` all have execute permission.
2. If necessary, use `chmod` to add execute permissions to these files.
3. Create the `/usr/local/svc/method` directory.
4. Copy the `banner-smf` file to the `/usr/local/svc/method` directory.
5. Copy the `banner-smf.xml` file to the `/var/svc/manifest/site` directory.
6. Verify that the `banner-smf` service is not yet part of SMF.
7. Import the new `banner-smf` service into the SMF repository.
8. Verify that the `banner-smf` service is now part of SMF.
9. Open a console window to view the output of the `banner-smf` service.
10. Use the `svcadm` command to disable the `banner-smf` service and view the output in the console window.
11. Use the `svcadm` command to enable the `banner-smf` service and view the output in the console window.
12. Shut down the system to run-state 0.

13. If you are using a SPARC system, boot the system with the `-m verbose` option.

    Does the startup message from `banner-smf` display?

14. If you are using an x86/x64-based system, press any key to initiate the boot process. Then:

    a. Enter `e` when the GRUB main menu displays.

    b. Select the kernel command line and enter `e` to use the GRUB editor to modify the line.

    c. Add the `-m verbose` option to the end of the `kernel` command line and press Return.

    d. Enter `b` to boot the system.

    Does the startup message from `banner-smf` display?

15. Log in as the `root` user and complete the following steps:

    a. Open a terminal window.

    b. Open a separate console window.

    c. Remove the `banner-smf` service from the system.

16. Copy the `banner-rc` script from the `/opt/ses/lab/smf` directory to the `/etc/init.d` directory.

17. Change directory to `/etc/init.d` and verify that the `banner-rc` script runs with both the `start` and `stop` arguments.

18. Change directory to `/etc/rc2.d` and create a hard link called `S22banner` that points to the same data as the `/etc/init.d/banner-rc` file.

19. Change directory to `/etc/rcS.d` and create a hard link called `K99banner` that points to the same data as the `/etc/init.d/banner-rc` file.

20. Reboot the system and watch for the output of the script you just installed.

    Does the startup message from `S22banner` display?.

21. Log in as the `root` user and complete the following steps:

    a. Open a terminal window.

    b. Use the `init` command to change to run level S.

    Does the shutdown message from `K99banner` appear?

22. Enter the password for the root user when it is requested and change to run level 3 by typing **`Control-D`**.

    Does the startup message from `S22banner` display?

23. Log in as the `root` user and complete the following steps:
    a. Open a terminal window.
    b. Change to the directory `/etc`.

24. Make a backup copy of the `/etc/system` file and name the backup file `system.orig`.

25. If your system uses a SCSI tape device, complete the following steps:
    a. Log in as the `root` user, and open a terminal window.
    b. Use the `prtconf` command to list instances of the `st` driver currently loaded.

       How many instances are reported?

       ____

    c. Edit the `/etc/system` file so that it includes the following line:

       `forceload: drv/st`

    d. Reboot the system.
    e. Log in as `root` and open a terminal window.
    f. Again, list instances of the `st` driver currently loaded.

       How many instances are reported?

       ____

26. Edit the `/etc/system` file so that it excludes the main disk driver for your system.

Note – On x86/x64 systems you must exclude both the 32-bit and 64-bit version of the driver.

---

27. Shut down the system to run level 0, and then attempt to boot it again.

    What happened?

28. If you are using a SPARC-based system, use the `boot -a` command to boot the system and complete the following steps:
    a. Supply the name of your backup file called `etc/system.orig`.

Note – There is *not* a slash before the `etc` directory name.

---

    b. Press Return to accept the default values for all other boot parameters.

29. If you are using a SPARC-based system, log in as the `root` user and complete the following steps:
    a. Open a terminal window.
    b. Copy the `/etc/system.orig` file to the `/etc/system` file.
    c. Reboot the system.

30. If you are using an x86/x64-based system, complete the following steps:
    a. When the GRUB main menu displays, press the space bar to cancel the timeout and automatic boot.
    b. Select the `Solaris failsafe` boot option, and boot the system.
    c. When prompted, choose to mount the Solaris instance below `/a`.
    d. Change directory to `/a/etc`, and copy `system.orig` to `system`.
    e. Change directory to root (/), and reboot the system.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Managing Services With SMF

In this exercise, you practice adding, using, and removing services in the Solaris Management Facility. You also demonstrate the function of the /etc/system file.

Preparation

This exercise requires the banner-rc, banner-smf, and banner-smf.xml files, found in the /opt/ses/lab/smf directory. Your instructor will indicate if the files are located elsewhere. Refer to the lecture notes, as necessary, to perform the tasks listed.

Task

Complete the following steps:

1. Change to the directory /opt/ses/lab/smf, and check that the files banner-rc, banner-smf, and banner-smf.xml all have execute permission.

```
# cd /opt/ses/lab/smf
# ls -l banner*
-rwxr-xr-x   1 root  root        347 Feb 28 14:26 banner-rc
-rwxr-xr-x   1 root  root        314 Feb 28 14:29 banner-smf
-rwxr-xr-x   1 root  root        989 Feb 28 14:30 banner-smf.xml
#
```

2. If necessary, use chmod to add execute permissions to these files.

```
# chmod a+x banner*
```

3. Create the /usr/local/svc/method directory.

```
# mkdir -p /usr/local/svc/method
```

4. Copy the banner-smf file to the /usr/local/svc/method directory.

```
# cp banner-smf /usr/local/svc/method
```

5. Copy the banner-smf.xml file to the /var/svc/manifest/site directory.

```
# cp banner-smf.xml /var/svc/manifest/site
```

6. Verify that the banner-smf service is not yet part of SMF.

```
# svcs site/banner-smf
svcs: Pattern 'site/banner-smf' doesn't match any instances
STATE          STIME    FMRI
#
```

7. Import the new banner-smf service into the SMF repository.

```
# svccfg import /var/svc/manifest/site/banner-smf.xml
#
```

8. Verify that the banner-smf service is now part of SMF.

```
# svcs site/banner-smf
STATE          STIME    FMRI
online          9:11:54 svc:/site/banner-smf:default
#
```

9. Open a console window to view the output of the banner-smf service.

```
# dtterm -C &
```

Or:

```
# xterm -C &
```

10. Use the svcadm command to disable the banner-smf service and view the output in the console window.

```
# svcadm -v disable site/banner-smf
svc:/site/banner-smf:default disabled.
# svcs site/banner-smf
STATE          STIME    FMRI
disabled   9:11:54 svc:/site/banner-smf:default
```

11. Use the svcadm command to enable the banner-smf service and view the output in the console window.

```
# svcadm -v enable site/banner-smf
svc:/site/banner-smf:default enabled.
# svcs site/banner-smf
STATE          STIME    FMRI
online     9:11:54 svc:/site/banner-smf:default
```

12. Shut down the system to run-state 0.

```
# init 0
```

SPARC-based systems display the ok prompt, x86/x64-based systems display the Press any key to reboot prompt.

13. If you are using a SPARC system, boot the system with the `-m verbose` option. Use the following command:

    ```
    ok boot -m verbose
    ```

    Does the startup message from `banner-smf` display?

    Yes.

14. If you are using an x86/x64-based system, press any key to initiate the boot process. Then:

    a. Enter `e` when the GRUB main menu displays.

    b. Select the kernel command line and enter `e` to use the GRUB editor to modify the line.

    c. Add the `-m verbose` option to the end of the `kernel` command line and press Return. For example:

```
grub edit> kernel /platform/i86pc/multiboot -m verbose <Return>
```

    d. Enter `b` to boot the system.

    Does the startup message from `banner-smf` display?

    Yes.

15. Log in as the `root` user and complete the following steps:

    a. Open a terminal window.

    b. Open a separate console window.

    ```
    # dtterm -C &
    ```

    Or:

    ```
    # xterm -C &
    ```

    c. Remove the `banner-smf` service from the system.

```
# svcadm -v disable site/banner-smf
svc:/site/banner-smf:default disabled
# svccfg delete -f svc:/site/banner-smf:default
# svcs site/banner-smf
svcs: Pattern 'site/banner-smf' doesn't match any instances
STATE          STIME    FMRI
# rm /var/svc/manifest/site/banner-smf.xml
```

16. Copy the `banner-rc` script from the `/opt/ses/lab/smf` directory to the `/etc/init.d` directory.

    ```
    # cp /opt/ses/lab/smf/banner-rc /etc/init.d
    ```

17. Change directory to /etc/init.d and verify that the banner-rc script runs with both the start and stop arguments.

```
# cd /etc/init.d
# ./banner-rc start
# ./banner-rc stop
```

18. Change directory to /etc/rc2.d and create a hard link called S22banner that points to the same data as the /etc/init.d/banner-rc file.

```
# cd /etc/rc2.d
# ln /etc/init.d/banner-rc S22banner
```

19. Change directory to /etc/rcS.d and create a hard link called K99banner that points to the same data as the /etc/init.d/banner-rc file.

```
# cd /etc/rcS.d
# ln /etc/init.d/banner-rc K99banner
```

20. Reboot the system and watch for the output of the script you just installed.

```
# init 6
```

Does the startup message from S22banner display?

Yes.

21. Log in as the root user and complete the following steps:
    a. Open a terminal window.
    b. Use the init command to change to run level S.

```
# init S
```

Does the shutdown message from K99banner appear?

Yes.

22. Enter the password for the root user when it is requested and change to run level 3 by typing **Control-D**.

```
# Control-d
```

Does the startup message from S22banner display?

Yes.

23. Log in as the root user and complete the following steps:
    a. Open a terminal window.
    b. Change to the directory /etc.

```
# cd /etc
```

24. Make a backup copy of the /etc/system file and name the backup file system.orig.

```
# cp system system.orig
```

25. If your system uses a SCSI tape device, complete the following steps:

    a. Log in as the root user, and open a terminal window.

    b. Use the prtconf command to list instances of the st driver currently loaded.

    ```
    # prtconf | grep "st, instance"
    ```

    How many instances are reported?

    None

    c. Edit the /etc/system file so that it includes the following line:

    ```
    forceload: drv/st
    ```

    d. Reboot the system.

    ```
    # init 6
    ```

    e. Log in as root and open a terminal window.

    f. Again, list instances of the st driver currently loaded.

    ```
    # prtconf | grep "st, instance"
    ```

    How many instances are reported?

    The number varies depending on how many SCSI controllers are present. You should see Instances 0 through 6 for a system with one controller.

26. Edit the /etc/system file so that it excludes the main disk driver for your system.

    - On SPARC systems using SCSI disks, add the following line:

    ```
    exclude: drv/sd
    ```

    - On SPARC systems using IDE disks, add the following line:

    ```
    exclude: drv/dad
    ```

    - On x86/x64 systems using SATA disks, add the following two lines:

    ```
    exclude: drv/cmdk
    exclude: drv/amd64/cmdk
    ```

If you are uncertain of the type of disk or disk driver used for your boot disk, use the following procedure to determine the diver in use:

a. Use the `df` command to display information for the root (/) file system.

```
# df -h /
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c1d0s0        4.8G   228M   4.5G     5%    /
#
```

b. Use the `ls -lL` command to display physical device file information for the root device.

```
# ls -lL /dev/dsk/c1d0s0
brw-r-----   1 root     sys      102,  0 Mar  1 11:31 /dev/dsk/c1d0s0
#
```

c. Search in `/etc/name_to_major` for the major device number listed in the previous command. This identifies the driver in use, in this example, `cmdk`. Specify this driver in the `exclude` directive that you create in `/etc/system`.

```
# grep 102 /etc/name_to_major
cmdk 102
#
```

27. Shut down the system to run level 0, and then attempt to boot it again.

```
# shutdown -y -i0 -g0
(shutdown messages)
```

On a SPARC system:

```
ok boot
```

On an x86/x64 system:

```
Press any key to reboot <press any key>
```

What happened?

The system is unable to boot. Excluding this driver prevents you from using the boot disk so long as you use the same `/etc/system` *file. You must boot using the* `-a` *option to be able to supply an alternative file for the* `/etc/system` *file, or boot from the failsafe boot archive on x86/x64 systems to correct the problem.*

28. If you are using a SPARC-based system, use the `boot -a` command to boot the system and complete the following steps:

    a. Supply the name of your backup file called `etc/system.orig`.

Note – There is *not* a slash before the `etc` directory name.

---

    b. Press Return to accept the default values for all other boot parameters. For example:

```
ok boot -a
Name of system file [/etc/system]: etc/system.orig
root filesystem type [ufs]: <Return>
Enter physical name of root device [/...]: <Return>
```

29. If you are using a SPARC-based system, log in as the `root` user and complete the following steps:

    a. Open a terminal window.

    b. Copy the `/etc/system.orig` file to the `/etc/system` file.

    c. Reboot the system.

```
# cd /etc
# cp system.orig system
# init 6
```

30. If you are using an x86/x64-based system, complete the following steps:

    a. When the GRUB main menu displays, press the space bar to cancel the timeout and automatic boot.

    b. Select the `Solaris failsafe` boot option, and boot the system.

    c. When prompted, choose to mount the Solaris instance below `/a`.

    d. Change directory to `/a/etc`, and copy `system.orig` to `system`.

```
# cd /a/etc
# cp system.orig system
#
```

    e. Change directory to root (/), and reboot the system.

```
# cd /
# init 6
(output omitted)
```

Notes:

Module 13

Performing User Administration

Objectives

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

- Describe user administration fundamentals
- Manage user accounts
- Manage initialization files

Introducing User Administration

An important system administration task is setting up user accounts for each user who requires system access. Each user needs a unique account name, a user identification (UID) number, a home directory, and a login shell. You also have to determine which groups a user may access.

Main Components of a User Account

The following is a list of the main components of a user account:

- User name – A unique name that a user enters to log in to a system. The user name is also called the login name.
- Password – A combination of up to 256 letters, numbers, or special characters that a user enters with the login name to gain access to a system.

Note – To enable 256 character passwords, the encryption policy in `/etc/security/policy.conf` needs to be changed to either `md5` or `blowfish`. The line that reads: `CRYPT_DEFAULT=_unix_` needs to be changed to `CRYPT_DEFAULT=2a (blowfish)`. The values `1`, `2a`, and `md5` are explained in `crypt.conf`.

- UID number – A user account's unique numerical identification within the system.
- Group identification (GID) number – A unique numerical identification of the group to which the user belongs.

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

- Comment – Information that identifies the user. A comment generally contains the full name of the user and optional information, such as a phone number or a location.
- User's home directory – A directory into which the user is placed after login. The directory is provided to the user to store and create files.
- User's login shell – The user's work environment is set up by the initialization files that are defined by the user's login shell.

System Files That Store User Account Information

The Solaris 10 OS stores user account and group entry information in the following system files:

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

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

The /etc/shadow file is a separate file that contains the encrypted passwords. To further control user passwords, you can enforce password aging. This information is also maintained in the /etc/shadow file.

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

The /etc/passwd File

Due to the critical nature of the /etc/passwd file, you should refrain from editing this file directly. Instead, you should use the Solaris™ Management Console or command-line tools to maintain the file.

The following is an example of an /etc/passwd file that contains the default system account entries.

```
root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webservd:x:80:80:WebServer Reserved UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
```

Each entry in the `/etc/passwd` file contains seven fields. A colon separates each field. The following is the format for an entry:

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

Figure 13-1 defines the requirements for each of the seven fields.

Figure 13-1 Fields in the `/etc/passwd` File

Field	Description
loginID	Represents the user's login name. It should be unique to each user. The field should contain a string of no more than eight letters (A–Z, a–z) and numbers (0–9). The first character should be a letter, and at least one character should be lowercase. **Note –** Even though some programs allow a maximum of 32 characters, as well as user names that contain periods (.), underscores (_), and hyphens (-), this practice is not recommended and might cause problems with other programs.
x	Represents a placeholder for the user's encrypted password, which is kept in the `/etc/shadow` file.
UID	Contains the UID number used by the system to identify the user. UID numbers for users range from 100 to 60000. Values 0 through 99 are reserved for system accounts. UID number 60001 is reserved for the `nobody` account. UID number 60002 is reserved for the `noaccess` account. While duplicate UID numbers are allowed, they should be avoided unless absolutely required by a program. **Note –** The maximum value for a UID is 2147483647. However, the UIDs over 60000 do not have full utility and are incompatible with some Solaris OS features. Avoid using UIDs over 60000 so as to be compatible with earlier versions of the operating system.

Figure 13-1 Fields in the `/etc/passwd` File (Continued)

Field	Description
GID	Contains the GID number used by the system to identify the user's primary group. GID numbers for users range from 100 to 60000. (Those between 0 and 99 are reserved for system accounts.)
comment	Typically contains the user's full name.
home_directory	Contains the full path name to the user's home directory.
login_shell	Defines the user's login shell. There are six possible login shells in the Solaris OS: the Bourne shell, the Korn shell, the C shell, the Z shell, the BASH shell, and the TC shell.

Table 13-1 shows the default system account data for entries in the `/etc/passwd` file.

Table 13-1 Default System Account Entries

User Name	User ID	Description
`root`	0	The `root` account that has access to the entire system. It has almost no restrictions and overrides all other logins, protections, and permissions.
`daemon`	1	The system daemon account that is associated with routine system tasks.
`bin`	2	The administrative daemon account that is associated with running system binary files.
`sys`	3	The administrative daemon account that is associated with system logging or updating files in temporary directories.
`adm`	4	The administrative daemon account that is associated with system logging.
`lp`	71	The line printer (`lp`) daemon account.
`uucp`	5	The daemon account associated with UNIX®-to-UNIX Copy Protocol (`UUCP`) functions.
`nuucp`	6	The account used by remote systems to log in to the host and start file transfers using `uucp`.

Table 13-1 Default System Account Entries (Continued)

User Name	User ID	Description
smmsp	25	The sendmail message submission daemon account.
listen	37	The network listener daemon account.
gdm	50	Gnome Display Manager daemon.
webservd	80	Account reserved for WebServer access.
nobody	60001	The anonymous user account that is assigned by a Network File System (NFS) server when an unauthorized root user makes a request. The nobody user account is assigned to software processes that do not need any special permissions.
noaccess	60002	The account assigned to a user or a process that needs access to a system through some application instead of through a system login procedure.
nobody4	65534	The anonymous user account that is the SunOS™ 4.*x* software version of the nobody account

Note – The nobody account secures NFS resources. When a user is logged in as root on an NFS client and attempts to access a remote file resource, the UID number changes from 0 to the UID of nobody (60001)

The /etc/shadow File

Due to the critical nature of /etc/shadow, don't edit it directly. Use the command-line tools. Only a root user can read the /etc/shadow file.

The following is an example /etc/shadow file that contains initial system account entries.

```
root:rJrdhjNWQQHoY:6445::::::
daemon:NP:6445::::::
bin:NP:6445::::::
sys:NP:6445::::::
adm:NP:6445::::::
lp:NP:6445::::::
uucp:NP:6445::::::
nuucp:NP:6445::::::
smmsp:NP:6445::::::
```

```
listen:*LK*:::::::
gdm:*LK*:::::::
webservd:*LK*:::::::
nobody:*LK*:6445::::::
noaccess:*LK*:6445::::::
nobody4:*LK*:6445::::::
```

Each entry in the `/etc/shadow` file contains nine fields. A colon separates each field.

Following is the format of an entry:

`loginID``:`*`password`*`:`*`lastchg`*`:`*`min`*`:`*`max`*`:`*`warn`*`:`*`inactive`*`:`*`expire`*`:`*`flag`*

Table 13-2 defines the requirements for each of the eight fields.

Table 13-2 Fields in the `/etc/shadow` File

Field	Description
loginID	The user's login name.
password	A 13-character encrypted password. The string `*LK*` indicates a locked account, and the string `NP` indicates no valid password. Passwords must be constructed to meet the following requirements: Each password must be at least six characters and contain at least two alphabetic characters and at least one numeric or special character. It cannot be the same as the login ID or the reverse of the login ID.
lastchg	The number of days between January 1, 1970, and the last password modification date.
min	The minimum number of days required between password changes.
max	The maximum number of days the password is valid before the user is prompted to enter a new password at login.
warn	The number of days the user is warned before the password expires.
inactive	The number of inactive days allowed for the user before the user's account is locked.

Table 13-2 Fields in the `/etc/shadow` File (Continued)

expire	The date (given as number of days since January 1, 1970) when the user account expires. After the date is exceeded, the user can no longer log in.
flag	To track failed logins. The count is in low order four bits. The remainder is reserved for future use, set to zero.

The `/etc/group` File

Each user belongs to a group that is referred to as the user's primary group. The GID number, located in the user's account entry within the `/etc/passwd` file, specifies the user's primary group.

Each user can also belong to up to 15 additional groups, known as secondary groups. In the `/etc/group` file, you can add users to group entries, thus establishing the user's secondary group affiliations.

Note – Line length limits of each group entry differ depending on the source. The maximum length for a line in the `/etc/group` file is 512 characters. The maximum length for a line in an NIS group is 1024 characters. The maximum length for a line in an NIS+ group is 4096 characters.

The following is an example of the default entries in an `/etc/group` file:

```
root::0:
other::1:root
bin::2:root,daemon
sys::3:root,bin,adm
adm::4:root,daemon
uucp::5:root
mail::6:root
tty::7:root,adm
lp::8:root,adm
nuucp::9:root
staff::10:
daemon::12:root
sysadmin::14:
smmsp::25:
gdm::50:
webservd::80:
```

```
nobody::60001:
noaccess::60002:
nogroup::65534::
```

Each line entry in the `/etc/group` file contains four fields. A colon character separates each field. The following is the format for an entry:

`groupname:group-password:GID:username-list`

Table 13-3 defines the requirements for each of the four fields.

Table 13-3 Fields in the `/etc/group` File

Field	Description
groupname	Contains the name assigned to the group. Group names contain up to a maximum of eight characters.
group-password	Usually contains an empty field or an asterisk. This is a relic of earlier versions of UNIX. **Caution –** A group-password is a security hole because it might allow an unauthorized user who is not a member of the group but who knows the group password, to enter the group. **Note –** The `newgrp` command changes a user's primary group association within the shell environment from which it is executed. If this new, active group has a password and the user is not a listed member in that group, the user must enter the password before the `newgrp` command can continue.
GID	Contains the group's GID number. It is unique on the local system and should be unique across the organization. Numbers 0 to 99, 60001, 60002 and 65534 are reserved for system group entries. User-defined groups range from 100 to 60000.

Table 13-3 Fields in the `/etc/group` File (Continued)

Field	**Description**
`username-list`	Contains a comma-separated list of user names that represent the user's secondary group memberships. By default, each user can belong to a maximum of 15 secondary groups. **Note –** The maximum number of groups is set by the `kernel` parameter called `ngroups_max`. You can set this parameter in the `/etc/system` file to allow for a maximum of 32 groups. Not all applications will be able to reference group memberships greater than 16. NFS is a notable example.

The `/etc/default/passwd` File

Set values for the following parameters in the `/etc/default/passwd` file to control properties for all users' passwords on the system:

- `MAXWEEKS` – Sets the maximum time period (in weeks) that the password is valid.
- `MINWEEKS` – Sets the minimum time period before the password can be changed.
- `PASSLENGTH` – Sets the minimum number of characters for a password. Valid entries are 6, 7, and 8.
- `WARNWEEKS` – Sets the time period prior to a password's expiration to warn the user that the password will expire.

Note – The `WARNWEEKS` value does not exist by default in the `/etc/default/passwd` file, but it can be added.

The password aging parameters `MAXWEEKS`, `MINWEEKS`, and `WARNWEEKS` are default values. If set in the `/etc/shadow` file, the parameters in that file override those in the `/etc/default/passwd` file for individual users.

The Solaris 10 OS release introduces a number of new controls for password management. These controls are configured by setting values in the `/etc/default/passwd` file. These controls are commented out by default.

- NAMECHECK=NO – Sets the password controls to verify that the user is not using the login name as a component of the password.
- HISTORY=0 – Forces the passwd program to log up to 26 changes to the user's password. This prevents the user from reusing the same password for 26 changes. If the HISTORY value is set to another number other than zero (0), and then set back to zero, it causes the password log for a user to be removed on the next password change.
- DICTIONLIST= – Causes the passwd program to perform dictionary word lookups from comma-separated dictionary files.
- DICTIONDBDIR=/var/passwd – The location of the dictionary where the generated dictionary databases reside. This directory must be created manually.

Note – To pre-build the dictionary database, refer to the man page for mkpwdict(1M).

Complexity of the password can be controlled using the following parameters:

```
#MINDIFF=3
#MINALPHA=2
#MINNONALPHA=1
#MINUPPER=0
#MINLOWER=0
#MAXREPEATS=0
#MINSPECIAL=0
#MINDIGIT=0
#WHITESPACE=YES
```

By default, all of the above parameters are commented out.

Note – By forcing greater complexity of password structure, you may inadvertently cause the users to write down their passwords as they may be too difficult for the user to remember. When setting a password change policy, you must not underestimate the problems that too much complexity may cause.

Password Management

The Solaris 10 OS has new security enhancements. The `pam_unix_auth` module implements account locking for local users. Account locking is enabled by the `LOCK_AFTER_RETRIES` tunable parameter in `/etc/security/policy.conf` and the `lock_after-retries` key in `/etc/user_attr`.

The `LOCK_AFTER_RETRIES=YES|NO` parameter specifies whether a local account is locked after the number of failed login attempts for a user is equal to, or exceeds the allowed number of retries. The number of retries is defined by `RETRIES` in `/etc/default/login`.

Note – These files are discussed in greater detail in: *SA-202-S10; System Administration for the Solaris 10 Operating System, Part 2.*

The `passwd` command has two new options, `-N` and `-u`. The `-N` option creates a password entry for a non-login account. This option is useful for accounts that should not be logged in to, but must run `cron` jobs. The `-u` option unlocks a previously locked account. The `passwd -N` *`username`* command sets the password field in `/etc/shadow` to `NP` which is an unmatchable password. This effectively disables the account from logging in.

For more information, see the `passwd(1)` man page.

The following example shows how to prevent a user from reusing too many previous passwords.

```
# vi /etc/default/passwd
(output omitted)
```

Locate the line called `#HISTORY=0`, and remove the comment from the beginning of the line. Modify the number to `3`, so the line shows as `HISTORY=3`. Write and quit the file. As a regular user, log in and attempt to change your password a number of times, using different passwords and then one of the previous passwords.

```
# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
login: testuser
```

```
Password: 123pass
$ passwd
passwd: Changing password for testuser
Enter existing login password: 123pass
New Password: pass123
Re-enter new Password: pass123
passwd: password successfully changed for testuser
$ passwd
passwd: Changing password for testuser
Enter existing login password: pass123
New Password: 123pass
passwd: Password in history list.
Please try again
New Password: newpas1
Re-enter new Password: newpas1
passwd: password successfully changed for testuser
$
```

By uncommenting the `HISTORY=` line in the `/etc/default/passwd` file, prior password history is checked. By changing the value to `3`, the number of prior password changes to keep and check when a user changes passwords is set to three.

Setting Password Aging on a User Account

If you are using NIS+ or the `/etc` files to store user account information, you can set up password aging on a user's password. Starting in the Solaris 9 12/02 release, password aging is also supported in the LDAP directory service.

Password aging enables you to force users to change their passwords periodically or to prevent a user from changing a password before a specified interval. If you want to prevent an intruder from gaining undetected access to the system by using an old and inactive account, you can also set a password expiration date when the account becomes disabled. You can set password aging attributes with the `passwd` command.

Managing User Accounts

Each of the following sections present two sets of command-line tools for managing user accounts: the command-line tools used in the Solaris OS versions prior to the Solaris 10 OS, and the new set of command-line tools developed for the Solaris 10 OS.

Introducing Command-Line Tools

The Solaris OS provides these command-line tools, defined as follows:

- `useradd` – Adds a new user account on the local system
- `usermod` – Modifies a user's account on the local system
- `userdel` – Deletes a user's account from the local system
- `groupadd` – Adds a new group entry to the system
- `groupmod` – Modifies a group entry on the system
- `groupdel` – Deletes a group entry from the system

In addition to these standard command-line tools, the Solaris 9 and 10 OS has a set of command-line tools that accomplish the same tasks. They are the `smuser` and `smgroup` commands.

The `smuser` command enables you to manage one or more users on the system with the following set of subcommands:

- `add` – Adds a new user account
- `modify` – Modifies a user's account
- `delete` – Deletes a user's account
- `list` – Lists one or more user entries

The `smuser` and `smgroup` commands interact with naming services, can use autohome functionality, and are better suited for remote management.

Note – The `smuser` and `smgroup` commands are the command-line interface equivalent to the Solaris Management Console range of operation, and allow you to perform Solaris Management Console actions in scripts. Therefore, the `smuser` and `smgroup` commands have numerous subcommands and options designed to function across domains and multiple systems. This module describes only the basic commands.

The smgroup command enables you to manage one or more groups on the system with the following set of subcommands:

- add – Adds a new group entry
- modify – Modifies a group entry
- delete – Deletes a group entry
- list – Lists one or more group entries

Any subcommand to add, modify, list, or delete users with the smuser and smgroup commands requires authentication with the Solaris Management Console server and requires the initialization of the Solaris Management Console. For example, the following is the command format for the smuser command:

```
/usr/sadm/bin/smuser subcommand [auth_args] -- [subcommand_args]
```

The authorization arguments are all optional. However, if you do not specify the authorization argument, the system might prompt you for additional information, such as a password for authentication purposes.

The -- option separates the subcommand-specific options from the authorization arguments. The -- option must be entered even if an authorization argument is not specified because it must precede the subcommand arguments.

The subcommand arguments are quite numerous. For a complete listing of the subcommands, refer to the smuser man page. It is important to note that descriptions and other arguments that contain white space must be enclosed in double quotation marks.

Creating a User Account

Use the useradd or smuser add command to add new user accounts to the local system. These commands add an entry for a new user into the /etc/passwd and /etc/shadow files.

These commands also automatically copy all the initialization files from the /etc/skel directory to the user's new home directory.

The `useradd` Command Format and Options

The following is the command format for the `useradd` command:

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

Table 13-4 shows the options for the `useradd` command.

Table 13-4 Options for the `useradd` Command

Option	Definition
`-u` *uid*	Sets the UID number for the new user
`-g` *gid*	Defines the new user's primary group
`-G` *gid*	Defines the new user's secondary group memberships
`-d` *dir*	Defines the full path name for the user's home directory
`-m`	Creates the user's home directory if it does not already exist
`-s` *shell*	Defines the full path name for the shell program of the user's login shell
`-c` *comment*	Specifies any comment, such as the user's full name and location
loginname	Defines the user's login name for the user account
`-D`	Displays the defaults that are applied to the `useradd` command

The following example uses the `useradd` command to create an account for a user named `newuser1`. It assigns `100` as the UID number, adds the user to the group `other`, creates a home directory in the `/export/home` directory, and sets `/bin/ksh` as the login shell for the user account.

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

The `useradd` command has a preset range of default values. These values can be displayed using the `useradd -D` command. When this command has been used for the first time, the `useradd` command generates a file called `/var/sadm/defadduser` that contains the default values. If the contents of this file are amended, the new contents become the default values for the next time the `useradd` command is used.

```
# ls -l /usr/sadm/defadduser
/usr/sadm/defadduser: No such file or directory
# useradd -D
group=other,1  project=default,3  basedir=/home
skel=/etc/skel  shell=/bin/sh  inactive=0
expire=  auths=  profiles=  roles=  limitpriv=
defaultpriv=  lock_after_retries=
# ls -l /usr/sadm/defadduser
-rw-r--r--   1 root     root         286 Oct 17 09:04
/usr/sadm/defadduser
# cat /usr/sadm/defadduser
#   Default values for useradd.  Changed Sun Oct 17 09:04:27 2004
defgroup=1
defgname=other
defparent=/home
defskel=/etc/skel
defshell=/bin/sh
definact=0
defexpire=
defauthorization=
defrole=
defprofile=
defproj=3
defprojname=default
deflimitpriv=
defdefaultpriv=
deflock_after_retries=
```

User accounts are locked by default when added with the `useradd` command. This can be verified by viewing the contents of the `/etc/shadow` file:

```
# grep 'newuser1' /etc/shadow
newuser1:*LK*:12708::::::
```

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

Use the passwd command to create a password for the new account.

```
# passwd newuser1
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for newuser1
```

This password setting can be verified by viewing the contents of the /etc/shadow file:

```
# grep 'newuser1' /etc/shadow
newuser1:M0/jo1fmSbYio:12708::::::
```

The smuser add Command Format and Options

The following is the command format for the smuser add command:

```
smuser add [auth_args] -- [subcommand_args]
```

Table 13-5 shows some of the most common subcommand arguments for the smuser add command.

Table 13-5 Subcommand Arguments for the smuser add Command

Subcommand Argument	**Definition**
-c *comment*	A short description of the login, typically the user's name. This string can be up to 256 characters.
-d *directory*	Specifies the home directory of the new user and is limited to 1024 characters.
-g *group*	Specifies the new user's primary group membership.
Subcommand Argument	**Definition**
-G *group*	Specifies the user's secondary group membership.
-n *login*	Specifies the user's login name.
-s *shell*	Specifies the full path name of the user's login shell.

Table 13-5 Subcommand Arguments for the `smuser add` Command (Continued)

`-u` *`uid`*	Specifies the user ID of the user you want to add. If you do not specify this option, the system assigns the next available unique UID greater than 100.
`-x` *`autohome=Y|N`*	Sets the home directory to automount if set to Y.

The following example uses the `smuser add` command to create an account for a user named `newuser2`. It designates the login name as `newuser2`, assigns the UID number `500`, adds the user to the group `other`, creates a home directory in the `/export/home` directory, and sets `/bin/ksh` as the login shell for the user account.

Note – The `-x autohome=N` option to the `smuser` command adds the user without automounting the user's `home` directory. See the man page for `automount` for more information.

```
# /usr/sadm/bin/smuser add -- -n newuser2 -u 500 -g other -d
/export/home/newuser2 -c "Regular User Account 2" -s /bin/ksh -x
autohome=N
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02
was successful.
```

Users are added without a password by default with the `smuser` command. This can be verified by viewing the appropriate entry in the `/etc/shadow` file:

```
# grep 'newuser2' /etc/shadow
newuser2::12708::::::
```

Use the `passwd` command to create a new password for the user.

```
# passwd newuser2
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for newuser2
```

Confirm that the password change has been applied by viewing the entry for that user in the /etc/shadow file:

```
# grep 'newuser2' /etc/shadow
newuser2:FSMOsxncoc6yI:12708::::::
```

Modifying a User Account

Use the usermod or smuser modify command to modify a user's login account on the system.

The usermod Command Format and Options

The following is the command format for the usermod command:

```
usermod [ -u uid [ -o ] ] [ -g gid ] [ -G gid [ , gid . . . ] ]
[ -d dir ] [ -m ] [ -s shell ] [ -c comment ]
[ -l newlogname] loginname
```

In general, the options for the usermod command function the same as those for the useradd command.

Table 13-6 shows the key options to the usermod command.

Table 13-6 Key Options for the usermod Command

Option	Definition
-o	Allows a UID to be duplicated.
-m	Moves the user's home directory to the new location specified with the -d option.
-l *newlogname*	Changes a user's login name for the specified user account.
-f *inactive*	Sets the number of inactive days that are allowed on a user account. If the account is not logged in to for the specified number of days, it is locked.
-e *expire*	Sets an expiration date on the user account. Specifies the date (*mm/dd/yy*) on which a user can no longer log in and access the account. After that date, the account is locked.
loginname	Identifies the user's login name for the current user account.

The following example changes the login name and home directory for newuser1 to usera.

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

The smuser modify Command Format and Options

The following is the command format for the smuser modify command:

```
smuser modify [auth_args] -- [subcommand_args]
```

In general, the options for the smuser modify command function the same as for the smuser add command. Refer to the smuser(1M) man page for additional options.

Table 13-7 shows the options for the smuser modify command.

Table 13-7 Options for the smuser modify Command

Option	Definition
-n *login*	Specifies the user's login name
-N *login*	Specifies the user's new login name

The following example changes the login name and home directory for newuser2 to userb.

```
# /usr/sadm/bin/smuser modify -- -n newuser2 -N userb -d
/export/home/userb
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02 was
successful.
```

Deleting a User Account

Use the `userdel` command or `smuser delete` command to delete a user's login account from the system.

The following is the command format for the `userdel` command:

```
userdel -r login
```

The `userdel` command also removes the user's home directory and all of its contents if you request it to do so. Use the `-r` option to remove the user's home directory from the local file system. This directory must exist.

The following example removes the login account for a user named `usera`.

```
# userdel usera
```

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

```
# userdel -r usera
```

Note – This command does not remove all files owned by the user, just the home directory. The system administrator should run a `find` command to locate all files owned by the user to be backed up or removed.

The `smuser delete` Command Format and Options

The following is the command format for the `smuser delete` command:

```
smuser delete [auth_args] -- [subcommand_args]
```

The following example removes the `userb` account from the system:

```
# /usr/sadm/bin/smuser delete -- -n userb
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys-02 was
successful.
```

Note – Unlike the userdel command, the smuser delete command has no -r equivalent option for deleting the home directory. The user's home directory must be deleted manually.

Creating a Group Entry

As the root user, you create new group entries on the local system by using the groupadd or smgroup add command. These commands add an entry for the new group into the /etc/group file. Like the smuser command, the smgroup add command uses the same subcommands and authentication arguments derived from the Solaris Management Console.

The groupadd Command Format and Options

The following is the command format for the groupadd command:

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

Table 13-8 shows the options for the groupadd command.

Table 13-8 Options for the groupadd Command

Option	Description
-g *gid*	Assigns the GID number for the new group
-o	Allows the GID number to be duplicated

The following example uses the groupadd command to create the new group classroom on the local system:

```
# groupadd -g 301 classroom
UX: groupadd: classroom name too long.
```

Note – A warning message will be written if the string exceeds MAXGLEN, which is usually set at eight characters in the /usr/include/userdefs file.

This example will work without changing the defaults:

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

The smgroup add Command Format and Options

The following is the command format for the smgroup add command:

```
/usr/sadm/bin/smgroup subcommand [auth_args] -- [subcommand_args]
```

Table 13-9 shows the options for the smgroup add command.

Table 13-9 Options for the smgroup add Command

Option	Description
-g *gid*	Specifies the GID number for the new group
-m *group_member*	Specifies the new members to add to the group
-n *group_name*	Specifies the name of the new group

The following example uses the smgroup add command to create a new group called workgroup with a GID of 123, and to add usera to the group:

```
# /usr/sadm/bin/smgroup add -- -n workgroup -g 123 -m usera
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
was successful.
```

Modifying a Group Entry

You can use the following commands to modify a group entry:

- The groupmod command
- The smgroup modify command

The groupmod Command Format and Options

The following is the command format for the groupmod command:

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

Table 13-10 defines the options for the `groupmod` command:.

Table 13-10 Options for the `groupmod` Command

Options	Description
`-g` *gid*	Specifies the new GID number for the group
`-o`	Allows the GID number to be duplicated
`-n` *name*	Specifies the new name for the group

The following example changes the `class1` account group GID number to `400`:

```
# groupmod -g 400 class1
```

The `smgroup modify` Command Format and Options

The following is the command format for the `smgroup modify` command:

```
/usr/sadm/bin/smgroup subcommand [auth_args] -- [subcommand_args]
```

Table 13-11 shows the options for the `smgroup modify` command.

Table 13-11 Options for the `smgroup modify` Command

Option	Description
`-n` *name*	Specifies the name of the group you want to modify
`-m` *new_member*	Specifies the new members to add to the group
`-N` *new_group*	Specifies the new group name

The following example changes the group `workgroup` to `schoolgroup`:

```
# /usr/sadm/bin/smgroup modify -- -n workgroup -N schoolgroup
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: Enter_The_root_Password
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
was successful.
```

Deleting a Group Entry

Use the `groupdel` or `smgroup delete` commands to delete a group entry from the `/etc/group` file on the system.

The `groupdel` Command Format

The following is the command format for the `groupdel` command:

```
groupdel groupname
```

The following example removes the group entry `class1` from the local system:

```
# groupdel class1
```

The `smgroup delete` Command Format and Options

The following is the command format for the `smgroup delete` command:

```
/usr/sadm/bin/smgroup  subcommand [auth_args] -- [subcommand_args]
```

Use the *-n group_name* option with the `smgroup delete` command to specify the name of the group you want to delete.

The following example deletes the group entry `schoolgroup` from the local system:

```
# /usr/sadm/bin/smgroup delete -- -n schoolgroup
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
Login to sys-02 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys-02
was successful.
```

Troubleshooting Login Issues

Some of the most common problems you might encounter as a system administrator are user login problems. There are two categories of login problems: login problems when the user logs in at the command line and login problems when the user logs in from the Common Desktop Environment (CDE) or the GNOME desktop.

The CDE uses more configuration files, so there are more potential problems associated with logging in from the CDE. When you troubleshoot a login problem, first determine whether you can log in from the command line. Attempt to log in from another system by using either the `telnet` command or the `rlogin` command, or click Options from the CDE login panel and select Command Line Login. If you can log in successfully at the command line, then the problem is with the CDE configuration files. If you cannot log in at the command line, then the problem is more serious and involves key configuration files.

Login Problems at the Command Line

Table 13-12 presents an overview of common login problems that occur when the user logs in at the command line.

Table 13-12 Login Problems at the Command Line

Login Problem	Description
`Login incorrect`	This message occurs when there are problems with the login information. The most common cause of an incorrect login message is a mistyped password. Make sure the that correct password is being used, and then attempt to enter it again. Remember that passwords are case-sensitive, so you cannot interchange uppercase letters and lowercase letters. In the same way, the letter "o" is not interchangeable with the numeral "0" nor is the letter "l" interchangeable with the numeral "1."
`Permission denied`	This message occurs when there are login, password, or NIS+ security problems. Most often, an administrator has locked the user's password or the user's account has been terminated.

Table 13-12 Login Problems at the Command Line (Continued)

Login Problem	Description
`Password will not work at lockscreen`	A common error is to have the Caps Lock key on, which causes all letters to be uppercase. This does not work if the password contains lowercase letters.
`No shell`	This message occurs when the user's shell does not exist, is typed incorrectly, or is wrong in the `/etc/passwd` file.
`No directory! Logging in with home=/`	This message occurs when the user cannot access the home directory for one of the following reasons: An entry in the `/etc/passwd` file is incorrect, or the home directory has been removed or is missing, or the home directory exists on a mount point that is currently unavailable.
`Choose a new password` (followed by the `New password:` prompt)	This message occurs the first time a user logs in and chooses an initial password to access the account.
`Couldn't fork a process!`	This message occurs then the server could not fork a child process during login. The most common cause of this message is that the system has reached its maximum number of processes. You can either kill some unneeded processes (if you are already logged into that system as `root`) or increase the number of processes your system can handle.

Login Problems in the CDE

Problems associated with logging into the CDE range from a user being unable to login (and returning to the CDE login screen), to the custom environment not loading properly. In general, the system does not return error messages to the user from the CDE. The following is a list of files and directories that provide troubleshooting information about the CDE:

- `/usr/dt/bin/Xsession`

  This file is the configuration script for the login manager. This file should not be edited. The first user-specific file that the `Xsession` script calls is the *$HOME*/`.dtprofile` file.

- `$HOME/.dtprofile`

  By default, the file does not contain much content, except for examples. It contains a few `echo` statements for session logging purposes, and the `DTSOURCEPROFILE` variable is set. But it also contains information about how it might be edited. The user can edit this file to add user-specific environment variables.

- `DTSOURCEPROFILE=true`

  This line allows the user's `$HOME/.login` file (for `csh` users) or the `$HOME/.profile` (for other shell users) to be sourced as part of the startup process.

  Sometimes a `.login` or `.profile` file contains problem commands that cause the shell to crash. If the `.dtprofile` file is set to source a `.login` or `.profile` file that has problem commands, desktop startup might fail.

  Consequently, no desktop appears. Instead, the system redisplays the Solaris OS CDE login screen. Startup errors from the `.login` or `.profile` file are usually noted in the `$HOME/.dt/startlog` file. Use a Failsafe login Session or a command-line login to debug problem commands in the `.login` or `.profile` files.

- `$HOME/.dt/sessions`

  This directory structure contains files and directories that configure the display of the user's custom desktop and determine the applications that start when the user logs in. Look for recent changes to files and for changes to the directory structure. For example, examine the `home` directory and the `home.old` directory or a `current` directory and the `current.old` directory. Compare the changes. The changes could provide information on a new application or on changes in the saved desktop that cause the user's login to fail.

- `$HOME/.dt`

  Upon removing the entire `.dt` directory structure, log out, and log back in again for the system to rebuild a default `.dt` file structure. This action allows the user to get back into the system if the problem with the CDE files cannot be resolved.

Table 13-13 shows the locations of and information found in error logs for the CDE.

Table 13-13 CDE Error Log Locations

Location	Error Log
`/var/dt/Xerrors`	The Solaris OS CDE login window system errors that occur prior to user login
$HOME`/.dt/startlog`	The Solaris OS CDE errors that occur during the startup of the `Xsession` script, while processing the `.dtprofile`, `.login`, or `.profile file`
$HOME*`/.dt/errorlog.old` *$HOME`/.dt/errorlog.older`	The Solaris OS CDE errors that occur after the `Xsession` script start up
$HOME`/.dt/sessionlogs`	Directory of session logs for Session Manager and Window Manager errors

You may now perform the Adding User and Group Accounts exercise.

Login Problem in the GNOME Desktop

There may be a problem with the GNOME Desktop. A menu item was missing from the GNOME 2.0 Desktop and it may not appear in the login screen.

GNOME `dtlogin` resource files are currently only installed for the C locale. To make the GNOME login option appear in other locales, you must copy this resource file to the correct directory for your locale. Use the following command to copy the dtlogin file. Future releases will include localized versions of this file.

```
cp /usr/dt/config/C/Xresources.d/Xresources.Sun-gnome-
2.0* /usr/dt/config/{your locale}/Xresources.d Managing
Initialization Files
```

Managing Initialization Files

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

When users log in to the system, their login shells look for and execute two different types of initialization files. The first type controls the system-wide environment. The second type controls the user's environment. The six shells available in the Solaris 10 OS provide basic features and a set of variables which the `root` user or a regular user can set in the initialization files to customize the shell environment.

The shells support two types of variables:

- Environment variables – Variables that provide information about the user's environment to every shell program that is started.
- Local variables – Variables that affect only the current shell. Any subshell started would not have knowledge of these variables.

Introducing System-Wide Initialization Files

As the system administrator, you maintain the system-wide initialization files. These files provide an environment for the entire community of users who log in to the system. The Solaris OS provides the system initialization files. They reside in the `/etc` directory.

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

The Bourne, Korn, and BASH login shells look for and execute the system initialization file `/etc/profile` during login.

The C login shell looks for and executes the system initialization file `/etc/.login` during the login process.

Note – The default files `/etc/profile` and `/etc/.login` check disk usage quotas, print the message of the day from the `/etc/motd` file, and check for mail. None of the messages are printed to the screen if the `.hushlogin` file exists in the user's home directory.

Introducing User Initialization Files

As the system administrator, you set up the user initialization files that are placed in each user account's home directory when the user is created.

The primary purpose of the user initialization files is to define the characteristics of a user's work environment, such as the command-line prompt, the environment variables, and the windowing environment.

Only the owners of the files or the `root` user can change or customize the content of these files.

Table 13-14 shows the initialization files necessary for each primary shell available in the Solaris 10 OS.

Table 13-14 Initialization Files for the Primary Shells

Shells	System-Wide Initialization Files	Primary User Initialization Files Read at Login	User Initialization Files Read When a New Shell Is Started	Shell Path Name
Bourne	`/etc/profile`	`$HOME/.profile`		`/bin/sh`
Korn	`/etc/profile`	`$HOME/.profile` `$HOME/.kshrc`	`$HOME/.kshrc`	`/bin/ksh`
C	`/etc/.login`	`$HOME/.cshrc` `$HOME/.login`	`$HOME/.cshrc`	`/bin/csh`

For additional information about the Z, BASH, and TC shells available in the Solaris 10 OS, refer to the online manual pages.

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

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

Customizing the User's Work Environment

The Solaris OS provides a set of initialization file templates. The `/etc/skel` directory contains the initialization file templates. Table 13-15 shows the default initialization file templates and the user initialization files for the Bourne, Korn, and C shells.

Table 13-15 Default User Initialization Files

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

Note – The `useradd` command copies files from the `/etc/skel` directory to the `$HOME` directory. The `smuser` command copies files from the `/etc/skel` directory to the `$HOME` directory and renames them to the appropriate file names.

The `root` user can customize these templates to create a standard set of user initialization files. A standard set of user initialization files provides a common work environment for each user. When the `root` user creates new user accounts, some or all of these initialization files are automatically copied to each new user's home directory.

Users can then edit their initialization files to further customize their environments for each shell.

Table 13-16 shows some of the variables available for customizing a user's shell environment.

Table 13-16 Login Variables

Variable Name	Set By	Description
LOGNAME	Login	Defines the user's login name.
HOME	Login	Sets the path to the user's home directory. It is the default argument for the cd command.
SHELL	Login	Sets the path to the default shell.
PATH	Login	Sets the default path that the shell searches to find commands.
MAIL	Login	Sets the path to the user's mailbox.
TERM	Login	Defines the terminal.
LPDEST	Not set by default	Sets the user's default printer.
PWD	Shell	Defines the current working directory.
PS1	Shell	Defines the shell prompt for the Bourne or Korn shell.
prompt	Shell	Defines the shell prompt for the C shell.

Note – For complete information on all variables used by the default shells, see the following man pages: `sh(1)`, `ksh(1)`, `csh(1)`, `zsh(1)`, `bash(1)`, and `tcsh(1)`.

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

Table 13-17 shows how to set environment variables in the user initialization files of the Bourne, Korn, and C shells.

Table 13-17 Setting Environment Variables

Shell	User's Initialization File

Bourne or Korn	`VARIABLE=value ; export VARIABLE`
	For example:
	`PS1="$HOSTNAME "; export PS1`

C	`setenv variable value`
	For example:
	`setenv LPDEST laserprinter`

You may now perform the Modifying Initialization Files exercise.

Exercise: Adding User and Group Accounts

In this exercise, you use the Solaris Management Console, and the `smuser`, `smgroup`, `usermod`, `userdel`, `groupadd`, and `groupdel` commands, to create, modify, and delete multiple user accounts and group entries.

Preparation

This exercise assumes that no user accounts exist except for the default accounts created during a default Solaris OS installation. Use the `logins` command to display the list of user accounts. Look for accounts named using the names listed in Table 13-19. Use the `userdel -r` command to remove any of these accounts that you find.

Refer to the lecture notes, as necessary, to perform the tasks listed. Refer to Table 13-18 and Table 13-19 as needed.

Table 13-18 Group Specifications

Group Name	GID Number
class1	101
class2	102

Table 13-19 User Specifications

User Name	Password	Shell	UID	Primary Group	Secondary Group
user3	123pass	Korn	1003	10	class1
user4	123pass	C	1004	10	class1
user5	123pass	Bourne	1005	10	
locked1	Select Account is Locked	Bourne	2001	10	
cleared1	Select User must set password at next login	Bourne	2002	10	

Note – Some of the commands displayed in this section are quite long and will wrap to the next line. You should consider all of the bold typeface commands that follow a command line prompt to be all one line.

Task 1 – Disabling the Solaris OS Registration Window

Complete the following steps:

1. Log in as the `root` user and open a terminal window (or use the `su` command to change your identity to `root`).
2. Change to the `/etc/default` directory.
3. In the `/etc/default` directory, use the `vi` editor to create a file called `solregis`.
4. In the `solregis` file, create a line that reads:

   ```
   DISABLE=1
   ```

Note – The character "1" is the number one.

5. Save this file, and then exit the editor.

Task 2 – Adding Group Entries

Complete the following steps:

Note – Refer to Table 13-18 for details while adding groups with the tools used in this exercise.

1. As the `root` user, open a terminal window.
2. Use the `groupadd` and `smgroup` commands to add the groups `class1` and `class2`, respectively.

   The `smgroup` command will request a password for authentication. You should respond to the prompts as required.

Task 3 – Adding User Accounts

Complete the following steps:

Note – Refer to Table 13-19 for details while adding users with the tools used in this exercise.

---

1. If the `/export/home` directory does not exist, create it.
2. Use the `useradd` command to add a user named `user3`, and then set a password for `user3`.
3. Use the `smuser` command to add a user named `user4` and complete the following steps:
   a. Respond to `smuser` authentication prompts, as required.

Note – Bug ID 6478291 for Solaris 10 Update 3 identifies an issue with the `smuser` and `smrole` commands. If `smuser` reports this error: `Authentication failed to the Solaris Management Console server`, and you have specified the correct authentication information, use `useradd` instead to add the `user4` account.

---

   b. As an alternative, use `useradd` to create the `user4` account.
   c. Set a password for `user4`.
4. Launch the Solaris Management Console by typing **`smc&`** on the command line.

   After the Solaris Management Console displays, complete the following steps to create a user template and add user accounts that do not use automounted home directories:

   a. Select the This Computer toolbox in the Open Toolbox panel that displays when you start SMC, and click the Open button.
   b. Select This Computer, then System Configuration, then Users, and then User Templates to open the User Templates tool.

      You should respond to authentication prompts as required.
   c. From the Menu Bar, select Action, and then select Add User Template.

      The Add User Template window appears, containing blank fields for a template name and description.
   d. Enter the name `SolarisUser` in the User Template Name field and `Class Template` for the Description field.

  e. Click the Home Directory Tab and complete the following steps:
     1. Uncheck the Automatically Mount Home Directory check box.
     2. Enter the name of your system in the Home Directory Server field.
  f. Click OK to create your template.

5. Complete the following steps:
   a. Click User Accounts, and add the `user5` account by selecting Action, then Add User, and then selecting From Template on the menu bar.

      The Add User From Template window appears.
   b. Enter `user5` in the User Name field and enter `1005` in the User ID Number field.
   c. Click the User Must Use button, and then enter `123pass` in both password fields for password information.
   d. Click OK, and confirm your choice as requested.
6. From the Solaris Management Console, complete the following steps:
   a. Use the `SolarisUser` template to add the users `cleared1 and locked1`.
   b. Use the UID numbers as listed in Table 13-21.
   c. Leave the User Account Is Locked button selected for both users.
7. Double-click the `cleared1` user and complete the following steps:
   a. Select the Password tab.
   b. Select the password option User Must Set Password At Next Login.
   c. Click OK.
8. Double-click the `locked1` user and complete the following steps:
   a. Select the General tab.
   b. Under the Account Availability section, verify that the Account is Locked button is selected.
9. Exit the Solaris Management Console.

Task 4 – Examining Configuration Files

Complete the following steps:

1. Examine the contents of the `/etc/passwd` file.

   What are the full path names of the shells used by `user3`, `user4`, and `user5`?

   `user3`

   `user4`

   `user5`

2. Examine the contents of the `/etc/shadow` file.

   What text is found in the password field for the users `locked1` and `cleared1`?

   `locked1`

   `cleared1`

   You used the same password for `user3` through `user5`.

   Are the password strings the same in the `/etc/shadow` file?

   ____

3. Examine the contents of the `/etc/group` file, and then verify that `user3` and `user4` are both listed as secondary members of the `class1` group.

   Are they?

   ____

4. Log out of the window manager, and then attempt to log in as `locked1`.

   Are you able to log in?

   ____

5. Complete the following steps:

   a. Attempt to log in as `cleared1`.

      What happens?

   ____

b. Attempt to use the password abcdefg.

What are the system requirements for the password? You must not press Return when you are asked for an initial password.

____

____

c. Use the password abc123, and then log in as cleared1 after you establish a password to verify that the login works.

d. Log out, and then log in as the root user. Open a terminal window.

Task 5 – Establishing Password Aging

Complete the following steps:

1. Start the Solaris Management Console and complete the following steps:
   a. Go back into the User Accounts tool.
   b. Respond to authentication prompts as required.
   c. Select user5 from the list of users.
2. Complete the following steps:
   a. Change the password options information for user5 so that it matches the following information.
   b. Click OK when you are finished, and exit the Solaris Management Console.

User Must Keep For:	1 (one day)
Before Change Alert User:	2 (two days)
User Must Change Within:	2 (two days)
Expires If Not Used For:	1 (one day)

3. Log out of your `root` login session, and attempt to log in as `user5`. What happens?

   ____

Note – If the password expiry warning does not display when you log in, you can test the password aging feature by using the `su user5` command. Doing so will cause the message `Your password will expire in 2 days` to display in the terminal window where you run the `su user5` command.

---

4. Complete the login as `user5` and complete the following steps:
   a. Open a terminal window.
   b. Attempt to change the password for `user5`.

   What happens?

   ____

   ____

5. Log out, and then log in again as the `root` user. Open a terminal window.

Task 6 – Modifying User Accounts and Group Entries

Complete the following steps:

1. Use the `groupadd` command to create a new group called `class3` that uses GID number `103`.
2. Complete the following steps:
   a. Use the `usermod` command to change the following information for the `locked1` user:
      - The login name to `user6`
      - The UID to `3001`
      - The home directory to `user6`
   b. Verify that the changes you request are recorded in the `/etc/passwd` file and that the directory was moved.

3. Use the `smuser modify` command to change the login shell of `user5` to `/bin/ksh` and complete the following steps:

   a. Respond to authentication prompts as required.

Note – Bug ID 6478291 for Solaris 10 Update 3 identifies an issue with the `smuser` and `smrole` commands. If `smuser` reports this error: `Authentication failed to the Solaris Management Console server`, and you have specified the correct authentication information, use `usermod` instead to modify the `user5` account.

---

   b. As an alternative, use `usermod` to modify the `user5` account.

   c. Verify that the changes you request are recorded in the `/etc/passwd` file.

4. Complete the following steps:

   a. Use the `userdel` command to delete the user account `cleared1` and the related home directory.

   b. Verify that the `/export/home/cleared1` directory no longer exists.

5. Use the `smgroup` command to change the group name of `class1` to `group1` and complete the following steps:

   a. Respond to authentication prompts as required.

   b. Verify the change in the `/etc/group` file.

6. Complete the following steps:

   a. Use the `groupdel` command to remove the group entry `class2`.

   b. Verify that `class2` is no longer in the `/etc/group` file.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Adding User and Group Accounts

This section contains solutions to the exercises.

Refer to the lecture notes as necessary to perform the tasks listed. Refer to Table 13-20 and Table 13-21 as needed.

Table 13-20 Group Specifications – Solution Section

Group Name	GID Number
class1	101
class2	102

Table 13-21 User Specifications – Solution Section

User Name	Password	Shell	UID	Primary Group	Secondary Group
user3	123pass	Korn	1003	10	class1
user4	123pass	C	1004	10	class1
user5	123pass	Bourne	1005	10	
locked1	Select Account is Locked	Bourne	2001	10	
cleared1	Select User must set password at next login	Bourne	2002	10	

Note – Some of the commands displayed in this section are quite long and will wrap to the next line. You should consider all of the bold typeface commands that follow a command line prompt to be all one line.

Task 1 – Disabling the Solaris OS Registration Window

Complete the following steps:

1. Log in as the `root` user and open a terminal window (or use the `su` command to change your identity to `root`).
2. Change to the `/etc/default` directory.
3. In the `/etc/default` directory, use the `vi` editor to create a file called `solregis`.

   ```
   # vi solregis
   ```

4. In the `solregis` file, create a line that reads:

   ```
   DISABLE=1
   ```

Note – The character "1" is the number one.

---

5. Save this file, and then exit the editor.

Task 2 – Adding Group Entries

Complete the following steps:

Note – Refer to Table 13-20 for details while adding groups with the tools used in this exercise.

---

1. As the `root` user, open a terminal window.
2. Use the `groupadd` and `smgroup` commands to add the groups `class1` and `class2`, respectively.

   The `smgroup` command will request a password for authentication. You should respond to the prompts as required.

```
# groupadd -g 101 class1
# /usr/sadm/bin/smgroup add -- -n class2 -g 102
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: cangetin
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from host2
Login to host2 as user root was successful.
```

```
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from host2
was successful.
#
```

Task 3 – Adding User Accounts

Complete the following steps:

Note – Refer to Table 13-21 for details while adding users with the tools used in this exercise.

1. If the `/export/home` directory does not exist, create it.

```
# ls /export/home
/export/home: No such file or directory
# mkdir /export/home
#
```

2. Use the `useradd` command to add a user named `user3`, and then set a password for `user3`.

```
# useradd -u 1003 -g 10 -G class1 -d /export/home/user3 -m -s /bin/ksh
user3
64 blocks
# passwd user3
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for user3
```

3. Use the `smuser` command to add a user named `user4` and complete the following steps:

   a. Respond to `smuser` authentication prompts, as required.

```
# /usr/sadm/bin/smuser add -- -n user4 -u 1004 -g 10 -G class1 -d
/export/home/user4 -s /bin/csh -x autohome=N
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: cangetin
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from host2
Login to host2 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from host2 was
successful.
#
```

b. As an alternative, use useradd to create the user4 account.

```
# useradd -u 1004 -g 10 -G class1 -d /export/home/user4 -m -s /bin/csh
user4
64 blocks
#
```

c. Set a password for user4.

```
# passwd user4
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for user4
```

4. Launch the Solaris Management Console by typing **smc&** on the command line.

   After the Solaris Management Console displays, complete the following steps to create a user template and add user accounts that do not use automounted home directories:

   a. Select This Computer, then System Configuration, then Users, and then User Templates to open the User Templates tool.

      You should respond to authentication prompts as required.

   b. From the Menu Bar, select Action, and then select Add User Template.

      The Add User Template window appears, containing blank fields for a template name and description.

   c. Enter the name SolarisUser in the User Template Name field and Class Template for the Description field.

   d. Click the Home Directory Tab and complete the following steps:

      1. Uncheck the Automatically Mount Home Directory check box.

      2. Enter the name of your system in the Home Directory Server field.

   e. Click OK to create your template.

5. Complete the following steps:

   a. Click User Accounts, and add the user5 account by selecting Action, then Add User, and then selecting From Template on the menu bar.

      The Add User From Template window appears.

   b. Enter user5 in the User Name field and enter 1005 in the User ID Number field.

   c. Click the User Must Use button, and then enter `123pass` in both password fields for password information.
   d. Click OK, and confirm your choice as requested.
6. From the Solaris Management Console, complete the following steps:
   a. Use the `SolarisUser` template to add the users `cleared1 and locked1`.
   b. Use the UID numbers as listed in Table 13-21.
   c. Leave the User Account Is Locked button selected for both users.
7. Double-click the `cleared1` user and complete the following steps:
   a. Select the Password tab.
   b. Select the password option User Must Set Password At Next Login.
   c. Click OK.
8. Double-click the `locked1` user and complete the following steps:
   a. Select the General tab.
   b. Under the Account Availability section, verify that the Account is Locked button is selected.
9. Exit the Solaris Management Console.

Task 4 – Examining Configuration Files

Complete the following steps:

1. Examine the contents of the `/etc/passwd` file.

   What are the full path names of the shells used by `user3`, `user4`, and `user5`?

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

2. Examine the contents of the `/etc/shadow` file.

   What text is found in the password field for the users `locked1` and `cleared1`?

`locked1`	`*LK*`
`cleared1`	*no value*

   You used the same password for `user3` through `user5`.

   Are the password strings the same in the `/etc/shadow` file?

   No.

3. Examine the contents of the `/etc/group` file, and then verify that `user3` and `user4` are both listed as secondary members of the `class1` group.

   Are they?

   The names `user3` *and* `user4` *should be listed in the last field for the* `class1` *group.*

4. Log out of the window manager, and then attempt to log in as `locked1`.

   Are you able to log in?

   No, you get a message that says `login incorrect`*, no matter what you use as a password.*

5. Complete the following steps:
   a. Attempt to log in as `cleared1`.

      What happens?
   b. Attempt to use the password `abcdefg`.

      What are the system requirements for the password? You must not press Return when you are asked for an initial password.

      You must choose an initial password for this user and then log in again. The first six characters must contain at least two alphabetic characters and at least one numeric or special character.
   c. Use the password `abc123`, and then log in as `cleared1` after you establish a password to verify that the login works.
   d. Log out, and then log in as the `root` user. Open a terminal window.

Task 5 – Establishing Password Aging

Complete the following steps:

1. Start the Solaris Management Console and complete the following steps:
   a. Go back into the User Accounts tool.
   b. Respond to authentication prompts as required.
   c. Select `user5` from the list of users.
2. Complete the following steps:
   a. Change the password options information for `user5` so that it matches the following information.

User Must Keep For:	`1` (one day)
Before Change Alert User:	`2` (two days)
User Must Change Within:	`2` (two days)
Expires If Not Used For:	`1` (one day)

   b. Click OK when you are finished, and exit the Solaris Management Console.

3. Log out of your `root` login session, and then attempt to log in as `user5`.

   What happens?

   You can log in directly as user5, and a warning that indicates that your password expires in two days displays.

Note – If the password expiry warning does not display when you log in, you can test the password aging feature by using the `su user5` command. Doing so will cause the message `Your password will expire in 2 days` to display in the terminal window where you run the `su user5` command.

---

4. Complete the login as `user5` and complete the following steps:

   a. Open a terminal window.

   b. Attempt to change the password for `user5`.

   What happens?

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

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

5. Log out, and then log in again as the `root` user. Open a terminal window.

Task 6 – Modifying User Accounts and Group Entries

Complete the following steps:

1. Use the `groupadd` command to create a new group called `class3` that uses GID number `103`.

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

2. Complete the following steps:

   a. Use the `usermod` command to change the following information for the `locked1` user:

   - The login name to `user6`
   - The UID to `3001`
   - The home directory to `user6`

b. Verify that the changes you request are recorded in the /etc/passwd file and that the directory was moved.

```
# ls -d /export/home/locked1
/export/home/locked1
# usermod -u 3001 -d /export/home/user6 -m -l user6 locked1
48 blocks
# ls -d /export/home/locked1
/export/home/locked1: No such file or directory
# ls -d /export/home/user6
/export/home/user6
# grep user6 /etc/passwd
user6:x:3001:10::/export/home/user6:/bin/sh
#
```

The /etc/passwd file should reflect the new UID number and user name. The directory under /export/home should be renamed from locked1 to user6.

3. Use the smuser modify command to change the login shell of user5 to /bin/ksh and complete the following steps:

a. Respond to authentication prompts as required.

```
# grep user5 /etc/passwd
user5:x:1005:10::/export/home/user5:/bin/sh
# /usr/sadm/bin/smuser modify -- -n user5 -s /bin/ksh
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: cangetin
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from host2
Login to host2 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from host2 was
successful.
```

b. As an alternative, use usermod to modify the user5 account.

```
# usermod  -s /bin/ksh user5
#
```

c. Verify that the changes you request are recorded in the /etc/passwd file.

```
# grep user5 /etc/passwd
user5:x:1005:10::/export/home/user5:/bin/ksh
#
```

The /etc/passwd *file should show that the shell for* user5 *is* /bin/ksh.

4. Complete the following steps:

   a. Use the userdel command to delete the user account cleared1 and the related home directory.

   b. Verify that the /export/home/cleared1 directory no longer exists.

```
# userdel -r cleared1
# ls /export/home/cleared1
/export/home/cleared1: No such file or directory
#
```

The /export/home/cleared1 *directory should no longer exist.*

5. Use the smgroup command to change the group name of class1 to group1 and complete the following steps:

   a. Respond to authentication prompts as required.

   b. Verify the change in the /etc/group file.

```
# grep class1 /etc/group
class1::101:user3,user4
# /usr/sadm/bin/smgroup modify -- -n class1 -N group1
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: cangetin
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from host2
Login to host2 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from host2
was successful.
# grep class1 /etc/group
# grep group1 /etc/group
group1::101:user3,user4
#
```

6. Complete the following steps:

   a. Use the groupdel command to remove the group entry class2.

   b. Verify that class2 is no longer in the /etc/group file.

```
# groupdel class2
# grep class2 /etc/group
#
```

The group class2 *should not exist.*

Module 14

Controlling System Processes

Objectives

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

- View system processes
- Kill frozen processes
- Schedule an automatic one-time execution of a command
- Schedule an automatic recurring execution of a command

Viewing System Processes

A process is any program running on a system. All processes are assigned a unique process identification (PID) number, which is used by the `kernel` to track and manage the process. The PID numbers are used by the `root` and regular users to identify and control their processes.

Using the `prstat` Command

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

This command enables you to view information by specific processes, user identification (UID) numbers, central processing unit (CPU) IDs, or processor sets. By default, the `prstat` command displays information about all processes sorted by CPU usage. To use the `prstat` command, perform the command:

```
# prstat
   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
  1641 root     4864K 4520K cpu0    59    0   0:00:00 0.5% prstat/1
  1635 root     1504K 1168K sleep   59    0   0:00:00 0.3% ksh/1
     9 root     6096K 4072K sleep   59    0   0:00:29 0.1% svc.configd/11
   566 root       82M   30M sleep   29   10   0:00:36 0.1% java/14
  1633 root     2232K 1520K sleep   59    0   0:00:00 0.1% in.rlogind/1
   531 root     8200K 2928K sleep   59    0   0:00:12 0.1% dtgreet/1
   474 root       21M 7168K sleep   59    0   0:00:11 0.1% Xsun/1
   236 root     4768K 2184K sleep   59    0   0:00:03 0.0% inetd/4
    86 root     3504K 1848K sleep   59    0   0:00:01 0.0% nscd/24
     7 root     5544K 1744K sleep   59    0   0:00:06 0.0% svc.startd/12
   154 root     2280K  824K sleep   59    0   0:00:01 0.0% in.routed/1
   509 root     6888K 2592K sleep   59    0   0:00:02 0.0% httpd/1
   240 root     5888K 1256K sleep   59    0   0:00:01 0.0% sendmail/1
   145 root     2944K  816K sleep   59    0   0:00:01 0.0% httpd/1
   347 daemon   2608K  776K sleep   59    0   0:00:00 0.0% nfsmapid/3
   206 root     1288K  600K sleep   59    0   0:00:00 0.0% utmpd/1
   344 daemon   2272K 1248K sleep   60  -20   0:00:00 0.0% nfsd/2
   241 smmsp    5792K  960K sleep   59    0   0:00:00 0.0% sendmail/1
   107 root     2584K  784K sleep   59    0   0:00:00 0.0% syseventd/14
   123 root     3064K  880K sleep   59    0   0:00:00 0.0% picld/4
   146 lp       2976K  448K sleep   59    0   0:00:00 0.0% httpd/1
Total: 53 processes, 171 lwps, load averages: 0.02, 0.04, 0.07
```

```
#
```

To quit the `prstat` command, type **q**.

Table 14-1 shows the column headings and their meanings in a `prstat` report.

Table 14-1 `prstat` Report Column Headings

Default Column Heading	**Description**
`PID`	PID process number
`USERNAME`	Login name or UID of the owner of the process
`SIZE`	Total virtual memory size of the process
`RSS`	Resident process set size in kilobytes, megabytes, or gigabytes
`STATE`	Process sate • `cpu` – process is running on CPU. • `sleep` – process is waiting for event to complete. • `run` – process is in run queue • `zombie` – process terminated, and the parent is not waiting • `stop` – process stopped
`PRI`	Process priority
`NICE`	Value used in priority computation
`TIME`	Cumulative process execution time
`CPU`	Percentage of recent CPU time used by the process.
`PROCESS/NLWP`	Name of process/number of lightweight processes (LWPs) in the process.

Note – The `kernel` and many applications are now multithreaded. A thread is a logical sequence of program instructions written to accomplish a particular task. Each application thread is independently scheduled to run on an LWP, which functions as a virtual CPU. LWPs in turn, are attached to `kernel` threads, which are scheduled to run on actual CPUs.

Note – Use the `priocntl(1)` command to assign processes to a priority class and to manage process priorities. The `nice(1)` command is only supported for backward compatibility to previous Solaris OS releases. The `priocntl` command provides more flexibility in managing processes.

Table 14-2 shows the options for the `prstat` command.

Table 14-2 Options for the `prstat` Command

Option	Description
`-a`	Displays separate reports about processes and users at the same time.
`-c`	Continuously prints new reports below previous reports.
`-n` *`nproc`*	Restricts the number of output lines.
`-p` *`pidlist`*	Reports only on processes that have a PID in the given list.
`-s` *`key`*	Sorts output lines by *`key`* in descending order. The five possible keys include: `cpu`, `time`, `size`, `rss`, and `pri`. You can use only one key at a time.
`-S` *`key`*	Sorts output lines by *`key`* in ascending order.
`-t`	Reports total usage summary for each user.
`-u` *`euidlist`*	Reports only processes that have an effective user ID (EUID) in the given list.
`-U` *`uidlist`*	Reports only processes that have a real UID in the given list.

Killing Frozen Processes

Use the kill command or the pkill command to send a signal to one or more running processes. You would typically use these commands to kill an unwanted process.

Using the kill and pkill Commands

You use the kill or pkill commands to terminate one or more processes.

The format for the kill command is:

```
kill -signal PID
```

To show all of the available signals used with the kill command:

```
kill -l
```

The format for the pkill command is:

```
pkill -signal Process
```

Before you can terminate a process, you must know its name or PID. Use either the ps or pgrep command to locate the PID for the process.

The following examples uses the pgrep command to locate the PID for the mail processes.

```
# pgrep -l mail
  241 sendmail
  240 sendmail
#
# pkill sendmail
```

The following examples use the ps and pkill to locate and end sendmail.

```
# ps -e | grep sendmail
   241 ?         0:00 sendmail
   240 ?         0:02 sendmail
# kill 241
```

To terminate more than one process at the same time, use the following syntax:

```
# kill -signal PID PID PID PID
# pkill signal process process
```

Use the `kill` command without a signal on the command line to send the default Signal `15` to the process. This signal usually causes the process to terminate.

Table 14-3 shows some signals and names.

Table 14-3 Process Signal Numbers and Names

Signal Number	Signal Name	Event	Default Action
1	SIGHUP	Hangup	Exit
2	SIGINT	Interrupt	Exit
9	SIGKILL	Kill	Exit
15	SIGTERM	Terminate	Exit

- `1`, `SIGHUP` – A hangup signal to cause a telephone line or terminal connection to be dropped. For certain daemons, such as `inetd` and `in.named`, a hangup signal will cause the daemon to reread its configuration file.
- `2`, `SIGINT` – An interrupt signal from your keyboard—usually from a Control-C key combination.
- `9`, `SIGKILL` – A signal to kill a process. A process cannot ignore this signal.
- `15`, `SIGTERM` – A signal to terminate a process in an orderly manner. Some processes ignore this signal.

Use `kill -l`, or see the `signal` man page, for a complete list of signals that the `kill` command can send.

```
# man -s3head signal
```

Some processes can be written to ignore Signal `15`. Processes that do not respond to a Signal `15` can be terminated by force by using Signal `9` with the `kill` or `pkill` commands. You use the following syntax:

```
# kill -9 PID
# pkill -9 process
```

Caution – Use the `kill -9` or `pkill -9` command as a last resort to terminate a process. Using the `-9` signal on a process that controls a database application or a program that updates files can be disastrous. The process is terminated instantly with no opportunity to perform an orderly shutdown.

Performing a Remote Login

When a workstation is not responding to your keyboard or mouse input, the CDE might be frozen. In such cases, you may be able to remotely access your workstation by using the `rlogin` command or by using the `telnet` command from another system.

Killing the Process for a Frozen Login

After you are connected remotely to your system, invoke the `pkill` command to terminate the corrupted session.

In the following examples, the `rlogin` command is used to log in to `sys42`, from which you can issue a `pkill` or a `kill` command.

```
# rlogin sys-02
Password:
Last login: Sun Oct 24 13:44:51 from sys-01
Sun Microsystems Inc. SunOS 5.10 s10_68  Sep. 20, 2004
# pkill -9 Xsun
or
# ps -e | grep Xsun
   442 ?          0:01 Xsun
# kill -9 442
```

Scheduling an Automatic One-Time Execution of a Command

Use the `at` command to automatically execute a job only once at a specified time.

Using the `at` Command

The format for the `at` command is:

```
at -m -q queuename time date
at -r job
at -l
```

Table 14-4 shows the options you can use to instruct the `cron` process on how to execute an `at` job.

Table 14-4 Options for the `at` Command

Option	Description
`-m`	Sends mail to the user after the job has finished
`-r` *job*	Removes a scheduled `at` job from the queue
`-q` *queuename*	Specifies a specific queue
time	Specifies a time for the command to execute
`-l`	Reports all jobs scheduled for the invoking user
date	Specifies an optional date for the command to execute, which is either a month name followed by a day number or a day of the week

For example, to create an `at` job to run at 9:00 p.m. to locate and verify the file type of `core` files from the `/export/home` directory, perform the command:

```
# at 9:00 pm
at> find /export/home -name core -exec file {} \; >> /var/tmp/corelog
at> <EOT>
commands will be executed using /sbin/sh
job 1098648000.a at Mon Oct 25 21:00:00 2004
```

Note – The `at` command job numbers show the job's execution time as reflected in the amount of time in seconds since 12:00 A.M. January 1, 1970, which is referred to as the UNIX® epoch date.

To display information about the execution times of jobs, perform the command:

```
# at -l 1098648000.a
1098648000.a    Mon Oct 25 21:00:00 2004
```

To display the jobs queued to run at specified times by chronological order of execution, perform the command:

```
# atq
 Rank     Execution Date     Owner     Job          Queue   Job Name
  1st   Oct 25, 2004 21:00   root       1098648000.a     a     stdin
```

To view all the `at` jobs currently scheduled in the queue, perform the command:

```
# ls -l /var/spool/cron/atjobs
total 4
-r-Sr--r--   1 root     root        1044 Oct 25 13:48 1098648000.a
```

You can also use the `at` command to remove a job from the `at` queue.

For example, to remove job `1098648000.a` from the `at` queue, perform the command:

```
# at -r 1098648000.a
# atq
 Rank     Execution Date     Owner     Job          Queue   Job Name
```

Controlling Access to the `at` Command

As the `root` user, you control who has access to the `at` command with the `at.deny` and `at.allow` files.

The `/etc/cron.d/at.deny` File

By default, the Solaris OS includes the `/etc/cron.d/at.deny` file. This file identifies users who are prohibited from using the `at` command. The file format is one user name per line. The file initially contains:

```
daemon
bin
nuucp
listen
nobody
noaccess
```

A user who is denied access to the `at` command receives the following message when attempting to use the command:

```
at: you are not authorized to use at. Sorry.
```

If only the `/etc/cron.d/at.deny` file exists but is empty, then all logged-in users can access the `at` command.

The `/etc/cron.d/at.allow` File

The `/etc/cron.d/at.allow` file does not exist by default, so all users (except those listed in the `/etc/cron.d/at.deny` file) can create `at` jobs. By creating the `/etc/cron.d/at.allow` file, you create a list of only those users who are allowed to execute `at` commands. The `/etc/cron.d/at.allow` file consists of user names, one per line.

The interaction between the `at.allow` and the `at.deny` files follows these rules:

- If the `at.allow` file exists, only the users listed in this file can execute `at` commands. This rule also applies to the `root` user.
- If the `at.allow` file does not exist, all users, except for users listed in the `at.deny` file, can execute `at` commands.
- If neither file exists, only the `root` user can use the `at` command.
- If a user is listed in both files, the user is allowed to use `at`.

Scheduling an Automatic Recurring Execution of a Command

You can use the `cron` facility to schedule regularly recurring commands. Users can submit a command to the `cron` facility by modifying their `crontab` file.

All `crontab` files are maintained in the `/var/spool/cron/crontabs` directory and are stored as the login name of the user that created the `cron` job.

The `cron` daemon is responsible for scheduling and running these jobs.

Note – The clock daemon, `cron`, starts at system boot and runs continuously in the background.

Introducing the `crontab` File Format

A `crontab` file consists of lines of six fields each. The fields are separated by spaces or tabs. The first five fields provide the date and time the command is to be scheduled. The last field is the full path to the command.

Note – If the command field contains a percent (%) character, then all subsequent characters are passed to the command as standard input.

These first five fields are separated by spaces and indicate when the command will be executed. See Figure 14-1.

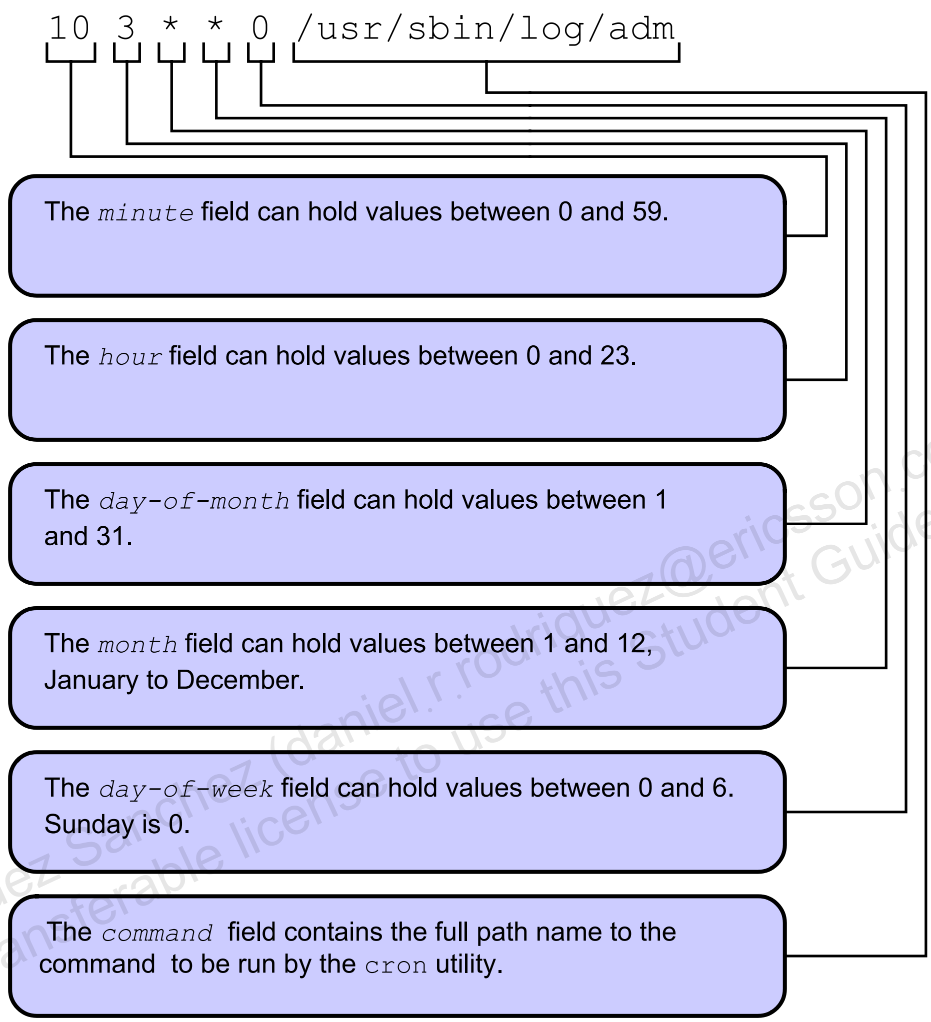


Figure 14-1 First Five Fields in a `crontab` File

The first five fields follow the format rules shown in Table 14-5.

Table 14-5 Rules for the `crontab` Fields

Value	Rule	Example
n	Matches if field value is *n*	As shown in the preceding figure for hour or minute, a 3 or 10
n,p,q	Matches if field value is *n*, *p*, or *q*	Every 10 minutes would be represented by 0,10,20,30,40,50
n–p	Matches if field has values between *n* and *p* inclusive	The hours between 1:00 a.m. and 4:00 a.m. would be represented by 1-4
*	Matches all legal values	As in the preceding example for the month, representing every month.

Using the `crontab` Command

The `crontab` command enables the user to view, edit, or remove a `crontab` file.

Viewing a `crontab` File

To view the contents of the `root` `crontab` file, run the `crontab -l` command as the `root` user.

```
# crontab -l
#ident  "@(#)root       1.21    04/03/23 SMI"
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] &&
/usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

This is the same command that users run to view the contents of their own `crontab` file.

As the `root` user, you can view the contents of any regular user's `crontab` file by performing the command:

```
# crontab -l username
```

Editing a `crontab` File

Caution – If you accidentally enter the `crontab` command on the command line without an option (`-l`, `-e`, `-r`), press the interrupt keys Control-C to exit. Do not press Control-D, this action overwrites the existing `crontab` file with an empty file.

To create or edit a `crontab` file, follow these steps:

1. Check that the `EDITOR` variable is set to the editor you want to use. This instructs the `cron` utility which editor to use to open the file.

```
# EDITOR=vi
# export EDITOR
```

2. Run the following `crontab` command to open your `crontab` file, and add the appropriate entry.

```
# crontab -e
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console
:wq
```

Note – If the users do not redirect the standard output and standard error of their commands in the `crontab` file, any generated output or errors are mailed electronically to the user.

Removing a `crontab` File

The correct way to remove a `crontab` file is to invoke the command:

```
# crontab -r username
```

Typical users can remove only their own `crontab` file. The `root` user can delete any user's `crontab` file.

Controlling Access to the crontab Command

You can control access to the crontab command with two files in the /etc/cron.d directory—the cron.deny file and the cron.allow file.

These files permit only specified users to perform crontab tasks, such as creating, editing, displaying, or removing their own crontab files.

The /etc/cron.d/cron.deny File

The Solaris OS provides a default cron.deny file. The file consists of a list of user names, one per line, of the users who are not allowed to use cron. The following is an example of the contents of a cron.deny file:

```
daemon
bin
nuucp
listen
nobody
noaccess
```

The /etc/cron.d/cron.allow File

The /etc/cron.d/cron.allow file does not exist by default, so all users (except those listed in the cron.deny file) can access their crontab file. By creating a cron.allow file, you can list only those users who can access crontab commands.

The file consists of a list of user names, one per line.

The interaction between the cron.allow and the cron.deny files follows these rules:

- If the cron.allow file exists, only the users listed in this file can create, edit, display, or remove crontab files.
- If the cron.allow file does not exist, all users, except for users listed in the cron.deny file, can create, edit, display, or remove crontab files.

- If neither file exists, only the `root` user can run the `crontab` command.

Note – The Solaris OS provides commands that check authorizations. The batch job-related commands, such as at, atq, batch, and crontab are such commands

---

Exercise: Using Process Control

In this exercise, you complete the following tasks:

- Use the Process Manager Tool and the `prstat` command to monitor and kill processes.
- Create an `at` job and create an entry in a `crontab` file.

Preparation

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

Task

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Press Ctrl+Alt+Del to start the Process Manager or invoke the appropriate command from the command line.
2. In the Process Tool display, sort the listing according to `CPU%`, and change the sample time to five seconds.
3. Open a second terminal window, and run the `prstat` command.
4. Position the Process Tool and the window in which the `prstat` command is running so that you can observe both simultaneously. In an available window, run the `find` command to list all files on your system. Observe how the Process Tool and the `prstat` command display statistics for the `find` command.

   What is the maximum percentage of recent CPU time used by the `find` command as it executes?

   ____

5. Open a third terminal window, and run the `ps` command to determine the PID of the shell. Record the PID you find.

   ____

6. In the Process Tool, locate and select the shell process you identified in the previous step. Select the Show Ancestry option from the Process menu in the Process Tool. What is the name and PID of the first process listed?

   ____

7. Close the Show Ancestry window. Again, select the shell process you identified in Step 4. From the Process menu in the Process Tool, select the Kill option. What happens?

   ____

8. In the Process Tool, use the Find function to locate the `prstat` process. Select the Signal option from the Process menu. In the Signal fill-in field, enter the `TERM` signal, and click OK. What happens to the `prstat` process? Close the Process Tool when you are finished.

   ____

9. Identify the device associated with your current terminal by using the `tty` command, and display the current time of day.

   ____

10. Submit an `at` job that echoes `Test Complete` to your current window. Have the job run about five minutes from the current time, and submit it to the queue called `x`.
11. Display the `at` job in the queue.
12. Open a new window and set and export the `EDITOR` environment variable to use the `vi` editor to edit `crontab` files.
13. Use the `crontab` command to view the current `crontab` file for the `root` user.
14. When is the `logadm` process scheduled to run?

    ____

15. Use the `crontab` command to edit the `crontab` file for the `root` user. Add an entry that sends the message `It works!` to your current window five minutes from now. For example, if the current time is 10:25, make an entry in your `crontab` file for the 30th minute of the same hour.

    Save the file, and quit the `vi` edit session. In about five minutes, you should see the result in your window.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Using Process Control

This section contains solutions to the exercise.

Task

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Press Ctrl+Alt+Del to start the Process Manager or invoke the appropriate command from the command line.

   ```
   # /usr/dt/bin/sdtprocess &
   ```

   In the Process Tool display, sort the listing according to `CPU%`, and change the sample time to five seconds.

2. Open a second terminal window, and run the `prstat` command.

   ```
   # prstat
   ```

3. Position the Process Tool and the window in which the `prstat` command is running so that you can observe both simultaneously. In an available window, run the `find` command to list all files on your system. Observe how the Process Tool and the `prstat` command display statistics for the `find` command.

   ```
   # find /
   ```

   What is the maximum percentage of recent CPU time used by the `find` command as it executes?

   This varies according to your system configuration. Some systems might display values in the 20-percent range.

4. Open a third terminal window, and run the `ps` command to determine the PID of the shell. Record the PID you find.

   ```
   # ps
   ```

   Your value appears here.

5. In the Process Tool, locate and select the shell process you identified in the previous step. Select the Show Ancestry option from the Process menu in the Process Tool. What is the name and PID of the first process listed?

   On systems running JDS, the first process should be `gnome-terminal`*.*

6. Close the Show Ancestry window. Again, select the shell process you identified in Step 4. From the Process menu in the Process Tool, select the Kill option. What happens?

The process stops, and the window no longer appears.

7. In the Process Tool, use the Find function to locate the `prstat` process. Select the Signal option from the Process menu. In the Signal fill-in field, enter the `TERM` signal, and click OK. What happens to the `prstat` process? Close the Process Tool when you are finished.

   The `prstat` *process terminates, and the prompt appears in the window in which it ran.*

8. Identify the device associated with your current terminal by using the `tty` command, and display the current time of day.

```
# tty
/dev/pts/2
#
# date
Tue Mar 27 18:07:52 MDT 2007
#
```

9. Submit an `at` job that echoes `Test Complete` to your current window. Have the job run about five minutes from the current time, and submit it to the queue called `x`.

```
# at -q x 18:13
at> echo "Test Complete" > /dev/pts/# (# is from the tty command)
at> <Control-D>
commands will be executed using /bin/ksh
job 1175040780.x at Tue Mar 27 18:13:00 2007
#
```

10. Display the `at` job in the queue.

```
# atq
# atq
 Rank     Execution Date     Owner      Job            Queue   Job Name
  1st   Mar 27, 2007 18:13   root       1175040780.x     x     stdin
#
```

11. Open a new window and set and export the `EDITOR` environment variable to use the `vi` editor to edit `crontab` files.

    If you are using the Bourne or Korn shell, perform the command:

```
# EDITOR=vi
# export EDITOR
```

    If you are using the C shell, perform the command:

```
# setenv EDITOR vi
```

12. Use the `crontab` command to view the current `crontab` file for the `root` user.

```
# crontab -l
```

```
                  #ident  "@(#)root       1.21    04/03/23 SMI"
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
(output omitted)
```

13. When is the `logadm` process scheduled to run?

    Ten minutes after 3:00 a.m. on all days

14. Use the `crontab` command to edit the `crontab` file for the `root` user. Add an entry that sends the message `It works!` to your current window five minutes from now. For example, if the current time is 10:25, make an entry in your `crontab` file for the 30th minute of the same hour.

```
# tty
/dev/pts/2
# date
Tue Mar 27 18:16:02 MDT 2007
# crontab -e
```

Add the following line, but substitute the correct time and terminal device:

```
30  10 * * * /usr/bin/echo "It works!" > /dev/pts/2
```

Save the file, and quit the `vi` edit session. In about five minutes, you should see the result in your window.

Module 15

Solaris™ 10 Operating System Installation Requirements

Objectives

Upon completing this module, you should be able to:

- Identify hardware requirements for Solaris 10 OS installation
- Identify the fundamentals of the Solaris 10 OS installations
- Identify the Solaris 10 OS software components
- Define guidelines to install the Solaris 10 OS from a DVD
- Describe the Secure by Default installation enhancement

Identifying the Hardware Requirements for Solaris 10 OS Installation

A Solaris 10 OS installation requires the following:

For SPARC platform systems:

- 512 Mbytes of memory is recommended, 256 Mbytes minimum
- At least 5 Gbytes of disk space
- Access to a bootable DVD drive or an installation server

For x86/x64 platform systems:

- 120–MHz or faster processor is recommended. Hardware floating-point support is required.
- 512 Mbytes of memory is recommended, 256 Mbytes minimum.
- At least 5 Gbytes of disk space.
- Access to a bootable DVD drive or an installation server.
- If you are installing on a system that is not produced by Sun Microsystems, Inc., a system listed in the Hardware Compatibility List (HCL) found at: `http://www.sun.com/bigadmin/hcl`

Two-Terabyte Disk Support

Starting with the Solaris 10 10/09 release, you can install and boot the Solaris OS from a disk that is up to 2 Tbytes in size. In previous Solaris releases, you could not install and boot the Solaris OS from a disk that was greater than 1 Tbyte in size.

When installing and booting the Solaris OS on a 2 Tbyte disk it must be connected to a system that runs a 64-bit kernel.

When booting an x86 system with a target disk greater than 1 Tbyte, a minimum of 1 Gbyte of memory is required.

Identifying the Fundamentals of Solaris 10 OS Installations

This section introduces the various methods used to install or upgrade the Solaris OS.

Solaris 10 OS Installation Options

There are two methods available to install the Solaris 10 OS on your system; `install-solaris` and Flash Archive installation. Procedures that use `install-solaris` install groups of software packages onto your system. Flash Archive installations use flash archives to install reference configurations of the Solaris OS.

The following installation procedures use `install-solaris`:

- Solaris installation Graphical User Interface (GUI)
- Solaris installation Command Line Interpreter (CLI)
- Solaris Custom JumpStart™ software (JumpStart) installation

The following installation procedures use Flash archives:

- Solaris Flash Archive installation
- Solaris WAN boot installation

This module provides an overview of the Solaris OS installation concepts.

GUI Installation

The default installation method uses a GUI if the host has sufficient memory and a display capable of displaying GUI screens. Otherwise, installation proceeds using the CLI. To run the installation GUI program requires a minimum of 768 Mbytes of memory.

These memory requirements relate to installation processes, and may represent values that are insufficient to support specific workloads. Memory requirements for systems should be determined according to anticipated production workloads.

On SPARC systems, initiate the GUI-based installation from DVD by issuing the following command:

```
ok boot cdrom
```

On x86/x64 systems, the GUI-based installation begins automatically when you insert the Solaris OS software DVD, power-on the system, and accept the default choices from the first two menus that display.

CLI Installation

Hosts that do not have a graphical screen or that have insufficient memory cannot run the GUI installation, and consequently use the CLI. You can specify that you want to use the CLI when you start the installation process. Text based installation requires a minimum of 384 Mbytes of memory. On SPARC systems, using the `nowin` option when you boot from DVD initiates the CLI. For example:

```
ok boot cdrom - nowin
```

Windowing x86/x64 systems present sets of menus when you boot from DVD. Choosing the `Solaris Interactive Text (Console session)` option causes the installation software to use the CLI on the system console. Choosing the `Solaris Interactive Text (Desktop session)` option causes the installation software to use the CLI in a terminal window within the Java Desktop System (JDS).

When using the CLI, installation options display in a menu-driven format. Use the space bar to select options, and the F2 key (or the equivalent escape key sequence), to accept selected options.

Custom JumpStart Installation

The Solaris JumpStart procedure installs Solaris OS software on a system through use of a user-defined profile found on a JumpStart server. You can customize profiles for different system types.

A JumpStart installation provides a hands-off installation across the network, and is based on a centrally-configured server. or set of servers. Custom JumpStart configurations allow you to incorporate shell scripts into the installation process, and customize software beyond simply installing the Solaris OS. Custom JumpStart shell scripts may perform both pre-installation and post-installation tasks. On a SPARC system, use the following command to initiate a custom JumpStart installation:

```
ok boot net - install
```

On a x86/x64 systems, you initiate a custom JumpStart installation by holding the F12 key during the initial power-on sequence. This causes the system to initiate the Intel Preboot Installation Environment (PXE) installation process, that requires a server to respond to DHCP and other requests made by the x86/x64 system. No boot media is required on the client system.

Flash Archive Installation

The Solaris Flash Archive Installation enables you to install many systems based on a reference configuration that you create on a master system. After you have installed and configured the master system, you then create a flash archive from that system. You associate the flash archive with a JumpStart profile to allow JumpStart clients to use the flash archive as their installation source. You can create many flash archives to capture many master system configurations.

Solaris Flash Archive installations can offer speed advantages over custom JumpStart installations. Custom JumpStart installations install each Solaris OS package individually. Solaris Flash Archive installations use a single archive source, and avoid individual package installations.

You initiate Solaris Flash Archive Installations in the same way that you initiate custom JumpStart installations. The server configuration determines which type of installation occurs.

Solaris WAN Boot

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

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

Solaris 10 OS Upgrade Options

The two Solaris 10 OS Upgrade options are as follows:

- Standard Upgrade
- Solaris Live Upgrade

Standard Upgrade

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

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

Solaris Live Upgrade

Solaris Live Upgrade is the default method for upgrading your system in the recent releases of Solaris 10 OS. The Solaris Live Upgrade software upgrades a duplicate boot environment while the active boot environment is still running. This method eliminates downtime of the production environment. First, create a duplicate boot environment. After that has been created, upgrade or install a Solaris Web Start Flash archive on the inactive boot environment. When you are ready, activate the inactive boot environment. During the next reboot, the inactive boot environment becomes the active boot environment. If there is a failure, you can recover your original boot environment by reactivating it and rebooting the system.

Troubleshooting Solaris 10 OS Upgrades

Check the log files to troubleshoot any problems encountered during an upgrade of Solaris 10 OS. The output files and upgrade process results are stored in log files, for example:

- `/var/sadm/system/logs/upgrade_cleanup`
- `/var/sadm/system/logs/upgrade_log`

Following an upgrade, log files help you identify problems and determine if any cleanup is needed.

Identifying the Solaris 10 OS Software Components

Solaris OS software is organized into three components:

- Software packages
- Software clusters
- Software groups

Software Packages

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

Software Clusters

During the software installation process, software clusters group logical collections of software packages together. Table 15-1 shows the software packages that are grouped into the CDE software cluster.

Table 15-1 Packages Included in the CDE Software Cluster

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

Some software clusters contain only one software package.

Solaris OS Software Groups

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

- Reduced Network Support Software Group
- Core System Support Software Group
- End User Solaris Software Group
- Developer Solaris Software Group
- Entire Solaris Software Group
- Entire Solaris Software Group Plus Original Equipment Manufacturer (OEM) Support

Reduced Network Support Software Group

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

A system installed with the Reduced Network Support software group could, for example, be used as a *thin-client* host in a network. At least 2 Gbytes of space is recommended to install this software group.

Core System Support Software Group

The Core System Support software group (`SUNWCreq`) contains the minimum software required to boot and run the Solaris OS in a minimum configuration, without the support to run many server applications. The Core System Support software group includes a minimum of networking software, including Telnet, file transfer protocol (FTP), network file system (NFS), network information service (NIS) clients, and domain name service (DNS). This software group also includes the drivers required to run the Common Desktop Environment (CDE) but does not include the CDE software. The Core System Support software group also does not include online manual pages. At least 2 Gbytes of space is recommended to install this software group.

End User Solaris Software Group

The End User Solaris software group (`SUNWCuser`) contains the Core System Support software group, the recommended software for an end user, and the CDE. At least 5.3 Gbytes of space is recommended to install this software group.

Developer Solaris Software Group

The Developer Solaris software group (`SUNWCprog`) contains the End User Solaris software group. It also contains the libraries, the include files, the online manual pages, and the programming tools for developing software. At least 6.6 Gbytes of space is recommended to install this software group.

Entire Solaris Software Group

The Entire Solaris software group (`SUNWCall`) contains the Developer Solaris software group. It also contains additional server software. The Entire Solaris software group is the entire Solaris OS software release minus OEM support. At least 6.7 Gbytes of space is recommended to install this software group.

Entire Solaris Software Group Plus OEM Support

The Entire Solaris Software Group Plus OEM Support group (`SUNWCXall`) contains the entire Solaris OS software release. It also contains additional hardware support for OEMs and hardware not on the system at the time of installation. This software group is recommended when you are installing the Solaris OS software on non-Sun servers that use UltraSPARC processors. At least 6.8 Gbytes of space is recommended to install this software group.

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

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

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

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

Defining Guidelines for Installing Solaris 10 OS From a DVD

Following are some of the guidelines to be followed while planning to install the Solaris 10 OS from a DVD:

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

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

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

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

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

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

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

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

Make sure you have the required Solaris installation media. You will need the Solaris Operating System for SPARC Platforms DVD, or the Solaris Operating System for x86 Platforms DVD.

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

Secure by Default Installation Enhancement

Traditionally, Solaris systems have provided many network services by default. This open approach is convenient, but it also makes it easier to exploit vulnerabilities that may exist in the software that provides network services.

The Secure by Default (SBD) enhancement reduces this exposure by disabling as many network services as possible while still providing a useful system. The only network-listening service left enabled in the hardened configuration is SSH.

Installation and Secure by Default

During an initial DVD-based installation of the Solaris 10 OS, as of the 11/06 release, a GUI screen or text-based panel displays that allows you to select the Secure by Default option.

Following is an example of the text-based prompt that displays during a new Solaris 10 installation. Selecting "Yes" enables an open configuration profile. Selecting "No" enables a limited profile.

```
Would you like to enable network services for use by remote clients?

        [x] Yes
        [ ] No

Note: Selecting "No" provides a more secure configuration in which
Secure Shell is the only network service provided to remote clients.
Selecting "Yes" enables a larger set of services as in previous Solaris
releases. If in doubt, it is safe to select "No" as any services can
be individually enabled after installation.
```

Custom JumpStart and Flash Archive servers use the new `service_profile` keyword in their `sysidcfg` files to establish either a limited or open network service profile on installation clients.

Simulation: Solaris 10 Operating System Installation

This module uses a simulation to present an example of the installation process.

This simulation describes installing the Solaris 10 OS on both SPARC and x86/x64-based systems, using DVD installation media.

Your instructor will provide instructions for using the installation simulation.

Notes:

Appendix A

Backing Up a Mounted File System With a UFS Snapshot

Objectives

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

- Create a UFS snapshot
- Back up the snapshot file

Note – For information on ZFS snapshots, see the "ZFS File System Introduction" module.

Creating a UFS Snapshot

The UFS Copy on Write Snapshots feature provides administrators an online backup solution for `ufs` file systems. This utility enables you to use a point-in-time copy of a `ufs` file system, called a snapshot, to create an online backup. You can create the backup while the file system is mounted and the system is in multiuser mode.

Note – The UFS snapshots are similar to the Sun StorEdge™ Instant Image product. Instant Image allocates space equal to the size of the entire file system that is being captured. However, the file system data saved by UFS snapshots occupies only as much disk space as needed.

Using the `fssnap` Command

You use the `fssnap` command to create, query, or delete temporary read-only snapshots of `ufs` file systems.

The format for the `fssnap` command is:

```
/usr/sbin/fssnap -F FSType -V -o special_option(s) mount-point | special
```

Table 15-2 shows some of the options for the `fssnap` command.

Table 15-2 `fssnap` Command Options

Option	Description
`-d`	Deletes the snapshot associated with the given file system. If the `-o unlink` option was used when you built the snapshot, the backing-store file is deleted together with the snapshot. Otherwise, the backing-store file (which contains file system data) occupies disk space until you delete it manually.
`-F` *`FSType`*	Specifies the file system type to be used.
`-i`	Displays the state of an *`FSType`* snapshot.
`-V`	Echoes the complete command line but does not execute the command.
`-o`	Enables you to use *`special_options`*, such as the location and size of the backing-store (`bs`) file.

To create a UFS snapshot, specify a backing-store path and the actual file system to be captured. The following is the command format:

```
# fssnap -F ufs -o bs=backing_store_path /file-system
```

Note – The `backing_store_path` can be a raw device, the name of an existing directory, or the name of a file that does not already exist.

The following example uses the `fssnap` command to create a snapshot of the `/export/home` file system.

```
# fssnap -F ufs -o bs=/var/tmp /export/home
/dev/fssnap/0
```

The snapshot subsystem saves file system data in a file called a backing-store file before the data is overwritten. Some important aspects of a backing-store file are:

- A backing-store file is a bit-mapped file that takes up disk space until you delete the UFS snapshot.
- The size of the backing-store file varies with the amount of activity on the file system being captured.
- The destination path that you specify on the `fssnap` command line must have enough free space to hold the backing-store file.
- The location of the backing-store file must be different from that of the file system you want to capture in a UFS snapshot.
- A backing-store file can reside on different types of file systems, including another `ufs` file system or a mounted `nfs` file system.

The `fssnap` command creates the backing-store file and two read-only virtual devices. The block virtual device, `/dev/fssnap/0`, can be mounted as a read-only file system. The raw virtual device, `/dev/rfssnap/0`, can be used for raw read-only access to a file system.

These virtual devices can be backed up with any of the existing Solaris OS backup commands. The backup created from a virtual device is a backup of the original file system when the UFS snapshot was taken.

Note – When a UFS snapshot is first created, the file system locks temporarily. Users might notice a slight pause when writing to this file system. The length of the pause increases with the size of the file system. There is no performance impact when users are reading from the file system.

Limiting the Size of the Backing-Store File

Before creating a UFS snapshot, use the `df -k` command to check that the backing-store file has enough disk space to grow. The size of the backing-store file depends on how much data has changed since the previous snapshot was taken.

You can limit the size of the backing-store file by using the `-o maxsize=`*n* option of the `fssnap` command, where *n* (`k`, `m`, or `g`) is the maximum size of the backing-store file specified in Kbytes, Mbytes, or Gbytes.

Caution – If the backing-store file runs out of disk space, the system automatically deletes the UFS snapshot, which causes the backup to fail. The active `ufs` file system is not affected. Check the `/var/adm/messages` file for possible UFS snapshot errors.

Note – You can force an unmount of an active `ufs` file system, for which a snapshot exists (for example, with the `umount -f` command). This action deletes the appropriate snapshot automatically.

The following example creates a snapshot of the `/export/home` file system, and limits the backing-store file to 500 Mbytes.

```
# fssnap -F ufs -o bs=/var/tmp,maxsize=500m /export/home
/dev/fssnap/0
```

Displaying Information for a `ufs` File System Snapshot

Use either `fssnap` command to display UFS snapshot information. The following example displays a list of all the current UFS snapshots on the system. The list also displays the corresponding virtual device for each snapshot.

```
# fssnap -i
0    /export/home
1    /usr
2    /database
```

Use the `-i` option to the `/usr/lib/fs/ufs/fssnap` command to display detailed information for a specific UFS snapshot that was created by the `fssnap` command.

The following example shows the details for the `/export/home` snapshot.

```
# /usr/lib/fs/ufs/fssnap -i /export/home
Snapshot number                : 0
Block Device                   : /dev/fssnap/0
Raw Device                     : /dev/rfssnap/0
Mount point                    : /export/home
Device state                   : idle
Backing store path             : /var/tmp/snapshot0
Backing store size             : 0 KB
Maximum backing store size     : 512000 KB
Snapshot create time           : Mon Oct 11 08:58:33 2004
Copy-on-write granularity      : 32 KB
```

Backing Up the UFS Snapshot File

The virtual devices that contain the UFS snapshot act as standard read-only devices, which enable you to back up the virtual device in the same manner as you would back up a file system.

Performing a Backup of a UFS Snapshot

Use the `tar` command or the `ufsdump` command to back up a UFS snapshot.

Using the `tar` Command to Back Up a Snapshot File

If you use the `tar` command to back up the UFS snapshot, mount the snapshot before backing it up. The following procedure demonstrates how to do this type of mount.

1. Create the mount point for the block virtual device.

   ```
   # mkdir -p /backups/home.bkup
   ```

2. Mount the block virtual device to the mount point.

   ```
   # mount -F ufs -o ro /dev/fssnap/0 /backups/home.bkup
   ```

3. Change directory to the mount point.

   ```
   # cd /backups/home.bkup
   ```

4. Use the `tar` command to write the data to tape.

   ```
   # tar cvf /dev/rmt/0 .
   ```

Using the `ufsdump` Command

If you use the `ufsdump` command to back up a UFS snapshot, you can specify the raw virtual device during the backup.

```
# ufsdump 0uf /dev/rmt/0 /dev/rfssnap/0
```

Verify that the UFS snapshot is backed up.

```
# ufsrestore tf /dev/rmt/0
```

Performing an Incremental Backup Using a UFS Snapshot

Incremental backups of snapshots contain files that were modified since the last UFS snapshot. Use the `ufsdump` command with the `N` option to create an incremental UFS snapshot, which writes the name of the device being backed up, rather than the name of the snapshot device to the `/etc/dumpdates` file.

The following example shows how to use `ufsdump` to create an incremental file system backup.

Note – Use the `N` argument when backing up a snapshot. This argument ensures proper updates to the `/etc/dumpdates` file.

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdsk/c1t0d0s0 /dev/rfssnap/0
```

Next you would verify that the `UFS` snapshot is backed up to tape.

```
# ufsrestore tf /dev/rmt/0
```

To understand incremental backups of snapshots, consider the following:

1. Create a snapshot of the `/extra` file system that is going to be backed up while the file system is mounted.

```
# fssnap -o bs=/var/tmp /extra
/dev/fssnap/0
#
```

2. Verify that the snapshot was successful, and view detailed information about the snapshot.

```
# fssnap -i
   0    /extra
# /usr/lib/fs/ufs/fssnap -i /extra
Snapshot number               : 0
Block Device                  : /dev/fssnap/0
Raw Device                    : /dev/rfssnap/0
Mount point                   : /extra
Device state                  : idle
Backing store path            : /var/tmp/snapshot0
Backing store size            : 0 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Mon Oct 11 10:34:21 2004
```

```
Copy-on-write granularity     : 32 KB
```

3. Make a directory that will be used to mount and view the snapshot data.

```
# mkdir /extrasnap
#
```

4. Mount the snapshot to the new mount point, and compare the size of the file system and the snapshot device.

```
# mount -o ro /dev/fssnap/0 /extrasnap
# df -k |grep extra
/dev/dsk/c1t0d0s0    1294023       9 1242254     1%    /extra
/dev/fssnap/0        1294023       9 1242254     1%    /extrasnap
```

5. Edit a file under the /extra directory and make it larger, and then compare the size of the file system and the snapshot device.

```
# vi file1
(yank and put text, or read text in from another file)
# df -k |grep extra
/dev/dsk/c1t0d0s0    1294023      20 1242243     1%    /extra
/dev/fssnap/0        1294023       9 1242254     1%    /extrasnap
```

Observe that the file system grew in size while the snapshot file did not.

6. Perform a full backup with the N option of the ufsdump command.

```
# ufsdump 0ufN /dev/rmt/0 /dev/rdsk/c1t0d0s0 /dev/rfssnap/0
  DUMP: Writing 32 Kilobyte records
  DUMP: Date of this level 0 dump: Mon Oct 11 10:49:38 2004
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rfssnap/0 (sys41:/extrasnap) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 262 blocks (131KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 254 blocks (127KB) on 1 volume at 1814 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Mon Oct 11 11:03:46 2004
```

7. Verify the backup.

```
# ufsrestore tf /dev/rmt/0
         2      .
         3      ./file1
         4      ./file2
         5      ./file3
         6      ./file4
```

```
#
```

8. Unmount the back up device and remove the snapshot.

```
# umount /extrasnap
# fssnap -d /extra
# rm /var/tmp/snapshot0
#
```

9. Make some changes to the `/extra` file system, such as copying some files, and then re-create the snapshot.

```
# cp file1 file5
# cp file1 file6
# fssnap -o bs=/var/tmp /extra
/dev/fssnap/0
#
```

10. Re-mount the snapshot device, and compare the size of the file system and the snapshot device.

```
# mount -o ro /dev/fssnap/0 /extrasnap
# df -k |grep extra
/dev/dsk/c1t0d0s0    1294023      46 1242217     1%    /extra
/dev/fssnap/0        1294023      46 1242217     1%    /extrasnap
#
```

11. Perform an incremental backup with the `N` option of the `ufsdump` command.

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdsk/c1t0d0s0 /dev/rfssnap/0
  DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Mon Oct 11 13:13:03 2004
  DUMP: Date of last level 0 dump: Mon Oct 11 12:30:44 2004
  DUMP: Dumping /dev/rfssnap/0 (sys41:/extrasnap) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 294 blocks (147KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 254 blocks (127KB) on 1 volume at 1693 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 1 dump on Mon Oct 11 13:22:36 2004
#
```

12. Verify the backup.

```
# ufsrestore tf /dev/rmt/0
         2      .
         7      ./file5
         8      ./file6
```

```
#
```

The backup of the snapshot contains only the files that were added since the previous Level 0 backup.

Restoring Data From a UFS Snapshot Backup

The backup created from a virtual device is a backup of the original file system when the UFS snapshot was taken.

Restore a UFS snapshot from a backup tape in the same manner as you would the backup of an original file system. Data written to a tape by `ufsdump` is simply data, whether it is a snapshot or a file system.

To restore the `demo` directory from the snapshot backup of the `/usr` file system, complete the following steps:

1. Load the tape that contains the snapshot backup of the `/usr` file system into the tape drive.
2. Change to the `/usr` file system.
   ```
   # cd /usr
   ```
3. Perform the a `ufsrestore` command.
   ```
   # ufsrestore if /dev/rmt/0
   ufsrestore > add demo
   ufsrestore > extract
   Specify next volume #: 1
   set owner/mode for '.'? [yn] n
   ufsrestore > quit
   ```
4. Verify that the `demo` directory exists, and eject the tape.

Deleting a UFS Snapshot

Deleting a UFS snapshot from the system is a multistep process and order-dependant. First, unmount the snapshot device, and then delete the snapshot. Finally, remove the backing-store file.

```
# umount /dev/fssnap/0
# fssnap -d /export/home
# rm /backing_store_file
```

Exercise: Working With UFS Snapshots

In this exercise, you complete the following tasks:

- Create a UFS snapshot of the `/opt` file system
- View the contents of the backing-store directory
- Display detailed information for the UFS snapshot
- Remove the snapshot and backing-store file

Preparation

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

Task

Complete the following steps:

1. Create a snapshot of the `/opt` file system without specifying a file name for the backing-store file.
2. List the contents of the `/var/tmp` directory.

   What is the default name assigned to the backing-store file?

   ____

3. Display the detailed information about the snapshot.

   What is the maximum backing-store file size for the snapshot?

   ____

4. Delete the snapshot from the system.
5. List the contents of the `/var/tmp` directory. Has the backing-store file been removed?

   ____

6. Remove the backing-store file you created in Step 1.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Working With UFS Snapshots

This section contains solutions to the exercise.

Task

Complete the following steps:

1. Create a snapshot of the /opt file system without specifying a file name for the backing-store file.

```
# fssnap -F ufs -o bs=/var/tmp /export/home
/dev/fssnap/0
#
```

2. List the contents of the /var/tmp directory.

```
# ls -lh /var/tmp
total 862
(output ommitted)
drwx------   2 root     root        1.0K Oct  6 15:53 orbit-root/
drwx------   2 1000     1000        1.0K Sep 25 14:03 orbit-student/
drwx------   2 user5    staff       1.0K Oct  6 14:46 orbit-user5/
-rw-------   1 root     root        9.8G Oct  6 16:30 snapshot0
-rw-------   1 root     root        2.2K Oct  5 18:44 wscon-:0-M4ayyb
(output ommitted)
```

What is the default name assigned to the backing-store file?

snapshot0

3. Display the detailed information about the snapshot.

```
# /usr/lib/fs/ufs/fssnap -i /export/home
Snapshot number              : 0
Block Device                 : /dev/fssnap/0
Raw Device                   : /dev/rfssnap/0
Mount point                  : /export/home
Device state                 : idle
Backing store path           : /var/tmp/snapshot0
Backing store size           : 0 KB
Maximum backing store size   : Unlimited
Snapshot create time         : Tue Oct 06 16:30:04 2009
Copy-on-write granularity    : 32 KB
```

What is the maximum backing-store file size for the snapshot?

Unlimited

4. Delete the snapshot from the system.

```
# fssnap -d /export/home
Deleted snapshot 0.
#
```

5. List the contents of the /var/tmp directory. Has the backing-store file been removed?

```
# ls -l /var/tmp/snapshot0
-rw-------   1 root     root     10574196736 Oct  6 16:30
/var/tmp/snapshot0
#
```

No

6. Remove the backing-store file you created in Step 1.

```
# rm /var/tmp/snapshot0
```

Appendix B

Performing File System Backups

Objectives

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

- Identify backup fundamentals
- Back up an unmounted file system

Introducing Backup Fundamentals

Backing up file systems safeguards against data loss, damage, or corruption. Backup tapes are often referred to as dump tapes.

Routine File System Backups

To back up file systems, copy them to removable media, such as a tape. Perform backups regularly to prevent data loss from:

- Accidental file deletion
- Hardware failures
- Problems with re-installations or system upgrades
- System crashes
- System break-ins
- Natural disasters

Tape Drive Naming

All tape drives have logical device names that you use to reference the device on the command line. Figure 15-1 shows the format that all logical device names use.

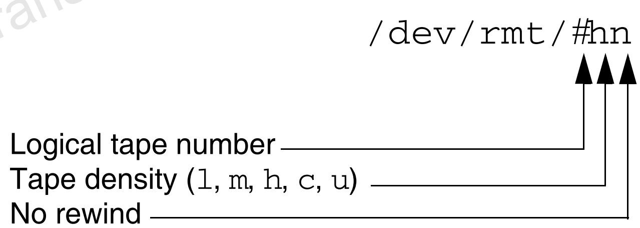


Figure 15-1 Logical Device Name Format

The logical tape numbers in the tape drive names always start with 0. For example:

- The first instance of a tape drive:

  /dev/rmt/0

- The second instance of a tape drive:

  /dev/rmt/1

- The third instance of a tape drive:

  /dev/rmt/2

Two optional parameters further define the logical device name:

- Tape density – Five values can be given in the tape device name: l (low), m (medium), h (high), c (compressed), or u (ultra compressed).

Note – The tape density optional parameters h, c, and u reflect the same density. Refer to the man page st(7D) for more information.

- No rewind – The letter n at the end of a tape device name indicates that the tape should not be rewound when the current operation completes.

Tape densities depend on the tape drive. Check the manufacturer documentation to determine the correct densities for the tape media.

Note – The st.conf file contains configuration entries for supported devices of Sun Microsystems and many third-party devices.

Tape drives that support data compression contain internal hardware that performs the compression. If you back up a software-compressed file to a tape drive with hardware compression, the resulting file may be larger in size.

Tape Drive Control

Use the magnetic tape control mt command to send instructions to the tape drive. Not all tape drives support all mt commands.

The format for the mt command is:

mt -f *tape-device-name command count*

Use the -f option to specify the tape device name, typically a no-rewind device name. If no -f option is used, the default tape device file /dev/rmt/0 is used.

Using the mt Command

Table 15-3 lists some of the mt commands that you can use to control a magnetic tape drive.

Table 15-3 Definitions of mt Commands

Command	Definition
mt status	Displays status information about the tape drive
mt rewind	Rewinds the tape
mt offline	Rewinds the tape and, if appropriate, takes the drive unit offline and if the hardware supports it, unloads
mt fsf *count*	Moves the tape forward *count* records

Assuming the tape was rewound to the start of tape, the following command positions the tape at the beginning of the third tape record.

```
# mt -f /dev/rmt/0n fsf 2
```

Scheduled Backup Strategies

Perform cumulative incremental backups daily.

To set up a backup schedule, determine:

- The file systems to back up
- A backup device (for example, tape drive)
- The number of tapes to use for the backup
- The type of backup (for example, full or incremental)
- The procedures for marking and storing tapes
- The time it takes to perform a backup

Determining File System Names to Back Up

1. Display the contents of the /etc/vfstab file.
2. View the mount point column to find the name of the file system that you want to back up.

Determining the Number of Tapes

1. Determine the number of tapes for a backup according to the size of the file system you are backing up.
2. Use `ufsdump`, with the `S` option, to determine the file system size as follows. The numeric option determines the appropriate dump level. The output is the estimated number of bytes that the system requires for a complete backup.

```
# ufsdump 0S filesystem_name
<number reported>
or
# ufsdump 3S filesystem_name
<number reported>
```

Note – To translate the output of the `ufsdump #S` *filesystem_name* to Mbytes, use the following calculation:
`# / (1024 x 1024).`

---

3. Divide the reported bytes by the capacity of the tape to determine how many tapes you need to backup the file system.

Determining Back Up Frequency and Levels

Determine how often and at what level to backup each file system. The level of a backup refers to the amount of information that is backed up.

Identifying Incremental and Full Backups

You can perform a full backup or an incremental backup of a file system. A full backup is a complete file system backup. An incremental backup copies only files in the file system that have been added or modified since a previous lower-level backup.

Use dump level `0` to perform a full backup. You use dump levels `1` through `9` to schedule incremental backups. The level numbers have no meaning other than their relationship to each other as a higher or lower number.

Figure 15-2 shows *an example* of a file system backup performed in incremental levels.

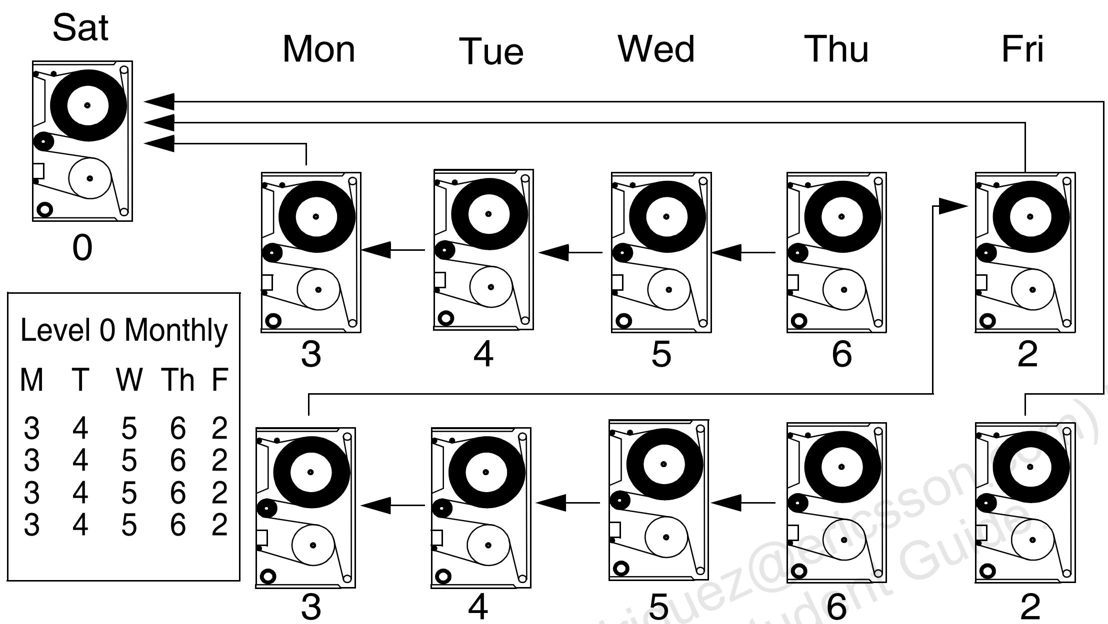


Figure 15-2 Incremental Backup Strategy Sample

Table 15-4 defines the elements of the sample incremental backup strategy shown in Figure 15-2 on page B-6. Use `crontab` to start a script that runs

Table 15-4 Incremental Back Up Level Definitions

Level	Example
0 (Full)	Performed once each month.
3	Performed every Monday. The backup copies new or modified files since the last lower-level backup (for example, `0`).
4	Performed every Tuesday. The backup copies new or modified files since the last lower-level backup (for example, `3`).
5	Performed every Wednesday. The backup copies new or modified files since the last lower-level backup (for example, `4`).
6	Performed every Thursday. The backup copies new or modified files since the last lower-level backup (for example, `5`).
2	Performed every Friday. The backup copies new or modified files since the last lower-level backup, which is the Level `0` backup at the beginning of the month.

The /etc/dumpdates File

Use `ufsdump -u`, to backup the `/etc/dumpdates` file records. Each line in the `/etc/dumpdates` file shows the file system that was backed up and the level of the last backup. It also shows the day, the date, and the time of the backup. The following is an example `/etc/dumpdates` file:

```
# cat /etc/dumpdates
/dev/rdsk/c0t2d0s6  0 Fri Nov 5  19:12:27  2004
/dev/rdsk/c0t2d0s0  0 Fri Nov 5  20:44:02  2004
/dev/rdsk/c0t0d0s7  0 Tue Nov 9  09:58:26  2004
/dev/rdsk/c0t0d0s7  1 Tue Nov 9  16:25:28  2004
```

When you perform an incremental backup, the `ufsdump` command consults the `/etc/dumpdates` file. It looks for the date of the next lower-level backup. Then, the `ufsdump` command copies to the backup media all of the files that were modified or added since the date of that lower-level backup.

When the backup is complete, the `/etc/dumpdates` file records a new entry that describes this backup. The new entry replaces the entry for the previous backup at that level.

View the `/etc/dumpdates` file to determine if the system is completing backups. If a backup does not complete because of equipment failure, the `/etc/dumpdates` file does not record the backup.

Note – When you restore an entire file system, check the `/etc/dumpdates` file for a list of the most recent dates and levels of backups. Use this list to determine which tapes are needed to restore the entire file system. The tapes should be physically marked with the dump level and date of the backup.

Backing Up an Unmounted File System

Determine if the file system is inactive, or unmounted, before you back up the file system. If the file system is active, the output of the backup can be inconsistent, and it may be impossible to restore some of the files correctly.

The `ufsdump` Command

Use `/usr/sbin/ufsdump` to back up a complete or a partial `ufs` file system. (Backups are often referred to as dumps.) The `ufsdump` command format is:

`ufsdump` *option(s) argument(s) filesystem_name*

`ufsdump` Command Options

Table 15-5 defines common options for the `ufsdump` command.

Table 15-5 `ufsdump` Command Options

Option	Description
`0-9`	Back up level. Level `0` is a full backup of the file system. Levels `1` through `9` are incremental backups of files that have changed since the last lower-level backup. When no backup level is given, the default is level 9.
`v`	Verify. After each tape is written, the system verifies the contents of the media against the source file system. If any discrepancies occur, the system prompts the operator to insert new media and repeat the process. Use this option only on an unmounted file system. Any activity in the file system causes the system to report discrepancies.
`S`	Size estimate. This option allows you to estimate the amount of space that will be needed on the tape to perform the level of backup you want.
`l`	Autoload. You use this option with an autoloading (stackloader) tape drive.
`o`	Offline. When the backup is complete, the system takes the drive offline, rewinds the tape (if you use a tape), and, if possible, ejects the media.

Table 15-5 `ufsdump` Command Options

Option	Description
`u`	Update. The system creates an entry in the `/etc/dumpdates` file with the device name for the file system disk slice, the backup level (`0`–`9`), and the date. If an entry already exists for a backup at the same level, the system replaces the entry.
`n`	Notify. The system sends messages to the terminals of all logged-in users who are members of the `sys` group to indicate that the `ufsdump` command requires attention.
`f` *device*	Specify. The system specifies the device name of the file system backup. When you use the default tape device, `/dev/rmt/0`, you do not need the `-f` option. The system assumes the default.

The `ufsrestore` Utility

The `ufsrestore` utility restores files from backup media created with the `ufsdump` command. `ufsrestore` actions are controlled by the key argument. The key is exactly one function letter (`i`, `r`, `R`, `t`, or `x`) and zero or more function modifiers (letters). The key string contains no SPACE characters. Function modifier arguments are listed on the command line in the same order as their corresponding function modifiers appear in the key string. For more information about the `ufsrestore` utility, see the `ufsrestore` man page.

Tape Backups

Use `ufsdump` to create file system backups to tape. Enter a dump level (`0`–`9`) to determine which files to back up.

Note – To backup file systems that you cannot unmount, such as the root (`/`) file system, it is recommended to shut down the system to single-user mode. For other file systems, unmounting the file system is sufficient.

Using the `ufsdump` Command

Perform the following steps to use the `ufsdump` command to back up an example file system (`/opt`) to tape:

1. Become the `root` user, and use the `shutdown` command to change the system to single-user mode.

```
# /usr/sbin/shutdown -y -g 300 "System is being shutdown for backup"

Shutdown started.    Wed Mar 28 09:17:53 MDT 2007

Changing to init state s - please wait
Broadcast Message from root (console) on sys-05 Wed Mar 28 09:17:53...
THE SYSTEM sys-05 IS BEING SHUT DOWN NOW ! ! !
Log off now or risk your files being damaged
System is being shutdown for backup
(output omitted)
```

2. Check if the `/opt` file system is still mounted. If it is, unmount it manually.

```
# mount | grep /opt
/opt on /dev/dsk/c0t0d0s5
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev
=220000d on Mon Mar  5 10:17:29 2007
# umount /opt
#
```

3. Check the integrity of the file system data with the `fsck` command.

```
# fsck /opt
** /dev/rdsk/c0t0d0s5
** Last Mounted on /opt
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
320 files, 35595 used, 444924 free (436 frags, 55561 blocks, 0.1%
fragmentation)
#
```

4. Perform a full (Level `0`) backup of the `/opt` file system.

```
# ufsdump 0uf /dev/rmt/0 /opt
  DUMP: Date of this level 0 dump: Wed Mar 28 09:42:05 2007
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s5 (sys-05:/opt) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
```

```
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 32 Kilobyte records
  DUMP: Estimated 71686 blocks (35.00MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 71678 blocks (35.00MB) on 1 volume at 811 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Wed Mar 28 09:42:05 2007
#
```

5. Return the system to multi-user mode.

```
# exit
svc.startd: Returning to milestone all.
checking ufs filesystems
/dev/rdsk/c0t0d0s5: is logging.
(output omitted)
```

Remote Backups to a Tape

Use the `ufsdump` command to perform a backup to a remote tape device. The format for the `ufsdump` command is:

`ufsdump` *options remotehost:tapedevice filesystem*

To perform remote backups across the network, the system with the tape drive must have an entry in its `/.rhosts` file for every system that uses the tape drive.

Using the `ufsdump` Command

The following example shows how to perform a full (Level `0`) backup of the `/opt` file system on the `sys-06` system, to the remote tape device on the `sys-05` system.

```
sys-06# ufsdump 0uf sys-05:/dev/rmt/0 /opt
  DUMP: Date of this level 0 dump: Wed Mar 28 09:51:22 2007
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c1d0s5 (sys-06:/opt) to sys-05:/dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 32 Kilobyte records
  DUMP: Estimated 68668 blocks (33.53MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
```

```
  DUMP: Tape rewinding
  DUMP: 68606 blocks (33.50MB) on 1 volume at 1455 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Wed Mar 28 09:51:22 2007
sys-06#
```

Backups to Disk

Use ufsdump to create file system backups to files on disk. As with backups to tape, the dump level (0–9) specified in the ufsdump command determines which files to back up.

Use backups to disk if you need to create a backup quickly, for temporary use, or if you do not have a tape device available. A backup made to a file on disk is only as reliable as the disk it is on.

Use the ufsdump -f option to specify the name of the file that will receive the backup.

Using the ufsdump Command

Perform the following steps to use the ufsdump command to back up an example file system (/opt) to a file on disk:

1. Become the root user, and use the shutdown command to change the system to single-user mode.

```
# /usr/sbin/shutdown -y -g 300 "System is being shutdown for backup"

Shutdown started.    Wed Mar 28 09:17:53 MDT 2007

Changing to init state s - please wait
Broadcast Message from root (console) on sys-05 Wed Mar 28 09:17:53...
THE SYSTEM sys-05 IS BEING SHUT DOWN NOW ! ! !
Log off now or risk your files being damaged
System is being shutdown for backup
(output omitted)
```

2. Check if the /opt file system is still mounted. If it is, unmount it manually.

```
# mount | grep /opt
/opt on /dev/dsk/c0t0d0s5
read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev
=220000d on Mon Mar  5 10:17:29 2007
#  umount /opt
```

```
#
```

3. Check the integrity of the file system data with the `fsck` command.

```
# fsck /opt
** /dev/rdsk/c0t0d0s5
** Last Mounted on /opt
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
320 files, 35595 used, 444924 free (436 frags, 55561 blocks, 0.1%
fragmentation)
#
```

4. Check the space used on the file system you intend to back up, and the file system that will hold the backup file. Verify that enough space is available in the destination file system to hold the backup.

```
# df -h /opt /export/home
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0t0d0s5      470M    36M   387M     9%    /opt
/dev/dsk/c0t0d0s7       94G    64M    93G     1%    /export/home
#
```

5. Perform a full (Level `0`) backup of the `/opt` file system to a file in the `/export/home` file system.

```
# ufsdump 0uf /export/home/dump_opt_level_0 /opt
  DUMP: Date of this level 0 dump: Wed Mar 28 10:00:59 2007
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s5 (sys-05:/opt) to
/export/home/dump_opt_level_0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 32 Kilobyte records
  DUMP: Estimated 71686 blocks (35.00MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 71678 blocks (35.00MB) on 1 volume at 9977 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Wed Mar 28 10:00:59 2007
#
```

6. Return the system to multi-user mode.

```
# exit
svc.startd: Returning to milestone all.
checking ufs filesystems
```

```
/dev/rdsk/c0t0d0s5: is logging.
(output omitted)
```

Saving ZFS Data With Other Backup Products

In addition to the `zfs send` and `zfs receive` commands, you can use archive utilities, such as the `tar` and `cpio` commands, to save ZFS files. These utilities save and restore ZFS file attributes and ACLs. Check the appropriate options for both the `tar` and `cpio` commands.

For recent information about issues with ZFS and third-party backup products, see the Solaris 10 release notes or the ZFS FAQ, here:

`http://opensolaris.org/os/community/zfs/faq/#backupsoftware`

Exercise: Backing Up a File System to Tape

In this exercise, you complete the following tasks:

- Backup an available file system on your system to a tape drive

Preparation

This exercise requires a system that is configured with a tape drive and file system that is available to unmount. This exercise assumes that the /export/home file system exists on a separate partition from the / (root) file system and can be unmounted. Identify the slice on which the /export/home file system resides. Get a tape that is appropriate for your system from the instructor.

For classrooms without a tape drive, skip to the exercise: Backing Up a File System to an Extra Disk Drive.

Task

Complete the following steps:.

1. Recursively copy the contents of the /opt/ses directory to /export/home.
2. Unmount the /export/home file system. If your system reports that the /export/home file system is busy, use the umount -f command.
3. Insert a tape into your tape drive.
4. Use the mt command to rewind the tape to the beginning.
5. Use the ufsdump command to create a backup tape for the /export/home file system.
6. Mount the /export/home file system.
7. Copy the contents of the /etc/uucp directory to the /export/home directory.
8. Unmount the /export/home file system.
9. Move the tape to the next tape record.
10. Use the ufsdump command to create an incremental backup for the /export/home file system, using the non-rewinding tape device.
11. Rewind and eject the tape from the tape drive.

12. Review the contents of the `/etc/dumpdates` file.
13. Mount the `/export/home` file system.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Backing Up a File System to Tape

This section contains solutions to the exercise.

Task

Complete the following steps:.

1. Recursively copy the contents of the `/opt/ses` directory to `/export/home`.

   \# **cp -r /opt/ses /export/home**

2. Unmount the `/export/home` file system. If your system reports that the `/export/home` file system is busy, use the `umount -f` command.

   \# **umount /export/home**

3. Insert a tape into your tape drive.
4. Use the `mt` command to rewind the tape to the beginning.

   \# **mt rewind**

5. Use the `ufsdump` command to create a backup tape for the `/export/home` file system.

\# **ufsdump 0uf /dev/rmt/0 /export/home**

You should see output similar to this:

```
  DUMP: Date of this level 0 dump: Wed Mar 28 10:29:31 2007
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (sys-05:/export/home) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 32 Kilobyte records
  DUMP: Estimated 12130 blocks (5.92MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 12094 blocks (5.91MB) on 1 volume at 723 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Wed Mar 28 10:29:31 2007
#
```

6. Mount the `/export/home` file system.

   \# **mount /export/home**

7. Copy the contents of the `/etc/uucp` directory to the `/export/home` directory.

cp -r /etc/uucp /export/home

8. Unmount the `/export/home` file system.

   # **umount /export/home**

9. Move the tape to the next tape record.

   # **mt -f /dev/rmt/0n fsf 1**

10. Use the `ufsdump` command to create an incremental backup for the `/export/home` file system, using the non-rewinding tape device.

    # **ufsdump 1uf /dev/rmt/0n /export/home**

    You should see output similar to this:

```
  DUMP: Date of this level 1 dump: Wed Mar 28 10:31:39 2007
  DUMP: Date of last level 0 dump: Wed Mar 28 10:29:31 2007
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (sys-05:/export/home) to /dev/rmt/0n.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 32 Kilobyte records
  DUMP: Estimated 6072 blocks (2.96MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 6078 blocks (2.97MB) on 1 volume at 2284 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 1 dump on Wed Mar 28 10:31:39 2007
#
```

11. Rewind and eject the tape from the tape drive.

    # **mt -f /dev/rmt/0 offline**

12. Review the contents of the `/etc/dumpdates` file.

    # **more /etc/dumpdates**

    You should see one line showing information for the level 0 *dump and another line for the level* 1 *dump, for example:*

```
/dev/rdsk/c0t0d0s7               0 Wed Mar 28 10:29:31 2007
/dev/rdsk/c0t0d0s7               1 Wed Mar 28 10:31:39 2007
```

13. Mount the `/export/home` file system.

    # **mount /export/home**

Exercise: Backing Up a File System to an Extra Disk Drive

In this exercise, you complete the following tasks:

- Backup an available file system on your system to an extra disk drive

Preparation

This exercise requires a system that is configured with an extra disk drive and file system that is available to unmount. This exercise assumes that the `/export/home` file system exists on a separate partition from the `/` (root) file system and can be unmounted. Identify the slice on which the `/export/home` file system resides.

In this exercise, you use the `format` utility to partition your extra hard drive so that it has one partition that uses all of the disk space. You then create a file system on that partition, and mount the file system.

Task

Complete the following steps:

1. Use the `format` utility to partition the extra hard drive and create one large partition to use as a backup device. Use the `All Free Hog` method to set partition 0 to use all disk space.
2. Create a file system on the large partition on the extra disk.
3. Run a file system check on the new file system.
4. Create a mount point directory and mount the new file system.
5. Recursively copy the contents of the `/opt/ses` directory to `/export/home`.
6. Unmount the `/export/home` file system. If your system reports that the `/export/home` file system is busy, use the `umount -f` command.
7. Use the `ufsdump` command to backup the `/export/home` file system to a file below the newly created mount point. Review the man page for `ufsdump` for information on how to use a dump file instead of a tape device.
8. Mount the `/export/home` file system.
9. Copy the contents of the `/etc/uucp` directory to the `/export/home` directory.

10. Unmount the /export/home file system.
11. Use the ufsdump command to create an incremental backup for the /export/home file system. Save this data in a separate file on the spare disk.
12. Review the contents of the /etc/dumpdates file.
13. Mount the /export/home file system.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Backing Up File System to Extra Disk Drive

This section contains solutions to the exercise.

Task

Complete the following steps:

1. Use the `format` utility to partition the extra hard drive and create one large partition to use as a backup device. Use the `All Free Hog` method to set partition 0 to use all disk space.

   This is an example from a SPARC system:

```
sys-05# format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@0,0
       1. c0t1d0 <DEFAULT cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@1,0
Specify disk (enter its number): 1
selecting c0t1d0
```

   When you are through partitioning the extra disk drive, your partition table should look something like this:

```
Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0 - 57458      111.79GB    (57459/0/0) 234432720
  1       swap    wu       0                0         (0/0/0)             0
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0
```

   This is an example from an x86/x64 system:

```
sys-05# format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 39887 alt 2 hd 64 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
```

```
       1. c2d0 <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
Specify disk (enter its number): 1
selecting c2d0
```

When you are through partitioning the extra disk drive, your partition table should look something like this:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 - 10007       76.64GB    (10005/0/0) 160730325
  1       swap    wu       0                0         (0/0/0)             0
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130
```

2. Create a file system on the large partition on the extra disk. For example, on a SPARC system:

   # **newfs /dev/rdsk/*c0t1d0s0***

   On an x86/x64 system:

   # **newfs /dev/rdsk/*c2d0s0***

3. Run a file system check on the new file system.For example, on a SPARC system:

   # **fsck /dev/rdsk/*c0t1d0s0***

   On an x86/x64 system:

   # **fsck /dev/rdsk/*c2d0s0***

4. Create a mount point directory and mount the new file system. For example:

   # **mkdir /extra**
   # **mount /dev/dsk/*c0t1d0s0* /extra**

5. Recursively copy the contents of the `/opt/ses` directory to `/export/home`.

   # **cp -r /opt/ses /export/home**

6. Unmount the `/export/home` file system. If your system reports that the `/export/home` file system is busy, use the `umount -f` command.

   # **umount /export/home**

7. Use the `ufsdump` command to backup the `/export/home` file system to a file below the newly created mount point. Review the man page for `ufsdump` for information on how to use a dump file instead of a tape device.

```
# ufsdump 0uf /extra/dump_export_home_0 /export/home
  DUMP: Date of this level 0 dump: Wed Mar 28 11:07:04 2007
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (sys-05:/export/home) to
/extra/dump_export_home_0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 32 Kilobyte records
  DUMP: Estimated 12130 blocks (5.92MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 12094 blocks (5.91MB) on 1 volume at 8084 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Wed Mar 28 11:07:04 2007
#
```

8. Mount the `/export/home` file system.

   `#` **mount /export/home**

9. Copy the contents of the `/etc/uucp` directory to the `/export/home` directory.

   `#` **cp -r /etc/uucp /export/home**

10. Unmount the `/export/home` file system.

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

11. Use the `ufsdump` command to create an incremental backup for the `/export/home` file system. Save this data in a separate file on the spare disk.

```
# ufsdump 1uf /extra/dump_export_home_1 /export/home
  DUMP: Date of this level 1 dump: Wed Mar 28 11:08:35 2007
  DUMP: Date of last level 0 dump: Wed Mar 28 11:07:04 2007
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (sys-05:/export/home) to
/extra/dump_export_home_1.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 32 Kilobyte records
  DUMP: Estimated 6072 blocks (2.96MB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 6078 blocks (2.97MB) on 1 volume at 5525 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 1 dump on Wed Mar 28 11:08:35 2007
#
```

12. Review the contents of the `/etc/dumpdates` file.

    `#` **more /etc/dumpdates**

You should see one line showing information for the Level `0` *dump and another line for the Level 1 dump, for example:*

```
/dev/rdsk/c0t0d0s7                  0 Wed Mar 28 11:07:04 2007
/dev/rdsk/c0t0d0s7                  1 Wed Mar 28 11:08:35 2007
```

13. Mount the `/export/home` file system.

    # **mount /export/home**

Notes:

Appendix C

Performing File System Restores

Objectives

Upon completion of this module, you should be able to restore `ufs` file systems. For ZFS file system boot information, see the ZFS File System Introduction module.

Note – Solaris releases that support the SPARC Newboot feature might require SPARC systems with at least 512 Mbytes of memory.

Restoring a `ufs` File System

Restore a file system to rebuild a damaged file system, to reinstall or upgrade the Solaris OS software, or to reorganize file systems on existing or new disks.

Restoring a Regular File System

When you restore system data, consider the following questions:

- Can the system boot on its own (regular file system restore)?
- Do you need to boot the system from CD-ROM, DVD, or network (critical file system restore)?
- Do you need to boot the system from CD-ROM, DVD, or network and repair the boot drive (special case recovery)?

To restore files or file systems, determine the following:

- The file system backup tapes that are needed
- The device name to which you will restore the file system
- The name of the temporary directory to which you will restore individual files
- The type of backup device to be used (local or remote)
- The backup device name (local or remote)

To restore a regular file system, such as the `/export/home` or `/opt` file system, back up to the disk, you use the `ufsrestore` command. The `ufsrestore` command copies files to the disk, relative to the current working directory, from backup tapes that were created by the `ufsdump` command.

You can use the `ufsrestore` command to reload an entire file system hierarchy from a Level `0` backup and related incremental backups. You can also restore one or more single files from any backup tape.

The format for the `ufsrestore` command is:

```
ufsrestore option(s) argument(s) filesystem
```

Table 15-6 describes some options that you can use with the `ufsrestore` command.

Table 15-6 Options for the `ufsrestore` Command

Option	Description
`t`	Lists the table of contents of the backup media.
`r`	Restores the entire file system from the backup media.
`x` *`file1 file2`*	Restores only the files named on the command line.
`i`	Invokes an interactive restore.
`v`	Specifies verbose mode. This mode displays the path names to the terminal screen as each file is restored.
`f` *`device`*	Specifies the tape device name. When not specified, the `/dev/rmt/0` device file is used.

When you restore an entire file system from a backup tape, the system creates a `restoresymtable` file. The `ufsrestore` command uses the `restoresymtable` file for check-pointing or passing information between incremental restores. You can remove the `restoresymtable` file when the restore is complete.

Using the `ufsrestore` Command to Restore a Regular File System

The following procedure demonstrates how to use the `ufsrestore` command to restore the `/opt` file system on the `c0t0d0s5` slice.

1. Create the new file system structure.

```
# newfs /dev/rdsk/c0t0d0s5
```

2. Mount the file system to the `/opt` directory, and change to that directory.

```
# mount /dev/dsk/c0t0d0s5 /opt
# cd /opt
```

3. Restore the entire `/opt` file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```

Note – Always restore a file system by starting with the Level 0 backup tape, continuing with the next-lower-level tape, and continuing through the highest-level tape.

---

4. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

5. Unmount the new file system.

```
# cd /
# umount /opt
```

6. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdsk/c0t0d0s5
```

7. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s5
```

Note – The system administrator should always back up the newly created file system because the `ufsrestore` command repositions the files and changes the inode allocation.

---

8. Reboot the system.

```
# init 6
```

Restoring the `/usr` File System

To restore the `/usr` file system, boot from the Solaris 10 OS DVD, and then use the `ufsrestore` command to restore files back to the `/usr` partition.

Note – If the `/` (root), `/usr`, or `/var` file systems are unusable because of some type of corruption or damage, the system will not boot.

Using the `ufsrestore` Command to Restore a Critical File System

The following procedure demonstrates how to restore the `/usr` file system on Slice 6 of the boot disk.

To boot a SPARC-based system from DVD, perform the following:

1. Insert the Solaris 10 OS DVD.
2. From the OBP level, boot from DVD with the single-user mode option.

   ```
   ok boot cdrom -s
   ```

To boot an x86/x64-based system from DVD, perform the following:

1. Insert the Solaris 10 OS DVD.
2. Use the reset button, or required key sequence (e.g., Control-Alt-Delete) to reset the system. You can power-cycle the system as an alternative. Your system needs to be configured to boot from DVD, which might require changes to BIOS settings before booting.

   The system begins its boot process from the Solaris miniroot on DVD. The following list of boot choices displays:

   ```
   1. Solaris Interactive (default)
   2. Custom JumpStart
   3. Solaris Interactive Text (Desktop session)
   4. Solaris Interactive Text (Console session)
   5. Apply driver updates
   6. Single user shell

   Enter the number of your choice.
   ```
3. Select boot choice 6, the single user shell. If an instance of Solaris is found on disk, the system asks if you want to mount it on /a. For example:

```
Solaris 10 5/09 s10x_u7wos_08 X86 was found on /dev/dsk/c1d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?]
```

4. Enter n to avoid mounting the root file system. You will manually mount the file system you want to restore.

To restore the file system, do the following:

1. Create the new file system structure. For example, on a SPARC system:

   ```
   # newfs /dev/rdsk/c0t0d0s6
   ```

   Device names used for disk slices on x86/x64 systems differ slightly, for example, /dev/dsk/c1d0s6.
2. Mount the file system to the mount point /a, and change to that directory.

   ```
   # mount /dev/dsk/c0t0d0s6 /a
   ```

```
# cd /a
```

3. Restore the entire `/usr` file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```

Note – Restore a file system by starting with the Level `0` backup tape, continuing with the next-lower-level tape, and continuing through the highest-level tape.

---

4. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

5. Unmount the new file system.

```
# cd /
# umount /a
```

6. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdsk/c0t0d0s6
```

7. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s6
```

8. Reboot the system. On x86/x64 systems, eject the DVD to prevent the system from booting from it.

```
# init 6
```

Performing Special Case Recovery of `/` (Root) on SPARC

Perform a special case recovery of the `/` (root) file system if there is damage to the boot block. To restore `/` (root), boot from the Solaris 10 OS DVD, and use `ufsrestore`.

The following demonstrates restoring the `/` (root) file system on slice 0 of the boot disk.

1. Insert the Solaris 10 OS DVD, and boot with the single-user mode option.

```
ok boot cdrom -s
```

2. Create the new file system structure.

```
# newfs /dev/rdsk/c0t0d0s0
```

3. Mount the file system to the mount point `/a` and change to that directory.

```
# mount /dev/dsk/c0t0d0s0 /a
# cd /a
```

4. Restore the / (root) file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```

Note – Restore a file system by starting with the Level 0 backup tape, and continuing with the next-lower-level tape, and continuing through the highest-level tape.

---

5. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

6. Install the `bootblk` in Sectors 1 through 15 of the boot disk. To do this, change to the directory that contains the `bootblk`, and enter the `installboot` command.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs
# installboot bootblk /dev/rdsk/c0t0d0s0
```

7. Unmount the new file system.

```
# cd /
# umount /a
```

8. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdsk/c0t0d0s0
```

9. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s0
```

10. Reboot the system.

```
# init 6
```

Performing Special Case Recovery of /(Root) on x86/x64

If there is damage to the partitions within the active Solaris partition on an x86/x64 system, perform a special case recovery to recover the / (root) file system. If you reconstruct the entire boot disk, establish `fdisk` partitions before restoring the Solaris root file system or any additional file systems. To restore the / (root) file system, boot from the Solaris 10 OS DVD, and use the `ufsrestore` command.

The following procedure demonstrates how to restore the `/` (root) file system on slice 0 of the active Solaris partition. This procedure describes performing the file system restore, and installing the GRUB `stage1` and `stage2` programs on the active Solaris partition

1. Insert the Solaris 10 OS DVD.
2. Use the reset button or required key sequence (e.g., Ctrl-Alt-Delete) to reset the system. Power-cycle the system as an alternative.

Note – Your system needs to be configured to boot from a DVD. This might require changes to BIOS settings before booting.

---

3. The system begins its boot process from the Solaris miniroot on DVD. The following list of boot choices displays:

```
1. Solaris Interactive (default)
2. Custom JumpStart
3. Solaris Interactive Text (Desktop session)
4. Solaris Interactive Text (Console session)
5. Apply driver updates
6. Single user shell

Enter the number of your choice.
```

4. Select boot choice 6, "Single user shell." If an instance of Solaris is found on the disk, the system asks if you want to mount it on `/a`. For example:

```
Solaris 10 5/09 s10x_u7wos_08 X86 was found on /dev/dsk/c1d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?]
```

5. Enter `n` to avoid mounting the root file system. You will manually mount the file system you want to restore.
6. Create the new file system structure.

```
# newfs /dev/rdsk/c1d0s0
```

7. Mount the file system to the mount point `/a` and change to that directory.

```
# mount /dev/dsk/c1d0s0 /a
# cd /a
```

8. Restore the `/` (root) file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```

Note – Always restore a file system by starting with a Level 0 backup tape, and continuing with the next-lower-level tape. Then continue through the highest-level tape.

9. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

10. Install the GRUB `stage1` and `stage2` programs on the Solaris partition. The `installgrub` command installs the GRUB `stage1` program in sector 0, and GRUB `stage2` program beginning at sector 50, (using 233 sectors) of the partition you specify. These sectors fall within the reserved first cylinder of the partition, and depending on the nature of the damage, may not require re-installation.

    To install the `stage1` and `stage2` programs found in miniroot from the Solaris OS DVD, use the following command:

```
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1d0s0
```

    To install the `stage1` and `stage2` programs found in the restored root file system, use the following command:

```
# /sbin/installgrub /a/boot/grub/stage1
/a/boot/grub/stage2 /dev/rdsk/c1d0s0
```

11. Unmount the new file system.

```
# cd /
# umount /a
```

12. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdsk/c0t0d0s0
```

13. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s0
```

14. Reboot the system. Eject the DVD to prevent the system from booting from it.

```
# init 6
```

Invoking an Interactive Restore

The `ufsrestore i` command invokes an interactive interface. Through the interface, you can browse the directory hierarchy of the backup tape and select individual files to extract. The term *volume* is used by `ufsrestore` and should be considered a single tape.

Using the ufsrestore i Command

The following procedure demonstrates how to use the ufsrestore i command to extract individual files from a backup tape.

1. Become the root user, and change to the temporary directory that you want to receive the extracted files.

   ```
   # cd /export/home/tmp
   ```

2. Perform the ufsrestore i command.

   ```
   # ufsrestore ivf /dev/rmt/0
   Verify volume and initialize maps
   Media block size is 64
   Dump   date: Mon Oct 11 12:30:44 2004
   Dumped from: the epoch
   Level 0 dump of /export/home on sys43:/dev/dsk/c0t0d0s7
   Label: none
   Extract directories from tape
   Initialize symbol table.
   ```

3. Display the contents of the directory structure on the backup tape.

```
ufsrestore > ls
.:
     2 *./          13  directory1    15  directory3      11  file2
     2 *../         14  directory2    10  file1           12  file3
```

4. Change to the target directory on the backup tape.

```
ufsrestore > cd directory1
ufsrestore > ls
./directory1:
  3904  ./          2 *../      3905  file1    3906  file2    3907  file3
```

5. Add the files you want to restore to the extraction list.

   ```
   ufsrestore > add file1 file2
   Make node ./directory1
   ```

   Files you want to restore are marked with an asterisk (*) for extraction. If you extract a directory, all of the directory contents are marked for extraction.

   In this example, two files are marked for extraction. The ls command displays an asterisk in front of the selected file names, file1 and file2.

```
ufsrestore > ls
./directory1:
  3904 *./          2 *../      3905 *file1    3906 *file2    3907  file3
```

6. To delete a file from the extraction list, use the `delete` command.

```
ufsrestore > delete file1
```

The `ls` command displays the `file1` file without an asterisk.

```
ufsrestore > ls
./directory1:
  3904 *./          2 *../      3905  file1    3906 *file2    3907  file3
```

7. To view the files and directories marked for extraction, use the `marked` command.

```
ufsrestore > marked
./directory1:
  3904 *./          2 *../      3906 *file2
```

8. To restore the selected files from the backup tape, perform the command:

```
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
```

Note – The `ufsrestore` command has to find the selected files. If you used more than one tape for the backup, first insert the tape with the highest volume number and type the appropriate number. Then repeat this step, working towards Volume #1 until all files are restored.

```
extract file ./directory1/file2
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
```

Note – Answering `y` sets ownership and permissions of the temporary directory to those of the mount point on the tape.

9. To exit the interactive restore after the files are extracted, perform the command:

```
ufsrestore> quit
```

10. Move the restored files to their original or permanent directory location, and delete the files from the temporary directory.

```
# mv /export/home/tmp/directory1/file2 /export/home
# rm -r /export/home/tmp/directory1
```

Note – You can use the `help` command in an interactive restore to display a list of available commands.

Performing an Incremental Restore

When performing incremental restores, start with the last volume and work towards the first. The system uses information in the `restoresymtable` file to restore incremental backups on top of the latest full backup.The following procedure demonstrates how to restore the `/export/home` file system from incremental tapes.

Note – This procedure makes use of the interactive restore to assist in showing the concept of incremental restores. You would typically use a command, such as `ufsrestore rf`, for restoring entire file systems.

---

1. View the contents of the `/etc/dumpdates` file for information about the `/export/home` file system.

```
# more /etc/dumpdates |grep c0t0d0s7
/dev/rdsk/c0t0d0s7       0 Wed Apr 07 09:55:34 2004
/dev/rdsk/c0t0d0s7       1 Web Apr 07 09:57:30 2004
```

2. Create the new file system structure for the `/export/home` file system.

```
# newfs /dev/rdsk/c0t0d0s7
```

3. Mount the file system and change to that directory.

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

4. Insert the Level `0` backup tape.

5. Restore the `/export/home` file system from the backup tapes.

```
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Wed Apr 07 09:55:34 2004
Dumped from: the epoch
Level 0 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
```

```
Calculate extraction list.
Make node ./directory1
Make node ./directory2
Make node ./directory3
Extract new leaves.
Check pointing the restore
extract file ./file1
extract file ./file2
extract file ./file3
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
#
```

6. Load the next lower-level tape into the tape drive.

```
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Wed Apr 07 09:57:30 2004
Dumped from: Wed Apr 07 09:55:34 2004
Level 1 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Begin incremental restore
Initialize symbol table.
Extract directories from tape
Mark entries to be removed.
Calculate node updates.
Make node ./directory4
Make node ./directory5
Make node ./directory6
Find unreferenced names.
Remove old nodes (directories).
Extract new leaves.
Check pointing the restore
extract file ./file4
extract file ./file5
extract file ./file6
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
#
```

Alternative Steps

The following steps are alternatives to the previous Steps 5 and 6.

5. Restore the /export/home file system from the backup tapes. (This example uses an interactive, verbose restore to provide more detailed information.)

```
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Mon Oct 11 13:10:12 2004
Dumped from: the epoch
Level 0 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
     2 *./                 8  directory2        5  file2
     2 *../                9  directory3        6  file3
     7  directory1         4  file1             3  lost+found/
```

The system lists files from the last Level 0 backup.

```
ufsrestore > add *
Warning: ./lost+found: File exists
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./file1
extract file ./file2
extract file ./file3
extract file ./directory1
extract file ./directory2
extract file ./directory3
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
Directories already exist, set modes anyway? [yn] n
ufsrestore > q
#
```

6. The information in the /etc/dumpdates file shows an incremental backup that was taken after the Level 0 backup. Load the next tape and perform the incremental restore.

```
# ufsrestore iv
Verify volume and initialize maps
Media block size is 64
Dump   date: Wed Apr 07 09:57:30 2004
Dumped from: Wed Apr 07 09:55:34 2004
Level 1 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
     2 *./             13  directory4      15  directory6      11  file5
     2 *../            14  directory5      10  file4           12  file6
ufsrestore > add *
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./file4
extract file ./file5
extract file ./file6
extract file ./directory4
extract file ./directory5
extract file ./directory6
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
ufsrestore > q
#
```

Exercise: Restoring Files and File Systems

In this exercise, you complete the following tasks:

- Backup the root (/) file system to tape
- Backup the root (/) file system to a file on disk
- Restore a selective file from a `ufsdump` archive
- Restore the root (/) file system from tape or disk

Preparation

Some steps in this lab use a tape drive. If your system has an attached tape drive, obtain an appropriate tape from your instructor. A tape drive is not necessary to complete the exercise.

This exercise requires a system with a `/` (root) file system that is separate from the `/usr` and `/var` file systems, and that has an unused spare disk with the capacity to store the entire root (/) file system.

Identify the device name of the `/` (root) file system. For example:

```
# df -h /
Filesystem            size   used  avail capacity  Mounted on
/dev/dsk/c0t0d0s0     4.8G   175M   4.6G     4%    /
```

Record the device name of the root file system, `/dev/dsk/c0t0d0s0` in this example.

Exercise Summary

This exercise is composed of four tasks that illustrate how to back up and restore the root (/) file system on SPARC and x86/x64 systems. The tasks in this exercise describe the following concepts:

- Back Up the root (/) File System to Tape - demonstrates the method used to create a `ufsdump` archive of the root file system on a tape device.
- Back Up the root (/) File System to a File on Disk - demonstrates the method used to create a `ufsdump` archive of the root file system on a file in a UFS file system.

- Restore the `/etc/inet/hosts` File From Tape or From a File in Disk - describes how to use an interactive restore session to select and restore a file from a `ufsdump` archive found either on tape, or in a file on disk.
- Destroy and Restore the root (/) File System - demonstrates destroying the root (/) file system by removing critical system files, booting SPARC and x86/x64 systems from DVD, creating a new root (/) file system, restoring the root file system from a `ufsdump` archive on tape or on disk, and installing boot programs.

Task 1- Back Up the Root (/) File System to Tape

Complete the following steps:

If your system has an attached tape drive, perform the following steps to back up the root (/) file system to tape. Obtain a tape for your tape drive from your instructor.

1. Log in as the root user, and open a terminal window. Shut down the system to run level `0`. Then boot the system to run level `S` (`single-user` milestone). Supply the root password as required to enter run level `S`.
2. Verify that a tape is in your tape drive.
3. Use the `ufsdump` command to create a backup tape for the root (/) file system.
4. Verify that the `/` (root) file system is on the tape.
5. Allow the system to remain at the `single-user` milestone.

Task 2 - Back up the / (Root) File System to a File on Disk

Perform the following steps to back up your root (/) file system to a file on your spare disk.

1. Use the `format` utility to partition the extra hard drive and create one large partition to use as a backup device. Use the `All Free Hog` method to set partition 0 to use all disk space.

Note – This procedure assumes that on x86/x64 systems, the spare disk has one Solaris `fdisk` partition that consumes the entire disk.

---

2. Create a file system on the large partition on the extra disk.
3. Mount the new file system on `/mnt`.
4. Use the `ufsdump` command to backup the root (`/`) file system to a file below the `/mnt` directory. For example:
5. Verify that the `/` (root) file system has been backed up to the file on disk.
6. Allow the system to continue to boot to run level 3 (`multi-user` or `all` milestone).

Task 3 – Restore `/etc/inet/hosts` File From Tape or File on Disk

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change to the `/var/tmp` directory.
2. Verify that the new file system that contains your backup file is still mounted on `/mnt`, and that the backup file exists in it.
3. If the file system is no longer mounted, mount the new file system on `/mnt`.
4. Enter the `ufsrestore if` command to retrieve the `/etc/inet/hosts` file from the tape or the backup file on disk.
5. Change to the `/etc/inet` directory in the `ufsdump` archive, and list the files in the directory.
6. Add the `hosts` file to the list of files to extract, and display the list.
7. Extract the `hosts` file from the `ufsdump` archive. Specify volume number 1. Do not set the owner and mode for `.`, and then quit the `ufsrestore` command.
8. Verify that the `etc/inet/hosts` file exists below the `/var/tmp` directory.
9. Unmount the file system mounted on `/mnt`.

Note – Be certain to unmount the `/mnt` file system before you destroy your root file system in the next task; especially if your only root (`/`) file system backup is in a file below `/mnt`.

Task 4 – Destroy and Restore the Root (`/`) File System

Complete the following steps:

Destroy the Root File System

1. Change to the root (/) directory, and remove the following critical system directories and their contents: `/platform` and `/devices`.
2. Attempt to shut down the system to run-state 0.
3. If the system fails to shut down:
   a. On SPARC systems, press the Stop-A key sequence to abort the operating system.
   b. On x86/x64 systems, press the reset button, or use a key sequence (such as Control-Alt-Delete) to reset the system. Power-cycle the system as an alternative.
4. Attempt to boot the system from the boot disk.

   What happens?

   ____

   ____

Boot the SPARC system from DVD

1. Insert the Solaris 10 OS DVD
2. Boot the system from the DVD with the single-user mode option.

Boot the x86/x64 System from DVD

1. Insert the Solaris 10 OS DVD.
2. If the `Press any key to continue` prompt displays, press any key, and allow the system to boot from DVD.

Note – Your system needs to be configured to boot from DVD, which might require changes to BIOS settings before booting.

3. If the system is not responsive, use the reset button, or required key sequence (e.g., Control-Alt-Delete) to reset the system. You can power-cycle the system as an alternative.

4. The system begins its boot process from the Solaris miniroot on DVD. The following list of boot choices displays:

```
1. Solaris Interactive (default)
2. Custom JumpStart
3. Solaris Interactive Text (Desktop session)
4. Solaris Interactive Text (Console session)
5. Apply driver updates
6. Single user shell

Enter the number of your choice.
```

5. Select boot choice 6, the single user shell. If an instance of Solaris is found on disk, the system asks if you want to mount it on `/a`. For example:

```
Solaris 10 5/09 s10x_u7wos_08 X86 was found on /dev/dsk/c1d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?]
```

6. Enter `n` to avoid mounting the root file system. You will manually mount the file system you want to restore.

Create a New Root File System and Prepare for the Restore

1. Use the `newfs` command to create a new file system on the `/` (root) slice.

Note – Be certain to specify the correct device name for the root slice. Be sure you do not specify a slice on the spare disk.

---

2. Mount the new file system on the `/a` directory.
3. Verify that your root backup tape is in the tape drive.
4. Mount the file system on your spare disk that contains your root backup as `/mnt`.
5. Verify that the file system that contains your backup file is mounted on `/mnt`, and that the backup file exists in it.

Restore the Root (`/`) File System from Tape or Disk

1. Change directory to `/a`.
2. To restore the root (`/`) file system from tape, use the `ufsrestore` command and specify the tape device.
3. To restore the root (`/`) file system from disk, use the `ufsrestore` command and specify the backup file on disk.

4. Remove the `restoresymtable` file.
5. Change directory to root (`/`).
6. Unmount the file systems mounted on `/a` and `/mnt`.

Install the Boot Block or GRUB Boot Programs

Depending on the damage incurred on disk, it may or may not be necessary to install the boot block on SPARC systems, or the GRUB boot programs (`stage1` and `stage2`) on x86/x64 systems. If it is necessary to install these programs, perform the following steps:

1. On a SPARC system, to install a new boot block in sectors 1 through 15 of the root (`/`) slice, change directory to the location of the boot block, and use the `installboot` command.
2. On an x86/x64 system, to install the GRUB `stage1` and `stage2` programs found in miniroot from the Solaris OS DVD, use the `installgrub` command.

Reboot the System

1. On x86/x64 systems, eject the Solaris OS DVD to prevent the system from booting from it.
2. Reboot the system.
3. Login as root and verify that the system is operational.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications

Exercise Solutions: Restoring Files and File Systems

This section contains solutions to the exercise.

Note – Disk and file system device names used in these procedures represent devices found on both SPARC and x86/x64 systems, and are provided as examples only. Use correct device names for your specific system.

Task 1- Back Up the Root (/) File System to Tape

Complete the following steps:

If your system has an attached tape drive, perform the following steps to back up the root (/) file system to tape. Obtain a tape for your tape drive from your instructor.

1. Log in as the root user, and open a terminal window. Shut down the system to run level `0`. Then boot the system to run level `S` (`single-user` milestone). Supply the root password as required to enter run level `S`.

```
# init 0
(output omitted)
```

On a SPARC system, use the `boot -s` command at the `ok` prompt:

```
ok boot -s
```

On an x86/x64 system, perform the following steps:

a. Press any key at the `Press any key to reboot` prompt.

b. When the GRUB main menu displays, enter `e` to display the list of commands for the default boot choice. For example:

```
GNU GRUB version 0.95 (572K lower / 3380808K upper memory)
+-----------------------------------------------------------------------+
| kernel /platform/i86pc/multiboot
| module /platform/i86pc/boot_archive
|
+-----------------------------------------------------------------------+
```

c. Highlight the `kernel` command in the command list, and enter `e` to edit the command. Add the `-s` option to the end of the `kernel` command line, and press Return. For example:

```
grub edit> kernel /platform/i86pc/multiboot -s
```

d. Enter `b` to boot the system.

2. Verify that a tape is in your tape drive.
3. Use the `ufsdump` command to create a backup tape for the root (`/`) file system.

   ```
   # ufsdump 0uf /dev/rmt/0 /
   ```
4. Verify that the `/` (root) file system is on the tape.

   ```
   # ufsrestore tvf /dev/rmt/0
   ```

   The screen should list directory structures under root (/) first, followed by files.
5. Allow the system to remain at the `single-user` milestone.

Task 2 - Back up / (Root) File System to a File on Disk

Perform the following steps to back up your root (/) file system to a file on your spare disk.

1. Use the `format` utility to partition the extra hard drive and create one large partition to use as a backup device. Use the `All Free Hog` method to set partition 0 to use all disk space. This is an example from a SPARC system:


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


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <HDS722512VLAT80 cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@0,0
       1. c0t1d0 <DEFAULT cyl 57459 alt 2 hd 16 sec 255>
          /pci@1e,600000/ide@d/dad@1,0
Specify disk (enter its number): 1
selecting c0t1d0
```

When you are through partitioning the extra disk drive, your partition table should look something like this:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 57458      111.79GB    (57459/0/0) 234432720
  1       swap    wu       0                0         (0/0/0)             0
  2     backup    wu       0 - 57458      111.79GB    (57459/0/0) 234432720
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
```

```
  5 unassigned    wu       0                 0          (0/0/0)             0
  6 unassigned    wu       0                 0          (0/0/0)             0
  7 unassigned    wu       0                 0          (0/0/0)             0
```

This is an example from an x86/x64 system:

Note – This procedure assumes that on x86/x64 systems, the spare disk has one Solaris `fdisk` partition that consumes the entire disk.

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


AVAILABLE DISK SELECTIONS:
       0. c1d0 <DEFAULT cyl 39887 alt 2 hd 64 sec 63>
          /pci@0,0/pci-ide@7/ide@0/cmdk@0,0
       1. c2d0 <DEFAULT cyl 10008 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide@7/ide@1/cmdk@0,0
Specify disk (enter its number): 1
selecting c2d0
```

When you are through partitioning the extra disk drive, your partition table should look something like this:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0   reserved    wm       3 - 10007       76.64GB    (10005/0/0) 160730325
  1       swap    wu       0                0         (0/0/0)             0
  2     backup    wu       0 - 10007       76.67GB    (10008/0/0) 160778520
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 alternates    wm       1 -     2       15.69MB    (2/0/0)         32130
```

2. Create a file system on the large partition on the extra disk. For example, on a SPARC system:

   # newfs /dev/rdsk/*c0t1d0s0*

   On an x86/x64 system:

   # newfs /dev/rdsk/*c2d0s0*

3. Mount the new file system on `/mnt`. For example, on a SPARC system:

   # mount /dev/dsk/*c0t1d0s0* /mnt

   On an x86/x64 system:

```
# mount /dev/dsk/c2d0s0 /mnt
```

4. Use the `ufsdump` command to backup the root (`/`) file system to a file below the `/mnt` directory. For example:

```
# ufsdump 0uf /mnt/dump_root_0 /
```

5. Verify that the `/` (root) file system has been backed up to the file on disk. For example:

```
# ufsrestore tvf /mnt/dump_root_0
```

The screen should list directory structures under root (/) first, followed by files.

6. Allow the system to continue to boot to run level 3 (`multi-user` or `all` milestone).

```
# <Control-D>
```

Task 3 – Restore `/etc/inet/hosts` From Tape or File on Disk

Complete the following steps:

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

```
# cd /var/tmp
```

2. Verify that the new file system that contains your backup file is still mounted on `/mnt`, and that the backup file exists in it.

```
# ls /mnt
dump_root_0  lost+found
#
```

3. If the file system is no longer mounted, mount the new file system on `/mnt`. For example, on a SPARC system:

```
# mount /dev/dsk/c0t1d0s0 /mnt
```

On an x86/x64 system:

```
# mount /dev/dsk/c2d0s0 /mnt
```

4. Enter the `ufsrestore if` command to retrieve the `/etc/inet/hosts` file from the tape or the backup file on disk.

For example, to restore the file from tape:

```
# ufsrestore if /dev/rmt/0
```

For example, to restore the file from a file on disk:

```
# ufsrestore if /mnt/dump_root_0
```

```
ufsrestore > ls
```

See files and directories for the / (root) file system.

Change to the `/etc/inet` directory in the `ufsdump` archive, and list the files in the directory. See files and directories for the `/etc/inet` file system.

```
ufsrestore > cd /etc/inet
ufsrestore > ls
```

Add the `hosts` file to the list of files to extract, and display the list. See the `hosts` file listed.

```
ufsrestore > add hosts
ufsrestore > marked
./etc/inet:
*hosts
ufsrestore >
```

5. Extract the `hosts` file from the `ufsdump` archive. Specify volume number 1. Do not set the owner and mode for ., and then quit the `ufsrestore` command.

```
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] n
ufsrestore > q
```

6. Verify that the `etc/inet/hosts` file exists below the `/var/tmp` directory.

```
# ls etc/inet/hosts
etc/inet/hosts
```

7. Unmount the file system mounted on /mnt.

```
# umount /mnt
#
```

Note – Unmount the `/mnt` file system before you destroy your root file system in the next task, especially if your only root (`/`) file system backup is in a file below `/mnt`.

Task 4 – Destroy and Restore the Root (/) File System

Complete the following steps:

Destroy the Root File System

1. Change to the root (/) directory, and remove the following critical system directories and their contents: /platform and /devices.

```
# cd /
# rm -r /platform /devices
```

2. Attempt to shut down the system to run-state 0.

```
# init 0
```

3. If the system fails to shut down:
   a. On SPARC systems, press the Stop-A key sequence to abort the operating system.
   b. On x86/x64 systems, press the reset button, or use a key sequence (e.g., Ctrl-Alt-Delete) to reset the system. Power-cycle the system as an alternative.

4. Attempt to boot the system from the boot disk. You will see that the system fails to boot. SPARC systems display the message:

```
Boot load failed.
The file just loaded does not appear to be executable
```

   x86/x64 systems display the message:

```
Error 20: Multiboot kernel must be loaded before
modules
```

Boot the SPARC system from DVD

1. Insert the Solaris 10 OS DVD
2. Boot the system from the DVD with the single-user mode option.

```
ok boot cdrom -s
```

Boot the x86/x64 System from DVD

1. Insert the Solaris 10 OS DVD.
2. If the `Press any key to continue` prompt displays, press any key, and allow the system to boot from DVD.

Note – Your system needs to be configured to boot from DVD, which might require changes to BIOS settings before booting.

3. If the system is not responsive, use the reset button, or required key sequence (e.g., Control-Alt-Delete) to reset the system. You can power-cycle the system as an alternative.
4. The system begins its boot process from the Solaris miniroot on DVD. The following list of boot choices displays:

```
1. Solaris Interactive (default)
2. Custom JumpStart
3. Solaris Interactive Text (Desktop session)
4. Solaris Interactive Text (Console session)
5. Apply driver updates
6. Single user shell

Enter the number of your choice.
```

5. Select boot choice 6, the single user shell. If an instance of Solaris is found on disk, the system asks if you want to mount it on `/a`. For example:

```
Solaris 10 5/09 s10x_u7wos_08 X86 was found on /dev/dsk/c1d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?]
```

6. Enter `n` to avoid mounting the root file system. You will manually mount the file system you want to restore.

Create a New Root File System and Prepare for the Restore

1. Use the `newfs` command to create a new file system on the `/` (root) slice.

Note – Be certain to specify the correct device name for the root slice. Be sure you do not specify a slice on the spare disk.

For example, on a SPARC system:

`# `**`newfs /dev/rdsk/`*`c0t0d0s0`***

On an x86/x64 system:

`# `**`newfs /dev/rdsk/`*`c1d0s0`***

2. Mount the new file system on the `/a` directory.

For example, on a SPARC system:

`# `**`mount /dev/dsk/`*`c0t0d0s0`*` /a`**

On an x86/x64 system:

```
# mount /dev/dsk/c1d0s0 /a
```

3. Verify that your root backup tape is in the tape drive.
4. Mount the file system on your spare disk that contains your root backup as `/mnt`.

   For example, on a SPARC system:

   ```
   # mount /dev/dsk/c0t1d0s0 /mnt
   ```

   On an x86/x64 system:

   ```
   # mount /dev/dsk/c2d0s0 /mnt
   ```

5. Verify that the file system that contains your backup file is mounted on `/mnt`, and that the backup file exists in it.

   ```
   # ls /mnt
   dump_root_0  lost+found
   ```

Restore the Root (/) File System from Tape or Disk

1. Change directory to `/a`.

   ```
   # cd /a
   ```

2. To restore the root (/) file system from tape, use the `ufsrestore` command and specify the tape device:

   ```
   # ufsrestore rf /dev/rmt/0
   ```

3. To restore the root (/) file system from disk, use the `ufsrestore` command and specify the backup file on disk:

   ```
   # ufsrestore rf /mnt/dump_root_0
   ```

4. Remove the `restoresymtable` file.

   ```
   # rm restoresymtable
   ```

5. Change directory to root (/).

   ```
   # cd /
   ```

6. Unmount the file systems mounted on `/a` and `/mnt`.

   ```
   # umount /a
   # umount /mnt
   ```

Install the Boot Block or GRUB Boot Programs

Depending on the damage incurred on disk, it may or may not be necessary to install the boot block on SPARC systems, or the GRUB boot programs (stage1 and stage2) on x86/x64 systems. If it is necessary to install these programs, perform the following steps:

1. On a SPARC system, to install a new boot block in sectors 1 through 15 of the root (/) slice, change directory to the location of the boot block, and use the installboot command. For example:

```
# cd /usr/platform/`uname -m`/lib/fs/ufs
# installboot bootblk /dev/rdsk/c0t0d0s0
```

2. On an x86/x64 system, to install the GRUB stage1 and stage2 programs found in miniroot from the Solaris OS DVD, use the installgrub command.

```
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1d0s0
```

Reboot the System

1. On x86/x64 systems, eject the Solaris OS DVD to prevent the system from booting from it.
2. Reboot the system.

```
# init 6
```

3. Login as root and verify that the system is operational.