HP-UX
System Administrator Manual

Volume 1

HP 9000 Series 300 Computers

HP Part Number 98594-90060

HEWLETT
PACKARD

Hewlett-Packard Company
3404 East Harmony Road, Fort Collins, Colorado 80525

Please print or type your name and address.

Name: ______________________________

Company: ______________________________

Address: ______________________________

City, State, Zip: ______________________________

Telephone: ______________________________

Additional Comments: ______________________________

HP-UX System Administrator Manual
HP Part Number 98594-90060
E1288

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 37 LOVELAND, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard Company
Attn: Learning Products Center
3404 East Harmony Road
Fort Collins, Colorado 80525-9988

Win an HP Calculator!

Your comments and suggestions help us determine how well we meet your needs. **Returning this card with your name and address enters you into a quarterly drawing for an HP calculator***.

HP-UX
System Administrator Manual

	Agree				Disagree
The manual is well organized.	○	○	○	○	○
It is easy to find information in the manual.	○	○	○	○	○
The manual explains features well.	○	○	○	○	○
The manual contains enough examples.	○	○	○	○	○
The examples are appropriate for my needs.	○	○	○	○	○
The manual covers enough topics.	○	○	○	○	○
Overall, the manual meets my expectations.	○	○	○	○	○

You have used this product:

__ Less than 1 week __ Less than 1 year __ More than 2 years
__ Less than 1 month __ 1 to 2 years

fold ——

Please write additional comments, particularly if you disagree with a statement above. Use additional pages if you wish. The more specific your comments, the more useful they are to us.

Comments: ____

____
____
____
____
____
____
____
____
____
____
____
____

*Offer expires 1/1/1991. (Manual: 98594-90060 E1288)

Please Tape Here

Legal Notices

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

December 1988 ... Edition 1. Documents the 6.5 release of HP-UX.

This edition of the *HP-UX System Administrator Manual* replaces part number 97597-90060 (First Edition; April 1988).

Contents

1

Getting Started

Welcome

This manual is written for you, the Series 300 HP-UX system administrator. Although some familiarity with computers is assumed, this manual will serve people with varying levels of expertise. The HP-UX operating system is composed primarily of AT&T Bell Laboratories' System V.2 UNIX.[1]
However, Hewlett-Packard has incorporated its own extensions as well as features from the University of California at Berkeley Unix 4.1 and 4.2 BSD (Berkeley Systems Distribution) systems and from AT&T's System V UNIX.

Who is the system administrator? The system administrator is the person responsible for installing the HP-UX operating system software, updating the software, tuning the system for optimum performance, maintaining the system, and repairing the system if something goes wrong. Additionally, the system administrator should become the local expert to whom other HP-UX users go for help.

---

1 UNIX is a registered trademark of AT&T in the USA and other countries.

What's in this Manual?

This manual is a guide to help you fulfill your duties as system administrator. The following is an overview of the chapters in this manual:

- *Chapter 1: Getting Started*

  This chapter provides an overview of the *System Administrator Manual,* explains the conventions the manual uses, mentions other manuals that will aid you in administrative tasks, points out differences between single-user and multi-user systems, and discusses the system administrator's responsibilities.

- *Chapter 2: System Management Concepts*

  This chapter discusses the HP-UX directory structure, processes, IDs, the super-user, file system implementation, magnetic tape, memory management, HP-UX cluster concepts, and device input and output.

- *Chapter 3: System Startup and Shutdown*

  This chapter discusses system startup functions, performing a controlled shutdown of the system, and recovering from a power failure or a disk crash.

- *Chapter 4: Customizing the HP-UX Operating System*

  Arranged alphabetically by task, this chapter contains instructions for accomplishing tasks to customize the system.

- *Chapter 5: Periodic System Administrator Tasks*

  There are a variety of tasks that the System Administrator must perform on a regular basis to maintain the integrity and functionality of the system. This chapter defines these tasks.

- *Chapter 6: Kernel Customization*

  Various attributes of the kernel (operating system) may be customized to enhance the performance of your system. The configuration changes that are explained in this chapter will help you in this task.

- *Chapter 7: System Accounting*

  As system administrator you may want to periodically evaluate how well your Series 300 HP-UX system is operating, as well as how many resources those logging onto your system are using. This chapter discusses the various accounting features available on HP-UX, how to install them, and how to produce various useful reports.

- *Chapter 8: Trusted Systems*

  This chapter describes concepts and tasks associated with security, as defined by the government, at the C2 level.

Conventions Used in this Manual

The following font style conventions are used throughout this manual:

- Computer font indicates a literal either typed by the user or displayed by the system. You must press the [Return] key after typing the command. A typical example is:

  ```
  fsck /dev/dsk/0s0
  ```

  Computer font also indicates files, HP-UX commands, system calls, subroutines, etc. Examples include `/etc/profile`, `date`, and `getpid`.

- Boldface is used when a word is first defined (as **flebnee**).

- Italics is used when terms are referenced to the *HP-UX Reference*. Italics is also used to indicate the titles of manuals, as well as for symbolic items either typed by the user or displayed by the system as discussed below.

  Note that when a command and/or file name is part of a literal, it is shown in computer font and not italics. However, if the command or file name is symbolic (but not literal), it is shown in italics as the following example illustrates:

  `fsck` *device_file_name*

  Here you would type in your own *device_file_name*.

  Italics is also used for general emphasis (*do not touch*).

- When the use of softkeys (or function keys) is discussed, the softkey will be shown in heavy bold type with a box. For example, a softkey labeled "PREVIOUS" will be printed as `PREVIOUS`.

- Environment variables such as PATH or MAIL are represented in uppercase characters.

- Quotation marks are used for chapter and section titles. Quotation marks are also used to emphasize unusual terms that may be typical for general computer usage, but are not standard English.

- Unless otherwise stated, all references such as "refer to the *login(1)* entry for more details" refer to entries in the *HP-UX Reference* manual. The parenthetic number refers to the section in the *HP-UX Reference*. Some of these entries will be under an associated heading. For example, the *chgrp(1)* entry is under the *chown(1)* heading. If you cannot find an entry where you expect it to be, use the *HP-UX Reference* manual's permuted index.

Using Other HP-UX Manuals

Besides this manual, the HP-UX manuals listed below will aid you in your system administrator tasks:

- The installation guide for your specific Series 300 computer contains instructions for installing the computer hardware, interface cards, and peripherals. The guide supplies all the hardware-specific information needed to set up the HP-UX system.
- The *HP-UX Installation Manual* provides step-by-step instructions for installing the HP-UX operating system software and explains what to do after the system has been successfully installed.
- There are several beginner guides to help you get started with HP-UX, the vi editor, and shells.
- The *HP-UX Reference* contains the syntactic and semantic details of all commands and application programs, system calls, subroutines, special files, file formats, miscellaneous facilities, and system maintenance procedures available on the Series 300 HP-UX operating system. Use this manual when looking for complete specifications of a command or a special file.
- The multi-volume set of *HP-UX Concepts and Tutorials* contains information on a broad range of HP-UX topics and tools. You may be particularly interested in the sections on UUCP and on the different shells. Refer to the *Documentation Roadmap* or the *HP-UX Concepts and Tutorials* for a list of topics.
- Manuals provided with optional products or applications, such as Local Area Network (LAN) or XWindows, provide information specific to the product.
- *Peripheral Installation Guide* describes how to set switches and connect peripherals to your Series 300 computer.

The Administrator's Responsibilities

This section contains a brief discussion of the system administrator's responsibilities and tells you where to find related information.

Installing and Testing the Hardware

As system administrator, you should make sure that your computer is installed and operating properly by using the instructions and tests in the installation guide supplied with your computer. The computer hardware must function properly before HP-UX is installed.

Evaluating Users' Needs

You must analyze the intended uses of the system. Knowledge of the number of users, the characteristics of each user, the system resources and peripherals required by each user, and the data/programs that must be shared by various user groups, will help you set up HP-UX for optimum performance. This also applies to single-user systems.

To aid you in this analysis, a sample user-survey form is provided at the end of this chapter. You may want to change this survey to fit your particular needs. Most users think in terms of "I need to do this job" not "I need FORTRAN, Graphics, a plotter, and 500 000 bytes of data storage." The survey should help you identify the needs of the system users and translate those needs into data relevant to system configuration.

Installing the HP-UX Operating System

The HP-UX operating system is supplied either on a 1/4 -inch cartridge tape or on 3 1/2 inch flexible disks. The operating system is installed on a hard disk drive. As system administrator, you are responsible for installing HP-UX. Instructions for accomplishing this task are provided in the *HP-UX Installation Manual.*

Configuring the HP-UX kernel

How the operating system uses computer resources depends on certain values and configurations that you control. Configuring the system influences its efficiency and response time. Once familiar with the system, you can use the instructions in Chapter 6 of this manual to alter the system configuration.

Allowing Users Access to the System

Once HP-UX is installed, you are responsible for allowing access by other users. This involves providing each user with a user name, a password, and a portion of the file system for his use. Instructions for adding users and assigning passwords are contained in Chapter 5 of this manual.

Adding and Moving Peripheral Devices

Another responsibility is to add peripherals (for example, printers, terminals, mass storage devices) to the HP-UX system as they are required. Directions for installing the peripherals are in Chapter 4 of this manual.

Monitoring File System Use and Growth

As HP-UX is used, files are added to the file system. If unused files are not removed, the amount of space required to store files eventually exceeds available space. One of your responsibilities is to monitor the size of the file system and identify unused files. Unused files should be archived (if needed in the future) and then removed from the file system. Also, you should watch for files that continually increase in size. Ask the file's owner to see if the file is needed, and to see if its size can be reduced. Instructions for monitoring the use and growth of the file system are supplied in Chapter 5 of this manual.

Updating the HP-UX System

You will receive software updates by purchasing HP support services that provide periodic updates. These updates change existing capabilities and add new capabilities, ensuring that your system contains the latest version of the software.

As system administrator, you are responsible for installing each software update. You should update the manuals to include the documentation changes provided with each update and keep a log showing when the update was installed. Notify all system users of the changes caused by the update. Because each update depends on changes made by the previous update, you must install each update when it arrives. Instructions for installing updates are in Chapter 4 of this manual.

Back Up, Recovery, and Restoring the System

The HP-UX operating system, programming languages, and applications software represent a large investment of time and money. Files can be unintentionally removed and each access to the system provides an opportunity for error. A critical error can cause additional errors in the file system and, if the system becomes corrupt enough, file system errors increase rapidly.

Loss of the system can also occur through unwelcome circumstances (such as spilled coffee, smoke contamination, dust, or fire) that damage a mass storage device, its media, and/or the data it contains.

As system administrator, you should make a file system backup and a recovery system. A **backup** is a copy of the HP-UX operating system, file system, and programming languages. A **recovery system** is a bootable subset of HP-UX and is created with the `mkrs` command.

If your system is destroyed, you can recover by using a combination of your recovery system and your latest backup. If a user accidentally removes a needed file, the file (or a previous version of it) can be recovered by copying it back into the file system from the backup. Note that a system backup is the *only* way to recover a deleted or destroyed file.

Instructions for backing up the system are given in Chapter 5 of this manual. Instructions for creating a recovery system are given in Chapter 4.

Detecting/Correcting File System Errors

Every day the system is used, numerous files are created, modified, and removed; each action requires an update to the file system. With each update to the file system it is possible that one or more of the updates could fail (for example, because of abnormal system shutdown). When an update fails, the file system can become corrupt.

HP-UX provides the `fsck` command—a program that checks the integrity of a file system and (optionally) repairs that system. Each time you boot your

HP-UX system, HP-UX automatically checks to see if your system was improperly shutdown. If HP-UX detects an improper shutdown, it will automatically check (and, if necessary, repair) the file system. Additionally, you should check the file system whenever you observe unexpected system behavior. Continuing to use a corrupt file system can further corrupt the files and can potentially crash your system. Instructions for verifying and repairing the file system are located in Appendix A of this manual and the *fsck(1M)* entry in the *HP-UX Reference*.

Assisting Other Users

Since you carry the title "System Administrator", users may come to you for help with the system. You should plan to allocate a portion of your time for consulting and problem solving.

If you have purchased certain support services, you have access to direct technical support from Hewlett-Packard. As the system administrator, you are the only person authorized to use this service. If other system users have difficulty with the system, they should direct their questions to you. If you cannot solve the problem, then call your support person at Hewlett-Packard.

Providing a "Back-up" Administrator

At least one other person should be trained as the system administrator to handle your responsibilities if you are gone.

To ease your job as system administrator and the job of the "back-up" system administrator, you should automate as many of your tasks as possible.

User Survey

Name _ _ _ _ _ _ _ _ _ _ _ _ _   Location _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Phone _ _ _ _ _ _ _ _ _ _ _ _ _

Location where you will use the system: _

User Category (please check all that apply):

_ Engineers and Managers — Run existing application programs; enter date, create models.

_ Technical Data Entry Operator — Run existing application programs; enter data or automatically read data from instrumentation.

_ Secretary - Word Processing Operator — Run existing application programs; enter data/text.

_ General Programmer — Develop application programs.

_ System Programmer Support Personnel — Develop programs for improving computer system performance or for use by other programmers.

Describe your application._ _

_ _

_ _

What programming language(s) will you use?_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _

What applications software (such as graphics) will you use? _ _ _ _ _ _ _ _ _

_ _

_ _

What computer hardware or peripherals will you need to access?

_ Inkjet printer
_ Impact printer
_ Graphics terminal
_ Laser printer
_ Plotter
_ Removable mass storage devices
_ Other _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Are there other users with whom you want to share programs or data? _ _ _ _ _

If so, list them. _

_ _

Will you generate or use large amounts of data? _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

If so, how much must be "on-line" (accessible at all times)?

_ _

_ _

What long-term data storage does your application requrire? _ _ _ _ _ _ _ _ _

_ _

How many programs/processes will you be running at one time?_ _ _ _ _ _ _ _ _

_ _

Which programs are interactive, which will you run in a background mode?

Can any programs be run overnight?_ _

_ _

Contents

2

System Management Concepts

This chapter discusses several essential concepts needed to manage an HP-UX system. These concepts include:

- the HP-UX directory structure
- processes
- IDs
- the superuser
- file system implementation
- magnetic tape
- memory management
- device input and output
- HP-UX clusters

It is not necessary to understand all of these concepts in depth; however, you should at least be familiar with the terms.

The HP-UX Hierarchy

The file system of HP-UX is organized in a tree structure. The base of the tree is the root of the file system, and the file name / is associated with the root. Under the root are several directories created when you installed your system: `bin`, `dev`, `etc`, `lib`, `system`, `tmp`, `users`, `disc`, `lost&found`, and `usr`.

This section describes the basic purpose of the major directories in your HP-UX hierarchy. This discussion includes not only the eight standard directories mentioned above but also many directories below the standard directories in the file system tree structure. You will find this useful as you add files and modify your system in the future. A brief description of the major directories follows.

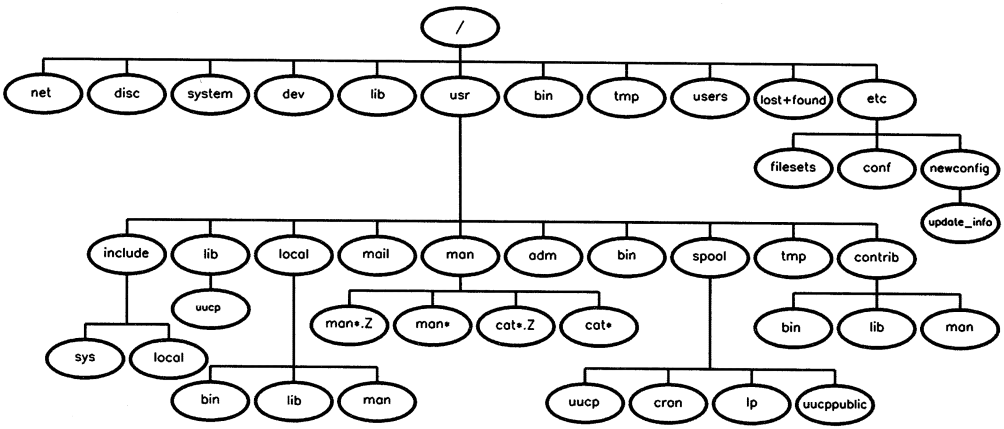


Figure 2-1. HP-UX Tree Structure for Major Directories

Directory	Description
/bin	Contains frequently used commands.
/dev	contains special device files used to communicate to peripherals. For more information, refer to *mknod(1M)*.
/disc	empty directory left by Install. This is a common place to mount other file systems.
/etc	all system administrative commands and configuration files reside here.
/etc/newconfig	new versions of customizable configuration files and shell scripts are stored here following an update. You should keep these files intact here for future reference.
/etc/newconfig/Update_info	information on your release of HP-UX and optional software products.
/etc/conf	kernel configuration files.
/etc/filesets	contains a list of all filesets loaded onto your system. *Do not remove anything from this directory.*
/lib	frequently used object code libraries and related utilities are placed in this directory.
/system	contains revision lists and customize scripts from updates, installs, and the `reconfig` program. *Do not remove anything from this directory.*
/tmp	a place to put temporary files (those normally with short lifetimes that may safely be removed after a few days).
/users	user home directories go below this directory.

Directory	Description
/usr	less frequently used commands and other miscellaneous files are stored under this directory.
/usr/adm	system administrative data files reside here.
/usr/bin	less frequently used commands and those not required to boot, restore, recover, and/or repair the system go here.
/usr/contrib	contains any contributed files and commands (from user groups).
/usr/contrib/bin	any contributed commands are placed here.
/usr/contrib/lib	any contributed object libraries are placed here.
/usr/contrib/man	the on-line documentation for any contributed files, is placed in this directory.
/usr/include	high-level C language header files (shared definitions).
/usr/include/local	localized (site-specific) C language header files.
/usr/include/sys	low-level (kernel-related) C language header files.
/usr/lib	less frequently used object code libraries, related utilities, lp commands, and miscellaneous data files go here.
/usr/local	localized (site-specific) files should be placed here.
/usr/local/bin	localized (site-specific) commands should go here.
/usr/local/lib	localized (site-specific) object code libraries are placed here.
/usr/local/man	put any on-line manual pages for localized (site-specific) systems in this directory.

Directory	Description
/lost+found	where `fsck` puts orphaned files and directories. This directory is automatically created by `newfs` when you create a file system.
/usr/mail	where your mail box resides.
/usr/man	all on-line documentation shipped with your system can be found here.
/usr/man/cat1 ... cat9	*man(1)* pages already processed to speed access go here.
/usr/man/cat1.Z ... cat9.Z	compressed version of the `cat` directories.
/usr/man/man1 ... man9	the unformatted version of *man(1)* pages.
/usr/man/man1.Z ... man9.Z	compressed version of the unformatted *man* pages.
/usr/spool	spooled (queued) files for various programs.
/usr/spool/cron	spooled jobs for `cron` and `at`.
/usr/spool/lp	control and working files for the lp spooler go here.
/usr/spool/uucp	queued work files, lock files, log files, status files, and other files for `uucp`.
/usr/spool/uucppublic	used for free access of files to other systems via `uucp` or LAN.
/usr/tmp	an alternative place (to `/tmp`) in which to place temporary files; this directory is usually used when there are many files and/or the temporary files may be very large.

Processes

A process is an environment in which a program executes. It includes the program's code and data, the status of open files, the value of all variables, and the current working directory. Each process is associated with a unique integer value (called the process ID) which is used to identify the process.

Process Creation (Parent and Child Processes)

A process consists of a single executing program at any given time. However, a process can create another process to:

- concurrently execute another program
- execute another program and wait for its completion

A new process is created when a program executes either the `fork` or the `vfork` system call. The terms **parent process** and **child process** refer to the original process and the process which it created, respectively.

The following sections explain the use of `fork`, `exec`, and `vfork` system calls you initiate from your program. They are also documented in section 2 of the *HP-UX Reference*.

Using fork

When a child is created with a `fork` system call, nearly all code and data is copied from the parent to the child. Only shared code and shared memory segments are not copied (the child process uses the same shared code as the parent process instead of creating a separate copy for itself). Thus, the child process is nearly identical to the parent process (with the exception of its process ID); it has exact copies of the parent's code, data and current variable values.

When the `fork` system call is executed, the system must have enough free swap space to duplicate the parent process or the call to `fork` fails. Once the child process is created, both processes begin execution from the completion of the call to `fork` (at the program statement immediately following the call to `fork`).

The `fork` system call returns the actual process ID of the child (a non-zero value) to the parent process, while the identical call in the child's copy of the code always returns zero. Since the process IDs returned by the `fork` system calls are distinguishable, each process can determine whether it is the parent process or the child process.

For example, suppose that a process consists of a program that tests the life of car batteries. The program has read 1000 data values from a voltmeter and is ready to print and plot the data. The program could have been written to do one task completely (such as printing the data) and then perform the other task. However, the programmer has included a `fork` system call in his program at a location after the data has been read.

When the program completes the statement containing the `fork` system call, two nearly identical processes exist. Each process examines the value returned by its `fork` system call to determine whether it is the child process or the parent process. Following the `fork` statement is a conditional branch statement that states: "If the process is the child process, it should print the data. If the process is the parent process, it should plot the data." Because of the inclusion of the `fork` statements and the conditional branch statement, both printing and plotting are done simultaneously. And because each process has its own copy of the test data, each can modify the data without affecting the other process.

Using exec

One modification which often follows the `fork` system call is to `exec` to another program. `exec` is a system call which overlays separate code and data on top of already existing process code and data. In this manner a parent process is able to create a new process using `fork`, and subsequently execute an entirely different program via `exec`.

As an example, let's suppose we are writing a text editor. We would like to let the user of our editor pause and list directories on the system—say before choosing a file to edit. One way of doing this would be to `fork` a different process, and then immediately `exec` the program `ls`. Let's look next at the `vfork` system call for a more efficient way of doing this.

Using vfork

Copying a parent process's code and data to a child process can be time consuming when a large program or a large amount of data is involved. The `vfork` system call provides an alternate way to create a new process in situations where generating a separate copy of the parent process's code and data is not necessary. `vfork` differs from `fork` in that the child process borrows the parent process's memory and thread of control until the child executes either an `exec` or `exit` system call, or it terminates abnormally. The parent process is suspended while the child uses its resources.

In situations where the child process is simply going to call `exec`, the parent's code and data is not required by the child. If `fork` is used to create the child process, time is wasted copying the unneeded code and data. Depending on the size of the parent's code and data space, using `vfork` instead of `fork` can result in a significant performance improvement.

Like `fork`, `vfork` returns the actual process ID of the child process to the parent process and returns a zero to the child.

Process Termination

A process terminates when:

- The program that is executing in the process successfully completes.
- The process intentionally terminates itself by calling the `exit` or `_exit` system call.
- The process receives, from any process, a signal for which the default action is taken (if the default action is fatal).

When a process "dies" (terminates), all open files associated with the process are closed. System resources associated with the process are de-allocated.

Process Groups

A **process group** is a set of related processes, such as a parent process, its child processes and its children's child processes.

A process group is established when a process calls the `setpgrp` system call. The calling process becomes the **process group leader;** it and all of its future descendants (such as its child processes and grandchild processes) are members of only that process group. Process group membership is inherited by a child process. Descendants already in existence are not placed in the new process group. Each active member of the process group is identified by the process ID of the process group leader. The `init` process is the parent process of all processes. It initially sets up process groups as it executes commands from the command field of `/etc/inittab`.

A signal sent to a process may also be sent to all other members of its process group. Typically, process groups are used to ensure that when an affiliated process group leader terminates, all members of its process group also terminate.

Terminal Affiliation

Process groups and process group leaders have significance in that a process group leader can become "affiliated" with a terminal. All standard input, standard output, and standard error generated by process group members is, by default, directed to the affiliated terminal (unless redirected). Affiliation is caused by an unaffiliated process group leader opening an unaffiliated terminal. Only a process group leader can become affiliated. At the time of affiliation, the process group leader cannot be affiliated with any other terminal and the terminal cannot be affiliated with any other process group. The terminal sends signals to the members of its affiliated process group in response to the interrupt character ([DEL]), QUIT (as set by the `stty` command, by default [CTRL] [|]), the [Break] key, or a modem hangup signal.

A child process inherits terminal affiliation when it is created. Thus, if an unaffiliated process group leader creates a child process, the child process is unaffiliated, even if the parent process becomes affiliated later.

Open Files in a Process

For a process to access files, it must first open them. HP-UX limits the number of files that one process can have open to 60. A process inherits all open files from the parent. Three files that are usually open are: **standard input** ('stdin''), **standard output** (`stdout`), and **standard error** (`stderr`). When a process terminates, the system closes any files that this process has open.

IDs

As previously mentioned, each process is assigned a process ID (a unique integer value) which identifies that process. The process also has associated with it a **real user ID,** a **real group ID,** an **effective user ID,** and an **effective group ID.**

A **real user ID** is an integer value which identifies the owner of the process. Similarly, a **real group ID** is an integer value which identifies the group to which the user belongs. The real group ID is a unique integer identifier that is shared by all members of a group. It is used to enable members of the same group to share files without allowing access to these files by non-group members. The real user ID and real group ID are specified by the file `/etc/passwd` and are assigned to the user at login. You can read the `/etc/passwd` file either from the shell (using `cat`, `grep`, or an editor) or from your program (using the `getpwent` call). Refer to the entries for *cat(1), grep(1), getpwent(3)* and your editor in the *HP-UX Reference* for more information.

Effective user and group IDs allow the process executing a program to appear to be the program's owner for the duration of its execution. The effective user ID and group ID are separate entities and can be set individually. The effective IDs are usually identical to the user's corresponding real IDs. However, a program can be protected such that when executed, the process's effective IDs are set equal to the real IDs of the program's owner. The new effective ID values remain in effect until:

- The process terminates.
- The effective IDs are reset by an "overlaying" process (a process is "overlaid" via the `exec` system call).
- The effective IDs are reset by a call to the `setuid` system call or the `setgid` system call. These calls are both described in the *setuid(2)* entry in the *HP-UX Reference.*

The primary use of effective IDs is to allow a user to access/modify a data file and/or execute a program in a limited manner. When the effective user ID is zero, the user is allowed to execute system calls as the superuser (described in the following section).

For example, suppose that the dean of a university keeps all of his student's records in a file on the system. He wishes to enable a professor to modify a student's record only for that professor's class (an English professor shouldn't be allowed to modify a student's grade in physics). The dean first enables the file containing the student's records such that only he may read or write to it. He then writes a program which receives the modifications requested by a user, checks to see that the user is allowed to make such changes, and then modifies the record if allowed. Finally, the dean protects the program such that the effective IDs of the user are set equal to the dean's real IDs when the program is executed. Then when the program accesses the student record file, the system allows the program to read from or write to the file because it believes that the dean is accessing the file (the effective user and group IDs are that of the dean).

Each process also has a **group access list** associated with it. A group access list is a list of up to 20 groups to which the process belongs. A process is permitted to access the files of any group in this list as though that group was the process's effective group ID. The access list is assigned at login based on the group memberships specified in the file `/etc/logingroup`.

Security If you have converted to a trusted system you also have an audit ID associated with each user. This audit ID is not changed, even when executing programs that use a different effective user ID. Refer to Chapter 8 for more information.

The Superuser

The term superuser describes the system users whose effective user ID equals 0. Users with effective user IDs equal to 0 are provided with special capabilities by HP-UX (hence the name "superuser"). Many commands and system calls can be accessed only by a superuser. Other commands and system calls provide additional features that can be accessed only by a superuser. A superuser is granted the ability to:

- execute any command in the system, as long as *any* execute permission bit is set in the command file's mode
- override any protections placed on user files
- modify any system configuration files
- add (and remove) users to the system
- perform other system functions

Some superuser commands and some system calls (those used heavily by the system administrator) require the user's name to be `root` and real user ID to be zero. You should maintain a superuser on the system whose user name is `root` and whose real user ID is zero. (This user is often referred to as "the root user".) Log in as this user when acting as system administrator, and use it *only* when necessary. To prevent other users from accessing superuser capabilities, assign a password to `root`. Only you and the "back-up" system administrator(s) should know this password.

Commands that can damage the system are restricted to the superuser. You may have users that need to use some of these commands to perform their work. While it is dangerous to allow users full use of superuser commands, the **privileged group** feature of HP-UX allows you to assign a subset of privileged commands to groups of users. All user processes whose effective group ID matches the ID of the privileged group, or whose group access list contains the privileged group, have access to those commands.

For example, someone using the `rtprio` command can demand prime CPU time. Or, someone who has unrestricted use of the `chown` or `chgrp` commands can defeat the accounting processes. Refer to the *setprivgrp(1M)* entry in the *HP-UX Reference* for a list of privileges which can be assigned to privileged groups.

File System Implementation

Series 300 uses a file system called the High Performance File System (HFS). the UNIX tutorial supplied with your HP-UX system discusses the structure of the file system at the user level and introduces some basic concepts and terms. This section expands on those concepts and introduces new concepts which are unique to HFS. This information is useful when verifying, maintaining, and repairing the HFS file system(s). For details on how to create your file system, refer to the section, "Creating a New File System", in Chapter 4 of this manual.

The files of the HFS file system are stored on a formatted mass storage medium, usually a disk. A file is specified by the user with a path name. The method in which files are stored in HFS is explained in this chapter.

Disk Layout

Each hard disk drive used for the file system begins with an 8 Kbyte volume header area. The rest of the disk holds the file system and swap area. Each file system begins with the primary copy of the superblock and consists of one or more cylinder groups (see Figure 2–2). If you have a hard disk that supports "hard" partitions, such as the HP 9133H, then you can address each partition separately (the volume field in the minor address indicates the partition).

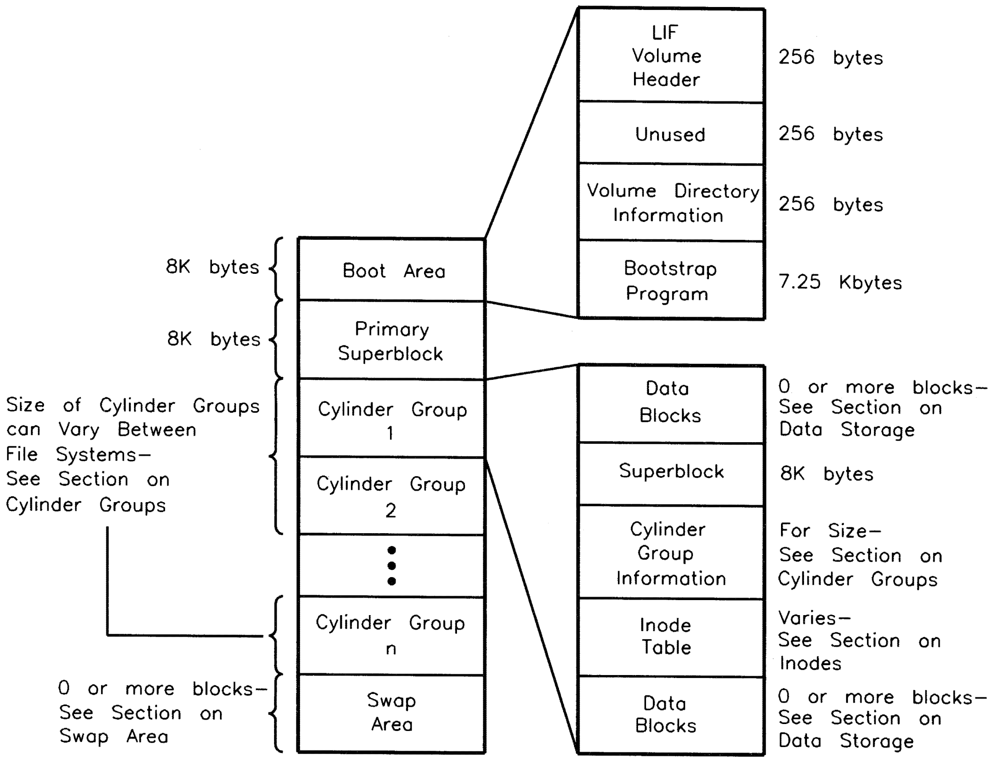


Figure 2–2. File System Layout on Disk

The Boot Area

The **boot area** is reserved on the mass storage medium (usually a disk) during the installation process. Information in the boot area is used only if the disk is used for booting (boot disk), but the space is reserved on all disks. The boot area resides on the first 8 Kbytes of the disk, and contains a volume header, volume directory information, and a small secondary loader used when the system is loaded. This area is reserved exclusively for use by the boot ROM.

If you created your file system using the `newfs` command the boot area will always be on your disk. If you created your file system using `mkfs` the boot area will not be there. In this case, if you will use the file system to boot your system, you must explicitly put the boot area on your disk using the following command (replace `0s0` for you disk's actual device file name):

```
dd if=/etc/boot of=/dev/dsk/0s0 count=1 bs=8k
```

Each boot disk must have a **volume header** in the boot area to identify the volume format. On the Series 300 the format is Hewlett-Packard's LIF (Logical Interchange Format). The volume header is checked by the boot ROM in its examination of bootable mass storage media when the computer is powered up.

The **volume directory information** contains 3 names: SYSHPUX, SYSDEBUG, and SYSBCKUP. SYSHPUX corresponds to the file `/hp-ux`, which is your kernel. SYSDEBUG corresponds to the file `/SYSDEBUG`. This file is used only when writing device drivers; its use is described in the device driver writing manual, *Series 300 HP-UX Driver Development Guide*. SYSBCKUP corresponds to `/SYSBCKUP`, which is used as a backup kernel. All three are assumed by HP-UX to be object files.

The last 7 1/4 Kbytes on the boot area contain the **secondary loader.** The boot ROM loads and passes control to the secondary loader which in turn loads and passes control to the file `/hp-ux` (or the backup kernel if you are using SYSBCKUP). `/hp-ux` (or the backup kernel) then completes the task of bringing up HP-UX.

If you are on a diskless cnode in an HP-UX cluster this process is slightly different. The boot ROM loads and communicates across LAN with the root server's `/etc/rbootd` program. It is the `rbootd` program that coordinates between the diskless cnode, secondary loader, and eventually the chosen kernel.

The Cylinder Group

Each cylinder group contains a copy of the superblock, a cylinder group information structure, an inode table, and data blocks. The superblock is located in each cylinder group so that any single track, cylinder, or platter can be lost without losing all copies of the superblock. If a superblock is lost, the file system can be repaired by using `fsck` with an alternate superblock. If major reconstruction is necessary use `fsdb` (with caution). Any extra space before or after the superblock, cylinder group information, and inode table is filled with data blocks.

There is a primary superblock at the beginning of the file system, and a copy of the superblock in each cylinder group. The superblock contains static information known at file system creation: block size, fragment size, and disk characteristics. The primary superblock also keeps track of file system update information in its summary information area.

Eight Kbytes are reserved for each copy of the superblock. The layout of the superblock data structure is defined in `/usr/include/sys/fs.h`.

The **cylinder group information** contains the dynamic parameters of the cylinder group:

- number of inodes and data blocks
- pointers to the last used block, fragment, and inode
- number of available fragments
- used inode map
- free block map

A bit map in the cylinder group information keeps track of available data blocks and fragments. Data blocks can be divided into 1 Kbyte, 2 Kbyte, or 4 Kbyte fragments. Data block and fragment allocation are described in the section "Data Storage" later in this chapter.

The cylinder group information data structure's size is between 1 fragment and 1 block (a block can be either 4 Kbytes or 8 Kbytes). The size depends on the number of data blocks per cylinder group. The layout of the cylinder group information is defined in `/usr/include/sys/fs.h`.

The **inode table** contains per-file information (see Figure 2–3). A static number of inodes is allocated for each cylinder group when the file system is created. HFS uses a default such that there are more inodes per cylinder group than will be needed for average usage. Refer to `/usr/include/sys/inode.h` for more details on the inode structure.

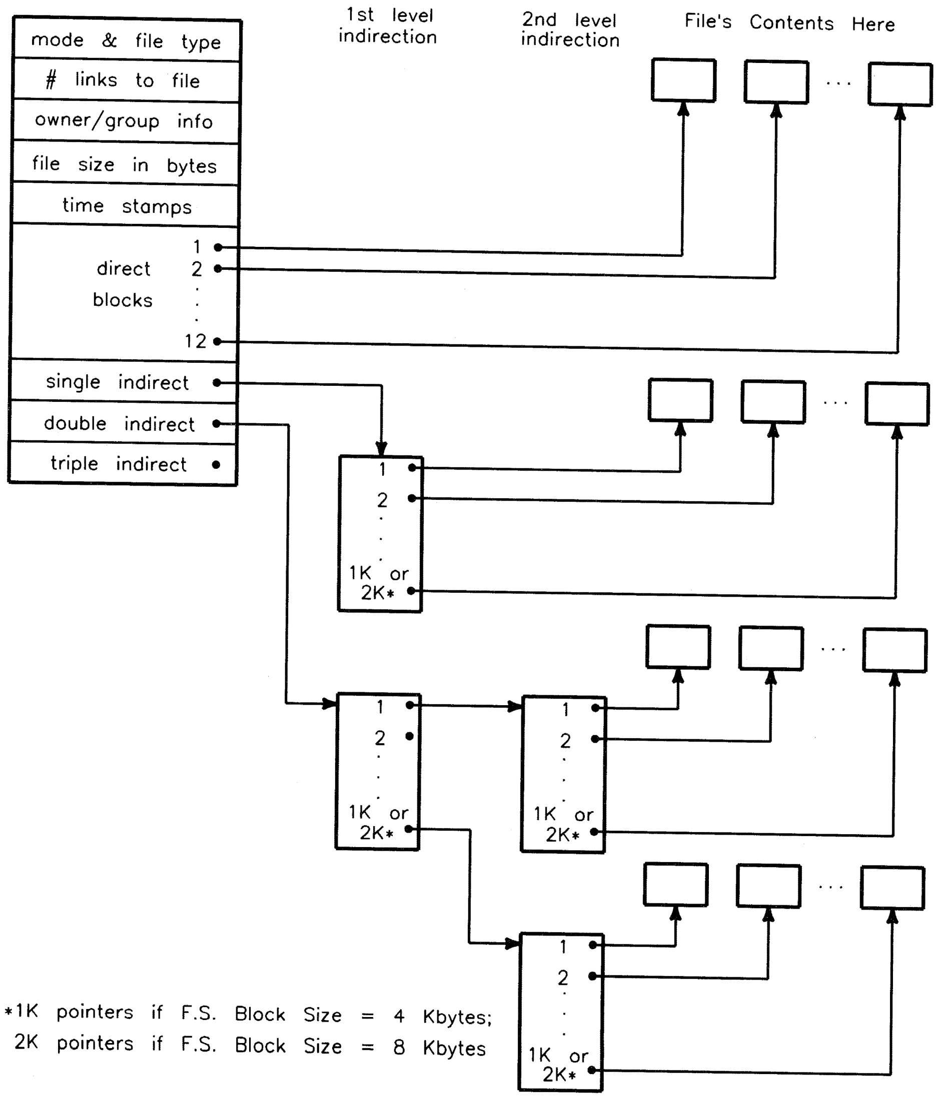


Figure 2–3. Regular File Mapping Scheme and the Inode Structure

A file system uses blocks of either 4 Kbytes or 8 Kbytes: for the rest of the discussion on inode pointers the size of blocks will be referred to as **fs_bsize.** You can replace the variable with 4K or 8K, depending upon what block size your file system uses.

The first 12 pointers in an inode point directly to the first 12 blocks or fragments containing the file's data. If the file is larger than 12 blocks (greater than 12 $\times$ *fs_bsize*), indirect reference is made to the file's data. A group of indirect pointers is contained in one data block. Each pointer is 4 bytes long, so there can be either 1024 pointers (4096/4) or 2048 pointers (8192/4) in each block of indirect pointers.

The 13th block address points to a block containing 1024 or 2048 additional pointers to data blocks (from now on the number of indirect pointers in a block will be called *num_ip*). Thus, the 13th (single indirect) block address handles files up to 4 243 456 bytes in a 4 Kbyte block file system or 16 875 520 bytes in an 8 Kbyte block file system ($fs\_bsize \times (12 + num\_ip)$) . If the file is larger than this, the 14th inode block address points to *num_ip* indirect blocks, each of which contains pointers to an additional *num_ip* actual data blocks.

If the file cannot be contained in this space, the 15th inode block address points to *num_ip* double-indirect blocks. With the 15th (triple-indirect) block address, the size of a file is limited to $fs\_bsize \times (12 + num\_ip + num\_ip^2 + num\_ip^3)$. Your disk drive probably doesn't have this much space and files cannot cross disk drive boundaries.

Inode pointers hold the address of a fragment. The address can be interpreted as referencing a whole block or as referencing one or more fragments, depending on the number of bytes stored at the address. Whether a block or a fragment is used depends on the following information in the inode: the file size, file system block size, and the pointer's index number. A partial block (one or more fragments) will be allocated only at the end of a file, so if there are three pointers to data, pointers 1 and 2 will point to full blocks, but pointer 3 may point to a partial block.

Figure 2–4 shows an example of a 20 Kbyte file stored in 8 Kbyte blocks with 1 Kbyte fragments. The number of blocks needed is 20 ÷ 8 (file size ÷ block size): 2 full blocks with a remainder of 4 fragments. Therefore, the first and second pointers point to full blocks, but the third pointer points to the remaining 4 fragments.

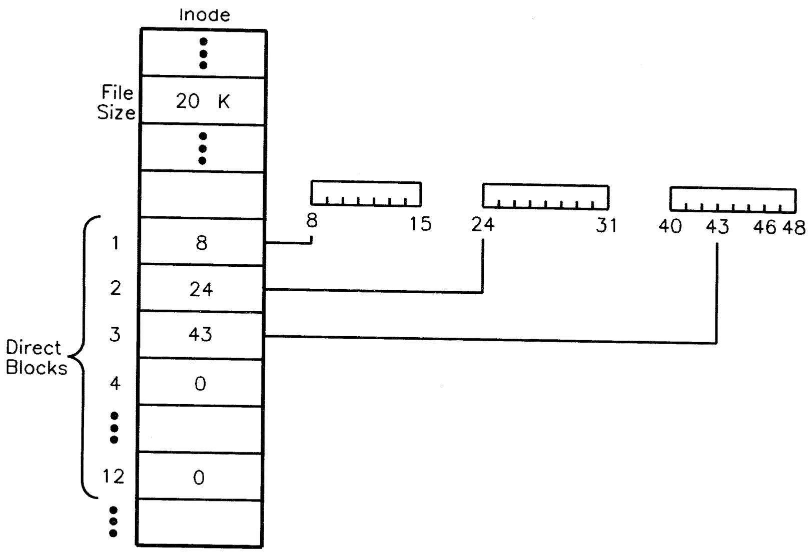


Figure 2–4. Inode Addressing Example

All indirect blocks are referenced only as full blocks; no pieces of the file are addressed at the fragment level beyond the 12 direct pointers.

If the file described by the inode is not a regular file, then some fields of the inode are interpreted differently (see Figure 2–3 for the regular file mapping scheme). The differences are:

- FIFO and pipes

  The space reserved for indirect block pointers contains information about the current state of a FIFO or pipe.

- Character or block device files

  The first direct block address is actually the major and minor number of the device. The rest of the direct block addresses are 0.

- Directory

  The pointers point to regular file system data blocks, but the blocks contain specifically formatted data. A description of the data is in the file `/usr/include/sys/dir.h`.

The inode table's size can vary between file systems. To determine the amount of space used by the inode table you need the following information: number of bytes per cylinder group, average number of data bytes per inode refer to "Creating a New File System" in Chapter 4 for information on how this is determined), inode size (always 128), and block size. For example, in a file system with 8 Kbyte blocks, 2 Mbyte cylinder groups, and 2048 data bytes per inode, there are 1000 inodes per cylinder group (2 Mbytes ÷ 2048 bytes). The 1000 inodes × 128 bytes per inode gives 128 000 bytes for the entire inode table. 128 000 ÷ 8192 (block size) gives 15.625 blocks needed for the table. Since a partial block will not be allocated for the inode table, the system rounds up to 16 blocks and "inode fills" the 16th block: an additional 24 inodes were added to fill the last block so no space is wasted.

Data Storage

In each cylinder group, the areas before and after the superblock, cylinder group information, and inode table contain the blocks used to store data for regular files, directories, pipes, symbolic links and FIFOs (see Figure 2–2). Indirect blocks filled with pointers to data blocks also reside in this part of the cylinder group.

Free space is allocated primarily in block sizes. Blocks can be either 4 Kbytes or 8 Kbytes. Block size is set at file system creation.

Having a large block size has both benefits and costs. In big files, a large block size significantly reduces the number of disk accesses, thereby increasing file system throughput. The problem is that most HP-UX files are small; using a large block size for small files creates wasted space. To circumvent the wasted space problem, a block can be divided into either 1 Kbyte, 2 Kbyte, or 4 Kbyte fragments.

Fragment size is bounded on the lower end by 1024 and on the upper end by the block size, and must be an even multiple of 1024. Fragment size is specified at file system creation.

Allocation of Disk Space

Free space availability is determined from a bit map associated with each cylinder group. The bit map contains one bit for each fragment. To determine if a block is available, consecutive fragments are examined. A piece of the bit map from a file system using 1024 byte fragments and 8192 byte blocks is shown in Table 2–1.

Table 2–1. Example Free Block Bitmap in an 8192/1024 File System

bit map	00000000	00000011	11111100	11111111
Fragment numbers	0-7	8-15	16-23	24-31
Block numbers	0	1	2	3

The free fragments in this example are fragment numbers 14-21 and 24-31, indicated by *1*s in the bit map. The allocated fragments are fragment numbers 0-13 and 22-23, indicated by *0*s in the bit map. Fragments in adjacent blocks cannot be used to create a full block; only 8 contiguous fragments starting on a block boundary can be used to allocate a full block. In this example, fragments 24-31 can be coalesced to form a full block, but fragments 14-21 cannot be. Also, if a partial block is allocated, the fragments must be consecutive and not cross a block boundary. For example, if three fragments are needed, fragments 16-18 can be allocated, but fragments 14-16 cannot be.

In an already existing file, each time additional data needs to be written to the file, the system checks to see if file size must increase. If file size must increase, one of three conditions exists:

1. There is enough space in the existing block or fragment. In this case the new data is written into the already allocated space.
2. The file contains only whole blocks, and there is not enough room in the last block to hold the additional data. If more than a full block of data needs to be written a new block is allocated and the first additional block of data is written there. This process is repeated until less than a block of new data needs to be written. When this happens, a block containing enough contiguous fragments is located and the new data is written there.

3. The file contains fragments, but not enough fragments to hold the new data. If the size of the existing data in fragments, plus the new data, exceeds the size of a full block, a new block is allocated. Both the old and the new data are written to the new block following the process in condition 2 above. If the size of the old and new data is less than a full block, a block with enough contiguous fragments (or a full block) is located and allocated.

When a block or fragment has been located, the address is recorded in the inode table and the free block bit map is updated.

A certain percentage of free space must always be available in the file system. This minimum free space percentage is specified at file system creation using the `-m` option of the `newfs` command or the `minfree` argument of the `mkfs` command. The default is 10 percent. The percent of free space may be changed at any time using the `-m` option of the `tunefs` command. The reserved free space is inaccessible to the normal user; once this threshold has been met only the superuser can continue to allocate blocks. When the percentage of free space drops below the threshold, system throughput (to and from newly-created files) usually drops because the file system can no longer localize the blocks for a file. Accessing a file is quicker if the whole file is grouped together (localized).

Allocation Policies

Allocation is performed on both a global level to determine placement of new directories and files, and on a local level to determine the actual placement of data in blocks.

A decision is made at the global level to determine which cylinder group will contain a given file or directory. An attempt is made to put all files from a single directory in the same cylinder group. When a new directory is created, it is put in the cylinder group that has the greatest number of free inodes and the smallest number of directories.

Global policy specifies that once the file size reaches MAXBPG (MAXBPG is defined in `/usr/include/sys/fs.h`), HFS will allocate blocks from another cylinder group. This helps to enforce the grouping of all files within one directory into a single cylinder group by causing the less common larger files to be spread over several cylinder groups.

The global allocation routines call local allocation routines with requests for specific data blocks. The global information, however, isn't always aware of the status of every data block. It is the local allocation routines, therefore, that make the decision of which blocks to allocate. Block(s) are allocated in the following order:

1. Allocate block requested.
2. Allocate a block on the same cylinder that is rotationally closest to the requested block.
3. Allocate any block within the same cylinder group.
4. Use a quadratic hash to find a new cylinder group; allocate a block somewhere in the new cylinder group.
5. Use brute force search to find an available block.

Updating the HFS File System

Every time a file is modified, the HP-UX operating system performs a series of file system updates. These updates are designed to ensure a consistent file system.

When a program does an operation that changes the file system, such as a write, the data to be written is copied into an in-core buffer, called the buffer cache. The physical disk update is handled asynchronously from the buffer write. The data, along with the inode information reflecting the change, is written to the disk sometime later unless the file was opened in the synchronous mode (refer to the description of `O_SYNCIO` in the *open(2)* and *fcntl(2)* entries in the *HP-UX Reference*). The process is allowed to continue even though the data has not yet been written to the disk. If the system is halted without writing the in-core information to disk, the file system on the disk is left in an **inconsistent** state.

Updates occur to the superblock, inodes, data blocks, and cylinder group information in the following ways:

Superblock — The superblock of a mounted file system is written to the disk whenever a `umount` command is issued, or when a `sync` command is issued and the file system has been modified. The root file system is mounted during boot and cannot be unmounted.

Inodes	An inode contains information specific to the file it describes. An inode is written to the file system upon closure of the file associated with the inode, when a `sync` or `fsync` command is issued, when the file system is unmounted, or as soon as the file is written if `O_SYNCIO` is set for the file.
Data blocks (directories, indirect blocks, files, pipes, symbolic links, and FIFOs)	In-core blocks are written to the file system whenever they have been modified and released by the operating system. More precisely, they are buffered or queued for eventual writing. Physical I/O is deferred until the buffer is needed by HP-UX, a `sync` command is issued, an `fsync` is issued for the file, or `O_SYNCIO` is set for the file. If a file is opened with the `O_SYNCIO` flag set, the `write` system call does not return until completed.
Cylinder group information	The cylinder group information is updated whenever a `sync` is executed, or when the system needs a buffer and the cylinder group is written.

Do not `fsck` *mounted file systems,* and always reboot the system after altering the root device with `fsck`. A file system inconsistency can also occur if you execute `fsck` (file system consistency check—described in Appendix A, Volume 2) on a mounted file system other than the root file system. If you perform a file system check on a mounted file system, information could be in the buffer, but not yet written to the file system. A subsequent flushing of the buffer cache could overwrite the corrections which `fsck` has just made.

Corruption of the File System

Although the HFS file system on your Series 300 computer is very reliable, it is possible to become corrupt. The most common ways for the file system to become corrupt are *improper shutdown procedures* and *hardware failures.*

Improper System Shutdown and Startup

File systems may become corrupt when proper shutdown procedures are not observed:

- not using the `reboot` or `shutdown` command to halt the CPU
- physically write-protecting a mounted file system
- taking a mounted file system off-line

File systems may become further corrupted if proper startup procedures are not observed:

- not checking a file system for inconsistencies
- not repairing inconsistencies

Allowing a corrupted file system to be further modified can be disastrous.

Hardware Failure

While your Hewlett-Packard Series 300 computer system and disks are highly reliable, it is good to remember that any piece of hardware can fail at any time. This isn't a prediction of gloom, but merely a word of caution to you, the system administrator, to take small steps of precaution. By following the preventative maintenance outlined in your installation guides and in this manual, you should be able to avert any serious problems. Failures can be as subtle as a bad block on a disk pack, or as blatant as a non-functional disk controller.

Detection and Correction of Corruption

You can check the root file system for structural integrity after performing a system shutdown. You can check non-root file systems any time as long as they are unmounted. The `fsck` command verifies the structural integrity by checking data which is intrinsically redundant in a file system. The redundant data is either read from the file system or computed from known values. A quiescent state is important during the checking of a file system because of the multi-pass nature of the `fsck` program. `init` run-level s (the system administrator run-level) is the only safe state in which to check the root file system.

When an inconsistency is discovered, `fsck` reports the inconsistency. Refer to Appendix A, Volume 2, "Using the fsck Command", for an explanation of the actions `fsck` takes in response to these inconsistencies, based on different run options.

Superblock Consistency

The summary information associated with the superblock may become inconsistent. The summary information is prone to error because every change to the file system's blocks or inodes modifies the summary information.

The superblock and its associated parts are most often corrupted when the computer is halted and the last command involving output to the file system is not a `reboot` or `shutdown` command.

The superblock can be checked for inconsistencies involving:

- File-system size—this rarely happens.
- Free-block count—this is fairly common.
- Free inode count—this is fairly common.

If `fsck` detects corruption in the static parameters of the primary (default) superblock (rarely happens), `fsck` requests the system administrator to specify the location of an alternate superblock. The alternate superblock addresses were listed during file system creation. If the last time you created a file system was during the installation, a list of addresses will be given in a file called `/etc/sbtab`. An alternate superblock will always be found at block number 16. If this superblock is also corrupted, you must supply the address of another superblock.

File System Size. The superblock is checked for inconsistencies involving file system size, number of inodes, free block count, and the free inode count. The file system size must be larger than the number of blocks used by the superblock and the number of blocks used by the list of inodes. The file system size and layout information are critical pieces of information to the `fsck` program. While there is no way to actually check these sizes, `fsck` can check for them being within reasonable bounds. All other checks of the file system depend on the correctness of these sizes.

Free-Block Checking. A check is made to see that all the blocks in the file system were found.

`fsck` checks that all the blocks marked as free in the free-block map are not claimed by any files. When all the blocks have been accounted for, a check is made to see if the number of blocks in the free-block map plus the number of blocks claimed by the inodes equals the total number of blocks in the file system.

If anything is wrong with the free-block maps, `fsck` will rebuild them, excluding all blocks in the list of allocated blocks.

The summary information contains a count of the total number of free blocks within the file system. `fsck` compares this count to the number of blocks it found free within the file system. If they don't agree, `fsck` will replace the count in the summary information by the actual free-block count.

Inode Checking. The summary information contains a count of the total number of free inodes within the file system. `fsck` compares this count to the number of inodes it found free within the file system. If they don't agree, `fsck` will replace the count in the summary information by the actual free inode count.

Inodes

An individual inode is not as likely to be corrupted as the summary information. However, because of the great number of active inodes, a few inodes may become corrupted.

The list of inodes is checked sequentially starting with inode 2 (inode 0 marks unused inodes and inode 1 is reserved for future use) and going to the last inode in the file system.

There are two major types of inodes: primary and continuation. Continuation inodes contain only a mode (which is of type continuation), a link count, and ACL entries. You will have continuation inodes only of the file has optional ACL entries associated with it. `fsck` will check the continuation inode's mode, link count, and the reference from the primary inode. It will not check for consistency in ACL information.

Each primary inode can be checked for the following inconsistencies:

- format and type
- link count
- duplicate blocks

- bad blocks
- inode size
- block count
- bad continuation inode number

Format and Type. Inodes may be one of the following types:

- regular file
- directory
- block device
- character device
- network device
- FIFO
- symbolic link
- continuation

Inodes may be found in one of three states:

- unallocated
- allocated
- neither unallocated nor allocated

This last state indicates an incorrectly formatted inode. An inode can get in this state if bad data is written into the inode list through, for example, a hardware failure. The only possible corrective action is for `fsck` to clear the defective inode.

The following sections describe information that `fsck` checks in primary inodes.

Link Count. Contained in each inode is a count of the total number of directory entries linked to the inode. `fsck` verifies the link count stored in each inode by traversing the total directory structure (starting from the root directory) and calculating an actual link count for each inode.

If the stored link count is non-zero and the actual link count is zero, it means that no directory entry appears for the inode. `fsck` can link the disconnected file to the `/lost+found` directory. If the stored and actual link counts are non-zero and unequal, a directory entry may have been added or removed without the inode being updated. `fsck` can replace the stored link count by the actual link count.

Duplicate Blocks. Contained in each inode is a list, or for large files, pointers to lists (indirect blocks), of all the blocks claimed by the inode. `fsck` compares each block number claimed by an inode to a list of already allocated blocks. If a block number is already claimed by another inode, the block number is added to a list of duplicate blocks. Otherwise, the list of allocated blocks is updated to include the block number. If there are any duplicate blocks, `fsck` will make a partial second pass of the inode list to find the inode of the duplicated block.

This condition can occur by using a file system with blocks claimed by both the free-block list and by other parts of the file system.

`fsck` will prompt the operator to clear both inodes. Often clearing only one inode will solve the problem, but the data in the other inode is suspect.

Bad Blocks. Contained in each inode is a list or pointer to lists of all the blocks claimed by the inode. `fsck` checks each block number claimed by an inode for a value outside the range of the file system (lower than that of the first data block, or greater than the last block in the file system). If the block number is outside this range, the block number is a bad block number.

`fsck` will prompt the operator to clear the inode.

Inode Size. Each inode contains a sixty-four bit (eight-byte) size field. This size indicates the number of characters in the file associated with the inode. Inode size can be checked for inconsistencies (for example, directory sizes in a file system using standard, 14-character, filename limits should be a multiple of thirty-two characters, and the number of blocks actually used should match that indicated by the inode size).

A directory inode within the HP-UX file system has the mode word set to `directory`. The directory size in a file system using standard, 14-character, filename limits must be a multiple of thirty-two because a directory entry contains thirty-two bytes of information. `fsck` will warn of such directory misalignment. This is only a warning because not enough information can be gathered to correct the misalignment.

A rough check of the consistency of the size field of any inode can be performed by computing from the size field the number of blocks that should be associated with the inode and comparing it to the actual number of blocks claimed by the inode. `fsck` calculates the number of blocks that should be claimed by an inode by dividing the number of characters in the file by the number of characters per block and rounding up. `fsck` then counts actual direct and indirect blocks associated with the inode. If the actual number of blocks does not match the computed number of blocks, `fsck` will warn of a possible file-size error. This is only a warning because HP-UX does not fill in blocks in sparse data files.

Indirect Blocks. Indirect blocks are owned by an inode. Therefore, inconsistencies in indirect blocks directly affect the inode that owns it. Inconsistencies that can be checked are:

- blocks already claimed by another inode
- block numbers outside the range of the file system

Detection and correction of the inconsistencies associated with indirect blocks follows the same scheme used for direct blocks, and is done iteratively. Direct and indirect blocks were discussed in the section "The Cylinder Group".

Data Blocks. The two types of data blocks are:

- Ordinary data blocks which contain the information stored in a file. `fsck` does not attempt to check the validity of the contents of an ordinary data block.
- Directory data blocks which contain directory entries.

Each directory data block can be checked for inconsistencies involving:

- directory inode numbers pointing to unallocated inodes
- directory inode numbers greater than the number of inodes in the file system
- incorrect directory inode numbers for "." and ".." (current and parent directories, respectively)
- directories which are disconnected from the file system hierarchy

To remove files with illegal characters, find out their inode number, then remove them by typing in the following sequence:

```
ls -i
find . -inum inode_number -exec rm {}\;
```

If a directory entry inode number points to an unallocated inode, then `fsck` can remove that directory entry.

If a directory entry inode number is pointing beyond the end of the inode list, `fsck` can remove that directory entry. This condition occurs if bad data is written into a directory data block.

The directory inode number entry for "." should be the first entry in the directory data block. Its value should be equal to the inode number for the directory data block.

The directory inode number entry for ".." should be the second entry in the directory data block. Its value should be equal to the inode number for the parent of the directory entry (or the inode number of the directory data block if the directory is the root directory).

If the directory inode numbers for "." and ".." are incorrect, `fsck` can replace them by the correct values.

`fsck` checks the general connectivity of the file system. If directories are found not to be linked into the file system, `fsck` will link the directory back into the file system in the `/lost+found` directory.

Uncorrectable File System Corruption

`fsck` may not be able to proceed in certain instances, such as if all copies of the superblock are lost. The `fsdb` (file system debugger) command is provided for such situations. `fsdb` should only be used by an HP-UX file system expert, since it can easily destroy the entire file system. Refer to the *fsdb(1M)* entry in the *HP-UX Reference* for details.

File Format and Compatibility

The format of the mass storage media on which Series 300 HP-UX files are stored is High performance File System (HFS). This is not necessarily the format for other operating systems patterned after the UNIX operating systems. For example, the Series 200 HP-UX 2.x uses the Bell System III file system (BFS), and the Series 500 HP-UX system uses Structured Directory Format (SDF). The following list provides guidelines on how to transfer files between your systems:

- Transfer files between two UNIX machines
  - `cpio`
  - `tar`
  - `uucp`
  - LIF utilities: `lifcp` and `lifinit` (HP-UX to HP-UX only)
  - mounted file system (same system, same version, only)
  - LAN
  - SRM

- Transfer files between Basic or Pascal and HP-UX on Hewlett-Packard machines
  - LIF utilities
  - terminal emulator running on the workstation, uploading or downloading files
  - SRM
- Transfer files between the current version of HP-UX and either the Integral Personal Computer or a 2.x version of HP-UX on the Series 200.
  - BIF utilities
  - terminal emulator running on the workstation, uploading or downloading files
- Transfer files between a Series 300 HP-UX machine and a Series 500 HP-UX machine
  - SDF utilities
  - terminal emulator running on the workstation, uploading or downloading files
  - all the methods of transfer between two UNIX machines shown above

File Protection

When each file in the file system is created, it is assigned a set of file protections stored in the file permissions bits (often called the file's mode). The file permission bits determine which classes of users may read from the file, write to the file, or execute the program stored in the file. Read, write, and execute permissions for a file can be set for the file's owner, all members of the file's group (other than the file's owner), and all other system users. These three classes of users (user, group, and other) are mutually exclusive—no member of one class of users is included in any other class of users. When a file is created, it is associated with an owner and a group ID. These values specify which user owns the file and which group has special access capability.

The default permissions of a file are initially determined by `umask`, or by parameters passed to `creat`, `mknod`, or `mkdir`, when the file is created. The permissions may be changed with the `chmod` command. The permissions of the file are represented as the binary form of four octal digits as shown in Figure 2–5. The initial discussion deals with only the three least significant digits. When the most significant digit is not specified, its value is assumed to be zero (0).

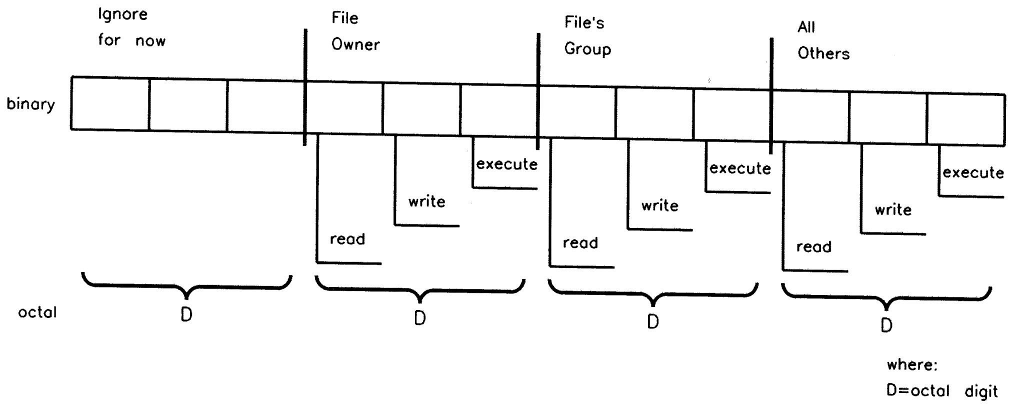


Figure 2-5. File Permission Bits

Each octal digit represents a three-bit binary value: one bit specifies read permission, one bit specifies write permission, and one bit specifies execute permission. If the bit value is one, then permission is granted for the associated operation. Similarly, if the bit value is zero, permission is denied for the associated operation.

For example, assume a file's permission bits are set to 754 (octal). Octal 754 is equivalent to 111 101 100 binary. Using Figure 2-6, you can see that this grants the owner of the file read, write, and execute permission. A file permission of 754 grants read and execute permission to all users who are members of the file's group. This includes any user (except the file's owner) whose effective group ID is equal to the ID of the file's group, or whose group access list includes the file's group ID. A file permission of 754 grants read permission to all other system users. The `ll` command represents this as `rwxr-xr--`.

Note that if there are optional Access Control List (ACL) entries on the file, a + is printed following the permissions. Also, by default, `chmod` deletes any optional ACL entries. You can use the `-A` option to preserve them. For more information on ACLs, refer to *A Beginner's Guide to HP-UX.*

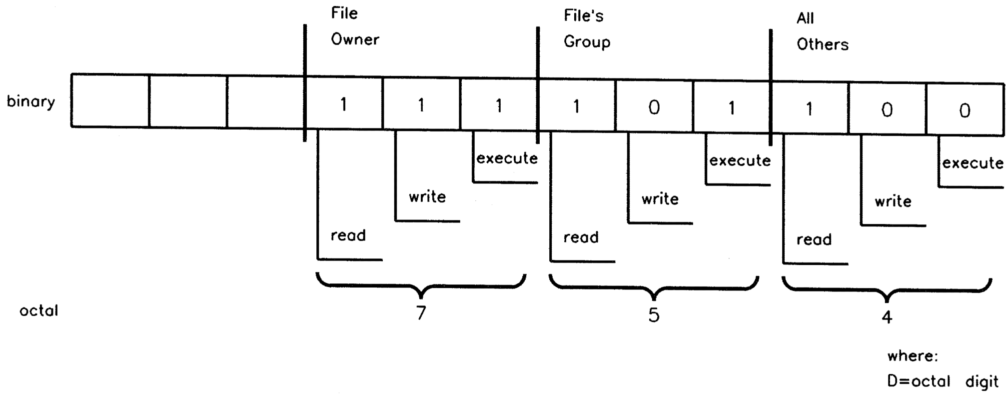


Figure 2-6. File Permission Bits of 0754

You can also use a **symbolic mode** to change permissions with `chmod`. To change protections using the symbolic mode, type:

`chmod` *who operation permission*

where *who*, *operation*, and *permission* can be:

who	`u` (user), `g` (group), or `o` (other)
operation	`+` (add the following permission), `-` (remove the following permission), or `=` (assign *only* the following permission—all other permissions will be taken away)
permission	`r` (read), `w` (write), `x` (execute), `s` (set owner or group ID), or `t` (set sticky bit)

For example, to deny write permission to `/users/bilbo/file1` to the user group, others, type:

```
chmod o-w /users/bilbo/file1
```

To make the same file executable for everyone, type:

```
chmod +x /users/bilbo/file1
```

Protecting Directories

Directories, like all files in the HP-UX file system, have permissions. The format of a directory's permission bits is identical to that of an ordinary file; however, the read, write, and execute permissions have a slightly different meaning when applied to a directory.

- Read permission provides the ability to list the contents of a directory.
- Write permission provides the ability to add a file to the directory, rename a file within the directory, and remove a file from the directory. It does not allow a user to directly write the contents of the directory itself. This capability belongs to the HP-UX system only.
- Execute permission provides the ability to search a directory for a file. If execute permission is not set for a directory, the files below that directory in the file system hierarchy cannot be accessed even if you supply the correct path name for the file.

Setting Effective User and Group IDs

The section "IDs" earlier in this chapter discussed effective user and group IDs. Through the use of user and group IDs, a file can be protected such that, when executed, the process's effective IDs are set equal to the file owner's IDs. This capability is specified through the most significant digit of the four octal file protection digits. (Refer to the previous sections for a discussion of the three least significant digits in the file protection bits.) The most significant digit is represented by a three-bit binary value. When its most significant bit is 1, the effective user ID of the process executing the file is set equal to the user ID of the file's owner. This bit is called the **set user ID bit** (`suid`). Similarly, if the middle bit of the most significant octal digit is 1, then the effective group ID of the process executing the file is set equal to the group ID of the file's group. This bit is called the **set group ID bit** (`sgid`).

If the `sgid` bit is on for an ordinary file, and the file does not have group execute permission, then the file is in enforcement locking mode. Refer to the section "File Sharing and Locking" later in this chapter, or to the *lockf(2)* entry in the *HP-UX Reference*.

If the `suid` bit is on for a directory, the directory is **hidden.** A hidden directory is part of the CDF structure used if you are running an HP-UX cluster. For more information on CDFs, refer to the section in this chapter called "Cluster Concepts".

For example, suppose that the file's permission bits are 6754. The binary equivalent of octal 6754 is 110 111 101 100. The meaning of these permissions is illustrated in Figure 2–7 and explained in Table 2–2.

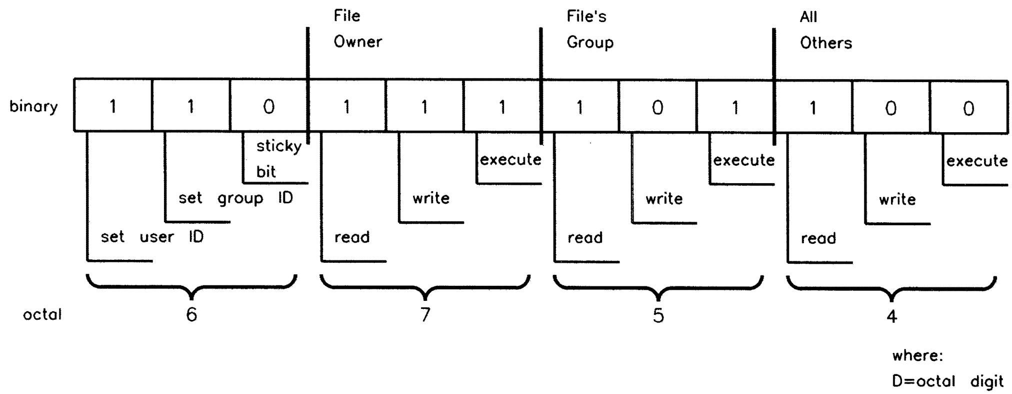


Figure 2–7. File Permission Bits of 6754

Table 2-2. Explanation of File Permission Bits of 6754

Octal Digit	Binary Form	Permission	Meaning
6	1	set user ID	Effective user ID of the process executing this file is set equal to the real user ID of the file's owner.
	1	set group ID	Effective group ID of the process executing this file is set equal to the group ID of the file's group.
	0	sticky bit	The stick bit is discussed in the section that follows.
7	1	read	File owner may read the file.
	1	write	File owner may write to the file.
	1	execute	File owner may execute the file.
5	1	read	Members of the file's group (other than the file's owner) may read the file.
	0	write	Members of the file's group (other than the file's owner) cannot write to the file.
	1	execute	Members of the file's group (other than the file's owner) may execute the file.
4	1	read	Any other user may read the contents of the file.
	0	write	Other users cannot write to the file.
	0	execute	Other users cannot execute the program contained in the file.

The Sticky Bit

Although the sticky bit can be set for all programs, *setting the sticky bit affects a program only if it is shared* (refer to the section on "Memory Management" discussed later in this chapter). The following discussion assumes that all files marked sticky are also shared.

The least significant bit of the upper octal digit is called the **sticky bit.** If the sticky bit is set and the program is executed, the data structures and swap space associated with the shared text is not released when the program terminates. This reduces start-up time if the program is executed again. Once a program is in the swap area (via the sticky bit), it can be removed onlyby changing the file's permissions such that the sticky bit is no longer set, then executing (and terminating) the program again.

Only the superuser can set the sticky bit.

File Sharing and Locking

In a multi-user, multi-tasking environment such as HP-UX, it is often desirable to control interaction with files. Many applications share disk files, and the status of information contained in them could have serious implications to the user (such as lost or inaccurate information).

For example, imagine we are responsible for maintaining on-line technical reports for a myriad of projects, and we have many different people who must have simultaneous access to these reports. The content of a given report at a given time could significantly affect a company decision, and so we want a way to control how records are accessed.

One potential problem could arise if one person (let's call him George) adds to or modifies information in a report while someone else (Sarah) is working on it. Sarah is unaware of changes that George has just made in the report. And once she is done, Sarah overwrites the information George added. The result is that we have lost *all* of George's information, and when Sarah added data she was unaware of information which could have been pertinent.

Advisory Locks

A solution to this common problem of file sharing is called **file locking.** On your Series 300, file locking is done with the `lockf` or `fcntl` system calls, and it handles two modes of functionality. **Advisory locks** are placed on disk resources to inform (warn) other processes desiring to access these same resources that they are currently being accessed or potentially being modified. Advisory locks are only valuable for cooperating processes which are both aware of and use file locking.

In our example, the programs used to access the on-line technical reports could use advisory locks. When George begins to work on the FubNibWitz project his program could call `lockf` and set an advisory lock. A few minutes later when Sarah tries to access records in the FubNibWitz report, she would get an error message informing her that the report is busy. Her program could wait until George is done and then access the report, by virtue of doing a call to `lockf`.

Enforcement Mode

Even if we use advisory locks in our example, Sarah would still be able to overwrite the FubNibWitz report if she uses commands or utilities which don't check for advisory locks. She needs some way to insure that no records are written until George is done with the report. HP-UX does this with **enforcement mode.** When a process attempts to read or write to a locked record in a file opened in enforcement mode, the process will sleep until the record is unlocked. Enforcement mode can only be used on regular files.

Enforcement mode is enabled when the set-group-id bit (`sgid`) is set while not having the group execute bit set. For example, if we opened a file which normally has its file permission bits set to 644, a long listing of the file would look something like:

```
-rw-r--r--   1 George   LubHood      512 May  7 16:11 FubNibWitz
```

Enforcement mode could be turned on by typing the following command:

```
chmod g+s FubNibWitz
```

This command would turn on the `sgid` bit resulting in file protection of 2644. Enforcement mode could also be turned on from a program with the `chmod` system call. After enforcement mode is enabled, a long listing would show:

```
-rw-r-Sr--   1 George   LubHood      512 May  7 16:11 FubNibWitz
```

By now using enforcement mode, George could prevent Sarah from overwriting his changes, and Sarah would have the data which George has added.

Caution It is possible to cause a system deadlock in enforcement mode. By calling the `wait` or `pause` system calls immediately after locking a record, the locking process could hang one or more processes which attempt to access the locked record.

When attempting to access a file which is locked under enforcement mode, your shell sleeps until the file is released. This provides a means for one script to control execution of another, separate, script. Be careful when doing this, because as just noted a system deadlock is possible.

Locking Activities

All file locking is controlled with the `lockf` or `fcntl` system calls. There are essentially four activities which `lockf` controls:

- Testing file accessibility by checking to see if another process is present on a specific file record.
- Attempting to lock a file. If the record is already locked by another process, `lockf` will put the requesting process into a sleep state until the record is free again.
- Testing file accessibility, locking the record if it is free, and returning immediately if it is not.
- Unlocking a record previously locked by the requesting process.

When the locking process either closes the locked file or terminates, all locks placed by that process are removed. For more details on how specific locking activities work on HP-UX, refer to the *lockf(2)* and *fcntl(2)* sections of the *HP-UX Reference* manual.

The File System Buffer Cache

Program code and the data which it uses must be transferred from disk into main memory before it can be executed. The manner in which code is transferred depends on the attributes of the code and the manner in which the code is executed. The file system buffer cache is used for all file system I/O operations, plus all other block I/O operations in the system (for example `exec`, `mount`, inode reading, and some device drivers).

The file system buffer cache is a collection of one or more buffers which the system uses as a temporary holding place for code/data being transferred between the file system and user's main memory. The number of buffers in the cache is determined when you power up your system, and is based upon the amount of available RAM (refer to the `nbuf` entry in Appendix D). As the code and data are moved into the buffer cache, the system copies the information from the buffer cache into user's main memory. If a user requests information that is already in the buffer cache, the information is copied from the cache to user's main memory, eliminating the I/O operation to bring it in from the file system disk.

The primary benefit of the buffer cache is faster transfers of data from the file system to the user address space. Transferring information from the buffer cache to the process's executing space in main memory is much faster than transferring information from the file system on the disk. Thus by increasing the size of the buffer cache, more information can be held in memory and the apparent system response time improves. However, memory used by the system cache is unavailable for other uses, such as executing processes. When the file system buffer cache exceeds a certain size, system performance begins to decrease since less memory is available for other system functions.

A major factor in determining the size of the file system buffer cache is the amount of memory in the system. By default, the system chooses a reasonable buffer cache size based on the available memory in the system at powerup. You can alter the default size by changing the `nbuf` system parameter (refer to "Configuring Operating System Parameters" in Chapter 6).

Magnetic Tape

Since computers are sometimes used to process massive amounts of data, there must be a way to store large files on-line. Applications such as atmospheric studies which, minute by minute, record megabytes of information and then sort it out, require cheap media on which to store data. Even with the advent of larger capacity hard disk drives, they are still too small and far too expensive for such purposes.

Perhaps the closest to an industry standard for mass media, 9-track (1/2 inch) magnetic tape serves as a low cost, high capacity media to store information. And beyond this, magnetic tape is also the most interchangeable media between different hardware and operating systems.

In addition to 9-track tapes, Hewlett-Packard manufactures a series of 1/4-inch data cartridge tapes which are used for the installation and updates of HP-UX on the Series 300. The cartridge tapes can also be used for inexpensive backups. These data cartridges, model HP 88140, have most of the benefits of 9-track magnetic tape but are cheaper and easier to handle. However, they don't offer the same level of data interchange between non-HP-UX machines as the 9-track tapes.

Magnetic Tape Definitions

Here are some common terms and concepts used in the discussion of magnetic tape. Consider them required reading if you use magnetic tape.

Coding

Tape is recorded in several ways. Older systems use **Non Return to Zero Immediate** (NRZI) coding, and record with a tape density of either 200, 556, or 800 bpi (bits per inch). Newer tapes use **Phase Encoding** (PE) and record at 1600 bpi, or they use **Group Coded Recording** (GCR) and record at 6250 bpi. There may be other forms of coding as well, but these are the most common. The HP 7971 supports a density of 1600 bpi, the HP 7974 and HP 7979 support both 1600 bpi and (optionally) 800 bpi, and the HP 7978 and HP 7980 magnetic tape drives support a density of 1600 and 6250 bpi.

The higher the density, the more information can be stored on a tape. On a 2400 foot tape, an HP 7974 at 800 bpi can only store 22 Mbytes of data, at 1600 bpi the HP 7974 can store 43 Mbytes, while an HP 7978 storing at 6250 bpi can write up to 140 Mbytes of data to a tape at a rate of up to 16 Mbytes per minute.

bpi

The most common measure of tape density, **bpi** is an abbreviation for bits per inch.

Cyclic Redundancy Check

When writing a tape, a number of frames are written by the drive in a single transaction. This collection of frames is called a **record.** Part of the record, but invisible to the user, is a **cyclic redundancy check** (CRC). The CRC is recorded as some additional frames on the tape. There is a very short blank section between the true record and the CRC. Following the CRC is a nominal 1/2 -inch gap of unrecorded tape, known as the **inter-record gap** or IRG. The next record follows the gap. If either the frame parity or the CRC is incorrect when the tape drive reads the tape, an error is generated by the drive. Newer formats (1600 bpi and above) generate a preamble and postamble to help synchronize the read logic.

End of Tape

There is both a logical end of tape (EOT) and a physical EOT (see Figure 2–8). Logical EOT is two consecutive file marks. Physical EOT is a foil mark about 25 feet from the end of the reel. Pre-5.0 drivers (Series 200 only) handle physical EOT differently than post 5.0 drivers. (Refer to the discussion on pre-5.0 drivers later in this chapter.)

Note that the distance between the EOT detector and the read/write head may vary among different model tape drives. So, one drive may return an EOT indication associated with the 1000th record on the tape, while another drive may return an EOT indication with the 999th or the 1001st record. For small records this variation may be large; for large records this variation is probably small.

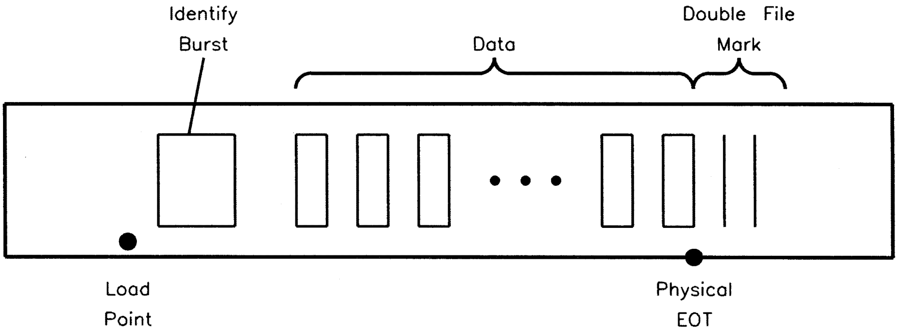


Figure 2-8. Magnetic Tape Format

File Marks

A **file mark** is a special type of record that can be written to the tape. A file mark is recognized by the drive and reported as a boolean condition during reading. It is not possible to write a file mark as ordinary data; it requires a special command to the drive.

Single file marks are used to separate logical files on tape. Two consecutive file marks are used to signify the logical EOT. Data is undefined past the logical EOT.

Foil Mark

A foil mark is a short piece of silver tape that is placed on one edge of the tape on the non-recorded side. Both the load point and the physical end of tape are marked by a foil mark. Both marks are placed by the tape manufacturer.

Load Point

The load point, or beginning of tape, is a foil mark placed about 10 feet from the beginning of a tape. When you load a tape (put the tape in the drive, and press "load"), the drive searches forward until the load point is found and placed under the sensor. The first write is then treated specially: several inches of tape are skipped and then, when using PE or GCR formats, a special burst of data is written to the tape (which is invisible to the user). This is the **identify burst.** Data is recorded after the identify burst in the usual way. The first read expects the identify burst, and quietly skips over it. Some smart drives, such as the HP 7978, can determine the tape density from the identify burst (1600 and up).

Magnetic Tape (Magtape)

Magnetic tape is a media similar to an everyday home cassette tape, used to store digital information. All standard magtape is 1/2 -inch wide, and comes in three sizes: 600, 1200 and 2400 foot reels (for a rule of thumb, a 2400 foot reel is about 1 foot in diameter). The size of the reels, hubs, tape width and other mechanical properties are all specified by ANSI standard.

Operations

Several operations that a tape drive can be expected to perform are to read and write to the media, rewind to the load point, forward or back space one record, and forward or back space to the next file mark. A variation on the theme of rewind is to **unload** where the tape is rewound and taken off line. Some tape drives actually rewind the tape out of the threading path; others simply set an interlock that requires manual intervention to release the tape.

Records

A series of frames written to the media is known as a **record.** The physical record size is variable. The maximum limits on record size range from 16 Kbytes to 60 Kbytes, depending upon the tape drive. Beyond these limit, the drive rejects the request and there are *no* write/read retries. The maximum record sizes are:

HP 7971	1600 bpi—32 Kbytes
HP 7974	1600 bpi—16 Kbytes 800 bpi—16 Kbytes
HP 7978A	1600 bpi—16 Kbytes 6250 bpi—Kbytes
HP 7978B	1600 bpi—32 Kbytes 6250 bpi—60 Kbytes
HP 7980	1600 bpi— 6250 bpi—64 Kbytes

Tape Density

The measure of the amount of information which can be stored in a given area of tape is known as **tape density. Bits per inch** (bpi), a common measure of tape density, is the number of bits per track, recorded per inch on the tape. For

9-track tape, eight data bits and one parity bit are written across the width of the tape simultaneously. Thus for 9-track tape, bpi is synonymous with **characters per inch** (cpi). One of these characters is sometimes called a **frame.**

Tracks

When digital information is written to a tape, it is written in a series of tracks (a lot like an 8-track car stereo). Most magtape today is written in a 9-track format. Older systems often wrote only 6 tracks plus a parity bit, resulting in 7 tracks.

Write/Read Errors

Tape, in its usage for long-term archive and data interchange, is somewhat more prone to error than disks. When your tape drive is reading from, or writing to, a tape and it detects an error, the normal procedure is to backspace the tape over the record and retry the tape operation. An error message is reported to the user only after the driver gives up. Many more tape errors are caused by dirty tape heads than by real recording errors, so you should periodically clean your tape drive as outlined in its service manual.

Tape drives do a form of reading-while-writing, and if the data is not properly recorded, an error will be detected. The normal procedure is to backspace and retry writing the record once, and if that fails, to backspace, write a **long gap** and try again on a section of tape farther down. A **long gap** is several inches of erased tape. That's why we said an IRG is "nominally" 1/2 inch long.

Write Ring

On the back of the reel there is a removable soft plastic **write ring.** Every magtape drive has a sensor mechanism to detect the presence of this ring. When a ring is present the tape can be written to by the host, and cannot be written when absent (it is **write protected**). Normally, once a tape is written, the ring is removed and left out indefinitely except when being rewritten.

Preventive Maintenance

There are several maintenance procedures for tape. A tape can be completely erased (degaussed), or the beginning of the tape can be discarded and a new load point put on (stripped). There is also a tape cleaning and certifying machine that will knock off any loose oxide and check that the tape will record properly over its full length (certified). This always makes any data on the tape unusable. Commercial shops certify their tapes fairly often, and discard them if they get too short or fail to certify. It is also an excellent idea to clean the tape head and guides of your drive periodically as they tend to accumulate loose oxide and other crud.

Tape Streaming

The HP 7971 transfers data to and from your Series 300 with very little buffering between your computer and your drive's read/write head; the drive must stop the tape between records, and wait for the next record. HP 7971 is called a **start/stop** tape drive, and is designed to stop and restart the tape fairly quickly.

The HP 7974, HP 7978, and HP 7980 are **streaming** magnetic tape drives.

A streaming magnetic tape drive is designed to move continuously, reading data from a buffer or writing data to a buffer, not stopping between records like start/stop tape drives do. Streaming increases the rate at which a tape drive can write data onto tape. Before a tape drive can write data onto a tape, the drive read/write head must be positioned at the proper place on the tape, and the tape must move across the head at the proper speed. After writing a record on the tape, if a streaming drive has already received the data for the next record from the computer, it can continue to move the tape across the head without slowing down to write the next record.

If the drive has not received the data for the next record after writing a record on the tape, then the drive must reposition the tape. This involves stopping the forward motion of the tape, backspacing the tape to some point preceding the beginning of the next record to be written, stopping the tape, and waiting for your computer to send the data for the next record. *The average data transfer rate is much higher when the drive streams than when it repositions,* especially for the HP 7978. The HP 7974 supports both a start-stop and a streaming mode. The HP 7978 supports only a streaming mode. Both drives are much faster than the HP 7971 when they stream. When they do not receive data fast enough to stream, the HP 7974's performance is similar to the HP 7971; the HP 7978 and HP 7980 are much slower.

Immediate Response

To help your computer send data fast enough to permit the drive to stream, the HP 7974, HP 7978, and HP 7980 support **immediate response** mode. Ordinarily the actions of your computer and the drive are serialized. Your computer sends data to the drive. Then the drive writes the data to the tape. After the data is written, the drive returns status information to the host indicating whether the write succeeded or failed. When immediate response is enabled the drive returns status before it writes the data to the tape.

This is accomplished by the drive buffering the data it receives from your computer in high speed memory which is built into the drive. The transfer rate between the host and this buffer memory is much faster than the transfer rate would be if the drive transferred the data directly to the tape. Because the drive returns status to your computer very quickly, the host's and the drive's activities overlap, so the average transfer rate to the drive has a much better chance of being fast enough to permit the drive to stream. Even when the drive has to go through a reposition cycle, it can still be buffering additional records from the host.

Even with immediate response enabled, the HP 7974 and HP 7978 tape drives typically don't stream continuously because the programs running on the Series 300 don't collect their data from the disk fast enough to supply it to the tape drive. However, they still perform faster than the HP 7971 stop/start tape drive.

An identical concept applied to CS/80 cartridge tape is referred to as immediate report.

Pre-5.0 Drivers

Drivers in the HP-UX versions before 5.0 treat records written across or beyond the physical end of tape mark differently. The older version of the HP 7971 device driver reports an error on read or write if a record crosses physical EOT. When writing on multiple reels, the old driver will finish writing the record, but since writing that record generated an error, the application (for example, `cpio`) will re-write the record on the next tape. The record that crosses physical EOT is called a **phantom** record; though the record is written at the end of one tape, it is written again at the beginning of the next tape. Reading the phantom record also generates an error; applications using the pre-5.0 drivers will receive a read error, and will not use the phantom record.

5.0 and later Drivers

As of version 5.0 of the Series 300 HP-UX kernel, the HP 7974 and HP 7978 drivers support immediate response mode by default. For single-reel magtape archives, the only consequence of this change is that the drive streams more when it writes, and so it writes faster; you can still interchange tapes between older driver versions and the current drivers. For multiple-reel magtape archives, the consequence of this change is that you can no longer interchange tapes between older and current drivers without setting a compatibility mode bit (refer to "Backward Compatibility" below). Without compatibility mode, the phantom record of the older multiple-reel archive will be read from both tapes on drives using immediate response. In particular, `cpio` from one version will not correctly read multiple tapes created from the other version.

Backward Compatibility

As of version 5.0 of HP-UX, the HP 7970, HP 7974, and HP 7978 drivers support a non-default old (2.x) compatibility mode which the user may select by setting the third least significant bit in the device file minor unit number (that is, 0x000008). In this mode these drivers can read and write tapes with older end-of-tape semantics. The only time you need to set the compatibility mode bit is when you are reading pre-5.0-written tapes with a current driver, or when you are writing a tape with a current driver to be read by an older driver. When the compatibility mode bit is set, the HP 7974 and HP 7978 will have a slower writing rate. The reading rate is unaffected. The *Series 300 Configuration Reference Manual* has a description of all the bits in the tape drive minor number.

If you are *sure* you have a tape written by a Series 200, pre-5.0 driver, then you may have the phantom record. The *only* way you could have a phantom record is if you wrote the tape using an HP 7971 driver, version previous to 5.0. If you delete this phantom record, you will no longer need to run your current driver in compatibility mode.

Note Before you try to delete the phantom record, *make sure you have the phantom record.*

As an example, if you have only one file, and it crosses over more than one tape, then to delete the phantom record load the tape and type in the following (each line has a "#" followed by a description of what's happening):

```
mt rew        # rewind the tape
mt fsf        # skip past the first file marker
mt bsr 2      # backspace over the first file marker and the
              # phantom record
mt eof 2      # write a new logical EOT (new double file mark)
mt rew        # rewind the tape
```

Optical Technology

Rewritable Optical

Traditionally, mass storage solutions have fallen into one of two categories—primary or secondary storage. Primary storage is typically one or more fixed magnetic hard disks. It is fast, random-access storage with moderately high capacity and is used as the online system disk.

Primary storage is used to store applications, heavily accessed databases, or files on which you are currently working and using extensively. It is also used for applications needing virtual memory, fast data processing, report generation, and complex calculations.

Secondary storage has consisted of one or more offline storage devices—usually a 1/4-inch or 1/2-inch tape drive, or a flexible disk drive on smaller systems. These devices are used primarily to back up the system disks and are also used for logging transactions, distributing software, archiving historical data, and exchanging data between systems.

There's a gap between primary and secondary storage in terms of access time and cost per megabyte. Average access time for hard disks is measured in tens of milliseconds. Access time for information on tapes is measured in tens of seconds for a mounted tape, minutes to hours for a tape in the library. The cost of magnetic disk storage is a few dollars, while tape storage costs a few cents per megabyte.

Rewritable optical fills this gap. With an average access time in the .1- to 10-second range, and a cost of a few cents per megabyte more than tape storage, optical drives create a new layer in the storage hierarchy called **Direct Access Secondary Storage (DASS).** Figure 2–9 illustrates this concept.

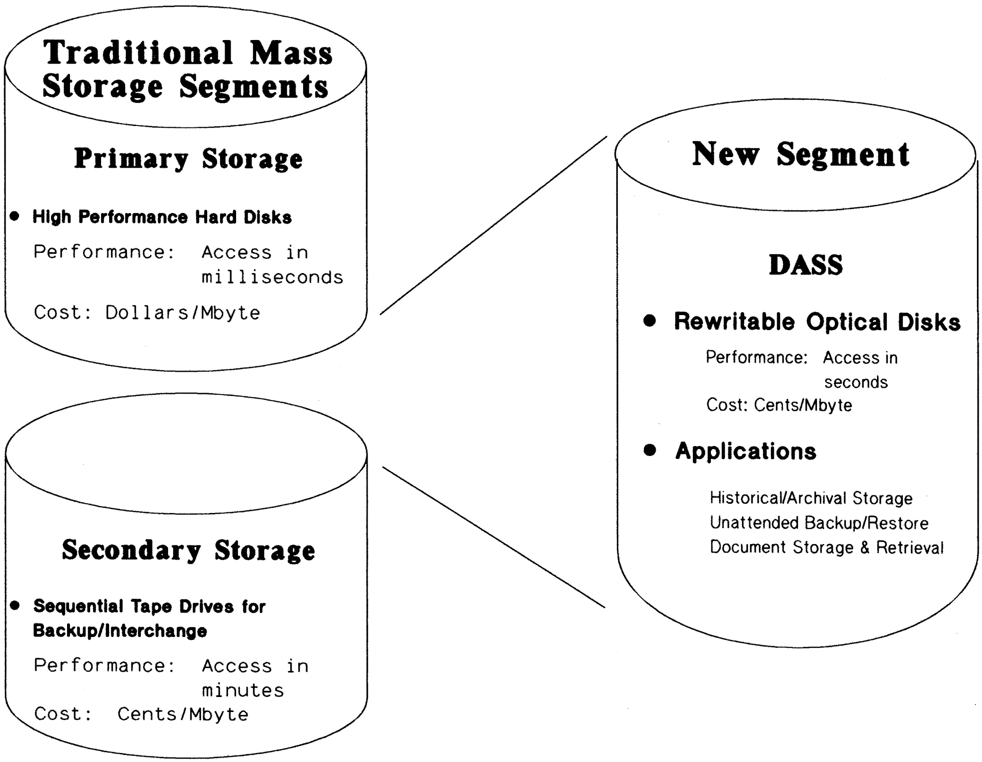


Figure 2-9. Direct Access Secondary Storage

Why Use Rewritable Optical?

Rewritable optical technology has many strengths. The disks are more durable than other media, have greater storage capacity, are more reliable, are removable, and cost far less per megabyte than magnetic disks. Rewritable optical disks are good when you need direct access to a large amount of traditionally "offline" information such as archival or backup files.

The major disadvantage of optical drives is an access time slower than that of hard disks. Due to the weight of the optical head, with its laser, lenses, and mirrors, today's access times for optical disks are 2-5 times slower than high performance magnetic hard disks. Therefore, the optical autochanger would not be good as a hard disk replacement.

Hewlett-Packard's Rewritable Optical Products

The Hewlett-Packard rewritable optical products[1] range from a standalone drive for small systems to an autochanger multi-disk setup for large systems and networks. The following products are supported on HP-UX 6.5 and subsequent releases of HP-UX.

HP Series 6300 Model 650/A - Optical Disk Drive

The Model 650/A is a stand-alone rewritable optical disk drive. It uses 5.25 inch (130 mm) magneto-optical disks (a type of rewritable optical technology and complies with the continuous composite (C*C) format. One optical disk holds 650 Mbytes of data (325 Mbytes per side). However, because there is only one read/write head assembly in the drive, you must eject the cartridge and flip it over to access the second side.

Using a 2400-rpm rotational speed, the Model 650/A achieves a data transfer rate of at least 340 Kbytes per second. It connects to the host system with a SCSI interface and can be accessed by SCSI commands as a conventional magnetic disk drive.

You can use an optical disk drive like any other disk drive. You can rewrite data an unlimited number of times and can store the disks for at least 10 years making data storage worry-free. Unlike 1/2-inch tape that you must re-tension to keep archived files readable, MO disks are maintenance-free and very durable. In addition, because of their small size MO disks require less storage space than 1/2-inch tapes.

---

1 While Hewlett-Packard intends to commercially release rewritable optical products, HP reserves the right at any time not to release the products or, even if released, to alter prices, features, specifications, capabilities, functions, release dates, general availability, or other characteristcs of the product.

Optical Disk Drive Guidelines

The Model 650/A can be used as a temporary or emergency boot device. If you use the optical disk drive as a boot device, you must insert the disk before you turn on the system, and you must not remove the disk until after you turn off the system.

If you use the optical disk drive as a part of your mounted file system, do not eject the disk until you have unmounted it.

HP Series 6300 Model 20GB/A - Optical Autochanger

The Model 20GB/A Optical Autochanger is a device which allows automatic, convenient access to a vast amount of information. The autochanger can have a library of up to 32 magneto-optical disks, offering a total capacity of 20.8 gigabytes.

The optical autochanger consists of several elements: one or more optical drives, slots to store the 32 disks, a mechanical picker, and a mailslot which lets you insert and remove disks.

Within the autochanger cabinet the mechanical picker, guided by the host computer, selects, moves, rotates, and inserts disk cartridges into the drive mechanism(s). The mechanical picker rotates and flips the disk because the drive is single-sided and the disk is double-sided. Figure 2–10 illustrates the optical autochanger concept.

Since the drive(s) are the same as in a Model 650/A, you can use the disk in either of the two devices.

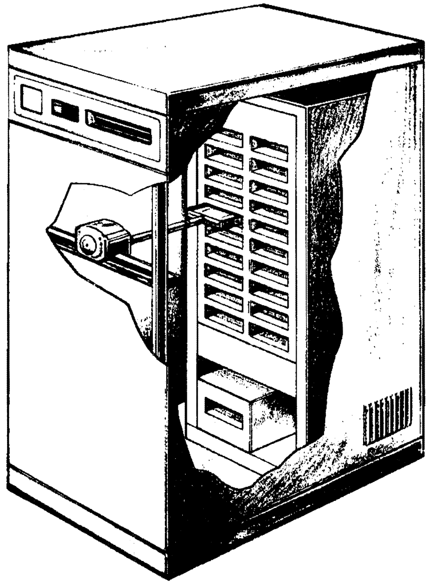


Figure 2-10. HP Series 6300 Model 20GB/A - Optical Autochanger

Optical Autochanger Guidelines

Each of the 64 surfaces in the autochanger (32 disks, double-sided) is treated like a hard disk. A surface can hold a file system, can be stored to and rewritten like a hard disk, has a device file associated with it, and can be accessed with the same HP-UX commands as a magnetic disk.

While each surface on the disk is treated like a magnetic disk, the optical autochanger is not a hard disk replacement. With this in mind, follow these guidelines when using an optical autochanger:

- Do not use the optical autochanger as a boot device.

  Do not use any surface in the optical autochanger as the boot device. It is not supported as a boot device.

- Do not use the optical autochanger's surfaces for swap space.

Do not put swap space on any surface in the optical autochanger. Swap space is not supported.

- Do not put directories that are on optical autochanger disk surfaces in your search path. This could cause performance delays.

  Note that you can define the path environment variable in a local environmental shell script or in an automatically-executed shell script such as `/etc/profile` or `/etc/csh.login`.

- Tips on mounting surfaces:
  - Minimize the number of surfaces mounted for writing.
    - If the system's power fails, all mounted surfaces will require a file system check (`fsck`). The time required to check the file system if you have many optical disk surfaces mounted could be excessive.
    - If you mount many surfaces readable/writable, the `syncer` process calls for more disk exchanges which decreases performance.
  - You can have as many read/write surfaces as you want; however, you should minimize the number of mounted read/write surfaces. This helps insure the product's stated performance quality.

    To get started, mount just two surfaces as read/write (the default). Mount the remaining surfaces as read-only using the `-r` option to the `mount` command.
  - Although you can mount surfaces anywhere in the file system, for best performance you should mount no more than one surface in any branch of the file tree. Additional disk exchanges will result when traversing these paths.
  - Do not put surfaces in `/etc/checklist`.

    This slows the booting process considerably since each surface must be exchanged to be mounted and possibly file system checked (`fsck`). A better strategy is to write a script to mount surfaces after the system is booted. This way, activity on the rest of the system can proceed while the surfaces are being mounted.

- If you have a one-drive optical autochanger system, don't copy large files straight from side A to side B of the same disk. This procedure requires many exchanges. Rather, copy the file intermediately to the hard disk and then to the other surface.

Memory Management

Overview

Series 300 computers use **demand paged virtual memory management** to allow the user process's **logical address space** to be larger than the actual physical memory. The Series 300 Model 310, equipped with the Motorola 68010, supports a logical address space of 16 Megabytes. Other Series 300 models use the Motorola 68020 processor and allow the logical address space to be as large as 4 Gigabytes.

The demand paged virtual memory management subsystem manages three types of resources: the logical address space, the physical memory, and the swap space.

This section provides information on how logical address space, physical memory, and swap space are managed. It is not intended to be a tutorial on virtual memory or HP-UX processes. The system parameter's default settings for demand page virtual memory management will support a broad range of users' applications and HP-UX utility programs.

Two major features of Hewlett-Packard's demand paged virtual memory management on the Series 300 computers are:

- HP series 300 computers use a Memory Management Unit (MMU) to provide protection against illegal accesses in a multi-tasking environment. The MMU also supports mapping between logical address space and physical memory at the page level. This means that protection and sharing is possible at the granularity of a page.

- Series 300 computers uses both a paging and a swapping mechanism to manage memory resources. A system process, known as the `pageout` daemon, tries to maintain the number of free pages of memory above a threshold. When the system is heavily loaded and the `pageout` daemon can't keep up with the memory demand, the swapping mechanism is enabled. The swapping mechanism selects and swaps entire processes to secondary storage; this will free memory.

Besides providing the fundamental support for virtual memory, the Series 300 computers provide several other important features:

- Series 300 computers have shared memory for high bandwidth interprocess communication (refer to *shmget(2)*, *shmat(2)*, and *shmctl(2)*).
- Series 300 computers provide device mapping for mapping physical addresses into logical address space. This allows direct access to I/O devices (refer to *iomap(7)* and *graphics(7)*).
- Series 300 computers provide process locking for locking all or part of the user process space for real time application needs (refer to *plock(2)*).

Logical Address Space Management

Logical address space management defines and controls the user process's structure. Each process that executes in the Series 300 computer consists of three logical segments:

- the code segment
- the data segment
- the stack segment

The Series 300 computer supports three execution formats:

- The normal format where the code segment is neither write-protected nor shared (`-N` option of `ld`).
- The shared format where the code segment is both write-protected and can be shared by more than one process (the `ld` default).
- The shared and demand load format where the code segment is write-protected and shared, and the entire file is demand loaded (`-q` option of `ld`).

Each of the three segments (code, data, and stack) is divided into equal size pages. The page size is 4 Kbytes. **Demand loading** means individual pages are brought into memory from the file system only when the information in the page is referenced. More details about shared and demand load characteristics and trade-offs will be discussed in later sections.

The code segment starts at logical address zero, followed by the data segment. The data segment can be dynamically expanded into higher addresses as required by a program's run time logic (using, for example, `brk` and `malloc`). The stack segment is mapped near the top (high address) of the logical address space. The system allocates stack towards the lower addresses if the process requires additional stack space to execute. The area at the top of the user logical address space is used by system overhead.

The code, data, and stack segments are limited by three configurable system parameters:

System Parameter	Use	Default
`maxtsiz`	Limits the size of the code segment.	16 Mbytes
`maxdsiz`	Limits the size of the data segment.	16 Mbytes
`maxssiz`	Limits the size of the stack segment.	2 Mbytes

These parameters are configurable using the `config` command (refer to "Configuring Operating System Parameters" in Chapter 6).

Shared memory segments can be attached (placed) between the data, and the stack segment using `shmat`. The configurable system parameter, `shmseg`, limits the maximum number of shared segments that can be attached to a process (default value is 10). The configurable system parameter, `shmmaxaddr`, gives the maximum address shared memory can be attached (default value is 16 Mbytes). Both of these parameters can be configured using `config`.

To prevent segments from overwriting each other, the system does not allow them to overlap. The upper limit of the data segment is the same as the address where the shared memory segment is attached. The lower limit of the stack address is set to be the address where the shared memory segment is attached, plus the size of the shared memory, plus a proper alignment. If the `sbrk` or `shmat` system calls, or stack growth, were to cause the segments to overlap, the user would either receive an error (such as ENOEM) or the user process will be killed.

Figure 2–11 illustrates the physical layout of the user process logical address space.

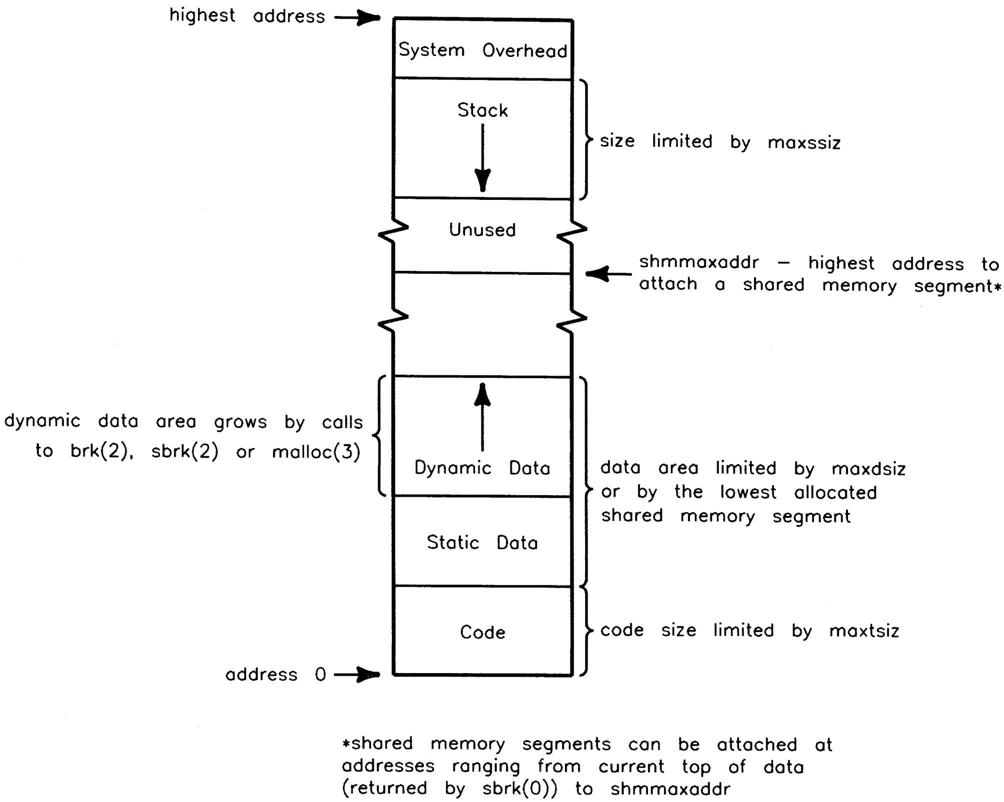


Figure 2–11. User Process Logical Address Space

Physical Memory Utilization

The maximum amount of physical memory you can install on your Series 300 computer is 7 1/2 Megabytes for Models 310 and 320, 4 Mbytes for the Model 318, 16 Mbytes for the Model 319, 8 Mbytes for the Model 330, and 32 Mbytes for Model 350. The minimum amount of RAM required for a non-networked single-user Series 300 HP-UX system is 2 Mbytes. The minimum amount of RAM for a Series 300 acting as the root server for an HP-UX cluster is 3 Mbytes. As more users are added on a multi-user system, more memory may be required for adequate performance. The computer's performance will also depend on the applications you run and on the peripheral devices attached to the system.

At powerup the system determines physical, available, and lockable memory:

physical memory
: The system displays the amount of physical memory on the system console as "real mem = xxxxxxx".

  HP-UX reserves part of this memory for use by the operating system code and its associated data structures; this part of memory is not pagable. The remaining physical memory is available for use by user processes.

available memory
: The system displays the amount of available memory on the system console as "avail mem = xxxxxxx".

  The number of kernel device drivers and the size of various kernel data structures can be reconfigured using `config` to increase/decrease the user available memory. For example, a larger value of `nproc` means the user process kernel data structure must be larger. Changing configurable system parameters is discussed in "Configuring Operating System Parameters" in Chapter 6. Note that this is available memory in the computer, not available free space on the file system. You can see the amount of free space in the file system by using the `df` command.

lockable memory — The system displays the amount of lockable memory on the console as "lockable mem = xxxxxxx".

All or part of available memory can be locked by a subsystem, or by user processes using `plock` or `shmctl` intrinsics.

Locked memory cannot be paged; if most of the available memory is locked the system may deadlock. Some unlockable memory must be available to prevent deadlock. The system parameter `unlockable_mem` reserves the amount of memory that cannot be locked. You can use `config` to set `unlockable_mem`. The default value for unlockable memory is 100 Kbytes.

The available memory minus the memory locked by subsystems or user processes is the memory that is actually available for virtual memory system usage.

As noted in the "Overview" part of this section, when the `pageout` daemon fails to keep up with the memory demand, the swapping mechanism will be turned on to select and swap some processes out to free memory. When swapping starts, the system performance is degraded. If this happens often, perhaps more physical memory should be installed. To find out if processes are swapped out use the `ps` command with `-l` option.

Swap Space Management

Swap space is a contiguous area on the secondary storage, usually a disk drive, reserved for use by demand page virtual memory management. This section explains some of the swap space concepts required to determine the amount of swap space required by your system. If you are running an HP-UX cluster, you should also review the concepts in the section called "Cluster Concepts".

The virtual memory management system keeps an image copy of all existing processes and shared memory objects. Swap space is separate from the file system. Series 300 computers support both single and multiple swap devices. The multiple swap device mechanism allows the swap space to be present on several disk drives for ease of expanding the swap space. Also, if your applications require above average number of I/O operations, multiple swap devices may increase throughput. This is especially true with HP-UX clusters.

At powerup the size and location of the swap space on each swap device is displayed in 512 byte blocks. "start = xxxxxx" indicates the swap space's starting disk block number and "size = xxxxxx" indicates size of swap space.

Swap space holds an image of code, data, stack, and shared segments. The storage size for each of these process segments is limited by the swap space. When any of those limits is exceeded the system either returns an error (such as ENOMEM) to the user process or kills the user process.

The default settings of the swap space system parameters allow the image on the swap space for each of the code, data, stack and shared memory segments to be as large as 20 Mbytes. These settings are bigger than necessary for the Model 310, but since they do not actually consume any extra resources, they are set to be the same for all Series 300 systems.

If these default settings are too small for your application program, you can change them using `config` and Table D-1 in Appendix D of Volume 2. Refer to the "Configuring Operating System Parameters" section in Chapter 6 and Appendix D for more detail.

Two of the operating system parameters you may change are `dmmin` and `dmmax`. Bigger values of `dmmin` and `dmmax` will result in more fragmentation in the swap space. *Do not reconfigure these value to be bigger than the default settings unless it is necessary to support large applications.*

The space for the entire image of every existing segment is allocated on the swap space; therefore, swap space must be large enough to hold all segments of all existing processes. If there is not enough swap space, the system will either return an error (such as ENOMEM) for some system calls, or it will kill the user process. If you need more swap space, you can add more swap devices or you can rebuild the file system and reserve more swap space on your existing swap device. Use the procedures in "Configuring Swap Space" in Chapter 6 to create more swap space.

If you have an HP-UX cluster, you must have enough swap space on your root server to accommodate the sum of swap requirements of all diskless cnodes swapping to the root server.

In an HP-UX cluster, cnodes that swap from the root server will adopt the same values for `dmmin` and `dmmax`, regardless of the values they have configured into their kernels. Cnodes that swap locally use the `dmmin` and `dmmax` values configured into their kernels.

The exec system call uses an area at the beginning of the swap space for a scratch area. While overlaying the old process image with the new process image, exec uses the scratch area to temporarily hold the arguments and environment variables. The size of the scratch area is determined by the configurable system parameter argdevnblks. The default size is 256 Kbytes. The size of the scratch area must be taken into consideration when reconfiguring swap space. Refer to the section "Determining the Amount of Swap Space Needed" in Chapter 6 for details on how to compute the swap size.

Shared Code

Often, several processes want to run the same program simultaneously (such as a text editor program). If the program is not shared, then each process running the program has a copy of the program's code and data. If the processes share one copy of the code, the amount of memory required for each user's process space dramatically decreases.

The term **shared code** describes user code which is loaded into the user text area. When a process executes shared code, it is directed to the copy of the code in the user text area. If the shared code is not yet loaded (no other process is currently accessing the code), the code is first loaded into user text area before the process begins execution. Only one copy of the code exists in memory regardless of the number of processes running the program.

The system knows how many processes are accessing shared code by maintaining a count (called the **use count**) of the number of processes accessing the code. When a shared program is loaded into the user text area, the use count for the program is set to one. When the process finishes executing the code, the use count is decremented.

For example, suppose that the text processor program ed is marked "shared". When a process first executes ed, its code segments are loaded into the user text area and its use count is set to one. Suppose that while the first process is executing ed, another process executes the ed program. Since the code already resides in main memory, no additional memory is allocated. The new process simply executes the copy of ed's code that resides in main memory; its use count is incremented from one to two. The first process now finishes editing and terminates the ed program. The system decrements the use count of ed's shared code. Since the use count is not yet zero, the shared code remains in memory. Finally, the second process finishes editing and terminates the ed program. The system decrements the use count of `ed`'s shared code segment and, finding its value to be zero, releases `ed`'s shared code data structure and its associated physical memory and swap space.

The shared text "image" can be swapped in or out (between memory and disk) like any other user process segments. Refer to *ld(1)* and *chatr(1)* for information about making programs shareable or shareable and demand loadable.

Shared Code and the Sticky Bit

If the sticky bit is set on a file containing shared code, then when the last process accessing the shared code terminates, the memory associated with the code is freed but the code still resides in the swap area.

For example, suppose that the code of the text editor program `ed` is marked "shared". Also suppose that the file in which `ed` resides has its sticky bit set. If two different processes execute the `ed` program and then terminate, the same actions occur as previously described (under the "Shared Code" heading above) with one exception: when the use count drops to zero, the swap space allocated for the `ed` program is *not* released (that is, it is not freed for other uses).

To release shared, "sticky" code, a superuser must change the protection/access bits on the file such that the sticky bit is no longer set. Before the swap area is released, the use count must drop to zero after the sticky bit is cleared.

Shared Code: Benefits vs. Cost

Shared code significantly reduces the amount of memory required for user process space when multiple processes are executing the same program. The only cost is the size of the data structure associated with shared code (refer to the `ntext` entry in Appendix D).

When shared code has its sticky bit set, start-up response time generally improves because bringing the code in from the swap area is generally faster than bringing it in from the file system. The trade-off for this is that the code occupies space in the system's swap area at all times (after the first invocation).

Shared Code in an HP-UX Cluster

Although you can mark code as shared in an HP-UX Cluster, code will never be shared between cnodes. If several users on one cnode are executing the same shared code, it will behave as in a standalone environment. However, if one user on cnode `donald` and one user on cnode `daisy` are using the same share code, there will be two complete copies, even if the two cnodes are sharing the root server's swap space.

Demand Load

Programs often contain routines and code which are rarely accessed. For example, error handling routines can comprise 80 percent or more of some program code and yet may be rarely accessed. When a program is loaded, `exec` normally copies the entire program into main memory. If the unused pages are a significant portion of the program, the memory allocated for that code is wasted.

With HP-UX, it is possible to mark programs as **demand loadable.** When a demand loadable program is executed, no pages are actually loaded. No memory is allocated for the non-loaded pages until those pages are actually accessed. Only when the program attempts to access a demand-loadable page is memory actually allocated and the page loaded from disk.

You can set demand loading at link time with the `-q` option of `ld`, or changed in the executable file with the `-q` option of `chatr`.

Making code demand-loadable provides faster program startup, and may reduce the amount of memory needed to run a program, since only the pages needed are loaded. However, the program will probably take slightly more file system space, and may actually take longer to execute than a program that is not demand-loadable if many of the demand-loadable pages are accessed, since loading a process one page at a time is slower than reading the whole image at once.

Device I/O

HP-UX treats I/O to a device in the same fashion as I/O to a file. In fact, before your computer can "talk" to a device, a file (called a device file) must be created. This file defines the location of the device and the manner in which the computer and the device must communicate. Device files are created with the `mknod` or `mkdev` commands and are usually stored in the `/dev` directory. To communicate with a device redirect input from, or output to, the device file. The computer then uses the information contained in the special file to manage all transfer of data between it and the device.

Device Classes

All I/O devices can be classified as block special, character special, or network special devices. Block devices are devices which transmit and receive data in blocks. The block size is defined as BLKDEV_IOSIZE in `/usr/include/sys/param.h`. Character devices include any device which is not a block device, including printers, plotters, terminals, magnetic tape drives, and pseudo devices. Disk mass storage devices, including rewritable optical devices, are used as both character devices and as block devices. Network devices are described in the *Using ARPA Services* manual.

Drivers

The `mknod` command creates a device file using a specified major number and minor number. A **device file** is the interface between a process and a driver. A **driver** is compiled code (supplied with your system) which defines the protocol and handshaking that allow an I/O device and the computer to communicate. For a list of the drivers available on your system, use the `lsdev` command. The `/etc/conf/dfile` contains a list of drivers currently installed in your system (refer to "Configuring Device Drivers and I/O Cards" in Chapter 6).

The **major number** is based on the device driver and on the access method (block or character). **Minor number** typically defines the device's address, but is different for different devices. Refer to the applicable section in the chapter on "Customizing the HP-UX System" for details on both the major number and the minor number.

Cluster Concepts

An HP-UX cluster is a group of workstations that are connected via Local Area Network (LAN). All workstations in the cluster share a single file system.

This section explains concepts of HP-UX clusters. Many terms, concepts, and tasks are different from standard HP-UX (or UNIX). If you are already familiar with HP-UX clusters you can skip much of this chapter. However, if you are new to HP-UX clusters you must read this chapter before creating, administering, and using your clustered system. Using and administering an HP-UX cluster without understanding the concepts could cause major problems such as data loss or a non-bootable system.

This chapter will introduce the following terms and concepts: diskless cnodes, clustered environment, Context Dependent Files (CDFs), and context. The first section will introduce you to the reasons you may wish to use an HP-UX cluster.

Why Use a Cluster?

To understand the advantages of HP-UX Clusters it is helpful to review traditional computer systems. Two main types of computer systems are used today: timeshared systems consisting of a single mainframe with attached terminals, and systems consisting of personal workstations connected with local area networks. Each type of system has its own advantages and disadvantages. These are outlined in Table 2–3 and Table 2–4.

Table 2–3. Advantages/Disadvantages of Timeshared Systems

Advantages	Disadvantages
transparent peripheral sharing	performance limitations: as number of users goes up, performance goes down.
transparent file sharing	limited reconfiguration ability
same login from any terminal	limited human interfaces (few or no bit-mapped displays)
no duplication of shared resources	large initial investment
only one system administrator required	

Table 2-4. Advantages/Disadvantages of Networked Workstations

Advantages	Disadvantages
better performance for each user	files not guaranteed to have same names on different systems
supports bitmapped graphics and windows displays	remote files must be accessed with different commands and system calls than local files
provides for incremental growth	performance is often significantly lower when accessing a remote resource
lower initial investment	each user must perform system administration functions for his workstation
	significant number of duplicate files on each workstation
	difficult to share peripherals

An HP-UX cluster combines the advantages of the timeshared system with the advantages of networked workstations. Some of the advantages are:

- same view of a global filesystem from each workstation
- single point system administration - only one system administrator required for a cluster
- flexibility of configuration
- dynamic reconfiguration
- maintains high performance for individual workstations
- provides sharing of costly resources
- a bit-mapped display per user becomes practical

What is an HP-UX Cluster?

An HP-UX **cluster** consists of one or more workstations linked together with a local area network (LAN) but having only one root file system. From the point of view of the file system, all the machines appear as one system. From the point of view of processors and processing space, each machine in the cluster is distinct.

A basic cluster consists of a **root server** and zero or more **diskless workstations.** Each computer in the cluster is referred to as a **cnode,** or cluster node. The **root server** is the cnode with the root file system. This cnode is capable of supporting other, diskless, workstations. A **diskless workstation** is a cnode that does not, and cannot, have a local file system. Diskless workstations are called **diskless cnodes.**

All diskless cnodes boot, over LAN, from the root server's file system. Diskless cnodes access all files from the root server's file system. Diskless cnodes also usually swap to the root server's swap area. If preferred, a diskless cnode can be set up to swap to a local disk.

Even though all nodes in a cluster share the same file system, there are some files that need to be different for different nodes. Some examples of files whose contents should not or cannot be shared are device files and files which contain system setup scripts, such as `/etc/inittab`.

To allow files specific to a cnode or class of cnodes, Hewlett-Packard has developed a structure called a **Context Dependent File** (CDF). A CDF is a mechanism for allowing different cnodes to see different contents for a file which has the same name for all cnodes. The CDFs that are required to run a cluster are automatically created when the reconfig program is used to create a cluster environment.

More detailed information on all the cluster concepts is found later in this section.

Refer to Figure 2–12 for an example system. The root server (the cnode with the file system) is called daisy. There are two diskless cnodes in the cluster: donald and dewey. donald has a second LAN card and serves as a networking gateway to computers not on the cluster's LAN.

Requirements

The following requirements must be met to have an HP-UX cluster:

- The root server must be a Model 319C+, 330, 350, 360, or 370.
- The root server must have enough swap area configured to satisfy the combined swap requirements of all the diskless cnodes that do not have local swap disks.
- Diskless cnodes must have Revision B or later Boot ROMs for booting over LAN.
- Diskless cnodes can be any Series 300 computer with a minimum of 3 Mbytes of RAM.
- The root server must have HP-UX version 6.0 or later installed. It must have the AXE, PE, and NS-ARPA Services/9000 products installed.

Setups

All file system disks must be mounted on the root server. Your backup device should also be connected to your root server. Modems for UUCP, spooled plotters, and spooled printers *must* be on the root server. A diskless cnode can have local devices such as HP-IB instruments, non-spooled printers and plotters, and non-file system mass storage devices, but the local devices can be used only by processes on that cnode. Cnodes can easily invoke processes on other cnodes with the `remsh` and `rlogin` commands.

All cnodes in the cluster must be connected via LAN. However, you can have more than one cluster per LAN. It is recommended, but not required, that the cluster be on a small local LAN. There will be less contention (and therefore better performance) than if the cluster were on the same LAN as many other computers.

On the cluster's LAN any system can be a diskless cnode, a root server for diskless cnodes, or a standalone system. Any one of these machines can also function as a networking gateway to computers located on another LAN.

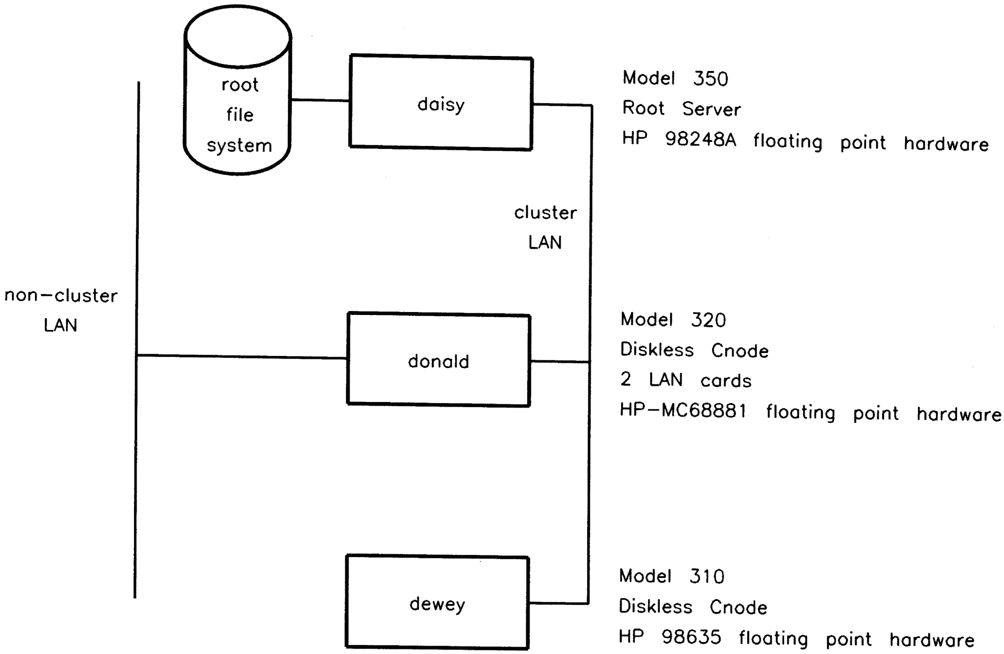


Figure 2-12. Example Cluster

Model of Cluster System Administration

Single-Point System Administration

All system administration can be done from one cnode (except for booting, which must be done from the specific cnode). You may find it most convenient to do system administration from the root server since cluster shutdown can only be done when physically logged into the root server as root. You cannot shut down the cluster using a remote login to the root server.

Much like a Standalone System

Many activities are the same as on a standalone system because the entire cluster uses a single, "global", file system.

Some Files Must Have Multiple Versions

Although there is a single file system, some files must have a copy for each cnode (or set of cnodes); these files cannot be shared among all cnodes on the cluster. For example, since each cnode on the cluster has its own console, the `/dev` directory needs to appear different for each cnode. This is accomplished through Context Dependent Files (CDFs), which are described later in this section. A list of system Context Dependent Files (CDFs) is also shown later in this section.

Tasks Restricted to Specific Cnodes

Although you have single-point system administration, you must execute some commands on specific cnodes. For example, you must add a cnode by executing the `reconfig` program on the root server, but you must create a cnode's kernel on that cnode if using the `reconfig` program. However, you can still accomplish this from a single point by using the `rlogin` and `remsh` NS-ARPA services commands.

For example, if you are on the root server, and wish to create a kernel for the cnode called `dewey`, you would execute the following series of commands to invoke the `reconfig` program properly:

> `$ rlogin dewey`
> *You will now see the copyright messages on dewey.*
> *When you see a system prompt, type:*
> `$ /etc/reconfig`
> *When you are finished with reconfig, enter a ^D or type:* `exit`
> *You will see:*
> `$ Connection closed.`

You are now back on the original system.

If you executed the `reboot` command using `rlogin`, you will not need to `exit` from the other system. If you have problems logging in or booting your cnode, refer to the "Troubleshooting" appendix in Volume 2 of this manual.

List of System Administration Tasks

System administration tasks are divided into four main groups: tasks that must be executed from the root server, tasks that should be executed from the root server for performance reasons, tasks that should be executed from the affected cnode, and tasks that can be performed from any cnode in the cluster (with either local or global results).

The following tasks must be executed from the root server, either directly or via an `rlogin` command from a diskless cnode onto the root server:

- `fsck` (When executing `fsck` on the root file system, you must be directly logged into the root server, not logged in via `rlogin`.)
- HFS `mount/umount` (NFS `mount/umount` can be done on any cnode)
- create file system
- create recovery system
- update HP-UX
- install or update application software
- remove filesets
- reboot cluster
- shutdown cluster
- configure line printer from reconfig
- add a cnode to the cluster
- remove a cnode from the cluster

In addition, the following tasks *should* be executed from the root server for performance reasons:

- backing up the file system
- restoring the file system from a backup

The following tasks should be performed from the affected cnode, either directly or via executing the `rlogin` command to the cnode:

- executing the `reboot` command (*must be done from the cnode*)
- accessing the CDF subfile for the cnode
- building a kernel (if using `reconfig` you *must* be on the cnode, if using `config` it is much *safer* to be on the cnode)
- making a device file (if using `reconfig` you *must* be on the cnode, if using `mknod` it is much *safer* to be on the cnode)
- running `reconfig` to set user access (*must* be on the cnode)

The following tasks can be performed from any cnode in the cluster, and the result is global to the cluster:

- NFS mounts/umounts
- adding or removing a user (again, perform this from only one cnode at a time)
- executing `cwall`
- setting the system clock
- modifying system files (whether CDF or regular files)

  System files should be edited by only one user at a time. This has always been true, but the point is emphasized again. In particular, the following files should not be modified by more than one user at a time: `/etc/rc`, `/etc/passwd`, `/etc/group`, `/etc/motd`, `/usr/news/*`, `/etc/issue`, `/usr/lib/tztab`.

The following tasks can be performed from any cnode in the cluster, and the result is local to the cnode:

- executing the `wall` command
- system accounting

  Some accounting operations can be performed on any cnode, others must be performed on each cnode. Refer to the "System Accounting" chapter in Volume 2 for details.

Context

Each cnode has a **context** (local to that cnode) that is set at boot time. All processes which run on that cnode inherit the cnode's context. The context is an ASCII string made up of the following attributes:

cnode name
: This attribute is set from field 3 of the `/etc/clusterconf` file. It is the name you entered when adding a cnode.

  If the cnode is not a part of a cluster, the cnode name will be set to "standalone".

floating point hardware type
: This context field will be empty if no floating point hardware exists on the cnode, otherwise it will be one or more of the following:

  - "HP98248A" for floating point accelerator
  - "HP-MC68881" for Motorola coprocessor
  - "HP98635A" for floating point math card

  If the cnode has more than one of these floating point hardware types they will be in the context attribute string in order of best performance to worst performance.

processor types
: There are three processor types:

  - "HP-MC68010" for the Model 310
  - "HP-MC68020 HP-MC68010" for models in the Series 300 using the MC68020 board. The MC68010 is included here since all Series 300 support the MC68010 instruction set.
  - "HP-MC68030 HP-MC68020 HP-MC68030" for the Series 300 using the MC68030 board.

cnode type
: "localroot" or "remoteroot". It is "localroot" if the root file system resides on the local machine (true for the root server or a standalone machine). It is "remoteroot" if the root file system is not on the local machine (true for diskless cnodes).

"default"
: All context strings end with the string "default".

Getting the Context

A new command, `getcontext`, shows the context string for a given cnode. The `getcontext` command returns the process context for the local system. There is a corresponding system call for programs. For example, if we were to inspect the context from system `dewey` from Figure 2–13 , we would see the following output:

```
$ getcontext
dewey HP-MC98635A HP-MC68010 remoteroot default
```

The `getcontext` command will return only the local cnode's context string. To get the context for other cnodes you must log into that cnode (or execute the command with a `remsh` command: `remsh` *cnodename* `getcontext`). For more information on the `getcontext` command, refer to the *getcontext(1)* page in the *HP-UX Reference*.

Context Dependent Files (CDFs)

Even though all cnodes in a cluster (by default) share files, there are some files that must not be shared. To allow files specific to a cnode or class of cnodes, Hewlett-Packard has developed the Context Dependent File (CDF). CDFs are a mechanism for allowing different cluster nodes to "see" different contents for a file which has the same pathname for every cnode in the cluster.

A CDF is used where a file, or files, cannot be shared by the nodes of a cluster. This mechanism uses the cnode's context to determine the view of the requested file for a particular system.

The files in a CDF (subfiles) are referred to by names which must match a part of the cnode's context. These names can be any of the possible parts of the context (i.e., cnode name, floating point hardware types, processor types, cnode type, or the string "default").

Figure 2–13 shows the CDF which allows the systems in the example cluster to have different versions of `/etc/inittab`. The `/etc/inittab` file uses the cnode name part of the context to match CDF subfiles.

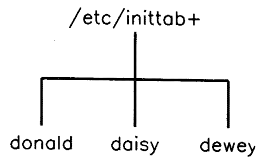


Figure 2-13. /etc/inittab in the example cluster

A user accessing `/etc/inittab` on system donald will see different contents than a user on system dewey.

Using CDFs

Many HP-UX system files have been converted to CDFs in your HP-UX cluster. These *system CDFs* are required for HP-UX clusters to work; they are not intended for general use. *System CDFs must not be modified.* A list of the system CDFs is given in the section "System CDFs".

IMPORTANT Do not change the structure of system CDFs.

In some cases, one or more CDFs must be created to support application software on a cluster. Such CDFs must be created and structured with a full understanding that CDFs are a last resort and should be used only if no other means exist. This is because CDFs contribute to making the administration of the file system much more difficult due to their transparent nature.

The rest of the information on CDFs is provided only to help you understand how the system CDFs work, and to help you create and use your own CDFs if you must create them.

Some Reasons for Having CDFs

Reasons for having different files with the same name:

- Some system files must correlate closely with each cnode's hardware configuration. For instance, `/etc/inittab` and `/etc/ttytype` must reflect how many and what terminals are present for logging in on each cnode. In our example cluster, daisy might have just a console, donald might have a console plus three terminals, while dewey might have a console plus one terminal. Daisy's console might be of type 300l, while donald's and dewey's consoles might be of type 300h.

- Some files are expected to correspond to only one process space (i.e., to one cnode). For example, the `who` command assumes `/etc/utmp` corresponds to one process space, so `/etc/utmp` must be a CDF so that there is one file per process space.
- To obtain the highest performance, you may need to differentiate between SPUs which have a common file system. For example, when setting up a clustered environment, `/hp-ux` (the kernel) is made into a CDF.

  Executables are particularly subject to this if the machines are not identical. CDFs provide a convenient naming convention for this scenario. For example, if you have different floating point hardware on the cnodes in your cluster, and need one version of a program for each type of floating point hardware, you can create a CDF to hold the various versions.
- You have an application which assumes complete control over some file or set of files (or directory, etc.) by using semaphores. If run on multiple cnodes, a copy on each cnode can get access to the files without synchronization with other cnodes.

CDF Implementation

A CDF is implemented as a special directory containing all the versions of a file needed by the different systems in the cluster. The final file name (located in the special directory) must match part of the cnode's context before the cnode can access the file. Using our example cluster (from Figure 2–13), and the `/etc/inittab` file (which is a CDF in a clustered environment), the directory would appear as in Figure 2–14.

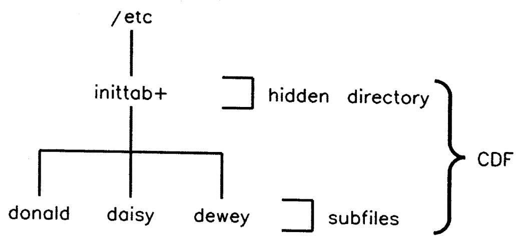


Figure 2-14. Structure of a CDF

In a CDF, the original file (that cannot be shared) becomes a special directory called a **hidden directory.** It is called hidden because the directory structure is normally not seen: you normally think of it as a file, and normally view it as a file, but it is actually a directory. The hidden directory has its set-user-id bit set to indicate to the file system that this is a CDF. Refer to *chmod(1)* for more details on the set-user-id bit.

In addition to the hidden directory, a CDF consists of zero or more subfiles. The subfiles can be any type of file, including another CDF or a directory. The subfiles are named and chosen according to the cnode's context attributes. If no subfiles match a cnode's context string, it will appear that the file (a CDF) does not exist.

To explicitly reference the hidden directory, "+" is appended to the CDF pathname, for example, `/etc/inittab+`. The + serves as an escape character which allows access to the hidden directory. You must append a + to the hidden directory name to refer to the directory. When you refer to the hidden directory without appending a +, the name refers to one of the subfiles in the CDF (or fails to match anything), depending on the context of the process making the reference.

To list the subfiles of the example CDF `/etc/inittab`, type:

```
ls /etc/inittab+
```

You would see:

```
daisy  donald   dewey
```

Creating CDFs

All CDFs required to run an HP-UX cluster are automatically created when you use the `reconfig` program to create a clustered environment. The section

"System CDFs" lists these CDFs. *Do not restructure the system CDFs* (i.e., do not change a non-CDF to a CDF, or vice-versa, and do not change the type of context attribute assigned to the subfiles).

Do not convert system files to CDFs, other than what the `reconfig` *program has already set up.* System files are HP-supplied files, commands, and directories.

CDFs must be managed carefully. The existence of CDFs is not obvious unless you are specifically looking for them. HP therefore recommends that you limit the use of CDFs. *Do not use CDFs unless there is no alternative.* The examples later in this chapter show several classes CDF of anomalies—ways in which CDFs seem to act differently than an ordinary file or directory.

If you need to create your own CDFs, use the `makecdf` program. Refer to the *makecdf(1M)* entry in the *HP-UX Reference* for information on specifics of the `makecdf` program. There is an example later in this chapter.

Hewlett-Packard recommends using only one type of context attribute in a given CDF, for example `localroot/remoteroot`, all cnode names, or all cpu types. Mixing context attributes can result in anomalous behavior as shown in one of the examples later in this chapter. Thus, using mixed context attributes should only be done if the associated anomalies are understood, and are acceptable for the intended usage.

Removing a CDF

If you need to remove one of your CDFs (never remove a system CDF), specify a + with the hidden directory name and recursively remove the files. You must recursively remove the CDF since it is really a directory. For example:

```
rm -rf cdf+
```

Finding CDFs

A CDF is a directory with the set-user-id bit set. CDFs can be nested, one inside the other. So, to find all CDFs in the system:

```
find / -hidden -type H -print
```

Tips and Cautions

When examining and/or modifying CDFs, the most straightforward way to access the entire set of subfiles is to change your current working directory to the CDF's hidden directory (e.g., `cd /cdf.file+`). Be aware of the cluster-wide impact of any changes you make. Some possible side-effects are shown in the examples that follow.

Do not change the way `reconfig` *sets up the CDFs.* Structural changes to system CDFs are not supported. You can change the contents of user-configurable files, but do not change the context attribute (i.e., the name of the subfile in the CDF).

In general, the fewer CDFs you create, the easier your system will be to administer. However, do not remove any of the CDFs created by the `reconfig` program (listed in the section "System CDFs").

Do not mix context attributes in a CDF unless you have no choice. You should avoid having a CDF that contains some files named for one attribute, and other files named for another attribute. Mixing context attributes can cause anomalous behavior when removing, moving, or creating files in the CDF. In particular, never mix context attributes in a CDF such that the CDF subfiles match more than one string in a cnode's context. For example, do not create the CDF shown in Figure 2–15, where the context attribute string for cnode `daisy` would match on the subfile `daisy` and on the subfile `localroot`.

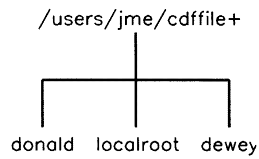


Figure 2–15. CDF with Mixed Context Attributes

The one exception to this is the `/dev` directory. This directory is set up with `localroot` (for the root server) and cnode names (but not an entry for the root server's cnode name). This is a special case and should not be repeated.

In particular, do not use "default" as a CDF subfile name unless you have no choice since this always causes multiple subfiles to match the context of at least one cnode. This can be unpredictable when removing, moving, or creating files in the CDF. The best method is to have subfiles for each cnode named with the cnode name.

Context Attributes and CDFs

Within a CDF, the search for a subfile is based on a predefined context attribute precedence. That is, the system searches for the correct file based on a priority search sequence. The sequence is:

1. cnode name
2. floating point hardware type
3. processor type
4. cnode type
5. the string "default"

Do not depend on the order of attributes within the context. Although it will always follow the above sequence for this release of HP-UX, the order may change in future releases. However, it is guaranteed that a CDF will be parsed in the same order as the attributes returned by the `getcontext` command.

The HP-UX file system tries to match context attributes to the hidden directory's subfile names to select the appropriate version of the file. In the example shown in Figure 2–13 , the following is matched:

- If you specify `/etc/inittab` on the root server you will access the file `/etc/inittab+/daisy`.
- If you specify `/etc/inittab` on the system dewey you will access the file `/etc/inittab+/dewey`.
- If you specify `/etc/inittab` on the system donald you will access the file `/etc/inittab+/donald`.
- on any system, if you specify `/etc/inittab+/dewey` you will access the file named `/etc/inittab+/dewey`.

Autocreation

If you specify only the path for the hidden directory when creating the file, it will automatically create a CDF subfile named after the cnode name attribute. This is called **autocreation.** The order of attributes in the context is not guaranteed to be the same from release to release, so do not depend on the specific autocreation type used by this release of HP-UX.

For example, on system donald, if you have a CDF named /users/jme/cdffile and execute the following:

```
cp /tmp/cdffile /users/jme/cdffile
```

you will create a new file in the CDF with the following pathname:

```
/users/jme/cdffile+/donald
```

CDF Examples

Example: Listing CDFs

Suppose you have the /etc/inittab file used in Figure 2–14, and are currently on the cnode named daisy. Using different options to the ll command, you would see the following responses (note the new option, -H, to view hidden directories):

```
$ ll /etc/inittab
-rwxr-xr-x  1 root    other     725 Dec 26 07:45 /etc/inittab
$
$ ll /etc/inittab+
-rwxr-xr-x  1 root    other     725 Dec 12 07:45 daisy
-rwxr-xr-x  1 root    other     650 May 26 15:53 donald
-rwxr-xr-x  1 root    other     650 May 18 11:47 dewey
$
$ ll -d /etc/inittab+
drwsr-xr-x  2 root    other     1024 Dec 26 07:30 /etc/inittab+
$
$ ll -H /etc/inittab
-rwxr-xr-x  1 root    other     725 Dec 12 07:45 daisy
-rwxr-xr-x  1 root    other     650 May 25 15:53 donald
-rwx--xr-x  1 root    other     625 May 18 11:47 dewey
```

There is also a new command, showcdf, that you can use to display the full pathname (including the CDF and subfile). The following example shows output if you are on the system donald:

```
$ showcdf /etc/inittab
/etc/inittab+/donald
```

Example: Displaying files within a CDF

If you are on the root server (daisy) in the sample cluster, the command

```
more /etc/inittab
```

will display the contents of the file /etc/inittab+/daisy. To view the inittab file for system dewey from the root server, use:

```
more /etc/inittab+/dewey
```

Example: Creating a CDF

Suppose the systems in the sample cluster frequently run a floating point intensive program: /usr/local/bin/floatprog. The systems have different floating point hardware and require a different version of the program to take advantage of this. A CDF to allow each system on the example cluster to execute the version of the floating point program best suited to its hardware configuration would look like Figure 2–16.

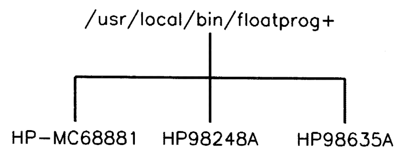


Figure 2–16. CDF Structure Using Different Floating Point Hardware

Assume you created the original floatprog on donald, the machine with the HP MC68881 processor. You have since created two additional versions of the program. These versions are in the files /users/progs/fp48A and /users/progs/fp35. A CDF was created by using the makecdf program, then moving the final compiled versions of the program into the floatprog file by the following commands:

```
makecdf -c HP-MC68881 /usr/local/bin/floatprog
mv /users/progs/fp48a /usr/local/bin/floatprog+/HP98248A
mv /users/progs/fp35 /usr/local/bin/floatprog+/HP98635A
```

If you tried to access this file from a cnode without any of the above floating point hardware you would receive the message:

```
floatprog: file not found
```

This shows where the `default` attribute might be helpful. If you had a version of the program compiled such that it requires no floating point hardware, you could place it in the CDF and use the name `default`. Refer to the *makecdf(1)* entry in the *HP-UX Reference* for more details on the command.

Example: Changing directories within a CDF

Directory changes are also context-sensitive. For example, if you have the directory structure shown in Figure 2-17, with nested CDFs, you may notice some anomalies with the `cd` command.

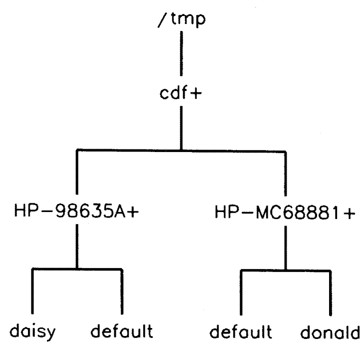


Figure 2-17. Nested CDFs

If you are on system daisy, and wish to change directories within the CDF, you will see the following behavior. Since "HP98635A" is in daisy's context, but "HP-MC68881" is not, `cd` appears to behave differently in the two cases:

```
$ cd /tmp
$ cd cdf+/HP98635A
$ /bin/pwd -H
/tmp/cdf+/HP98635A
$ cd ..
$ /bin/pwd -H
/tmp
$
$ cd cdf+/HP-MC68881
$ /bin/pwd -H
/tmp/cdf+/HP-MC68881
$ cd ..
$ /bin/pwd -H
/tmp/cdf+
```

Example: Removing files from a CDF

Consider the CDF shown in Figure 2–18, using the `localroot/remoteroot` attributes.

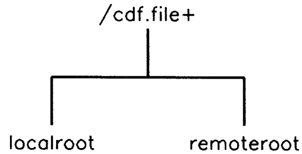


Figure 2–18. CDF with localroot and remoteroot Context Attributes

On system dewey,

```
rm /cdf.file
```

will remove `/cdf.file+/remoteroot`. If you now do the following (also on system dewey):

```
cp /tmp/newfile /cdf.file
```

`/cdf.file+/dewey` will be created (using the autocreation mechanism) leaving the CDF shown in Figure 2–19.

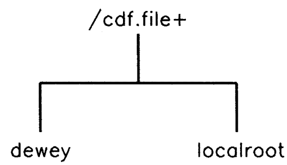


Figure 2–19. CDF with Mixed Attributes

If a user on system donald now tries to access the file, the file does not exist:

```
$ ll /cdf.file
/cdf.file: not found
$
```

Example: Problems from using the "default" context

Using the "default" attribute for a subfile name can be confusing and is not recommended. Suppose the CDF shown in Figure 2–20 exists.

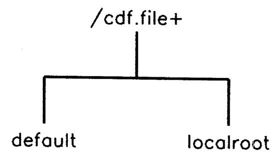


Figure 2–20. CDF with the "default" Attribute

If you are on the root server, the command `rm /cdf.file` removes the file called `/cdf.file+/localroot`. However, after removing this file, `/cdf.file` still exists on the root server because the context then matches the file `/cdf.file+/default`.

Example: Removing a subfile in a CDF containing multiple context attributes

Multiple context attributes in the same CDF can be confusing and are not recommended. Assume you have the CDF shown in Figure 2–21 .

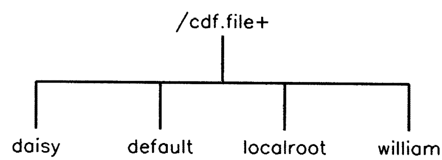


Figure 2–21. CDF containing multiple context attributes

If you were on the root server (`daisy`), you would need to remove `/cdf.file` three times before it actually disappeared from daisy! `rm /cdf.file` would first remove the cnode-name subfile, `/cdf.file+/daisy`, the next time it would remove the cnode type attribute, `/cdf.file+/localroot`, and the third time it would remove the default. Until you had removed all three context subfiles, you could still access a `/cdf.file` file from the root server.

Cnode-Specific Device Files

Starting with the 6.5 release of HP-UX, each device file has a new attribute associated with it: a cnode ID. The cnode ID limits access of that file strictly to processes running on the specified cnode, thus making the file what is termed **cnode-specific.** The cnode ID is the cnode's name or ID number as specified in the `/etc/clusterconf` file. If the device file's cnode ID is 0, it allows access from all cnodes, thus making the file **generic.**

Prior to the 6.5 release, all device files were generic, allowing access from all cnodes in a cluster. Starting with the 6.5 release, all newly created device files are cnode-specific by default, and all device files under `/dev` are automatically converted at update time to appropriate cnode-specific device files. This will improve security in the clustered environment.

Because pre-6.5 releases used generic device files, any device files on archives created before 6.5 will not have the correct cnode-specific information. All device files restored from a pre-6.5 archive will be recovered as a cnode-specific device file using cnode information from the cnode on which you are performing the recovery. This means that if you use your root server to recover a device file for a diskless cnode, the diskless cnode will not be able to access the device file until you re-create it from the appropriate cnode using the `mknod` command.

You can be affected by this change only if you have an HP-UX cluster and:

- You recover device files from pre-6.5 archives.
- You, for some reason, depend on device files being generic.
- You create device files using methods other than what is recommended in this manual.

Note To prevent unexpected behavior when accessing a device file, either do not recover device files from a pre-6.5 archive or recover them only from the cnode that will access them.

To view the cnode ID, use the -H option to ll. In the following example, first you see the listing for the /dev directory for the whole example cluster. Next you see a partial listing of the /dev directory for the cnode named donald, and a partial listing of the /dev directory for the cnode named dewey:

```
# ll -H /dev+
total 6
drwxr-xr-x   6 root    root       1024 Dec  6 14:33 daisy/
drwxr-xr-x   6 root    root       1024 Dec  6 14:33 dewey/
drwxr-xr-x   6 root    root       1024 Dec  6 14:33 donald/
drwxr-xr-x   7 root    root       2048 Dec  6 14:36 localroot/

# ll -H /dev+/donald
total 20
crw--w--w-   3 root    other     0 0x000000 donald   Dec  6 15:58 console
crw-rw-rw-   1 root    other    12 0x000000 donald   Dec  6 14:33 crt
crw-rw-rw-   1 root    other    19 0x150000 donald   Dec  6 14:33 ether
crw-rw-rw-   1 root    other    24 0x000010 donald   Dec  6 14:33 hil1
crw-rw-rw-   2 root    other    24 0x000020 donald   Dec  6 14:33 hil2
      ⋮

# ll -H /dev+/dewey
total 20
crw--w--w-   3 root    other     0 0x000000 dewey    Dec  6 15:58 console
crw-rw-rw-   1 root    other    12 0x000000 dewey    Dec  6 14:33 crt
crw-rw-rw-   1 root    other    19 0x150000 dewey    Dec  6 14:33 ether
crw-rw-rw-   1 root    other    24 0x000010 dewey    Dec  6 14:33 hil1
crw-rw-rw-   2 root    other    24 0x000020 dewey    Dec  6 14:33 hil2
      ⋮
```

Note the extra field where you see the name donald. If this were a generic device file, you would see a 0.

The mknod(1m) command has been enhanced to allow the creation of generic device files or cnode-specific device files whose cnode ID differs from the cnode where the mknod command is being executed. This new enhancement is not required for any of the operations described in the System Administrator Manual; however, sophisticated cluster administrators might find this capability useful. The required syntax is documented in *HP-UX Reference Manual* under the *mknod(1m)* entry.

Use discretion in creating generic device files, since the same generic device file accessed from different cnodes will apply to different devices, and the file's ownership and permissions may not be appropriate in the context of all the cnodes.

If you havae device files in directories other than `/dev`, you probably should re-create them as cnode-specific.

Cluster Server Processes

Cluster Server Processes (CSPs) are kernel processes that handle requests received from remote cnodes (or by certain local activities) which cannot be handled by the kernel under interrupt. CSP functions include:

- all file system requests (e.g., opens, reads, writes, and mount table updates)
- swap space allocation requests
- sync requests
- PID allocation

Requests that do not require CSPs include some network protocol messages and clock synchronization.

There are 3 types of Cluster Server Processes: Limited CSPs (LCSP), General CSPs (GCSP), and User CSPs (UCSP). The limited and general CSPs are kernel processes. Even though these are processes which run in the kernel, they are shown by the `ps` command. A User CSP is a special program which runs in user address space to perform some operation on behalf of the kernel, such as `/etc/read_cct`.

Limited CSP

There is one LCSP on each cnode in the cluster. It is automatically spawned by the kernel at cluster time (on the root server) or at boot time (for diskless cnodes). **Cluster time** is when the root server executes the `cluster` command. The LCSP handles certain essential requests if no GCSP is available. It performs limited specific operations like syncs and mounts.

An LCSP is the only CSP required by a diskless cnode. This one CSP is sufficient to handle incoming requests. The LCSP always has a process ID of 3 on a diskless cnode.

General CSP

The GCSPs are created when the /etc/csp command is executed (generally from /etc/rc). The number of GCSPs created is given as an argument to the /etc/csp command. The /etc/csp command with no argument will read the /etc/clusterconf entry for the cnode to determine the number of GCSPs which should be running on that cnode. (The clusterconf file is a file describing your cluster's configuration; it is described later in more detail.) The number of CSPs is the last field of an /etc/clusterconf entry. In the default setup (using the reconfig program), the /etc/csp command in /etc/rc has no arguments. The reconfig program added the number of desired CSPs to the /etc/clusterconf file when you created your cluster. The diskless cnodes are set up so that this entry is 0.

There should be a pool of GCSPs running at all times on the root server to handle remote requests from diskless cnodes. GCSPs are not needed on diskless cnodes.

If, for example, the /etc/clusterconf entry for your root server was:

```
08000900399d:1:daisy:r:1:4
```

the /etc/csp command at cluster time will fork 4 GCSPs.

If you execute the command: /etc/csp 5, the system will either fork additional CSPs to bring the total up to five, or terminate CSPs to bring the total down to five.

The command /etc/csp 0, when executed on the root server, will terminate all GCSPs on the root server and will cause all diskless cnodes to stop functioning and ultimately crash. This is the only way to terminate GCSPs; CSPs cannot be terminated with the kill command.

Because the GCSPs will finish servicing all existing requests in progress, the effect of issuing the /etc/csp command may not be immediate.

The number of CSPs can never exceed the number specified in the ngcsp configurable operating system parameter. You must set the num_cnodes parameter when you set up your cluster kernel. For more information about ngcsp and its effect, refer to the chapter on kernel configuration and the ngcsp entry in Appendix D.

User CSP

UCSPs perform requests that cannot be handled in the kernel. They are created on demand when specific types of requests are received. The UCSP terminates after servicing the request.

Process IDs

Processes executing on different cnodes in a cluster must have unique Process IDs because many UNIX programs use the PID in temporary filenames to guarantee unique file names. In standard UNIX PIDs range from 0 to 30 000. To maintain UNIX semantics the PID on any cluster will not exceed 30 000. The cluster will cycle through 30 000 before reusing a PID just like a single multi-user system.

In a cluster, the root server is the PID allocator (server) for the cluster. PIDs are allocated in "chunks" of 50. These chunks are used so there is less network traffic than would occur if a diskless cnode needed to go to the root server for each PID required. The PID chunk is not returned to the server until all 50 have been released. After they are returned to the server, they may be reallocated to another cnode.

All cnodes keep track of the PID chunks allocated to them. They also maintain an array of available PIDs. PIDs in the available PID table are not recycled; when they are used, they are returned to the server.

Not all system process PIDs are unique in a cluster. On all cnodes the following PIDs are reserved:

PID #	Process
0	scheduler
1	init
2	page daemon
3	limited csp (diskless cnode)

Swapping in a Cluster

This section assumes you know the general swapping information in the section "Memory Management". This section covers some additional swapping information specific to HP-UX clusters.

The root server always swaps to the swap device(s) attached to it; this is the same as for a standalone system. There are two methods of swapping in a cluster for diskless cnodes:

- the diskless cnode swaps to the root server's swap area (**remote swapping**). This must be indicated in the `/etc/clusterconf` file. The `reconfig` program automatically sets up the diskless cnode to swap to the root server.
- the diskless cnode has a local disk for swapping (**local swapping**). This must be indicated in the `/etc/clusterconf` file. You must create a new kernel for cnodes that use local swap devices.

Remote swapping

When a diskless cnode uses remote swapping, it uses swap space from a pool of space found on the root server's swap devices. Swap space is dynamically allocated to a cnode as needed. When it is no longer needed, the swap space is returned to the pool by a daemon process run as a CSP.

The amount of swap space any cnode can have is controlled by two configurable operating system parameters: `maxswapchunks` and `minswapchunks`. Refer to Appendix D for more information on configurable operating system parameters.

Local swapping

When a diskless cnode uses local swapping, the swap space is found on swap devices attached to the cnode. Refer to the kernel configuration chapter for information on setting up local swapping.

Tips on Customizing your Cluster

Using NFS in an HP-UX Cluster

If NFS is configured into/out of the root server's kernel, it must also be configured into/out of all the diskless cnode kernels. If not, the cnode will not cluster. If NFS is loaded onto your system when you create a clustered environment with the `reconfig` program, `reconfig` will include NFS into the root server's kernel. Once NFS is included in the root server's kernel, `reconfig` will add NFS to each new diskless cnode's kernel.

Syncing the System

The `/etc/rc` script runs the `syncer` command on the root server. This automatically syncs all systems on the cluster. Although syncing your system regularly is important, syncs use resources. Because of this, you should to sync regularly, but no more than necessary. If you wish to do a sync that syncs only the local system, type `sync -l` (lowercase L).

Previously, the `sync` command was run from the crontab file. Check your crontab file to make sure you are not syncing the system from cron. Running `syncer` from `/etc/rc` is sufficient.

Using reconfig in an HP-UX Cluster

In general, do not use the `reconfig` or `config` program on different cnodes simultaneously. In particular, *never create a kernel simultaneously from multiple cnodes.*

CSPs

The `reconfig` program will put the number of CSPs you request into the `/etc/clusterconf` file. The `cluster` command automatically starts the limited CSP, so the total number of CSPs running will be one more than what you request with the `cluster` command.

If, in your `/etc/rc` file, you specify the number of CSPs with the `/etc/csp` command, the value in `/etc/rc` will be used instead of the value given in the `/etc/clusterconf` file. When trying to tune your cluster, you may wish to execute the `csp` command with a parameter. When you have decided on a number, you should put it in the `/etc/clusterconf` file to make the change permanent.

Changes to Commands and Files

This chapter lists the system administration commands and files that have changed for Diskless HP-UX. The commands documented here come primarily from section 1M of the HP-UX reference. The "System CDFs" section describes the HP-UX system files that must be CDFs. These are automatically converted to CDFs by the `reconfig` program when you create a clustered environment.

System Administration Commands and the Root Server

The following commands can be executed only on the root server. Note that HFS mounts and unmounts are not allowed on a diskless cnode. NFS mounts are allowed if you are running the NFS services.

```
fsck        update
fsclean     mkfs
fsdb        newfs
fuser       tunefs
mount       sysrm
umount
```

System Administration Commands that Change

This section lists system administration commands that changed to support Diskless HP-UX features; it also describes why they changed. For more information on these commands, refer to the appropriate entry in the *HP-UX Reference*.

`/etc/rc` — The `/etc/rc` script performs additional functions for the root server. For a full description of the functionality of `/etc/rc`, refer to the chapter "System Startup and Shutdown".

`/etc/brc` — The `/etc/brc` script removes a new startup flag called `/etc/rcflag` (which is a CDF). Refer to the chapter "System Startup and Shutdown" for a description of `/etc/brc`'s functionality.

`/etc/reboot`	This command has been modified to perform a clusterwide reboot when executed from the root server. When executed from a diskless cnode, only that cnode is rebooted.
`/etc/shutdown`	The `/etc/shutdown` script will perform an orderly shutdown of the entire cluster when executed from the root server. When executed from a diskless node, `/etc/shutdown` will shutdown only that cnode.
`/bin/sync`	The `sync` command causes all file system updates to be written to disk. The `lsync` command is new; it will flush all the buffers on the local system to both the file system disk and remote cnodes. It will not flush the buffers on remote cnodes.
`/etc/wall`	There is a cluster-wide version of `wall`, called `cwall`.
`/bin/who`	There is a `-c` option which displays information about the entire cluster.
`/etc/fuser`	There is a cluster-wide version of `fuser`, called `cfuser`.
`/bin/ps`	There is a cluster-wide version of `ps`, called `cps`.

New System Administration Commands

`/etc/cluster`	The `cluster` command is executed by the root server to allow other cnodes to join a cluster. Refer to *cluster(1M)*.
`/etc/csp`	The `csp` command starts the cluster server processes needed by each cnode to communicate within the cluster. Refer to *csp(1M)*.
`/bin/cnodes`	The `cnodes` command lists the cnodes in a cluster. Some scripts which should execute different commands on different cluster nodes may be modified to use the cnodes command instead of being made into CDFs. Refer to *cluster(1)*.
`/bin/getcontext`	Returns the process's context. Refer to *getcontext(1)*.
`/usr/bin/makecdf`	The `makecdf` command converts a file to a CDF. Refer to *makecdf(1M)*.

showcdf — The /usr/bin/showcdf command displays the actual path names, including any CDFs. Refer to *showcdf(1)*.

Examples using the cnodes command Some examples of how the cnodes command might be used follow:

To get the cnodename of the local system:

```
$ cnodes -m
donald
$
```

To list all cnodes in the cluster:

```
$ cnodes
daisy dewey donald
$
```

To get the cnodename of the root server:

```
$ cnodes -r
daisy
$
```

To list the status of all cnodes configured in the /etc/clusterconf file:

```
$ cnode -alC
daisy       1    daisy      ROOTSERVER
donald      2    daisy
dewey*      3    daisy
```

"*" indicates a cnode which has not joined the cluster (i.e., is not booted up).

New Files: /etc/clusterconf

The biggest change from standalone HP-UX to an HP-UX cluster is the file system setup using CDFs. These are listed in the "System CDFs" section later in this chapter. In addition to CDFs, there is a new file called /etc/clusterconf, created just for the HP-UX cluster. *This file must not exist on a standalone system.*

The /etc/clusterconf file is the cluster configuration file. It is used by many utilities and system processes to obtain information about the cnodes in the cluster. It is created and modified by the reconfig command.

Each line in /etc/clusterconf has six fields. Each field is separated by a colon (:). *Do not remove the first non-comment line* (comment lines begin with a pound sign "#"): the system may not cluster without this line. Although it is currently unused, the `reconfig` program sets up the file so the first line contains the root server's link level address. The fields in the remaining lines are used as follows:

Field Number	Description
1	Link level address of the LAN card. You can get the link address from the boot ROM display when booting, off the LAN card, or from running the `landiag` program. Refer to the *Installing and Maintaining NS-ARPA Services* manual for information on the `landiag` program.
2	Cnode number. This is a unique but arbitrary number between 1 and 255. The `reconfig` program will always assign 1 to the root server, and will assign numbers sequentially for the other cnodes.
3	Cnode name. This is automatically set to be the same as the cnode's HP-UX hostname. If the cnode's hostname is not yet defined, you will set this name in the `reconfig` program. You cannot use the names: default, localroot, remoteroot, or anything beginning with HP because they are possible attributes in the context.

4	Type of cnode. This can be:
	`r = root server`
	`c = diskless cnode`

5	Cnode number of swap server. This can be either: 1 = root server (assuming the root server has cnode number 1) current cnode number = use local swap on this cnode.
6	Number of Cluster Server Processes (CSPs) to run. This will be some number (probably between 4 and 8) on the root server, and should be 0 on all diskless cnodes.

For example, you might have the following /etc/clusterconf file:

```
0800090039dd: # clustercast address.  Do not remove.
0800090039dd:1:daisy:r:1:8
080009000565:2:donald:c:1:0
08000900297c:3:dewey:c:1:0
```

The first line is set up by `reconfig`. Note that although this first line has a comment in it, you cannot comment any of the cnode entries.

System CDFs

Listed in this section are all the system CDFs the `reconfig` program will create for your cluster. The list uses the example cluster shown in Figure 2–13 earlier in this chapter, with the root server named `daisy`, and two cnodes named `donald` and `dewey`. The list shows the CDFs, the context attributes of the CDF's subfiles, and the permissions on all the files. This information will be useful if you need to recreate part of your cluster environment.

```
/dev+:
total 6
drwxr-xr-x   5 root     root         1024 Sep  4 09:34 donald
drwxr-xr-x   5 root     root         1024 Sep  4 10:34 dewey
drwxr-xr-x   6 root     root         2048 Sep  4 08:28 localroot
```

`/etc/btmp+:` *<subfiles exist for each cnode only if explicitly created, i.e., log into "donald" and type "touch /etc/btmp">*

```
total 0

/etc/checklist+:
total 2
-r--r--r--   1 root     other         156 Sep  4 08:40 daisy
-r--r--r--   1 root     other           0 Sep  4 10:08 dewey
-r--r--r--   1 root     other           0 Sep  4 09:08 donald

/etc/conf/dfile+:
total 4
-rw-rw-rw-   1 root     other         699 Sep  4 08:45 daisy
-rw-rw-rw-   1 root     other         585 Sep  4 10:13 dewey
-rw-rw-rw-   1 root     other         585 Sep  4 09:13 donald

/etc/inetd.conf+:
total 8
-rw-r--r--   1 root     bin          1187 Sep  2 08:36 daisy
-rw-r--r--   1 root     bin          1187 Sep  4 10:12 dewey
-rw-r--r--   1 root     bin          1187 Sep  4 09:12 donald

/etc/inittab+:
total 4
-rwxr-xr-x   1 root     other         778 Sep  4 08:26 daisy
-rwxr-xr-x   1 root     other         775 Sep  4 10:36 dewey
-rwxr-xr-x   1 root     other         775 Sep  4 09:36 donald
```

```
/etc/ioctl.syscon+: <entries for cnodes exist once the cnode has booted>
total 4
-rw-r--r--   1 root     root         35 Sep  4 08:12 daisy
-rw-r--r--   1 root     root         39 Sep  4 10:34 dewey
-rw-r--r--   1 root     root         39 Sep  4 09:34 donald

/etc/netstat_data+:
total 0

/etc/ps_data+: <entries exist for diskless cnode only after a "ps" is
                executed from the cnode>
total 146
-rw-rw-r--   1 root     root      36464 Sep  4 08:28 daisy
-rw-rw-r--   1 root     other     35684 Sep  4 10:39 dewey
-rw-rw-r--   1 root     other     35684 Sep  4 09:39 donald

/etc/rcflag+:
total 0
-rw-rw-rw-   1 root     root          0 Sep  4 08:29 daisy
-rw-rw-rw-   1 root     root          0 Sep  4 10:31 dewey
-rw-rw-rw-   1 root     root          0 Sep  4 09:31 donald

/etc/reboot+:
total 108
-r-xr--r--   1 root     other      1490 Sep  4 08:40 localroot
-r-xr--r--   1 root     other     52012 Sep  4 08:40 remoteroot

/etc/ttytype+:
total 4
-r--r--r--   1 root     other        48 Sep  4 08:40 daisy
-r--r--r--   1 root     other        48 Sep  4 10:08 dewey
-r--r--r--   1 root     other        48 Sep  4 09:08 donald

/etc/utmp+:   <entries for diskless cnodes are created once the cnode
               boots>
total 4
-rw-r--r--   1 root     root        288 Sep  4 08:50 daisy
-rw-r--r--   1 root     root        288 Sep  4 10:52 dewey
-rw-r--r--   1 root     root        288 Sep  4 09:52 donald
```

/etc/wtmp+: *<subfiles exist for each cnode only if explicitly created, i.e., log into "donald" and type "touch /etc/wtmp">*

```
total 4
-rw-rw-rw-   1 adm      adm          1440 Sep  4 08:50 daisy

/hp-ux+:
total 3254
-rwxr-xr-x   1 root     other      880637 Sep  4 08:46 daisy
-rwxr-xr-x   1 root     other      769844 Sep  4 10:14 dewey
-rwxr-xr-x   1 root     other      769844 Sep  4 09:14 donald

/usr/adm+:
total 4
drwxr-xr-x   3 adm      adm          1024 Sep  4 08:35 daisy
drwxr-xr-x   3 adm      adm          1024 Sep  4 10:31 dewey
drwxr-xr-x   3 adm      adm          1024 Sep  4 09:31 donald

/usr/bin/cancel+:
total 156
-r-sr-sr-x   1 lp       bin         77112 Sep  4 08:41 localroot
-r-xr-xr-x   1 bin      bin            71 Sep  4 08:41 remoteroot

/usr/bin/ct+:
total 200
-r-sr-xr-x   1 root     other       99876 Sep  4 08:44 localroot
-r-xr-xr-x   1 bin      bin            65 Sep  4 08:44 remoteroot

/usr/bin/cu+:
total 132
-r-sr-xr-x   1 root     other       64940 Sep  4 08:44 localroot
-r-xr-xr-x   1 bin      bin            66 Sep  4 08:44 remoteroot

/usr/bin/disable+:
total 156
-r-sr-sr-x   1 lp       bin         76912 Sep  4 08:41 localroot
-r-xr-xr-x   1 bin      bin            67 Sep  4 08:41 remoteroot

/usr/bin/slp+:
total 60
-r-xr-xr-x   1 bin      bin         28084 Sep  4 08:41 localroot
-r-xr-xr-x   1 bin      bin            55 Sep  4 08:41 remoteroot
```

```
/usr/etc/yp/ypserv.log+:
total 0

/usr/etc/yp/ypxfr.log+:
total 0
```

/usr/lib/cron/FIFO+: *<created for each cnode when "cron" is run from the cnode>*

```
total 0
prw-------    1 root     root           0 Sep  4 08:12 daisy
prw-------    1 root     root           0 Sep  4 10:12 dewey
prw-------    1 root     root           0 Sep  4 09:12 donald
```

/usr/lib/cron/log+: *<created for each cnode when "cron" is run from the cnode>*

```
total 4
-rw-rw-rw-    1 root     root         417 Sep  4 08:55 daisy
-rw-rw-rw-    1 root     root          61 Sep  4 10:32 dewey
-rw-rw-rw-    1 root     root          61 Sep  4 09:32 donald

/usr/lib/libgkssb.a+:
total 502
-r--r--r--    1 bin      bin       132820 Sep 21  1987 HP-MC68010
-r--r--r--    1 bin      bin       121344 Sep 21  1987 HP-MC68881

/usr/lib/libsb1.a+:
total 1926
-r--r--r--    1 bin      bin       528036 Sep 21  1987 HP-MC68010
-r--r--r--    1 bin      bin       446528 Sep 21  1987 HP-MC68881

/usr/lib/lpsched+:
total 212
-r-sr-sr-x    1 root     bin       105536 Sep  4 08:41 localroot
-r-xr-xr-x    1 bin      bin          202 Sep  4 08:41 remoteroot

/usr/lib/uucp/uucico+:
total 326
-r-sr-xr-x    1 uucp     other     162596 Sep  4 08:44 localroot
-r-xr-xr-x    1 uucp     other        199 Sep  4 08:44 remoteroot

/usr/spool/cron+:
total 4
drwxr-xr-x    4 root     sys         1024 Sep  2 08:43 daisy
drwxr-xr-x    4 root     sys         1024 Sep  4 10:12 dewey
drwxr-xr-x    4 root     sys         1024 Sep  4 09:12 donald
```

Subsystem Administration

System Accounting

System accounting has not changed. `/usr/adm` is a CDF, so system accounting will be done on a per-cnode basis within the cluster. Disk usage accounting is only done on the root server.

Mail

Mail can be sent via either `sendmail` or UUCP. In either case, the mail subsystem is set up so all mail transactions appear to occur on the root server. Mail from another system to a cnode in the cluster should be addressed to that same user on the root server. For example, if `joe` on a non-cluster system wishes to send mail to `jme` on the example system `donald`, it should be sent to the following address: `daisy!jme`, since `daisy` is the server for the cluster.

Non-local mail from a user on any cnode will appear to have originated on the server node. Local mail originating from any cnode will be handled as if it were local mail on the server node.

LP Spooling

All printers on the root server can be added to the lp spooler system. Requests can be spooled from any cnode, but the scheduler runs on the root server only. Spooled printers cannot be attached to diskless cnodes.

UUCP

Although all UUCP lines must be on the root server, UUCP transfers can be initiated from any cnode. UUCP is described in the *Concepts and Tutorials* article called "UUCP".

Cron

The `/usr/spool/cron` directory is a CDF, meaning each cnode has its own copy of cron files.

Contents

3

System Startup and Shutdown

From the time you switch on power to the computer until you have successfully logged in, many tasks are performed automatically by the system. These tasks include: testing the computer hardware, loading and initializing the operating system, communicating messages to the user(s), and running scheduled routines. To manage your HP-UX system effectively, you must understand which tasks are performed at which times.

This chapter provides you with a description of the computer's activities from power-up through successful completion of the `login` routine. This chapter also provides a step-by-step procedure for a controlled shutdown of your system and for a system recovery. Throughout this chapter, you will learn about features of HP-UX that can ease your role as system administrator.

System Startup Functions

System startup, often referred to as **booting the system,** is getting your computer from a powered down state to a functional state where HP-UX is running and ready to take input.This section of the chapter tells how to get your system to a usable state (how to boot the system), and discusses the sequence of events that happens internally (what you don't see on your screen, but what is happening to get your system to the usable state).

The bootup sequence is slightly different depending on whether you have a standalone machine or a diskless node on a cluster of machines. The sequence is the same for a standalone machine and for the root server on a cluster of machines. For information on booting a diskless cnode refer to the section "Booting a Cluster".

Booting the System (Standalone or Root Server)

This section explains the steps you must follow to boot your system. The details of each step (that is, what happens internally) are discussed in later sections.

System startup is made possible through a piece of software called the boot ROM (Read Only Memory). It was specifically developed to support a wide variety of present and future Hewlett-Packard operating systems. Different operating systems use different aspects of the boot ROM; the following description of the boot ROM's operation focuses on its use with HP-UX.

If you are unfamiliar with the Series 300 boot ROM, read the section later in this chapter called "The Boot ROM" before continuing. This will ensure that the correct operating system and system console are found during the boot procedure.

1. Turn power on to the hard disk drive containing the HP-UX operating system. Wait until the disk drive is ready before continuing with the next step. (Refer to the operator's manual for specifics on when your disk is ready.)

2. Turn power on to all peripherals connected to your computer. If your system has an I/O expander, make sure that power to the expander is also turned on.

3. Turn power on to your Series 300 computer.

   The boot ROM will send messages to your screen. Figure 3–1 shows a typical display of the boot ROM's operation. *The display varies depending on the version of the boot ROM.*

```
Copyright 1987,                         :HP7946, 1400, 0, 0
Hewlett-Packard Company.                1H SYSHPUX
All Rights Reserved.                    1D SYSDEBUG
                                        1B SYSBCKUP
BOOTROM Rev. C
Bit Mapped Display
MC68050 Processor
Keyboard
HP-IB
HP98620B
HP98644 at 9
HP98625 at 14
HP98643 at 21, 080009000001
4182016 Bytes




SEARCHING FOR A SYSTEM (RETURN To Pause)
RESET To Power-Up
```

Figure 3-1. Boot ROM Display

1. Choose the correct operating system. If you are booting in unattended mode this step is automatic. If you are booting in attended mode you will need to enter the correct operating system (for example by pressing `1H`). Refer to the section called "The Boot ROM" for information on finding an operating system and on finding the system console.

   Once the operating system (HP-UX) is found, the screen will clear and the messages, including "`Booting /hp-ux`", will appear. The screen will clear again, and a series of messages will appear. Figure 3–2 shows *parts* of a typical display.

```
CONSOLE is ITE
ITE + 1 port(s)
MC68020 processor
Internal HP-IB Interface - System controller at select code 7
HP98644 RS-232 Serial Interface at select code 9
HP98625B High-Speed HP-IB Interface - system controller at select code 14
HP98643 LAN/300 Link at select code 21
Bit Mapped Display at 0x560000
HP98620B DMA
real mem = 4182016
using 10 buffers containing 40960 bytes of memory
Local Link Address = (hex) 080009003626
Root device major is 0, minor is 0xe0210
Swap device table: (start & size given in 512-byte units)
entry 0 - auto-configured on root device; start=2840, size=127976
-- BATTERY BACKED REAL TIME CLOCK
avail mem=3624960
lockable mem=3522560
           copyright information
/etc/bcheckrc:
/etc/fsclean: /dev/dsk/0s0 (root device) ok
File system is OK, not running fsck
/etc/brc:
/etc/rc:
Is the date Fri Jun 19 10:53:58 MDT 1987 correct? (y or n, default: y)
```

Figure 3-2. Messages during Bootup

Take the time to read your display. It is informative and will tell you about many possible error conditions.

2. Input the date and time information requested.

   You will be prompted to confirm and, if incorrect, set the date. If the date and time are correct, press the [Return] key. If you do not respond within 10 seconds the system assumes the date and time are correct and continues the boot process. If the displayed date is incorrect, respond to the prompt with [n] [Return]. You will then be prompted to enter the correct date. Respond by typing the date in the format shown below:

   ```
   MMddhhmm{yy}
   ```

   For more information on setting the date, refer to the chapter on "Periodic System Administrator Tasks", the section "Setting the System Clock".

3. Log into your system.

   You will see a login prompt. The system is now ready for use.

On single-user systems or small multi-user systems, you may wish to allow any user to power up the system. If this applies to your system, write a short document that describes the procedure for booting HP-UX (and changing system run-levels if it applies) and distribute the document to all users. Knowing the specific details of your system—the hardware, configuration files, system states, and needs at your installation—should enable you to write a streamlined procedure for your users. This can ease your administration tasks and provide system users with more flexibility. Typically the system may be booted by simply turning on power to peripherals and the computer.

Booting a Cluster

This section deals with booting a cluster and booting a diskless cnode within a cluster. If you are not familiar with HP-UX clusters, refer to the section on "Cluster Concepts" in the Concepts chapter.

Cluster Startup

The root server must be booted first. Boot the root server the same way you boot a standalone system.

Booting the Diskless Cnode

Each diskless cnode on the cluster requires a Rev B or later boot ROM. This boot ROM enables the cnode to boot across LAN.

If you wish to boot in **unattended mode** (i.e., an automatic boot), make sure all the following are true:

- there is no disk attached to your diskless cnode that contains a bootable system.

  Although this is not normally the case, it may be that you run the workstation part-time as a standalone system and part-time as a diskless cnode.

- the cnode is part of only one cluster

  This means that only one root server on the cluster's LAN has and entry in the `/etc/clusterconf` file for your cnode.

Refer to the section "Selecting an Operating System" for a list of the operating system search sequence.

If you wish to boot in **attended mode,** press the space bar to force a selection menu on your screen. You must boot in attended mode if your cnode is part of more than one cluster on the LAN. That is, if more than one root server has an `/etc/clusterconf` file containing an entry for your cnode, you must boot in attended mode. This is because you cannot depend on the order of finding operating systems on the same LAN. Refer to the section "Selecting an Operating System" for details on attended mode.

Before you can boot across the LAN, you must add the cnode to the cluster using `reconfig`, and the root server must be booted, clustered, and have the boot server running. For more information on adding a cnode to the cluster, refer to the section "Adding a Cnode" in the chapter "Customizing your HP-UX System".

Overview of Internal Functions of System Startup

When you boot the system (described above), many things happen that you do not see on your screen. The "behind the scenes" startup for the typical HP-UX system follows the stages listed below (refer to Figure 3–3). Each is described later:

1. The boot ROM is started. It tests the hardware and loads a secondary loader.
2. Control transfers to the HP-UX operating system.
3. HP-UX starts a process called `init`.
4. `init` brings the system to the default run-level, as specified by the `initdefault` entry in the `/etc/inittab` file. As shipped, this will be run-level 2.

   The `init` process also runs the `/etc/bcheckrc`, `/etc/brc`, and `/etc/rc` command scripts.
5. `init` starts processes called **getty**s that give login prompts on terminals.
6. A user logs in.

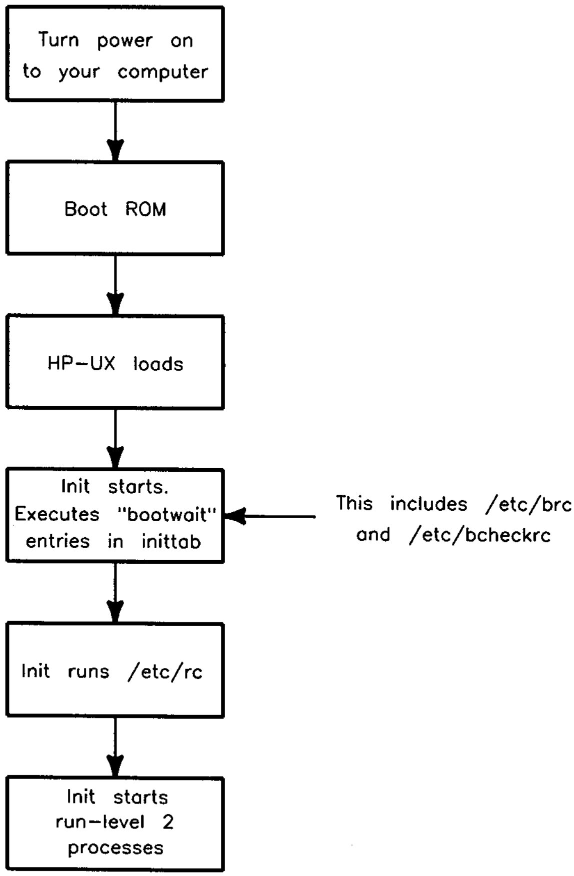


Figure 3-3. Boot-Up Sequence

The Boot ROM

When you turn your computer on, the boot ROM goes through the following sequence:

1. Checks for and tests interface cards, RAM, and internal peripherals (refer to your computer's installation guide for a description of these tests).
2. Searches for and assigns an input device (keyboard) and an output device (display) to use as a console (refer to the section "Selecting a System Console" later in this section).

3. Polls all supported mass storage devices and LANs connected to the computer to search for an operating system to boot (refer to the section "Selecting an Operating System" later in this section).
4. Loads the first bootable operating system found (unattended mode) or the operating system chosen (if attended mode).
5. Passes control to that operating system. Once the operating system (in this case, HP-UX) gains control, many tasks are performed. Later sections describe these tasks.

Selecting a System Console

The **system console** is the first keyboard and display (or terminal) found by the boot ROM. It is given a unique status by HP-UX and associated with the special device file `/dev/console`. All boot ROM error messages, HP-UX system error messages, and certain system status messages are sent to the system console. Under certain conditions (for example, the single-user state), the system console provides the only mechanism for communicating with HP-UX.

The boot ROM and HP-UX operating system assign the system console function according to a prioritized search sequence. HP-UX's search for a system console terminates as soon as one of the following conditions is met:

1. A built-in serial interface, HP 98626A, HP 98628A[1], HP 98642A, or HP 98644A RS-232C serial interface is present with the remote bit[2] set. If more than one serial interface card with its remote bit set is present, the one with the lowest select code is used. In the case of the HP 98642A (4-channel multiplexer), port 1 is used.

---

[1] The HP 98628A Datacomm Interface Card with its remote bit set is not supported as a remote console by the boot ROM; however, it is supported as the system console by the HP-UX operating system. Therefore, when an HP 98628A card is used and has its remote bit set, the boot ROM sends messages to the next console found, but HP-UX sends its messages to the terminal associated with the HP 98628A card. This configuration is not supported.

[2] On the HP 98626A Serial Interface board the remote switch is set by cutting a jumper as described in the installation manual supplied with your computer. On the HP 98628A, HP 98642A, HP 98644A, and built-in RS-232 interface cards the remote bit is set by setting a switch on the board as described in each board's installation manual.

2. An internal bit-mapped display is present (internal bit-mapped displays are not associated with any select code so there can be only one internal bit-mapped display). This is true if any of the following is present: Model 310 built-in video output, HP 98542A, HP 98543A, HP 98544A, HP 98545A, HP 98547A, HP 98548A, HP 98549A, HP 98720, or HP 98550A boards. Note that an HP 98700H display station can have its display interface card (HP 98287A or HP 98720) configured for internal or external control. If it is configured for external control, it is never chosen.
3. An external bit-mapped display with select code 132-255 is present. This can be any of the following: HP 98548, HP 98549, or HP 98550. They are selected in order of increasing select code.
4. An HP 98546A compatibility video interface is present.
5. A built-in serial interface, HP 98626A, HP 98628A, HP 98642A, or HP 98644A RS-232C serial interface is present without the remote bit set. If more than one is present, the one with the lowest select code is used. In the case of the HP 98642A (4-channel multiplexer), port 1 is used. The boot ROM *does not* recognize the HP 98628 serial interface card as console when this condition is met; however, HP-UX does.

If none of the above conditions are met, no system console exists. While the boot ROM tolerates this, HP-UX will not.

Line Control Switch Pack Settings

The boot ROM requires that the Line Control Switch Pack settings on the HP 98626A RS-232C Serial Interface card be set to the same setting as your remote terminal. HP-UX resets these values to system defaults on log in. These values are as follows:

Stop Bits	should be set to 1.
BaudRate	should be set to 9600 bps.
Parity/DataBits	should be set to 0's/7.
Enq/Ack	should be set to NO.
Pace (Handshake)	should be set to XON/XOFF.

To make the above settings on your HP 98626A Serial Interface card, set your cards' Line Control Switch Pack (U-2) switches as shown in Table 3–1:

Table 3–1. Line Control Switch Pack Settings

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
1	1	0	0	1	0	1	0

Additional settings for Handshake Type bits 6 and 7 of the Line Control Switch Pack are shown in Table 3–2:

Table 3–2. Additional Line Control Switch Pack Settings

Bit 6	Bit 7	Handshake Type
0	0	ENQ/ACK
1	0	XON/XOFF
0	1	NO HANDSHAKE
1	1	NO HANDSHAKE

Note that the other switch settings for the Line Control Switch Pack on the HP 98626A card are defined in the installation guide supplied with that card.

Selecting an Operating System

When more than one operating system is present both the location of the operating systems and the type of media on which they are stored determine which operating system is loaded. Operating systems can be on different mass storage media connected to the computer you are booting, or can be located on root servers on your computer's local area network (LAN).

The boot ROM has two modes for selecting an operating system: attended and unattended. In **attended** mode, you can select the operating system to boot from all the operating systems found on the mass storage devices. In **unattended** mode, the boot ROM automatically boots the first operating system in its search sequence.

If for any reason, the boot ROM is unable to find and load an operating system, informational and/or error messages are displayed on the system console. These messages are described in the "Boot ROM Error Messages" section of the hardware installation manual supplied with your HP-UX system.

Unattended Mode

Use the unattended mode of booting if you have only one bootable operating system on line, or if you know the operating system you wish to boot is the first operating system the boot ROM will find.

The boot ROM searches a prioritized list of select codes. This includes LAN cards, disks, and other peripheral devices. The first operating system found on one of these devices will be booted. If no operating system is found, the list will be searched again until a system is found. This means that disks not found at power-up will be found after their initialization is complete.

To boot the operating system:

- Make sure the desired operating system is the first system found, following the prioritized list below.
- Make sure the device holding the operating system is fully powered up and has achieved a ready state *before* powering up your Series 300 computer. If the disk drive has not completed its power-up sequence (which may require several minutes on some disk drives), the boot ROM will not be able to access the disk and load the system. If you are booting a diskless cnode, make sure the root server has completed the bootup sequence, including clustering and boot daemons.

To find an operating system, the boot ROM uses the following search sequence:

1. Any external disk at select codes 0-31, with bus address, unit number, and volume numbers of 0.
2. SRM at Select Code 21, Volume 8.
3. LAN at Select Code 21.

   If there are multiple remote servers at the same select code, the first server to respond is the first system listed on the menu (if attended mode) or the remote system selected (if unattended mode). This can change each time you boot your system.

4. HP 98259 (Bubble Memory) at Select Code 30.
5. HP 98255 (EPROM card) on Unit 0.
6. ROM systems (for ROM-based operating systems).
7. Other external disks at Select Codes 0 through 31, not bus address, unit number, and volume number of 0.
8. Other Shared Resource Managers at Select Codes 0 through 31 (not Select Code 21, Volume 8).
9. Other LANs, not at select code 21.
10. HP 98259 (Bubble Memory) on Select Code 0 through 29.
11. Remaining HP 98255 (EPROM cards).

Multiple units at the same select code and bus address are searched before moving to the next ascending select code or bus address. Thus, an HP-UX system on the root mass storage device at select code 14, bus address 0, unit 0, is found and loaded before any operating systems at select code 14, bus address 0, unit 1.

Note that you must have boot ROM rev C or later to boot from a SCSI disk drive. At rev C, SCSI disk drives are treated the same as any other disk drive.

You must have boot ROM rev B or later to boot across a local area network (LAN).

Attended Mode

If you do not want to default to the first operating system found by the boot ROM, you must enter the attended mode of selection.

You enter attended mode by typing a space, [Return], or any letter or number, during the time before a default system is found, but after the keyboard has been initialized. You can tell when the keyboard has been initialized because the word `Keyboard` appears on the screen. The character used to enter attended mode is used as part of the string to select the operating system.

The best way to enter the attended mode is to hold down the space bar until the word `Keyboard` appears in the installed interfaces list on the left side of the screen.

When attended mode is activated, the boot ROM displays a boot selection menu. This menu lists all accessible operating systems as the boot ROM finds them. The operating systems will be displayed by their names, listed under a mass storage device name and address. If you are booting a diskless cnode, the root server (or servers) will be listed similar to: `:LAN, 21, root_server_name`.

Using Attended Mode to Boot the Backup Kernel

On bootup press the space bar to get into the attended mode. On the right column of the bootup screen, you will see one or more operating systems. The possible HP-UX operating systems are SYSHPUX, SYSDEBUG, and SYSBCKUP. SYSHPUX contains a pointer to the file `/hp-ux` (the kernel). SYSBCKUP contains a pointer to the file `/SYSBCKUP` (a backup kernel). SYSDEBUG is used only for device driver writing. For example, if you have both an HP-UX kernel (shown as SYSHPUX) and an HP-UX backup kernel (shown as SYSBCKUP) on a 7946, at select code 14, bus address 0, unit number 0, the following will appear on the menu:

```
:HP7946, 1400, 0, 0
   1H SYSHPUX
   1B SYSBCKUP
```

Select the backup kernel by typing `1B`.

Diskless cnodes should not use the SYSBCKUP*option.* If your diskless cnode cannot boot from it's normal kernel, then you should create a new kernel for the cnode (from the root server). Kernel creation is described in the chapter "Kernel Customization". To create a kernel for a cnode from another cnode, you must use the `config` command, not the `reconfig` command. You must explicitly put the kernel into the correct version of `/hp-ux` using the guidelines given in "Cluster Concepts" in the chapter called "System Management Concepts".

If you are in unattended mode, SYSHPUX is the HP-UX operating system the boot ROM will load.

Note

If you depend on the boot ROM's search sequence to boot a default system, make sure the mass storage device containing the operating system has completed its power-up cycle and is ready for use before powering up the computer. The order in which the operating systems are found depends on both the search sequence and on the availability of the mass storage devices. You cannot depend on the order of finding multiple remote servers across LAN.

Using Attended Mode to Boot a Diskless Cnode

If you have a computer that has its own bootable system, and can be a member of a cluster, then you must get in to the attended mode to boot into the cluster.

For example, if you have an HP7946 disk drive with HP-UX, and you are able to boot to a cluster, you will see the following on the right part of the menu:

```
:HP7946, 1400, 0 0
   1H SYSHPUX
   1D SYSDEBUG
   1B SYSBCKUP

:LAN, 21, daisy
   2H SYSHPUX
   2D SYSDEBUG
   2B SYSBCKUP
```

To select the HP-UX operating system on the root server, named `daisy`, you would type 2H.

If you have trouble booting across the LAN, refer to the "Troubleshooting" appendix.

HP-UX Takes Control

Once the HP-UX operating system has been found and loaded successfully, many tasks are performed automatically by HP-UX. The first task is to search for the root file system. The **root file system** is the portion of the file system that forms the base of the file system hierarchy (that is, the portion of the file system on which other volumes can be mounted). The root file system contains the files required for HP-UX to properly run (for example, the kernel in `/hp-ux`). This is generally the disk you booted from. In the case of a diskless cnode, this is the root disk of your computer's root server.

Note The documentation which follows describes the operation of the system as shipped to you; however, by altering certain configuration or system files, any of the following procedures can change. If, for example, you write your own `/etc/rc` script, the paragraphs which follow may no longer apply.

HP-UX Starts the Init Process

After finding the root file system, HP-UX sets up its first process, `/etc/init`. The `/etc/init` process becomes process ID one (1) and has no parent. For more information on processes, refer to the chapter called: "System Management Concepts".

The `init` process reads a configuration file called `/etc/inittab`. Each line in the file `/etc/inittab` describes an activity for the system to perform when entering a given run-level. A **run-level** can be described as a set of processes allowed to run at a given time.

If initdefault is not specified, then `init` will prompt the user for a value when the system reaches this point. After `init` begins, as it makes the first transition into run-levels 0-6, all entries marked `boot` or `bootwait` are executed. This includes executing the `bcheckrc` and `brc` programs.

/etc/bcheckrc

The `/etc/bcheckrc` (Boot CHECK Run Command) program checks to see if the system was properly shutdown. To determine if the system was properly shut down, `bcheckrc` calls the `fsclean` program (refer to Figure 3–4. `fsclean` checks each file system of type `hfs` in `/etc/checklist` to see if there might be a consistency problem. To do this, `fsclean` looks at a flag called the **clean byte** in the primary superblock of each file system. When a file system is created, the clean byte flag is set to FS_CLEAN. When the file system is mounted (using the `mount` command), the clean byte flag is set to FS_OK. During a normal shutdown (that is, during execution of the `reboot` or `shutdown` command), the clean byte is reset to FS_CLEAN. So, under normal conditions, the file system can be unmounted and set to FS_CLEAN, or mounted and set to FS_OK. (Refer to "Shutting Down the System" later in this chapter for more information.)

If, when `fsclean` checks the clean byte, it finds the file system is unmounted and set to FS_OK, then the file system might be in an inconsistent state (due to a crash or other incorrect shutdown). In this case `bcheckrc` will run `fsck` automatically using the `preen` mode. This will correct most errors found. Refer to the discussion on the preen mode in Appendix A of this manual.

On a system with multiple file systems, you may see the system come up, run `fsck`, reboot, then run `fsck` again. The first `fsck` fixes the root volume, the second `fsck` fixes all the additional, as yet unmounted, disks. Anytime `fsck` makes changes to the root (/) file system the system must be rebooted to force the memory resident disk information to be consistent with the changes `fsck` made to the disk. The system will reboot automatically. The second `fsck` does not cause the system to reboot because the file systems it fixes are not mounted.

If the `fsck` command run from `bcheckrc` fails for any reason, `bcheckrc` starts a shell with the prompt `(in bcheckrc)#`, along with instructions to run `fsck` manually. If this occurs, *you must run* `fsck` *to ensure the integrity of your file system.*

Some file system problems must be fixed manually because of the risk of data loss. When you have completed running the manual `fsck`, you may be instructed to reboot the system. If you are instructed to reboot, *you must reboot the system, using* `reboot -n`*, to ensure the integrity of your system.* If `fsck` does not tell you to reboot, simply exit the shell by typing [CTRL][D]. The `bcheckrc` program will then proceed.

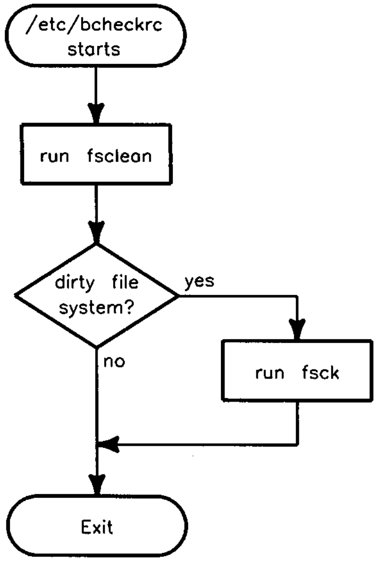


Figure 3-4. Flow of /etc/bcheckrc Program

/etc/brc

The `/etc/brc` program performs the following tasks (refer to Figure 3–5):

- sets the system PATH variable
- downloads the floating point microcode if your system is set up for the floating point accelerator
- removes the `/etc/rcflag` file (used later as a check for startup condition)
- checks for, and removes, the `/etc/mnttab` file on a standalone or root server system. The `/etc/mnttab` file contains a list of mounted file systems. This file will be re-created later when `/etc/rc` re-mounts the file systems.

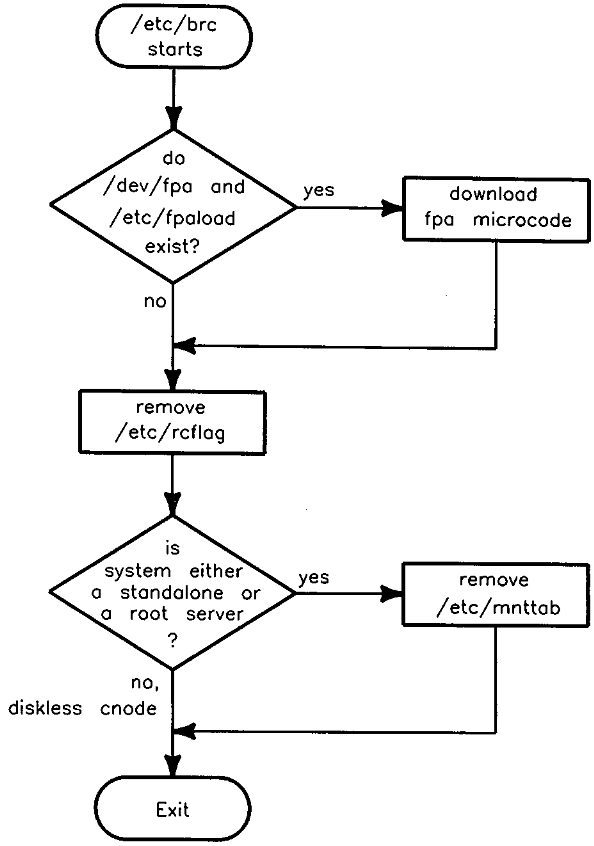


Figure 3-5. Flow of /etc/brc Program

Init Brings the System to Run-Level 2

Once the booting processes have run, `init` comes up in the initdefault run-level as defined in `/etc/inittab`. As shipped, the **initdefault run-level** is run-level 2. Refer to the *init(1M)* and *inittab(4)* entries in the *HP-UX Reference* for more information.

Each time `init` changes run-level, either at boot time or when invoked manually, `/etc/inittab` is read. After reading `/etc/inittab` and signaling processes as required, a line in `/etc/inittab` invokes `/etc/rc`.

A file called /etc/rcflag is removed at boot time (with the /etc/brc program) and during a system shutdown (with the /etc/shutdown script). The /etc/rc script checks for the existence of /etc/rcflag to determine whether to perform system initialization and to start various **daemon** (background) processes.

/etc/rc

The /etc/rc script performs many functions, depending on the status of the machine (refer to Figure 3–6 and Table 3–3).

The /etc/rc script consists of a main script program and several shell functions (a shell function is essentially a subroutine in a shell program). The main program checks to see if it is boot time. If so, it determines if you are running as a standalone system, a root server, or a diskless cnode. It then calls shell functions applicable to your system. When finished, it calls a shell function called localrc. This is the shell function you should use to customize the /etc/rc script. It should contain any tasks you wish to perform that are not part of the standard /etc/rc functions.

In the localrc shell function, you can add commands you wish to perform every time the system is booted or whenever there is a change in run-level which init does not handle.

The /etc/rc script sets your system host name in the variable called SYSTEM_NAME.

The /etc/rc script performs the functions shown in Table 3–3, depending on your system state (standalone, root server, or diskless cnode). Note that some commands (for example cron) may not be available or installed. The /etc/rc script checks for the existence of all such commands before attempting to run them.

Table 3-3. /etc/rc functions

Function	Stand-alone	Root Server	Diskless Cnode
initialization: set TZ and other variables, setup vt gateway	X	X	X
local functions: any functions you put into the script	X	X	X
set the date	X	X	
create the `/etc/setmnt` file	X	X	
mount all `hfs` volumes listed in the `/etc/checklist` file	X	X	
start the syncer	X	X	
start the lp scheduler	X	X	
clean uucp and editor files	X	X	
start networking	X	X	X
start swapping to all swap devices in `/etc/checklist`	X	X	X
start cron	X	X	X
start pty allocation daemon	X	X	X
start vtdaemon	X	X	X
list files found in `/tmp` and `/usr/tmp` directories	X	X	X
clean accounting logging files	X	X	
start CSPs		X	X
start the remote boot daemon		X	

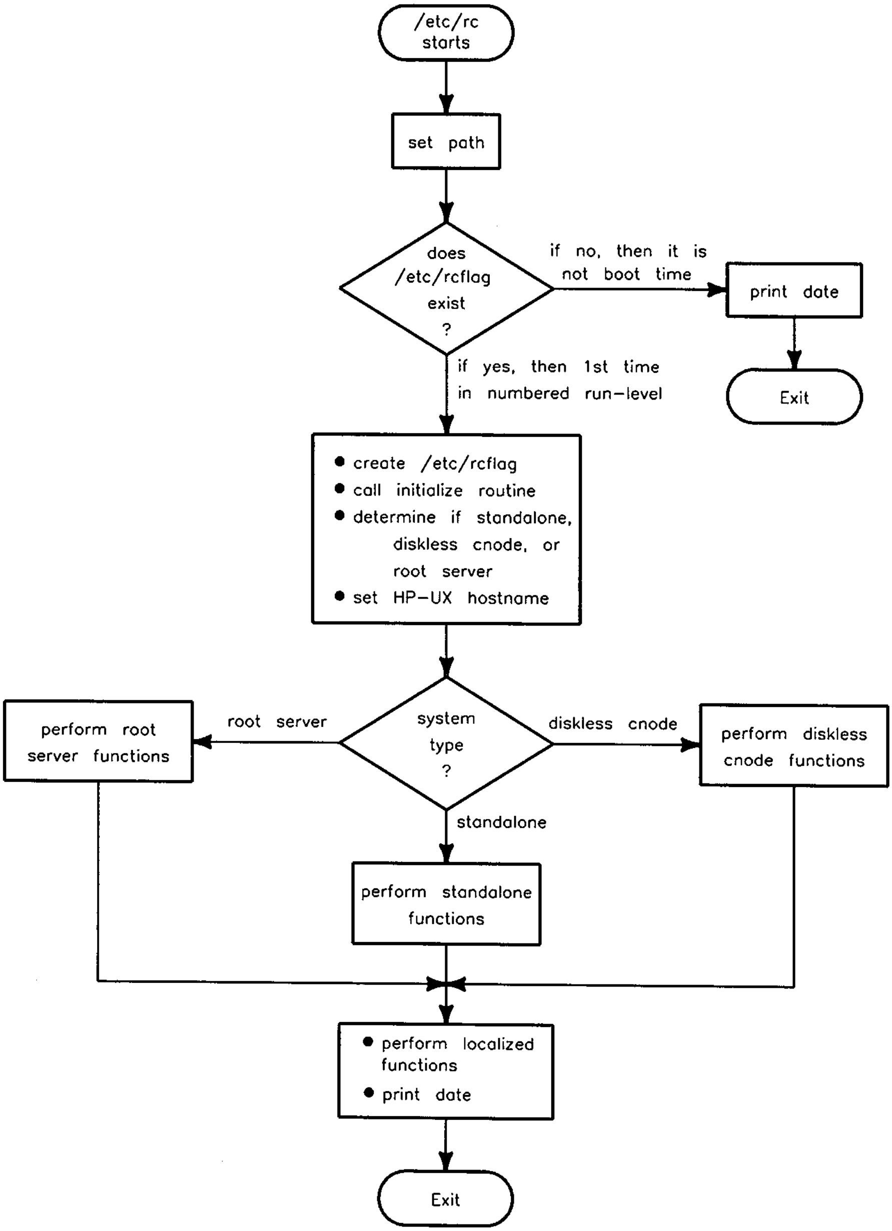


Figure 3-6. Flow of /etc/rc Program

Init Spawns gettys to Cause a Login Prompt

Once `/etc/rc` has finished its run-level 2 execution, control returns to `/etc/init` which now executes the commands from the command field of all run-level 2 entries in `/etc/inittab`. Typically, `/etc/inittab`'s run-level 2 command field entries consist of `/etc/getty` commands, one for each terminal on which users are to log in. This sets up, on each terminal, the process that runs the login program and eventually runs the shell program once someone successfully logs in.

The `/etc/inittab` entries are of the form:

```
id:rstate:action:process
```

where `id` is a unique two-character identification code, `rstate` specifies the run-levels to which this entry applies, `action` tells `init` what to do with the entry, and `process` is an HP-UX command to execute. Run-levels are described in *inittab(4)* of the *HP-UX Reference.*

The action `respawn` tells `init` to re-create a process at the console in the specified run-levels, each time the last invocation terminates. Leaving the `rstate` field empty (as shown below) will cause execution in all run-levels (0 through 6). The example below sets up a `getty` process for the console in all possible run-levels:

```
co::respawn:/etc/getty console H
```

As shipped to you, `/etc/getty` is invoked only for the system console in run-level 2. You will need to customize your system by adding additional `getty`s to `/etc/inittab` for each terminal supported by your system. (Refer to "Adding Peripheral Devices" in Chapter 4 for more information.)

The `/etc/getty` command is the first command executed for each login terminal. It specifies the location of the terminal and its default communication protocol, as defined in the `/etc/gettydefs` file. It prints the `/etc/issue` file (if present) and it causes the first `login:` prompt to be displayed. Eventually, the `getty` process is replaced by your shell's process (refer to the following section, "A User Logs In").

When that process is terminated (when you log out), the `/etc/init` process is signaled and takes control again. The `init` process then checks `/etc/inittab` to see if the process that signaled it is flagged as continuous ("respawn"). If the process is continually respawned, `init` again invokes the command in the command field of the appropriate `inittab` entry as described above (that is, the getty runs and a new login: prompt appears). If the process is not flagged as continuous, it is not restarted.

Note Do *not* add /etc/getty entries to /etc/inittab for terminals which are not present (unless action is "off". If you do, the getty process will repeatedly send an error message to the console, wait 20 seconds, and then exit.

A User Logs In

The tutorials supplied with your system describe how to log in (gain access to the system). This section describes the function of the operating system during that process.

1. The login process begins when you type in a user name in response to the login: prompt. Once the user name has been entered, /etc/getty executes the login program with the user name as an argument; /bin/login checks the name against the list of valid user names kept in /etc/passwd.

2. If the user name is valid, login checks to see if there is a password associated with the user name (the encrypted form of the password is stored in /etc/passwd). If there is a password associated with the user name, the system prompts for a password. The password you type in is encrypted and compared to the encrypted password stored in /etc/passwd. If a valid user name is entered and that name has no password associated with it, you are logged in without further prompting.

   For security reasons, if the user name entered is invalid (it is not found in /etc/passwd), the system still prompts you to enter a password before denying you access to the system. This makes it more difficult for an intruder to find and use a valid user name. Once access is denied, login displays its login: prompt and waits for another user name to be entered.

   If you wish to keep track of all bad login attempts, create a log file called /etc/btmp by entering (while the root user):

   ```
   touch /etc/btmp
   ```

If this file exists, the system uses it to log unsuccessful login attempts. You can read this file (using the `lastb` program) to help determine if unauthorized users are attempting to login.

The system also keeps track of all successful logins and logouts in a log file called `/etc/wtmp`: you can look at the login and logout information using the `lastb` command.

3. The `login` process sets your numeric user and group IDs. The values are taken from the values supplied in the user ID and group ID fields of the `/etc/passwd` file.
4. The `login` process sets the current working directory to that supplied in the home directory field in `/etc/passwd`.
5. The `login` process executes (using the `exec` system call) whatever command is present in the command field of your `/etc/passwd` entry. Any command may be placed in the command field of `/etc/passwd`. Typically, the command invokes a shell for the user. The most common shells are `/bin/sh`, `/bin/csh`, `/bin/ksh`, and `/bin/pam`. If no entry exists in the `/etc/passwd` command field, `/bin/sh` is executed by default.
6. Assuming a shell was generated in step 5, the shell now executes the system shell script. The **system shell script** sets up a user's environment. The possible system shell scripts are:
   - `/etc/profile` if you use the Bourne shell (`/bin/sh`), Korn shell (`/bin/ksh`), or the restricted shells (`/bin/rsh` and `/bin/krsh`)
   - `/etc/csh.login` if you use the Berkeley C shell (`/bin/csh`)
   - There is none for the PAM shell (`/bin/pam`).

   As shipped to you, these scripts define and export the environment variables PATH, TZ, and TERM. Since `/etc/profile` and `/etc/csh.login` execute for each user as he logs in and since the superuser (root) owns these scripts, you (as system administrator) can modify `/etc/profile` and `/etc/csh.login` to change and export each user's default settings for the environment variables. This is ideal for forcing the execution of commands that each user should execute at login.

7. The shell executes the user's local environment script if it exists in the user's home (login) directory (note the "." prefix which normally makes these hidden files):

   - `.profile` (for the Bourne shell and Korn shell),
   - `.login` (for the C shell),
   - `.environ` (for PAM).

   Typically, the system administrator creates a local environment file for each user. If you add users by using the `reconfig` program, `reconfig` places a default environment file in the users' home directory. Users may customize their local environment on the HP-UX system by modifying the local environment file.

   C shell - In addition to executing `.login`, the C shell also executes the file `.cshrc` (if it exists) each time a new C shell starts. Many programs (such as `vi`) allow you to start a shell from within the command. This is called a **shell escape.** `.chsrc` would be re-run for a shell escape. `.login` is executed only once, following the first execution of `.cshrc`.

   Korn shell - In addition to executing `.profile`, if you have the ENV environment variable defined, the Korn shell will execute the file defined by ENV each time a new `ksh` is started. This is similar to what the C shell does as explained above.

8. Now that you have successfully logged in, the shell prints a prompt and waits for your first command.

System Administration Mode

In addition to run-level 2, HP-UX comes with a system administration mode: run-level s. If the `initdefault` entry in `/etc/inittab` is s, then immediately you will get a Bourne shell at the console, logged in as root. The `bootwait` and `boot` entries in `/etc/inittab` will not execute. These include `/etc/bcheckrc` and `/etc/brc`. In this run-level no `gettys` are issued, nor are any actions taken by the script `/etc/rc`. Additionally, `syncer` and `fsck` do not get executed.

Run-level s is a system maintenance run-level. When you shut down your system you will be in run-level s. *Other than during system shutdown, run-level s is not* *recommended by Hewlett-Packard* since certain processes that monitor and check your system do not run in run-level s.

Booting Problems

Booting HP-UX (bringing up the system) should be a straight-forward process. However, in case of any difficulties, the following helpful suggestions are provided:

- If for any reason, the boot ROM is unable to find and load an operating system, informational and/or error messages are displayed on the system console. These messages are described in the "Boot ROM Error Messages" section of Series 300 hardware installation manual.
- Remember that the mass storage device containing the HP-UX system must be powered up and have achieved a ready state *before* powering up your Series 300 computer. If the disk drive has not completed its power-up sequence (which may require several minutes on some disk drives), the boot ROM will not be able to access the disk and load the system.
- The boot ROM follows a specific search for the system console. Refer to the section "Selecting a System Console". If unsuccessful, it boots without a system console (if autoboot is enabled).
- If you are attempting to boot across the LAN, refer to the "Troubleshooting" appendix.
- The `/etc/bcheckrc` script executes the `fsck -P` command at bootup to check the file systems when the system was incorrectly shutdown. The file system consistency check program (`fsck`) is vital to the maintenance of your file system. If the file system becomes corrupt (whatever the cause), continuing to use the corrupted file system invites disaster. For this reason, if `fsck` finds serious file system errors it will prompt you to re-run `fsck` interactively. Since it is the `/etc/bcheckrc` program prompting you, you will see the prompt:

  ```
  (in bcheckrc)#
  ```

  You must run `fsck` on the corrupt file system to correct the errors. Refer to Appendix A, Volume 2, "Using the fsck Command", in this manual for details on checking the file systems.

Shutting Down the System

Improperly powering down the computer (or an "on-line" mass storage device) can cause the file system to become corrupt. The `reboot` and `shutdown` commands terminate, in an orderly and cautious manner, all processes currently running on the system. This allows you to power down the system hardware without adversely affecting the file system.

This section contains instructions for shutting down a system. The last subsection discusses specifics on shutting down an HP-UX cluster. In general, standalone systems and cluster root servers are treated the same.

Shutting Down for System Maintenance

To eliminate processes and activities on the system which could interfere with system administration activities such as running `fsck`, the `shutdown` program terminates all but the essential processes on the system, unmounts any mounted disks listed in `/etc/checklist`, and puts the system in run-level s. It also will optionally halt or reboot the system, using the `reboot` command.

To shutdown the system perform the following steps (if you are already in run-level s you must use the `reboot` command as described in steps 4 and 5):

1. Log in as the superuser `root`.
2. Move to the root directory of the file system by entering the command:

   ```
   cd /
   ```

3. Execute the shutdown command. The syntax is:

   ```
   /etc/shutdown [ -h | -r ] [ -d device ] [ -f lif_file ] [ grace ]
   ```

   The most common use of the `shutdown` command is simply to bring your system to the system administration mode (run-level s). To do this, use no options other than the grace period. The **grace period** (*grace* in the syntax line) is the number of seconds you want `shutdown` to wait before terminating all processes.

   For example, if you wish to back up your system after giving your users two minutes to log off, you should change to run-level s by typing:

   ```
   shutdown 120
   ```

After backing up the system, bring the system to normal operating run-level with all the daemons and processes running by typing:

```
reboot
```

If the grace period is non-zero, `shutdown` prompts to see whether you wish to send the standard broadcast message or enter your own message. If you elect to send your own broadcast message, type the message on the terminal. When you are finished typing the message, press [Return]. Then press and hold the [CTRL] key as you press [D] to signify the end of the message.

If *grace_period* is omitted, then after waiting 60 seconds, `shutdown` asks if you want to continue.

Some additional tips on using the `shutdown` command:

- If you executed `shutdown -h`, the system will halt, printing a message on the system console that says "halted". You may now power down the system. The only way to reboot after halting is to cycle power on the system.
- If you have not executed `shutdown -h`, you *did not halt* the system. If you wish to power down the system, you can halt the system by typing:

  ```
  reboot -h
  ```

Halting or Rebooting the System

To prevent file system corruption, you should halt the system using the `reboot` command. The `reboot` command provides not only a clean shutdown, but also warns users of the event and allows time delays to permit users to complete work in progress and log out.

1. Log in as the superuser `root`.
2. Move to the root directory of the file system by entering the command:

   ```
   cd /
   ```

3. Type the `reboot` command with the appropriate options. Refer to the *reboot(1M)* page in the *HP-UX Reference* for a list and description of all options. The most commonly used are shown in the following examples:

   - If you wish to boot a new operating system (for example, to test your recovery system), you can use the -d option to reboot. If your recovery system is on the device associated with `/dev/dsk/xxx`, type:

     ```
     reboot -r -d /dev/dsk/xxx
     ```

     You cannot reboot across the LAN using this option; you must cycle power to reboot across LAN.

   - If you wish to halt the system from run-level s with no daemons or programs running, type:

     ```
     reboot -h
     ```

   - to halt the system at noon so you can install a new interface card or other hardware, login as root and type:

     ```
     reboot -h -t 12:00  -m "to install new expander"
     ```

     This will periodically warn users of the approaching system shutdown, then at 12:00 (noon) the system will halt. You can then turn the power off, install the hardware, and reboot the system by turning power back on.

   - To reboot the system in 15 minutes to incorporate added optional kernel features after running `config`, login as root and type:

     ```
     reboot -t +15 -m "to add magtape driver, back in a minute"
     ```

     This will warn users, and the system will reboot in 15 minutes.

Shutting Down a Cluster

The procedure for shutting down the root server differs from that of the diskless nodes.

Shutting down the root server (and/or the cluster)

Shutting down the cluster can be accomplished by shutting down the root server. Both `shutdown` and `reboot` perform an orderly shutdown of the cluster when executed from the root server.

Shutting down the root server will affect all other nodes in the cluster. Executing `shutdown` on the root server will perform an orderly shutdown of all the nodes in the cluster. When rebooting or setting the root server in single-user state, the diskless nodes will be placed in auto-reboot mode. That is, they will attempt to reboot and wait until the root server's boot server program responds. When halting the root server, each of the diskless nodes will be halted.

If you do not properly shut down your root server, or if the root server panics for some reason, all diskless nodes in the cluster will panic. A **panic** is when the kernel finds an unrecoverable error and sends a message to the console that begins with the word `panic`.

Shutting down a Cnode

Shutting down a diskless node will not affect other nodes in the cluster. When shutting down a diskless node, it is sufficient to notify system users, execute the `/etc/shutdown` command, and then power down the system. As with any HP-UX system, use the `-h` option if you wish to power down the system. If you do not use the `-h` option when halting your cnode, the system will automatically come back up if the root server is still running the boot daemon. `reboot` only acts upon the local machine when executed from a diskless cnode. When executed from the root server, however, it shuts down the entire cluster.

Power Fail or Disk Crash Recovery

Since you have invested a significant amount of time installing HP-UX and creating file systems, it is important to maintain the file system to ensure its integrity for your users. Simple daily checks and procedures and correcting problems before they become catastrophic will save you from remaking the entire system. Backup procedures are discussed in Chapter 5, the section "Backing Up and Restoring the File System". If these procedures are followed on a regular basis, a power failure or disk crash should not be catastrophic.

Likewise, during installation of HP-UX, you were encouraged to create a recovery system. The recovery system will be very important if your system problems are serious enough that the normal system will not even boot. Details on creating and using a recovery system are contained in Chapter 4 of this manual. *Unless your disk has already crashed, it is not too late to create a recovery system right now.* Take the time for this very important procedure. Note that you can create a recovery system only on cartridge tape.

If your electricity goes off or if you accidentally pull the plug on your standalone or cluster root server system, the computer simply stops. However, because the system was not shut down using the `shutdown` or `reboot` command, `fsck` will run during the next bootup.

If you unplug or power off a diskless cnode in a cluster, the rest of the cluster will continue transparently. A slight delay (hang) may occur on surviving cnodes (including the root server). This delay will happen only if the failed cnode had been claiming a resource at the time of the power fail.

If your hard disk crashes or the power fails, then try to boot and run `fsck`. (Refer to Appendix A in Volume 2 and to this chapter, the section "HP-UX Starts the Init Process".) If you can't boot, use your recovery system. If none of the above work, then call your HP support engineer or re-install and restore your system from backups.

If you receive a system panic message and the system produces a dump, you should write down the system panic message. If you do not know what caused the problem, you should call your Hewlett-Packard support personnel. This information, along with the configuration of the system, should be given to your local Software Engineer.

Contents

4

Customizing the HP-UX System

After you have installed your system, you may want to customize the system. Customizing your system means one or more of the following:

- Preparing your system to talk to a new peripheral device such as a printer or disk drive.
- Creating `init` run-levels that contain all procedures you wish executed.
- Changing any of several configuration files.
- Creating a recovery system.
- Adding or removing optional software.
- Adding security to protect your system.
- Creating an HP-UX cluster.
- Adding a node to an HP-UX cluster.

All of these are optional. However, it is strongly suggested that you do at least the following:

- Create a recovery system.
- Add security to protect your system.

There are four programs you can use to customize your system: `mklp`, `mkdev`, `reconfig`, and `config`. These programs overlap as shown in Figure 4–1. The procedures documented for each customization task will help you decide which customization program is best for your situation. In general, HP recommends using the `reconfig` program wherever possible, and using the other customization programs only where `reconfig` does not have the functionality.

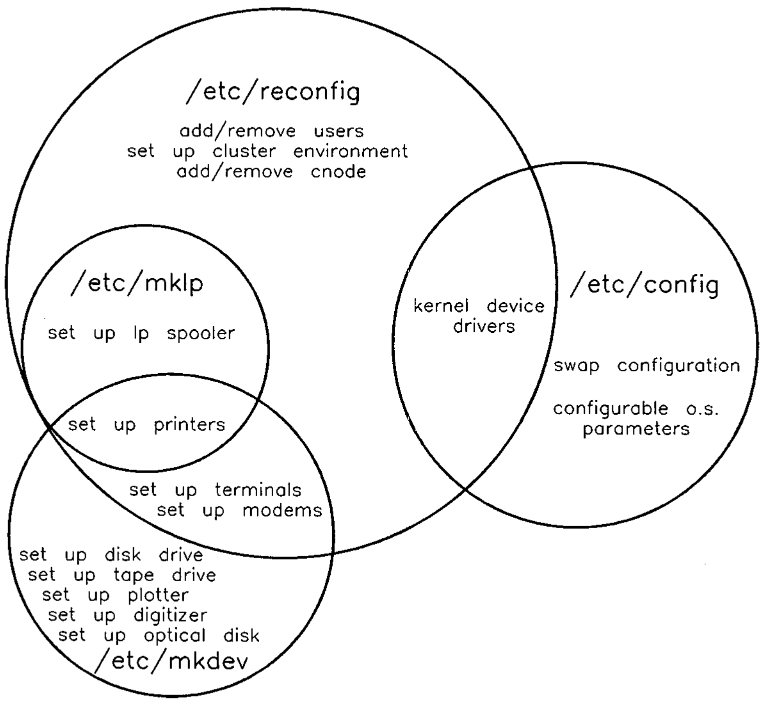


Figure 4–1. The Four Customization Programs

Changing the HP-UX Environment Files

The system boot and login processes provide many opportunities to customize the environment in which your system operates. Customization is achieved primarily by altering the contents of one or more files known as **environment** files. The following list summarizes the files that you may want to alter and identifies the types of changes you may want to make. Use these suggestions in conjunction with Chapter 3, the section "System Startup Functions", to determine which files to modify. All of the files listed here have versions that were shipped with your system. Unless stated otherwise, these are text files.

Because customers often change the contents of the files, update will not overwrite them. When you update your system to a new version of HP-UX the update program will put many of the environment files into the `/etc/newconfig` directory. When you update your system you should check the `/etc/newconfig` directory to compare the new files with your versions. You may wish to edit the new versions to customize them, then move them to the `/etc` directory. The `/etc/newconfig/README` file contains information about the files in `/etc/newconfig`.

Caution The system may not boot if some entries in `/etc/inittab`, `/etc/rc`, and `/etc/passwd` are modified. Also, do not modify the structure of CDFs in an HP-UX cluster.

/etc/inittab

This file contains entries for the different run-levels (supplied or created) on your system. Refer to "Creating System Run-Levels" in this chapter and to the "System Startup and Shutdown" chapter.

When you add a new remote terminal to your system you also must add a `getty` entry to `/etc/inittab`. The entry should be of the form:

```
04:2:respawn:/etc/getty tty04 H #Terminal at rob's desk
```

This ensures that every time you boot the system the remote terminals will receive a `login:` prompt. If you use the `reconfig` program to add a terminal, `reconfig` will add the correct `getty` line into `/etc/inittab`. For more information on `getty` entries refer to the section on adding terminals in the "Adding Peripheral Devices" section in this chapter.

If you are in an HP-UX cluster, this file must be a CDF. The CDF will contain one subfile for each cnode.

/etc/rc

This shell script defines miscellaneous actions to be taken during the bootup procedure. The "System Startup and Shutdown" chapter lists the tasks this script performs.

/etc/passwd

This text file identifies the user name, real user and group IDs, home (login) directory, and execution command for every valid user on the system. The execution command is the command executed when the user correctly logs in. You must add an entry to this file for each new user who is added to your system. Refer to "Adding/Removing Users" in Chapter 5.

/etc/group

This text file identifies the users that form a group. It associates group IDs with group names. It also contains a list of users and associates those users with a group name and a group ID. Refer to "Creating Groups/Changing Group Membership" in Chapter 5.

/etc/motd

This text file (*M*essage *O*f *T*he *D*ay) contains messages that are printed to each user when he logs in. If you have a message that you want every user to read (such as a message specifying a new system update), write the message in this file by using a text editor. As each user logs in, the message will be printed (assuming that the scripts `/etc/profile` and `/etc/csh.login` are not modified to remove the command that prints `/etc/motd`).

/usr/news

This is a directory owned by the user `root`. It is shipped as an empty directory. You, the system administrator, can use it to communicate with users on the system. You can also change the directory permissions to allow any user to put messages here. Place any message you want in a file contained in this directory. If there is a `news` command in either the file `/etc/profile` or in `/etc/csh.login`, the file you placed in the `/usr/news` directory will be announced when the user logs in. Depending on the options used with `news`, a user receives the message only once. Refer to the *news(1)* entry in the *HP-UX Reference* for details.

/etc/profile or /etc/csh.login

These shell scripts are automatically executed for users upon logging into their shell: `/etc/profile` is executed for users logging into the Bourne shell, Korn shell, or restricted shell, and `/etc/csh.login` is executed for users logging into the C shell. This is an ideal location to place commands that each user is required to execute. For example, you may want every user to read the message of the day file (`/etc/motd`) since it contains information that each user should see before beginning her work. This is accomplished by placing the statement:

```
cat /etc/motd
```

in the `/etc/profile` or `/etc/csh.login` shell scripts. These scripts are also an ideal location to define and export default environment variables (such as PATH and TZ) in case the user does not set them in his local environmental shell script. For more information on setting the TZ variable refer to the section called “Setting the System Clock”.

/etc/wtmp

This is a binary file which is used by the system to keep a history of logins, logouts, and date changes. The system automatically creates this file. This file will grow without bound. Therefore you should check it regularly and empty the file. The contents of this file are accessed with the `last` command. For more information refer to the *utmp(4)* entry in the *HP-UX Reference*.

This file must be a CDF for HP-UX clusters. The CDF must contain a subfile for each cnode.

/etc/btmp

A binary file which, if it exists, is used by the system to keep track of bad login attempts. You must explicitly create this file to use this feature. To create it, type `touch /etc/btmp`. This file will grow without bound. Therefore you should check it regularly and empty the file. The contents of this file are accessed with the `lastb` command. For more information refer to the *utmp(4)* entry in the *HP-UX Reference.*

This file is a CDF for HP-UX clusters. The CDF must contain a subfile for each cnode.

/etc/utmp

A binary file which is used by the `who` command. It is automatically created by the system. It contains a list of current users and system startup information. *Do not remove this file.* For more information refer to the *utmp(4)* entry in the *HP-UX Reference.*

This file is a CDF for HP-UX clusters. The CDF must contain a subfile for each cnode.

/etc/securetty

A text file which, if it exists, specifies the `tty` files on which the `root` user can log in. You must explicitly create this file and place the `tty` device file names in it to use this feature.

If you are on an HP-UX cluster, you must create this file such that it is a CDF. The CDF must contain a subfile for each cnode. The entries in each cnode's subfile may differ.

$HOME/.profile, $HOME/.cshrc, $HOME/.login, or $HOME/.environ

These shell scripts are usually put into each users' home (or login) directory. $HOME is an environment variable that maps to the home directory. These shell scripts execute at the following times:

- `$HOME/.profile` executes each time the user successfully logs in using the Bourne shell, Korn shell, or restricted shell
- `$ENV` (in Korn shell) is executed each time a new Korn shell is started. $ENV is normally defined by the user in `$HOME/.profile`.
- `$HOME/.cshrc` (C shell) executes each time a new C shell is started. A new C shell is started each time the user successfully logs in, and each time the user starts a new C shell, such as using the shell escape feature of `vi`
- `$HOME/.login` (C shell) executes each time the user successfully logs in, but after `$HOME/.cshrc` is executed
- `$HOME/.environ` (PAM) executes each time the user successfully logs in using the Pam shell

For example, they may include a definition of the default shell prompt (the `PS1` and `PS2` environment variables in the Bourne and Korn shells, `prompt` in the C shell) or the default search path (the PATH environment variable). It also generally includes the execution of one or more commands such as the `export` command—to export environment variable definitions, the `who` command—to identify who is logged in on the system, and the `mail` command—to automatically display mail that has been received.

Examples of `.profile`, `.login`, and `.environ` are shipped under the names `/etc/d.profile`, etc. You may find it useful to customize these files and provide them to new users by default. If you add a user with the `reconfig` program, `reconfig` will place the appropriate environment file into the user's home directory.

$HOME/.exrc

This file maps terminal characteristics and sets up new key definitions so that features like arrow keys can be used with the ex family of HP-UX editors (`vi`, `ex`, etc.). The file `.exrc` must exist in the user's home directory ($HOME) to use these features. The editor searches for `$HOME/.exrc` and, if it exists, uses the definitions to create extra editor features.

Note that the `.exrc` file is functional *only if the EXINIT environment variable is not defined.* EXINIT can be defined and exported from either `/etc/profile` or `$HOME/.profile`. The `.exrc` file serves a function similar to EXINIT. Refer to the appendix to "The vi Editor" article in the *HP-UX Concepts and Tutorials* manual for further details.

An example of `.exrc` is shipped under the name `/etc/d.exrc`. You may find it useful to customize the file and provide it to new users by default.

/usr/lib/terminfo

This subsystem identifies terminal capabilities for programs such as the `vi` text editor. It defines terminal attributes for all Series 300 models and HP-supported terminals. It also contains terminal attributes for terminals *not* supported by Series 300 HP-UX; these are provided for your convenience, but Hewlett-Packard does not support their use.

/etc/checklist

This text file contains a list of mountable file systems and swapping devices. When no device file specification is supplied with the `fsck` command, `fsck` performs its checks on the file systems listed in `/etc/checklist`. This file is also used by the system accounting `diskusg` command, and the `mount`, `umount`, `swapon`, and `fsclean` commands.

The file `/etc/checklist` is shipped with a single device file name: `/dev/dsk/0s0`. This file corresponds to the hard disk on which you installed the system and which contains the root file system. You should add entries for each additional disk drive containing a file system which you want automatically mounted, and for each disk drive used as a swap device. Refer to "Adding to /etc/checklist" in this chapter for more information.

This file must be a CDF if you have an HP-UX cluster. If you are on a diskless cnode this file should be empty unless you have local swapping on your diskless cnode.

/etc/catman

Executing the `catman -z` command (`-z` for uncompressed) expands the `nroff` (formatted) versions of manual pages (used by the `man` command) into their "processed" form. Subsequent accesses via `man` use the processed manual page, significantly improving response time. The price for the improved speed is disk space—the expanded files will use about the same storage space as the originals. This doubles the disk usage for manual pages because the original files remain intact.

By default, running `catman` causes manual pages in all the `/usr/man/manX`, `/usr/local/man/manX`, and `/usr/contrib/man/manX` directories (where `X` is a number corresponding to sections in the *HP-UX Reference*) to be processed and stored in the corresponding `/usr/man/catX`, `/usr/local/man/catX`, and `/usr/contrib/man/catX` directories. If you run `catman` without the `-z` option the pages are put in compressed form in the corresponding .z directories. The `catman` command creates the directories if they do not exist.

You have three alternatives for creating on-line documentation:

- Create all the processed manual pages by executing `/etc/catman` with no parameters. This process can take as long as five or six hours to complete so you might want to run it in the background and/or at night.
- Create selected sections of the processed manual pages by executing `/etc/catman` *sections* (where *sections* is one or more logical sections in the *HP-UX Reference* such as 1).
- Do not execute `/etc/catman` at all. If you create all the `/usr/man/cat` directories, the first time `man` is executed for manual entry, the entry is processed, added to the appropriate `cat` directory, and used in subsequent accesses. The following script creates the `cat` directories:

```
cd /usr/man
for num in 1 1m 2 3 4 5 7 8 9
do
 mkdir cat$num
done
```

The third alternative is recommended if you can spare some disk space but do not want to use any more than is necessary. With this "build-as-you-go" alternative, the system fills the `cat` directories as manual pages get accessed by `man`.

When the processed `man` pages exist, you can remove the `nroff` source files and thus recover much of the disk space required by the formatted version of the manual.

/etc/issue

This file contains information that is printed by the terminal's `getty` process prior to the login prompt. Messages that identify the computer or provide the system name might be stored in this file.

/etc/csh.login, /etc/rc, and /etc/profile

Edit these three files to set the correct date information in the environment. The format for setting the time zone environment variable is:

```
TZ=XXXHYYY
```

where:

XXX
: is an alphabetic abbreviation of the standard time zone, usually three letters in length.

H
: represents the difference between standard local time and Greenwich Mean Time, in hours. Fraction hours are indicated in minutes (for example, 3:30 for Newfoundland). Negative hours are allowed (for example, -9:30 for South Australia).

YYY
: is an alphabetic abbreviation of the daylight time zone for your area, usually three letters in length. YYY may be deleted if Daylight Savings Time is not observed in your geographic area.

Insert or modify the lines in `/etc/rc` and `/etc/profile`:

```
TZ=XXXHYYY
export TZ
```

Insert or modify the lines in `/etc/csh.login`:

```
setenv TZ XXXHYYY
```

For example:

- In Eastern time zone, use `TZ=EST5EDT`
- In Central time zone, use `TZ=CST6CDT`
- In Arizona, where Daylight Savings Time is not observed, use `TZ=MST7`

For more information pertaining to setting the system clock, refer to the section "Setting the System Clock" in Chapter 5.

/usr/lib/tztab

The `/usr/lib/tztab` file is used to correctly handle changes to and from summer time zones (Daylight Savings Time) in the United States and to accommodate future changes to these adjustments. This file is really a "Time Zone Table" that contains the value of the TZ variable described above, followed by lines detailing transitions in the time zone adjustment. Refer to *tztab(5)* for details on modifying this file.

/etc/ttytype

The `tset` command uses this file as a data base of terminal types on your system. Change this file when adding terminals and modems to your HP-UX system. Change the samples (for example, 300h console) to reflect the true terminal types attached to your system. Refer to the section "Adding Peripheral Devices" in this chapter for more information.

This file is a CDF in HP-UX Clusters.

Adding Peripheral Devices

The HP-UX operating system requires the existence of special files, called **device files,** to perform I/O to peripheral devices. **Peripheral devices** include disk drives, tape drives, and terminals. Information on setting up printers is in the section "Setting Up Printers and the LP Spooler" later in this chapter. This section introduces you to the tools necessary to set up peripherals and their associated device files. Following a general discussion of the procedures involved in adding peripherals to your HP-UX system, this section provides detailed information regarding each of the following peripheral types:

- miscellaneous devices
- terminals and modems
- pseudo terminals
- hard disks
- optical disks
- flexible disks
- cartridge tapes
- magnetic tape devices
- plotters and digitizers
- HIL and GPIO

Many of these procedures can be accomplished more easily using the `reconfig` program. The `reconfig` program is sometimes limited in its functionality for setting up a peripheral. These limitations are discussed in the `reconfig` method of accomplishing a task.

Overview of the Task

There are several basic steps required to add peripheral devices to your system. Here is an overview of the tasks you will need to accomplish; they are explained in more detail later:

1. Determine the best place to locate the peripheral. **Locate** means hardware address location in terms of select codes, bus addresses, etc. The best location depends on several things, including shared sets of I/O resources and expected usage. The *Peripheral Installation Guide* supplied with your computer and the installation manual supplied with the peripheral device will help you to identify an appropriate location.

2. Connect the peripheral device. This may entail two important tasks: installing an interface card and connecting the peripheral. If the peripheral device requires an interface card, set the appropriate switches on the card and install the card in the computer. *Never install or remove an interface card while the computer is powered up.* Then set any required switches on the device and connect it to the computer (or interface card). If you ever change the switch settings on an HP-IB device, be sure to cycle power on the device before attempting to address it because many devices only look at the switch settings when they are first turned on.

3. Ascertain whether the peripheral device will be addressed as a block or character device, or both (disk drives will require both types of access). Block and character I/O are discussed in Chapter 2 of this manual, and examples are provided later in this section.

4. Determine if the device file necessary to communicate with the peripheral device already exists on your HP-UX system. The device files shipped with your system are shown in Table 4–1 later in this chapter. Default device files reside in the `/dev` directory and follow the naming conventions explained in the *intro(4)* entry in the *HP-UX Reference* manual.

   Note that if you do not wish to use these device names, you can link them to the name you wish. However, do not link the file `/dev/dsk/0s0` to anything.

   If you are in an HP-UX cluster, the `/dev` directory is a CDF. In most cases this will not affect you, but you must be aware of it when creating device files for cnodes other than the cnode you are logged into.

5. If the appropriate device file does not exist for the device in question, you must create one. If you are using the `reconfig` program to install the peripheral device, then `reconfig` will create the device file.

   If you are not using `reconfig`, use the `mknod` command to create a particular device file or edit and execute the `/etc/mkdev` shell script to create one or more device files. Use the `mkdev` script if you need to create many device files. If you are creating only a few, use the `mknod` command. The `mkdev` script contains templates for `mknod` commands. *You must execute these commands on the appropriate cnode.*

6. Make sure your kernel contains the correct drivers (kernel code) to use the peripherals. By HP-UX convention, your kernel should contain all drivers listed in `/etc/conf/dfile`.

   To find out what drivers are necessary to use your peripherals, look in `/etc/master` for the list of available drivers and a product number/driver alias table. Look up your product number in the alias table; you will need the associated driver number. For example, if you look up 9122, you will see that you need a `cs80` driver.

Determining the Peripheral's Location

The peripheral's **location** is its hardware address. The hardware address is referred to as the **minor number.** The minor number can reflect information other than the hardware address, however. For example, magnetic tape drive minor numbers reflect addressing information, tape density information, and behavioral information. The minor number is a six-digit hexadecimal number (base 16 rather than decimal which is base 10).

All peripherals connect to an interface card, either built-in or optional. The interface card has switch settings on it. These switch settings are referred to as the **select code.** Any peripheral connected to a particular interface card will have that interface's select code as the first field of the minor number.

All other fields in the minor number depend on the type of device. The *Peripheral Installation Guide* gives specific instructions for setting up devices and interface cards to a specific address. It also provides example `/etc/mknod` command lines that correspond to the address. This is a good source of information for determining the peripheral's location or setting up the peripheral. Refer also to the section in this chapter specific to the type of peripheral device to determine the other fields of the minor number.

The following list provides some guidelines for HP-IB interface selection:

- The system root device (hard disk) is usually located at select code 14, bus address 0 on a (high-speed) HP 98625A, HP 98625B, HP 98262A high-speed disk interface card, or SCSI interface card (HP 98265A).
- The built-in (internal) HP-IB is always at select code 7.
- The system printer (if present) should be on a low-speed HP-IB interface, separate from the system root device. A bus address of 1 is typical.

- An HP 7971 9-track tape must be placed on a low speed HP-IB. A bus address of 3 is typical.
- An HP 7974 or 7978 9-track tape drive should be placed on a hi-speed disk HP-IB, if possible. You can also use that same bus for the root device. A bus address of 3 is typical.
- Avoid putting flexible disk drives, cartridge tape drives, or 9-track tape drives on the same interface as the root device.
- Plotters and the HP 9111 graphics tablet should be placed on separate low-speed HP-IB interfaces when possible. Typical bus addresses are 5 and 6 for plotters and graphics tablets, respectively.

Connecting the Peripheral

Connect the peripheral to your computer at the location you have just determined. If the peripheral device requires an interface card, the computer's installation guide provides instructions for its installation and for identifying its select code. Following these instructions, set the appropriate switches on the card and install the card in the computer. *Never install or remove an interface card while the computer is powered up.* The manual supplied with the peripheral details the procedure for connecting the peripheral to the computer and for setting its address, if it has one. This procedure may require setting switches on the peripheral device before the device is connected to the computer or to the interface card. If the switch settings on an HP-IB device are changed, be sure to cycle power on the device before attempting to address it.

An additional caution to remember while using your system: do not connect or disconnect an HP-IB device while the system is running, or turn power on or off to an HP-IB device while connected to a powered-up system. This could result in bad data on the HP-IB bus.

Terminal hardware configuration is covered in the *Peripheral Installation Guide.* You must create the associated device files for the terminal as well as follow the instructions at the end of this section to set up the software aspects of terminal configuration.

Caution DO NOT attempt to unpack and connect the following disk drives yourself: HP 7911, HP 7912, HP 7914, HP 7933, HP 7935, HP 7936, or HP 7937. These CS/80 disk drives are packed to prevent damage during shipment. To prevent damage to the device, an HP customer engineer must unpack, install, and test the device.

Block versus Character Special Files

Determine whether you should create a **block device file** or a **character device file.** Disks should have both block and character device file entries. Any cartridge tape or flexible disk drives that will have mounted file systems should have entries for both block and character device files. All other devices should have character device file entries only. For more information on block vs. character devices, refer to the "System Management Concepts" chapter.

Configuring the Kernel Drivers

Verify (and possibly add) new information to your kernel. This task can be accomplished using either the `reconfig` or the `config` command. Please refer to the chapter called "Kernel Customization" for more information. The appropriate kernel device drivers are given in the section specific to the peripheral you are setting up.

Creating Device Files

There are basically two ways to create device files: the `mknod` command and the `/etc/mkdev` script. The `mknod` command creates a particular device file. The `/etc/mkdev` script contains example `mknod` command lines. By modifying these lines to reflect the correct addressing information for your devices, one or more device files can be created. The `/etc/mkdev` script is shipped with your system. You can also create it (or re-create it) by executing `config -a`, and view it by executing `more /etc/mkdev`.

If you need to create many device files, you may wish to edit, then execute, the `mkdev` script. If you have only a few device files to create, use the `mknod` command described next.

Each mknod line has the form:

```
/etc/mknod path_name file_type major minor
```

path_name — The pathname of the device file to be created. Select a name for the device file which easily identifies the associated peripheral. The entry *intro(4)* in the *HP-UX Reference* manual describes a naming convention for device files. Use this naming convention to make your system easier to support and maintain. Put all device files into the /dev directory; many commands expect to find device files in /dev and will fail if the required device file is not there.

file_type — File type is represented by a single character: b, c, n or p. b specifies that the file is a block device file, c specifies that the file is a character device file, n specifies that the file is a network special file, and p specifies that the file is a named pipe (FIFO). Refer to *mknod(1M)* for making network special files.

major — The major number is a pointer to the kernel driver used to communicate with the peripheral. For devices needing both a character and a block device file, there are different character major numbers and block major numbers.

minor — The minor number specifies the address on the I/O bus. It is made up of the select code and other information depending on the driver and device you are using. If you set up your peripherals using the *Peripheral Installation Guide,* you already have all the correct minor numbers written on your worksheet. If not, use the guidelines in the *Peripheral Installation Guide* and in the section specific to the peripheral device to determine what the address is.

If you are in an HP-UX cluster, the /dev directory is a context-dependent CDF. You must create device files while logged in to the appropriate cnodes.

If you make a mistake, delete the device files you wish to change and re-create them by executing the /etc/mknod script again.

Note Do not delete the device files listed in Table 4–1. If you delete these files you system may not boot.

Device File Format

In the /dev directories, if you execute an ll command, you should see entries similar to the following:

```
crw--w--w-   1 root     other      0 0x000000 May 20 09:30 console
```

The first character in the entry tells you whether the device file is a character (c) or block (b) device and the next series of characters represent the file's access permissions. The major and minor numbers are the two numbers contained in the size field, in this case 0 and 0x000000.

Miscellaneous Devices

The miscellaneous device class includes the device files that the system needs to run properly. Each HP-UX installation must have the device files /dev/null, /dev/console, /dev/mem, /dev/kmem, /dev/dsk/0s0, /dev/swap and /dev/tty. The device file /dev/null is a null file (a "bit bucket") used by many HP-UX commands. The device file /dev/console identifies the system console and the device file /dev/tty is a synonym for the control terminal associated with a process group.

These miscellaneous device files are copied to your system when HP-UX is installed. *They should not be changed or modified.* If one or more of these files is accidentally deleted or otherwise destroyed, you can recreate it by editing the /etc/mkdev script and removing the comment sign (the # character) in front of the corresponding entry. Alternatively, recreate it with the mknod command using the character/blocked designation, major, and minor numbers given below.

Although there are additional device files created when HP-UX is installed, only the ones listed in Table 4–1 are vital to booting and running HP-UX.

Table 4–1. Default Special Device Files

Special File	C/B	Major	Minor	Device Description
console	c	0	0x000000	System message port
syscon	c	0	0x000000	System console (linked to console)
systty	c	0	0x000000	System tty (linked to console)
tty	c	2	0x000000	Process group control terminal
null	c	3	0x000002	Null file ("bit bucket")
mem	c	3	0x000000	Physical memory image
kmem	c	3	0x000001	Kernel virtual memory image
swap	c	8	0x000000	Swap device
dsk/0s0	b	255	0xffffff	Root pseudo device file

Note The `/dev/dsk/0s0` device file (the root device) *must not* be removed from the system or linked to another file name, if HP-UX is to operate properly.

There needs to be a `/dev/systty` (which is linked to `/dev/console`), and a `/dev/syscon` (which is linked to some terminal—usually the console). This is explained in *init(1M)*.

Terminals and Modems

Adding a terminal allows you to log in from a terminal other than your system console. Adding a modem allows you to dial into your computer from a remote terminal and/or dial out of your computer to another computer. Refer to Figure 4–2 for a diagram of the possible configurations. In Figure 4–2, your computer (the local computer) is equipped with the 4-channel MUX card, so both a direct-connect terminal and a dial-in/dial-out modem are attached to it.

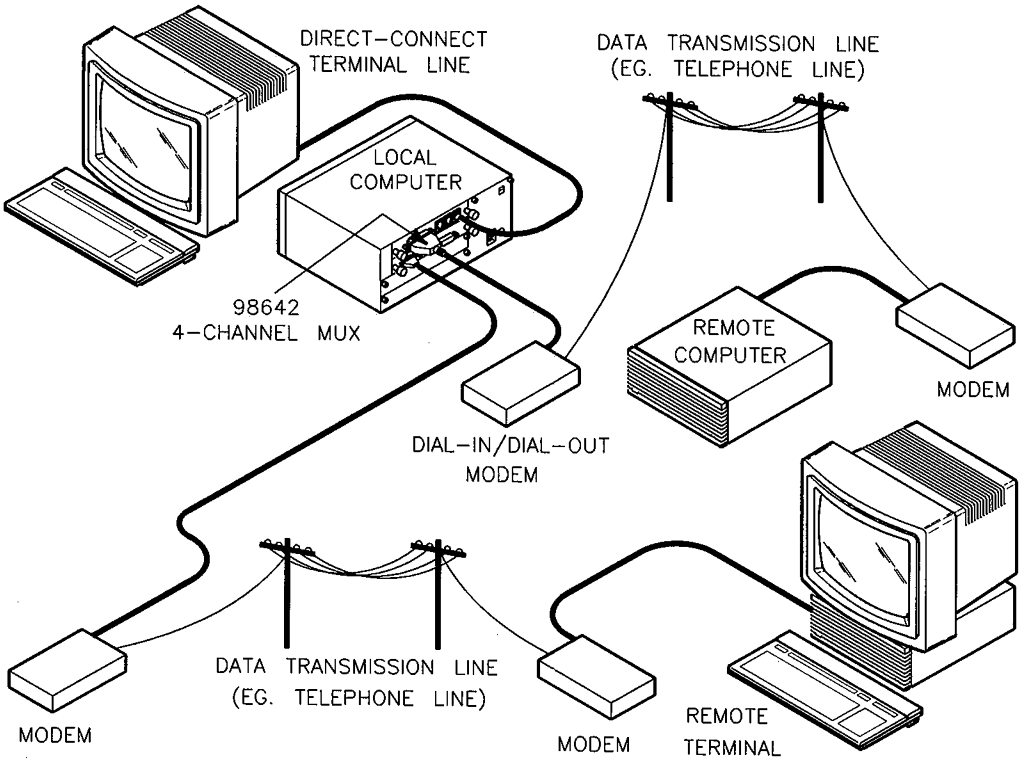


Figure 4-2. Terminal and Modem Connections

Use the procedure in this section to set up a terminal or a modem. The installation guide supplied with your computer discusses the hardware aspects of adding a terminal to your system. This section offers the software configuration information you need.

In this section, the term **port** refers to an entry point to your system. This entry point can be used for modems or direct-connect terminals. The address of the entry point is based on the type of interface card your terminal or modem is connected to. This will be described later in the procedure.

You can add your terminal or modem using `reconfig` or by completing the steps manually. HP recommends using `reconfig`. If, however, you wish to add your terminal or modem manually, the section "Manual Method" provides the information you need.

Reconfig Method

Note	
	If you are adding a terminal or modem to a diskless cnode (HP-UX cluster only), you must execute the `reconfig` command from that cnode (either physically or with the `rlogin` command). If you execute it from the cluster's root server the device file will be set up with the wrong context, and may overwrite existing root server device files.

To add support for a new terminal or modem, follow these steps:

1. Follow the guidelines in the *Peripheral Installation Guide* to install the RS232 or MUX interface card (HP 98626, HP 98642, or HP 98628) to which you will connect your remote terminal.
2. Log in as the superuser `root`.
3. Start the `reconfig` program by typing:

   ```
   /etc/reconfig
   ```

   The main menu of `reconfig` will appear on your screen (Figure 4–3).

```
**************************************************
              RECONFIG -- MAIN MENU
**************************************************


 User Configuration
 Workstation Configuration
 Cluster Configuration



[        ] [        ] [  NEXT  ] [PREVIOUS]  [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 4–3. Reconfig Main Menu

Choose the item called `Workstation Configuration`. You will now see the screen shown in Figure 4–4.

```
**************************************************
      RECONFIG -- WORKSTATION CONFIGURATION
**************************************************


Add terminals and modems
Configure the line printer
Create a minimally loaded operating system
Create a fully loaded operating system
Create a custom operating system








[        ] [        ] [  NEXT  ] [PREVIOUS]  [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 4–4. Reconfig Workstation Configuration Menu

If you don't see the last three lines of the main menu on your screen you do not have the ACONFIG fileset loaded on your computer. You do not need the ACONFIG fileset to add remote terminal support unless you also need to add a new kernel driver for the terminal interface. As shipped, your kernel includes all kernel drivers for terminal interfaces.

4. Press the [SELECT] softkey (the cursor should already be highlighting the "Add terminals and modems" menu item).

   The form for adding a terminal will appear on your screen (Figure 4–5). You will see the `Remote terminal support ...` line only if you currently have one or more remote terminals set up. The values below this line indicate the select code and port address of existing terminals. For example, 12:0 indicates that there is a terminal at select code 12 and port address 0.

```
**************************************************
         RECONFIG -- ADD TERMINAL SUPPORT
**************************************************

Select Code:
Port Address:
Speed:                  9600
Port Usage:             direct connect


Remote terminal support is currently provided for the following:

12:0          12:1



Select Code? >>





RESTORE                                    HELP            PREVMENU
```

Figure 4–5. Configuring Terminal Support Menu

5. Type the select code and press the Return key.

   Each interface card installed in your system has a unique address called a select code. The select code is set with switches on the interface card and can be a value of 0 to 31. Sometimes a sticker displaying the select code is attached to the interface card. If you installed your interface card using the guidelines in the *Peripheral Installation Guide,* you will have the select code on your worksheet.

   If you cannot get the select code information from either the *Peripheral Installation Guide* or the back of the interface card, you can either see it listed in the boot ROM messages as you reboot, or you need to power off your computer, remove the card, and check the position of the select code switches.

Caution DO NOT remove the card while your computer is turned on. If you do, you can damage your hardware.

When you press the [Return] key, the `reconfig` program will check that the select code you entered is between 0 and 31, but it will not verify that the select code is correct.

6. Type the port address, then press the [Return] key.

   If you are adding terminal support to a single channel RS-232 interface card (HP 98626 or HP 98644) port, you must set this field to 0 (zero). If the port is a 4 channel MUX interface card (HP 98642), you should set this field to the port number (0 to 3) to which terminal support is being added. The interface card's product number is shown on the faceplate of the card.

   The `reconfig` program will check to see if the number is between 0 and 3, but will not verify that the number is correct.

7. Choose the speed for your terminal connection, then press the [SELECT] softkey.

   On the bottom of your screen you will see the message shown in Figure 4–6.

```
Following are the valid terminal speeds (baud rate):

300          1200         2400         4800
9600

  Speed? >> 9600

[RESTORE] [        ] [NEXT] [PREVIOUS]   [HELP] [SELECT] [PREVMENU] [        ]
```

Figure 4–6. Choosing Terminal Speed

Choose the default speed (or baud rate) at which this port should operate. Use the NEXT and PREVIOUS softkeys to choose the correct speed for your terminal. Terminals that are directly connected to ports are generally set to 9600 baud. If you are using a modem, the setting will depend on the modem.

Pressing just the SELECT softkey or just the Return key will select the displayed (default) value.

8. Select the port usage for your terminal connection, then press the SELECT softkey.

   You will see the following message on your screen:

```
Following are the valid terminal port usages:

us modem                 ccitt modem              direct connect


    Port Usage? >> direct connect


RESTORE  |        |  NEXT  | PREVIOUS |  HELP  | SELECT | PREVMENU |
```

Figure 4-7. Choosing Port Usage

It is important that you set this field to correctly reflect how the port will be used. Use the NEXT and PREVIOUS softkeys to choose the correct port usage based on the information below:

Choosing Port Usage

direct connect	If the terminal is directly connected to your interface card (you will not use a modem), select this option.
us modem	If you will use a modem and are in the United States or Canada, select this option. This setting will allow you to dial into your system. If you select "us modem" you will be prompted for the port direction: dial-in, dial-out, or dial-in/dial-out. Choose the correct direction for your modem and press Select.
ccitt modem	If you will use a modem and are not in the United States or Canada, select this option. This setting will allow you to dial into your system.

Pressing just the SELECT softkey or just the Return key will select the displayed (default) value.

9. Confirm your choices. You will see the following message:

```
Are the above values ok (y or n)? >>
```

If you type `n` you will be asked for each value again. If you type `y` then `reconfig` will perform the following tasks:

- create a file in the `/dev` directory with the correct port number (the port number is the xx at the end of the file name) and correct address. A direct connect terminal will have a file that looks like:`/dev/ttydxx`. A dial-in modem will have a file that looks like: `/dev/ttyxx`.

- add an entry for the device file in the /etc/inittab file.

  This entry will start a process, called a getty, each time you boot HP-UX. The getty is associated with the terminal.

10. The ADD TERMINAL SUPPORT menu is redrawn with the following message:

```
Next Select Code? >>
Terminal added at select code <SC> port <P>.
Press [PREVMENU] if there are no more terminals to add.
```

where <SC> and <P> are the select code and port address you just selected for your terminal.

You can now either add another terminal or exit the form.

To add another terminal, begin with step 5.

To exit press the PREVMENU softkey. The menu shown in Figure 4–4 will appear on your screen. Press the PREVMENU softkey to return to the main menu.

11. Press the QUIT softkey to exit reconfig and return to your shell.

12. Add an entry to the /etc/ttytype file.

The /etc/ttytype entries have the form:

model_number location

where *model_number* is the product number of the terminal or computer (as defined in /usr/lib/terminfo), and *location* is the device file associated with the terminal/computer and contained in the /dev directory (not the full path name, just the file name).

Here is a sample /etc/ttytype file:

```
300h    console    # Frodo's (administrator) system console
2622    tty00      # Bilbo's terminal
2622    tty01      # Gandalf's terminal
2623    tty02      # Strider's terminal
dialup  ttyd03     # Greybeard's dialup modem
```

If you are on an HP-UX cluster, this file is a CDF with one subfile for each cnode in the cluster.

13. Configure your HP-UX operating system to support the interface card you installed.

    If you haven't reconfigured your kernel to include the new interface card, you will receive a message on your screen similar to:

    ```
    Unable to access ttyxx
    ```

    The kernel driver needed by your terminal is the kernel driver for the terminal's interface card. The interface card kernel drivers to choose from are: 98628, 98642, or 98626. For example: if your terminal is connected to an HP 98642 Mux Card, you need kernel driver 98642.

    Use the `reconfig` program to add kernel drivers to your operating system (refer to "Creating a New Operating System" in Chapter 6).

    If you don't know what interface your terminal or modem is connected to, or can't decide which interface to connect it to, read the information for your peripheral in the *Peripheral Installation Guide*.

14. Set the terminal type when you login. For information on this, refer to *A Beginner's Guide to Using Shells*.

Manual Method

Following is a brief overview of how to add a terminal to an HP-UX system:

1. Follow the guidelines in the *Peripheral Installation Guide* to install the RS232 or MUX interface card (HP 98626, HP 98642, or HP 98628) to which you will connect your remote terminal.
2. Determine the terminal's location, in terms of hardware address, and connect the terminal to the HP-UX system.
3. Create the device files that are required. This could be as many as three files if the terminal will use both dial-in and dial-out access.
4. Add the necessary `getty` entries to the `/etc/inittab` file.
5. Add the necessary entries to the `/etc/ttytype` file.
6. Add the correct kernel driver for the terminal's (or modem's) interface.
7. When using the terminal, set the TERM environment variable and execute the `tset` command.

Creating Device Files for Terminals and Modems

Communication ports—user terminals as well as modems—need to be identified by one or more device files, depending on the intended use of the port. There are four considerations when creating device files for ports:

- *Name* of the device file.

  `tty` files are required for terminals (hard-wired ports). Ports that receive incoming signals ("dial in" modems) require a special naming convention, `ttyd`, for device files. Ports that transmit signals ("dial out") require both `cua` and `cul` device files.

  If you are on an HP-UX cluster, /dev is a CDF. This means you must create the device file from the cnode where the terminal is located.

- *Type* of device file.

  All device files for ports should be character device files: (use a `c` in the `mknod` command line).

- *Major number.*

  All terminals and modems require a major number of 1.

- *Minor number.*

  The information on minor numbers is in the following section "Minor Number for Ports".

Table 4–2 shows a general `mknod` template for ports where *xx* is a two-digit line identifier in the device file name:

Table 4–2. General Template for Ports

Device	C/B	Major	Minor	Notes
`tty`*xx*	c	1	0xScAdUV	hardwired ports (terminals)
`ttyd`*xx*	c	1	0xScAdUV	dial-in modems
`cua`*xx*	c	1	0xScAdUV	dial-out ports
`cul`*xx*	c	1	0xScAdUV	dial-out ports

Minor Number for Ports

The minor number is a hexadecimal (base 16) number with the following format:

0x*ScAdUV*

that represents the following:

0x	This indicates the number is hexadecimal.
Sc	The select code. The select code is a two-digit hexadecimal number determined by switches set on the terminal or modem's interface card.
Ad	The port address for each port. This two-digit hexadecimal number is set by switches on the device. Port address—always 00 if your terminal was connected to an HP 98626 or HP 98644 interface card, and can be 00, 01, 02, or 03 if the terminal was connected to an HP 98642 interface card.
U	Always 0 for terminals and modems.
V	The hexadecimal representation of a 4-bit binary number as defined below:

Bit	Value
3	Always 0
2	1=direct connect, 0=modem
1	1=CCITT protocol (Europe), 0=Simple protocol (U.S.)
O	1=dialout modem, 0=dial-in modem

For example, assume that you want to create device files for a modem at select code 20 (decimal 20 = hexadecimal 14), using an HP 98626 card, and associate it with line 20 (that is, `/dev/ttyd20`). Because the modem will be used as a dial-in and dial-out port, the `V` term of the minor number on the `cua` and `cul` files must be 1, and on the `ttyd` file must be 0. The following `mknod` command lines are needed:

```
mknod  /dev/cua20   c   1 0x140001
mknod  /dev/cul20   c   1 0x140001
mknod  /dev/ttyd20  c   1 0x140000
```

There are now three device files associated with the dial-in and dial-out modem at select code 20.

The following example will set up a direct-connect port for an HP 98642 on line 13 at select code 13 (13 decimal = hexadecimal d). The minor number ends with a 4 since this is a direct-connect port:

```
mknod /dev/tty13 c 1 0x0d0004
```

When a terminal is added to the system, you must add entries to the `/etc/ttytype` and `/etc/inittab` files. This allows a user to login from the terminal. Add entries to these files as described below.

/etc/ttytype entries

The `/etc/ttytype` file is a data base which contains the terminal type of the terminal associated with each port on the system. It is used by the `tset` and `login` commands. Based on the information in this file, `tset` will do some terminal-dependent processing, such as setting erase and kill characters, setting or resetting delays, and sending any sequences needed to properly initialize the terminal. `login` uses this file to set the TERM variable.

The `/etc/ttytype` entries have the form:

model_number location

where:

model_number is the product number of the terminal or computer (as defined in `/usr/lib/terminfo`). For more information on the model number to use here, refer to the *terminfo(4)* entry in the *HP-UX Reference*.

location is the device file associated with the terminal/computer and contained in the `/dev` directory (not the full path name, just the file name).

Here is a sample /etc/ttytype file:

```
300h    console    # Frodo's (administrator) system console
2622    tty00      # Bilbo's terminal
2622    tty01      # Gandalf's terminal
2623    tty02      # Strider's terminal
dialup  ttyd03     # Greybeard's dialup modem
```

If the entry is for a dialup port, the *model_number* should be `dialup`. This causes `tset` to request the proper terminal type during the login sequence.

If you are on an HP-UX cluster, this file is a CDF. There must be one subfile for each cnode.

/etc/inittab entries

The `/etc/inittab` file is described in Chapter 3, the section "System Startup Functions". For terminals, `/etc/inittab` entries will contain the `/etc/getty` command. This section discusses entries specific to terminals.

Most `/etc/inittab` entries for terminals have the form:

```
id:rstate:respawn:/etc/getty -t xxx device_file_name N # comment field
```

where:

`id` — is a unique two-character string. The value of the two-character string is arbitrary but must be unique for each entry. It is used to refer to the same entry/process in other states.

`rstate` — indicates the run-levels in which the `getty` will run. This field typically equals 2, meaning the terminal can be used in run-level 2 only.

`respawn` — The `respawn` flag specifies that the command in the command field (such as `getty`) is to be re-invoked once the process terminates (typically, when a user logs off the system).

`/etc/getty` — This is the command to execute. The fields of the `/etc/getty` command are described below.

The fourth field, the `process` field, must contain the `/etc/getty` command; it is immediately followed by three parameters for a `getty` command, as follows:

`-t xxx` — is the optional time-out option for use with modems.

`device_file_name` is the file name (`tty04`)—not the complete path name (`/dev/tty04`)—of the terminal's or modem's character device file. The named file must reside in the `/dev` directory.

`N` specifies a speed indicator for `getty`. A value of H is common for "hardwired" (9600 baud terminal) lines; a value of 3 is common for dial-up (300/1200 baud modem) lines.

For more information, refer to the *getty(1M)* and *gettydef(4)* entries in the *HP-UX Reference* manual.

On a multi-user system, be certain to set up `/etc/inittab` terminal entries *for each terminal* connected to the system. For example, to add a terminal on `/dev/tty04` for run-level 2 the `/etc/inittab` entry would be:

```
04:2:respawn:/etc/getty tty04 H #terminal at rob's desk
```

Note that the `id` field 04 corresponds to the last two digits of the terminal's device file (`tty04`). This convention is often used with "continuous" (respawn) `getty` processes that get killed in the single-user run-level but is *not* required syntax: any two-character string will suffice. After a user logs out, `getty` is "respawned", and the "login:" prompt is redisplayed . Refer to Chapter 3 in this manual, and to the *getty(1M), gettydef(4),* and *inittab(4)* entries in the *HP-UX Reference* for further details.

If you are on an HP-UX cluster, this file is a CDF. There must be one subfile for each cnode.

Kernel Drivers for Terminals

You must configure your kernel to support the terminal or modem's interface card. The interface card kernel drivers to choose from are: 98628, 98642, or 98626. The HP 98644 interface card requires the 98626 kernel device driver. For example, if your terminal is connected to an HP 98642 Mux Card, you need kernel driver 98642.

If you haven't reconfigured your kernel to include the new interface card, you will receive a message on your console similar to:

```
Unable to access ttyxx
```

If you don't know what interface your terminal or modem is connected to, or can't decide which interface to connect it to, read the information for your peripheral in the *Peripheral Installation Guide*.

Removing A Remote Terminal

If you remove a remote terminal from your system, you must clean up your system by performing the following steps:

1. Find the system's name for the terminal. If you have been receiving messages on your console that are similar to:

   ```
   Unable to access ttyxx
   ```

   then the system's name for the terminal is the `ttyxx` name.

   If you have not received the message, you must determine the name of the remote terminal by the following method:

   a. type `cd /dev`

   b. type `ls -l tty*`

      You will see lines similar to:

      ```
      crw--w--w-   1  jaci    axe      1 0x000000 1986    /dev/ttyxx
      ```

      If you have only one `tty` file, then that is the terminal you are removing. If you have several, then determine which one you wish to remove by using the information you used to set up the terminal. Refer to the section "Minor Number for Ports".

2. Edit the file `/etc/inittab`.

   Delete the line that has a field with the words:

   `id:rstate:respawn:/etc/getty` *ttyxx*

   where *ttyxx* is the port you identified in Step 1.

3. Notify the `init` process that `/etc/inittab` has changed by typing:

   ```
   telinit q
   ```

4. Remove the device file associated with the terminal you removed.

Pseudo Terminals

Some applications need a form of software support which enables them to act as though they are talking to a terminal. This is implemented as a facility called a pseudo terminal. A **pseudo terminal** is a pair of character devices: a **master** device and a **slave** device. The pseudo terminal is structured so that output from either process acts as input to the other. The slave device interacts with the application process. It provides processes (in this case, user applications) an interface identical to that described in *termio(4)* of the *HP-UX Reference* manual. The master device interacts with the server process controlling the application process. It interacts through the device as though it were a hardware terminal interface.

The difference between an HP-UX pseudo terminal and the interface described in `termio` is that the latter always has a hardware device behind it—like an HP 2623 terminal. A slave device has another process manipulating it through the master half of the pseudo terminal. Anything written on the master device is given to the slave device as input, and anything written on the slave device is presented as input on the master device.

Device Files for Pseudo Terminals

The four things to consider when creating device files are:

- *Name* of the device file.

  According to HP-UX naming conventions, all pseudo terminal devices are located in the directories `/dev/pty` (slaves), and `/dev/ptym` (masters). The master device file should be called `/dev/ptym/ptyXX`, and the slave side `/dev/pty/ttyXX`, where *XX* is an identifying letter from p to w, and a hexadecimal digit. *Do not change these naming conventions because some programs depend on them.*

  As an example, `/dev/ptym/ptyp0` (master) and `/dev/pty/ttyp0` (slave) would be the lowest numbered pseudo terminal pair; `/dev/ptym/ptywf` and `/dev/pty/ttywf` would be the highest ordered pair.

- *Type* of the device file.

  All pseudo terminals must be character device files.

- *Major number.*

  The master pseudo terminal device driver must have a major number of 16. The slave pseudo terminal device driver must have a major number of 17.

- *Minor number.*

  The minor number for both master and slave pseudo terminal device files is:

  0x00*YYYY*

  where *YYYY* is a unique hexadecimal value, in the range of 0 to `npty-1`, where `npty` is a configurable system parameter. (Refer to "Configuring Operating System Parameters" in Chapter 6 and to Appendix D in Volume 2 of this manual if you wish to read about this parameter.) This value is used to identify the relationship between master and slave.

Table 4–3 shows a general template for pseudo devices:

Table 4–3. General Template for Pseudo Devices

Device	C/B	Major	Minor	Notes
ptyXX	c	16	0x00YYYY	Master side of pseudo terminal
ttyXX	c	17	0x00YYYY	Slave side of pseudo terminal

Using the lowest numbered pair, an example `mknod` command would be:

```
mknod /dev/ptym/ptyp0 c 16 0x000000
mknod /dev/pty/ttyp0  c 17 0x000000
```

These commands would create a master and slave pair called `ptyp0` and `ttyp0`. The minor numbers, shown above as zeros, must be in the range of 0 to `npty-1` where `npty` is a configurable system parameter. Refer to "Configuring Operating System Parameters" in Chapter 6 and to Appendix D (Volume 2 of this manual) if you wish to read about this parameter.

Your application's documentation will tell you how many pseudo terminals you need. For example, HP Windows/9000 needs three master/slave pairs per window. Based on your needs, it is up to you, as System Administrator, to create the pseudo terminals that are required.

For more information on pseudo terminals, refer to both the *termio(5)* and *pty(5)* sections of the *HP-UX Reference* manual.

Kernel Drivers for Pseudo Terminals

Pseudo terminals require the `ptymas` and `ptyslv` kernel drivers. Refer to the section "Configuring Device Drivers and I/O Cards" in the "Kernel Customization" chapter.

Hard Disks

This section explains how to set up the device files for hard disk drives. You can use your hard disk as part of your swap space, part of the HP-UX file system, or both.

Procedure

The general procedure for adding file system space is as follows:

1. Attach the disk drive to your system.

   Use your disk drive's hardware manual and/or the *Peripheral Installation Guide* for details.

2. Create two device files (one character type and one block type) using the `mknod` command.

   If you are on an HP-UX cluster, `/dev` is a context-dependent CDF. This means you must be logged into the correct cnode when creating the device file.

   The `mknod` syntax is:

   ```
   mknod     name     type     major_num     minor_num
   ```

   The four items that must go into the `mknod` command line are:

   - *Name* of the file.

     Using HP-UX naming conventions, you will create a character device file in the `/dev/rdsk` directory and a block device file in the `/dev/dsk` directory. For example, the device file names for a new disk drive might be:

     ```
     /dev/dsk/1s0          (block)
     /dev/rdsk/1s0         (character)
     ```

   - *Type* of the device file.

     Each hard disk (except the root disk) must have two device files associated with it: a block special file and a character or **raw** special file. *Type* for block is `b`, for character is `c`. Note that the root device should not have a character device file.

- *Major number.*

  The major number is different for block and character files, and is different for each type of disk drive:

Type	Block	Character
CS80	0	4
amigo	2	11
SCSI	7	47

  If you don't know what type of disk drive you have, look up the product number of your mass storage device in the section "Mass Storage Device Kernel Driver and Major Numbers" at the end of this section.

- *Minor number.*

  The address-dependent minor number is the same for both block and character entries. It has the form: 0x*SCBAU*0

  The 0x designates this as a hexadecimal number. The select code (*SC*) is set on the interface card, the bus address (*BA*) is set on the disk drive, and the unit (*U*) designates which drive (always 0 if hard disk only, if it is an integrated device the hard disk unit is 0, the cartridge tape or flexible disk drive unit is 1). The minor number must be given in hexadecimal format.

For example if you have a CS80 hard disk drive, at select code 14, bus address 1, your mknod command lines would be:

```
mknod /dev/dsk/1s0 b 0 0x0e0100
mknod /dev/rdsk/1s0 c 4 0x0e0100
```

If this is an integrated device (hard disk drive with either a flexible disk drive or a cartridge tape drive) you would also create device files for the other drive. For example, if you were adding an integrated hard disk and flexible disk drive, you would type, in addition to the above, the following two lines (notice the last two digits are 01 instead of 00 because it is unit 1):

```
mknod /dev/dsk/2s0 b 0 0x0e0110
mknod /dev/rdsk/2s0 c 4 0x0e0110
```

Hint: To help your users (and yourself) remember the names of the disk drive you should label the disk drive with the device file path name.

3. Change access permission on the new device files.

   You must have restricted access permission on all device files that are associated with mountable file systems, giving read/write permission to the owner (`root`) only. This prevents someone from mounting unauthorized media on your system, and prevents everyone on the system from accidentally overwriting a file system residing on the device associated with this device file.

   For example:

   ```
   chown root /dev/dsk/1s0 /dev/rdsk/1s0
   chmod 600 /dev/dsk/1s0
   chmod 600 /dev/rdsk/1s0
   ```

4. Look in `/etc/conf/dfile` to see if your kernel includes the appropriate kernel driver for your disk drive and interface card. If it doesn't, you must add the driver to `/etc/conf/dfile` and remake the kernel (this involves a reboot of the system).

   If you do not know what kernel driver you need, look up your disk drive's part number in Table 4–4 at the end of this section. Table 4–4 shows the required kernel driver for your disk drive. Note that you also must check to see if you have the correct interface card configured. Refer to the section "Kernel Driver and Major Number Information for Hard Disk Drives, Cartridge Tape Drives, and Flexible Disk Drives" for details.

   For example, if you added a SCSI disk drive, and you do not have the `scsi` driver listed in `/etc/conf/dfile`, you need to add this driver and reconfigure your kernel. Refer to the section "Configuring Device Drivers and I/O Cards" in Chapter 6 for information on kernel configuration.

5. Initialize the hard disk.

   The initialization command is `mediainit`. The syntax is:

   ```
   mediainit [options] pathname
   ```

   The most common *option* is the interleave factor. Most hard disks require an interleave factor of 1 (which is the default). The *pathname* is the path of the character device file you created in Step 2.

   For example, using a disk drive at `/dev/rdsk/1s0` with the default interleave factor of 1, you would type:

   ```
   mediainit  /dev/rdsk/1s0
   ```

   Refer to the procedures in the section "Initializing Media" in Chapter 5 if you want more details on initializing your hard disk.

6. Create a file system.

   If you will use the hard disk as part of your file system, you must create a file system on the disk by using the `newfs` command. The basic syntax is:

   ```
   newfs [options] device_file disk_type
   ```

   where *device_file*is the character device file you created in Step 2, and *disk_type* is the disk drive's entry in the `/etc/disktab` file. For more details on the options refer to the "Creating a New File System" in this chapter.

   For example, if you wish to create a file system on an HP 7958 disk drive at `/dev/rdsk/1s0` with the default amount of swap space you would type:

   ```
   newfs /dev/rdsk/1s0 hp7958
   ```

   SCSI disk drives are treated the same as CS80 disk drives. An example for an HP 7959S SCSI disk drive at `/dev/rdsk/2s0` with the default swap space is:

   ```
   newfs /dev/rdsk/2s0 hp7959s
   ```

   Refer to the `/etc/disktab` file for additional entries.

7. If you will use this disk drive for swapping, configure your kernel to enable swap space (this involves a reboot of the system).

This involves adding a swap entry to your `dfile`. Refer to the section called "Configuring Swap Space" in Chapter 6 for instructions if you will use this disk for additional swap space.

8. If this disk drive will be used for file system space, create a mount directory.

   Create the mount directory and give the directory the correct access permissions. Mount directories used by everyone on the system should have read/write/execute permission for everyone. For example:

   ```
   mkdir /misc
   chmod 777 /misc
   ```

9. If this will be a permanent addition to your file system hierarchy, add an entry to the `/etc/checklist` file.

   For example, if you were going to use the example disk as both permanent swap and permanent file system (mounted on the `/misc` directory), you would add the following two entries to `/etc/checklist`:

   ```
   /dev/dsk/1s0     /misc    /hfs    rw    2    0     #7958 file system
   /dev/dsk/1s0     /misc    swap                      #7958 swap
   ```

   With entries in `/etc/checklist` these devices will be mounted or swap-enabled automatically at system bootup.

HP-UX Clusters

The `/dev` directory is a CDF. For this reason you must be aware of the following things:

- The root disk (`/dev/dsk/0s0`) must be under `/dev+/localroot`. It was automatically set up this way when you created the cluster environment. Do not change it.
- You cannot mount any disks that are not associated with the root server. For example, you cannot mount the disk at `/dev+/cnode1/dsk/1s0`.
- If you are creating device files on an HP-UX cluster, you must log into the correct cnode to create the device file.

The Root Disk

There is a default device file, named /dev/dsk/0s0, created during the installation process. This device file is necessary for certain operations of the HP-UX file system (such as checking file system integrity with the `fsck` command). This pseudo device's actual major number is 255 and its minor number is 0xFFFFFF. However, the HP-UX kernel dynamically replaces the device file's major and minor numbers with those of the device that the system is actually rooted from. Thus, this device file does not require change regardless of system configuration. You should replace this special file only if it is somehow destroyed. To do so use the mknod command as shown below *(you must create this file from the root server if you are on a cluster):*

```
mknod /dev/dsk/0s0 b 255 0xFFFFFF
```

Note that the major and minor numbers shown by performing a list directory command (`ls -l`) on this file will be the current root device's numbers, not 255 and 0xFFFFFF.

You cannot copy this file to a new system. You must use mknod *to create it on every system.*

Kernel Driver and Major Number Information for Hard Disk Drives

Table 4–4 shows the kernel driver and major device number for hard disk drives. Your hard disk drive can be of type CS80, amigo, or SCSI.

CS/80 and AMIGO drives. In addition to the `cs80` or `amigo` device driver (as shown in Table 4–4) you must also include the appropriate interface driver. If your drive is connected to the internal HP-IB you need the kernel driver 98624. If your drive is connected to the high-speed HP-IB you need the kernel driver 98625. Some of the Amigo drives can only be used with the 98624 kernel driver.

If you don't know what interface your peripheral device is connected to, or can't decide which interface to connect it to, read the information for your peripheral in the *Peripheral Installation Guide.*

SCSI drives. Any SCSI drive should be connected to a SCSI interface card. The SCSI disk drives require a `scsi` kernel driver. The kernel code for the SCSI interface card is automatically included when you include the `scsi` driver.

SCSI and CS80 disks are very similar. Although the information for `mknod` differs between the two, `mediainit`, `newfs`, and `mount` are used the same for both, as are all user-level commands.

Table 4-4. Kernel Driver & Major Numbers for Hard Disk Drives

Product Number or Name	Kernel Driver	Block-Type Major Number	Character-Type Major Number
HP 7907, HP 7908, HP 7911, HP 7912, HP 7914, HP 7933, HP 7935, HP 7936, HP 7937, HP 7941, HP 7942, HP 7945, HP 7946, HP 7957, HP 7957B, HP 7958, HP 7958B, HP 7959B, HP 9122, HP 9125, HP 9127, HP 9133D/H/L, HP 9134D/H/L, HP 9144, HP 9145, HP 9153, HP 9154	cs80	0	4
HP 7906, HP 7920, HP 7925, HP 82901, HP 82902, HP 9121, HP 9133A/B/V/XV, HP 9134A/B/XV, HP 9135, HP 9138A, HP 9895	amigo	2	11
HP 7957S, HP 7958S, HP 7959S	scsi	7	47

Rewritable Optical Disk Drives and Optical Autochangers

This section explains how to set up the device files for rewritable optical drives. You can use your optical disk setup, standalone or autochanger, as part of the file system or as secondary storage (e.g. backups, archives). Additionally, the standalone optical disk drive can be used as part of your swap space.

HP Series 6300 Model 650/A Optical Disk Drive

The general procedure for adding file system space is as follows:

1. Attach the disk drive to your system.

   Since this is a SCSI drive, it must be connected to a SCSI interface card. For detail, reference the Peripheral Installation Guide.

2. Create two device files (one character type and one block type) using the mknod command.

   The mknod syntax is:

   ```
   mknod        name       type       major_num       minor_num
   ```

   The four items that must go into the mknod command line are:

   - Name of the file

     Using HP-UX naming conventions, you will create a character device file in the `/dev/rdsk` directory and a block device file in the `/dev/dsk` directory. For example, the device file names for a new disk drive might be :

     `/dev/dsk/mo` *(block)*
     `/dev/rdsk/mo` *(character)*

     If you are on a cluster, `/dev` is a CDF. This means you must be logged into the correct cnode when creating the device file.

   - Type of the device file

     Each optical disk drive must have two device files associated with it, a block special file and a character (raw) special file. Type for block is b, for character is c.

- Major number

  The major number is 7 for block files and 47 for character files.

- Minor number

  The minor number is address-dependent and is the same for both block and character entries. It has the form 0x*ScBa*00

  The 0x designates this is a hexadecimal number. The select code (Sc) is set on the interface card, and the bus address (Ba) is set on the disk drive. The minor number must be given in hexadecimal format.

For example, if you have an optical disk drive, at select code 14, using bus address 1, your mknod command lines would be:

```
mknod /dev/dsk/mo b 7 0x0e0100
mknod /dev/rdsk/mo c 47 0x0e0100
```

3. Change access permission on the new device files.

   You must have restricted access permission on all device files that are associated with mountable file systems, giving read/write permission to the owner (root) only. This prevents someone from mounting unauthorized media on your system, and prevents everyone on the system from accidentally overwriting a file system that resides on the device associated with this device file.

   For example:

   ```
   chown root /dev/dsk/mo /dev/rdsk/mo
   chmod 600 /dev/dsk/mo
   chmod 600 /dev/rdsk/mo
   ```

4. Look in `/etc/conf/dfile` to see if the SCSI kernel driver is included. If it isn't, you must add the driver to `/etc/conf/dfile` and reconfigure your kernel (this involves a reboot of the system). Refer to the section "Configuring Device Drivers and I/O Cards" in Chapter 6 for information on kernel configuration.

5. Initialize the media.

Initialize the media by typing in the mediainit command using the character special device file's pathname (for example, use /dev/rdsk/mo rather than /dev/dsk/mo):

```
mediainit /dev/rdsk/mo
```

6. Create a file system.

   If you will be using the optical disk as part of your file system, create a file system on the disk and create a mount directory.

   a. Create a file system with the newfs command. The basic syntax is:

      newfs *[options] device_file disk_type*

      where *device_file* is the character device file you created in Step 2, and *disk_type* is the disk drive's entry in the /etc/disktab file. For more information on the options available refer to the "Creating a New File System" section in this chapter.

Note Since the media is removable, use the noswap entry in /etc/disktab.

      For example, if you are using a Model 650/A disk drive at /dev/rdsk/mo, you would type:

      ```
      newfs /dev/rdsk/mo hpS6300.650A_noswap
      ```

   b. Create the mount directory and give the directory the correct access permissions. Mount directories used by everyone on the system should have read/write/execute permission for everyone. For example:

      ```
      mkdir /misc
      chmod 777 /misc
      ```

   c. To mount the file system, use the mount command. For example, to mount an optical disk in the drive at /dev/dsk/mo onto the /misc directory you would type:

      ```
      /etc/mount /dev/dsk/mo /misc
      ```

Caution Do not eject the media until it has been unmounted.

Because optical media is removable, do not use the media as part of your automatically-mounted file systems (e.g. do not add this disk drive to `/etc/checklist`). For more information on mounting and unmounting, refer to "Mounting and Unmounting File Systems" in this chapter.

7. If you will use optical disks as miscellaneous file storage (i.e. for backups), you can read to and write from them using the `cpio` or `tar` commands.

Using the Model 650/A as a boot device. The Model 650/A can be used as a temporary or emergency boot device. To prepare the optical disk drive for use as a boot device you must first perform the following:

1. Backup your current boot device to the optical disk as follows:

```
dd if=/dev/rdsk/hd of=/dev/rdsk/mo bs=64k
```

2. Test the backup copy by rebooting the system using the optical disk drive. If you are unfamiliar with the boot process, refer to chapter 3, "System Startup and Shutdown."

Caution If you are using the Model 650/A as a boot device, insert the media *before* the system is powered up and do not remove until after the system is powered down.

HP-UX Clusters. The `/dev` directory is a CDF. For this reason you must be aware of the following things:

- You cannot mount any disks that are not associated with the root server. For example, you cannot mount the disk at `/dev+/cnode1/dsk/1s0`.
- If you are creating device files on an HP-UX cluster, you must execute the `mknod` command from the device's cnode.

HP Series 6300 Model 20GB/A Optical Autochanger

The general procedure for adding an optical autochanger is found in the following steps. Since you are probably not initially familiar with the decisions necessary to optimize the use of the optical autochanger for your users' particular needs, these steps will guide you to a standard setup. Later you will want to customize some processes for your environment.

1. Attach the unit to your system.

   Use your disk drive's hardware manual and/or the *Peripheral Installation Guide* for details.

2. Create two device files (one character type and one block type) for each optical disk surface.

   If you are on an HP-UX cluster, the device files are context dependent. This means you must create the device files from the appropriate cnode.

   Since there are 129 mknod commands to execute—32 disks, 64 surfaces, 2 device files per surface and 1 ioctl setup—use the `mkdev` script file for creating the device files. Before executing the `mkdev` script, you should understand the individual `mknod` commands.

   The mknod syntax is:

   ```
   mknod name type major_number minor_number
   ```

   The four items that must go into the mknod command line are:

   - Name of the file.

     Using HP-UX naming conventions, you will create a character device file in the /dev/rac directory and a block device file in the /dev/ac directory. For example, the device file names for a new optical disk drive might be:

`/dev/rac/1a`	*(character)*
`/dev/ac/1a`	*(block)*

   - Type of the device file.

     Each optical disk surface must have two device files associated with it: a block special file and character raw special file. Type for block is b, for character is c.

   - Major number.

     The major number is 10 for block and 55 for character files.

   - Minor number.

The address-dependent minor number is the same for both block and character entries. It has the form: 0x*ScISur*

The 0x designates this as a hexadecimal number. The select code (Sc) is set on the interface card, the address (I) is set on the optical autochanger, and the surface (Sur) is identified as 01, 02, ... 3f, 40.

For example, if you have a SCSI optical autochanger, at select code 14 and address 3, your `mknod` command lines for one disk, both sides, would be:

```
mknod /dev/ac/1a   b 10 0x0e3001  <-- Side 1a/disk one block device
mknod /dev/rac/1a  c 55 0x0e3001  <-- Side 1a/disk one character device
mknod /dev/ac/1b   b 10 0x0e3002  <-- Side 1b/disk one block device
mknod /dev/rac/1b  c 55 0x0e3002  <-- Side 1b/disk one character device
```

You would continue with `mknod` commands until you ended with the thirty-second disk, both sides, as in:

```
mknod /dev/ac/32a  b 10 0x0e303f  <-- Side 32a/disk 32 block device
mknod /dev/rac/32a c 55 0x0e303f  <-- Side 32a/disk 32 character device
mknod /dev/ac/32b  b 10 0x0e3040  <-- Side 32b/disk 32 block device
mknod /dev/rac/32b c 55 0x0e3040  <-- Side 32b/disk 32 character device
```

Note: The default SCSI select code is 14 and the default interface set on the optical autochanger controller is 3. In the event you have other SCSI peripherals set at interface 3, 4, or 5, you should check the optical autochanger installation manual for information on how to change these defaults.

Now that you know how the device files are created with the mknod commands, here is an automated way to set up the device files.

a. Move to the `/etc` directory, save any old copies of the `mkdev` script, and create a new copy of `mkdev` by typing:

```
cd /etc
cp mkdev mkdev.old
config -a
```

b. Customize the `/etc/mkdev` file.

To edit the `/etc/mkdev` script to create the device files for the optical autochanger, use an editor such as vi and perform the following steps:

i. Find the optical autochanger section of the script. (keyword search on "AUTOCHANGERS")

ii. Fill in the template's missing parameters (or placeholders for parameters) where the instructions indicate; this includes the major and minor numbers.

iii. Remove the comment sign (#) in front of the modified template so the line will be executed when you run the script.

c. After you have modified the file, execute the script by logging in as the user root and typing:

```
/etc/mkdev
```

The `/etc/mkdev` script performs the following:

- creates two subdirectories—`ac` and `rac`
- sets permissions on the subdirectories to 755
- creates 129 device files

3. Look in `/etc/conf/dfile` to see if your kernel includes the `scsi` and `ac` drivers. If not, you must add the drivers to `/etc/conf/dfile` and remake the kernel (this involves a reboot of the system). Refer to the section "Configuring Device Drivers and I/O Cards" in Chapter 6 for information on kernel configuration.

4. Initialize the optical disk surfaces.

The initialization command is `mediainit`. The syntax is:

`mediainit` *[options] pathname*

You must perform 64 "mediainits"—one for each side of the 32 disks that the autochanger can hold. Note: Use the mediainit command on the raw/character device only. Here's an example of the `mediainit` commands for the optical autochanger:

```
mediainit /dev/rac/1a
mediainit /dev/rac/1b
 :
mediainit /dev/rac/32a
mediainit /dev/rac/32b
```

Because initializing all 32 disks can be a lengthy process, you can choose to manually initialize only a few disks to get started using your system. However, if you choose to initialize all 32 disks, create a script that will perform all of the `mediainits`, and execute this script in background mode.

5. Create a file system.

   Before storing files on any of the surfaces of the optical disks, you must create a file system on each surface. The basic syntax is:

   `newfs` *[options] device_file disk_type*

   where *device_file* is the character device file you created in Step 2, and *disk_type* is the disk drive's entry in the `/etc/disktab` file. For more details on the options refer to the "Creating a New File System" in this chapter.

   Here is an example of creating a file system on the surfaces of an autochanger:.

```
newfs /dev/rac/1a hpS6300.650A_noswap
newfs /dev/rac/1b hpS6300.650A_noswap
 :
newfs /dev/rac/32a hpS6300.650A_noswap
newfs /dev/rac/32b hpS6300.650A_noswap
```

   The *disk_type* parameter of `hpS6300.650a_noswap` is the *disk_type* name found in the `/etc/disktab` file. Both the standalone optical disk drive and the autochanger use the same *disk-type* parameter. The optical autochanger should not be used for swap space so use the `_noswap` option.

If you wish to automate this task, edit your work file script from step 4 or create a new script file.

6. Create a mount directory.

   Create the mount directory and give the directory the correct access permissions. Mount directories used by everyone on the system have read/write/execute permission for everyone. For example:

   ```
   mkdir /mountpt1a
   chmod 777 /mountpt1a
   ```

   You need to create a unique mount directory for each surface.

7. Mount the surfaces.

   To assure timely product performance, you should mount only two surfaces at a time as readable/writable. All of the other surfaces should be read only. Additionally, none of the surfaces should be permanently mounted; therefore, do not include the optical surface file systems in the `/etc/checklist` file.

   Here are examples of the commands to mount the two surfaces as read/write:

   ```
   /etc/mount   /dev/ac/1a    /mountpt1a
   /etc/mount   /dev/ac/1b    /mountpt1b
   ```

   Here are examples of the commands to mount the remaining surfaces as read only:

   ```
   /etc/mount   /dev/ac/2a    /mountpt2a   -r
   /etc/mount   /dev/ac/2b    /mountpt2b   -r
   :
   /etc/mount   /dev/ac/32a   /mountpt32a  -r
   /etc/mount   /dev/ac/32b   /mountpt32b  -r
   ```

   As the system administrator you might want to build a script with these commands that you can execute upon system startup. You can also add this script to the `/etc/rc` file in the `localrc` shell function. The `/etc/rc` file is automatically executed at system startup.

8. Unmount the surfaces.

   In the event you need to unmount an optical disk surface, use the umount command as follows:

   ```
   /etc/umount  /dev/ac/1a   -or-
   /etc/umount /mountpt1a
   ```

   All surfaces are unmounted automatically when the system is shut down.

Flexible Disk Drive

This section explains how to set up the device files for flexible disk drives. You can use your flexible disk as part of the file system or as miscellaneous storage space (e.g. for backups). Flexible disks are often called floppies. There are two kinds of flexible disks: 5.25 inch and 3.5 inch. The 3.5 inch flexible disks are more common with an HP-UX system than the 5.25. They are treated the same.

Procedure

The general procedure for adding a flexible disk drive is as follows:

1. Attach the disk drive to your system.

   Use your disk drive's hardware manual and/or the *Peripheral Installation Guide* for details.

2. Create two device files (one character type and one block type) using the `mknod` command.

   If you are on an HP-UX cluster, `/dev` is a context-dependent CDF. This means you must be logged into the correct cnode when creating the device file.

   The `mknod` syntax is:

   `mknod` *name* *type* *major_num* *minor_num*

   The four items that must go into the `mknod` command line are:

   - *Name* of the file.

     Using HP-UX naming conventions, you will create a character device file in the `/dev/rdsk` directory and a block device file in the `/dev/dsk` directory. For example, the device file names for a new flexible disk drive might be:

     ```
     /dev/dsk/1s0        (block)
     /dev/rdsk/1s0       (character)
     ```

   - *Type* of the device file.

     Each flexible disk drive must have two device files associated with it: a block special file and a character or **raw** special file. *Type* for block is `b`, for character is `c`.

- *Major number.*

  The major number is 0 for block device files and 4 for character device files.

- *Minor number.*

  The address-dependent minor number is the same for both block and character entries. It has the form: 0x*SCBAU*0

  The select code (*SC*) is set on the interface card, the bus address (*BA*) is set on the flexible disk drive, and the unit (*U*) designates which drive on a dual or integrated disk drive. The minor number must be given in hexadecimal format.

For example, if you have a flexible disk drive at select code 7, bus address 1, your `mknod` command lines would be:

```
mknod /dev/dsk/1s0 b 0 0x070100
mknod /dev/rdsk/1s0 c 4 0x070100
```

If this is an integrated device (both hard disk and flexible disk drive in the same unit), the hard disk drive is unit 0 and the flexible disk drive is unit 1. If you have a dual disk drive, one is unit 0 and the other is unit 1. In either case, you would create four device files: one block and one character device file for each unit. For example, if you have a CS80 drive at select code 7, bus address 1, your `mknod` command lines would be:

```
mknod /dev/dsk/1s0 b 0 0x070100
mknod /dev/rdsk/1s0 c 4 0x070100
mknod /dev/dsk/2s0 b 0 0x070110
mknod /dev/rdsk/2s0 c 4 0x070110
```

Hint: To help your users (and yourself) remember the names of the disk drive you should label the flexible disk drive with the device file path name.

3. If you will use this flexible disk drive to hold a mounted file system, change access permission on the new device files.

   You must have restricted access permission on all device files that are associated with mountable file systems, giving read/write permission to the owner (`root`) only. This prevents someone from mounting unauthorized media on your system, and prevents everyone on the system from accidentally overwriting a file system residing on the device associated with this device file.

   For example:

   ```
   chown root /dev/dsk/1s0 /dev/rdsk/1s0
   chmod 600 /dev/dsk/1s0
   chmod 600 /dev/rdsk/1s0
   ```

4. Look in `/etc/conf/dfile` to see if your kernel includes the appropriate kernel drivers for your flexible disk drive (`cs80`) and interface card (either 98624 or 98625). If it does not, add the drivers to your `/etc/conf/dfile` and remake the kernel (this involves a reboot of the system).

   Refer to the section "Configuring Device Drivers and I/O Cards" in Chapter 6 for information on kernel configuration.

5. Insert and initialize the flexible disks.

   The initialization command is `mediainit`. The syntax is:

   `mediainit` *[options] pathname*

   The most common *options* are the interleave factor and format option. Most flexible disks require an interleave factor of 2. The format option depends on the type of flexible disk (refer to your disk drive's entry in `/etc/disktab`). The *pathname* is the path of the character device file you created in Step 2.

   For example, using an HP 9122 disk drive with a 788 Kbyte flexible disk at `/dev/rdsk/1s0`, you would insert the flexible disk and type:

   ```
   mediainit -i 2 -f 3 /dev/rdsk/1s0
   ```

   The formatting and interleave information (and more information) is in the `/etc/disktab` file under the part number of your disk drive. Refer to `/etc/disktab` or to the procedures in the section "Initializing Media" in Chapter 5 if you want more details on initializing your flexible disk.

6. If you wish to use this flexible disk as file system space, create a file system on the flexible disk and create a mount directory.

    a. Create a file system with the `newfs` command. The basic syntax is:

        `newfs` *[options] device_file disk_type*

        where *device_file* is the character device file you created in Step 2, and *disk_type* is the disk drive's `noswap` entry of the `/etc/disktab` file (you should not use a flexible disk as part of your swap area). For more details on the options refer to the "Creating a New File System" in this chapter.

        For example, using an HP 9122 disk drive at `/dev/rdsk/1s0`, you would type:

        ```
        newfs /dev/rdsk/1s0 hp9122_noswap
        ```

    b. Create the mount directory and give the directory the correct access permissions. Mount directories used by everyone on the system should have read/write/execute permission for everyone. For example:

        ```
        mkdir /misc
        chmod 777 /misc
        ```

    c. To mount the file system, use the `mount` command. For example, to mount a flexible disk in the drive at `/dev/dsk/1s0` onto the `/misc` directory you would type:

        ```
        /etc/mount /dev/dsk/1s0 /misc
        ```

        Because flexible disks are removable, do not use them as part of your automatically mounted file systems (e.g. do not add this disk drive to `/etc/checklist`). For more information on mounting and unmounting, refer to "Mounting and Unmounting File Systems" in this chapter.

7. If you will use this flexible disk as miscellaneous file storage (i.e. for backups), you can read to and write from the flexible disk using the `cpio`, `tar`, or `backupf` command.

HP-UX Clusters

The /dev directory is a CDF. For this reason you must be aware of the following things:

- You cannot mount any disks that are not associated with the root server. For example, you cannot mount the disk at /dev+/cnode1/dsk/1s0.
- If you are creating device files on an HP-UX cluster, you must execute the mknod command from the device's cnode.

Cartridge Tape Drive

This section explains how to set up device files for cartridge tape drives. Cartridge tapes are the 1/4 -inch tapes in plastic cartridges (as opposed to 9-track magnetic tape on reels). You can use your cartridge tape for miscellaneous storage space (e.g. for backups).

The general procedure for adding a cartridge tape drive is as follows:

1. Attach the tape drive to your system.

   Use your tape drive's hardware manual and/or the *Peripheral Installation Guide* for details.

2. Create a character-type device file using the `mknod` command.

   If you are on an HP-UX cluster, `/dev` is a context-dependent CDF. This means you must be logged into the correct cnode when creating the device file.

   The `mknod` syntax is:

   `mknod` *name* *type* *major_num* *minor_num*

   The four items that must go into the `mknod` command line are:

   - *Name* of the file.

     Using HP-UX naming conventions, you will create a character device file in the `/dev/rct` directory. For example, the device file name for a new cartridge tape drive might be:

     `/dev/rct/0s0`

   - *Type* of the device file.

     Each cartridge tape drive must be a character or **raw** special file. *Type* for a character file is `c`.

   - *Major number*

     The major number is 4.

   - *Minor number*

     The address-dependent minor number has the form: 0x*SCBAU*0

The select code (SC) is set on the interface card, the bus address (BA) is set on the cartridge tape drive, and the unit (U) designates which drive on an integrated disk/tape drive. The minor number must be given in hexadecimal format.

For example, if you have a CS80 cartridge tape drive at select code 14, bus address 1, your `mknod` command line would be:

```
mknod /dev/rct/0s0 c 4 0x0e0100
```

If this were an integrated device (both hard disk and cartridge tape drive in the same unit), the hard disk drive is unit 0 and the cartridge tape drive is unit 1. You would create three device files: one block device file for the hard disk, and one character device file for each unit. For example, if you have a CS80 drive at select code 14, bus address 1, your `mknod` command lines would be:

```
mknod /dev/dsk/1s0 b 0 0x0e0100
mknod /dev/rdsk/1s0 c 4 0x0e0100
mknod /dev/rct/0s0 c 4 0x0e0110
```

Hint: To help your users (and yourself) remember the names of the drive you should label the drive with the device file path name.

3. Look in `/etc/conf/dfile` to see if your kernel includes the appropriate kernel drivers for your cartridge tape drive (`cs80`) and interface card (either `98624` or `98625`). If not, add the drivers to your `/etc/conf/dfile` and remake the kernel (this involves a reboot of the system).

   For information on kernel configuration, refer to the section "Configuring Device Drivers and I/O Cards" in Chapter 6.

Nine-Track Magnetic Tape

When you set up your magnetic tape drive, you must create device files for them. The four items that must go into the mknod command line are:

- *Name* of the file.

  Using HP-UX naming conventions, the device file names will be created under special directories in the /dev/rmt directory.

- *Type* of the device file.

  All magnetic tape drive device files should be character type (c).

- *Major number.*

  The major numbers for magnetic tape drives is described in the section "Kernel Driver and Numbers for Magnetic Tape Drives".

- *Minor number* (refer to the next section for a description of magnetic tape minor numbers).

Magnetic Tape Minor Number

The minor number consists of the following fields:

0x*ScBaUV*

0x — This prefix indicates the number is hexadecimal.

Sc — This field is a two-digit hexadecimal representation of the select code. The select code is determined from the switch settings on the tape drive's interface card.

Ba — This field is a two-digit hexadecimal representation of the bus address. It is determined from the switch settings on the tape drive.

U — The single hexadecimal unit number (*U*) represents a four-bit binary value. Setting and clearing the bits of this binary value affects the manner in which the tape drive operates, as indicated in Table 4–5.

V — The volume number (*V*) field of the minor number also has special meaning when creating device files for magnetic tape drives. The single hexadecimal volume number represents a four-bit binary value. Setting and clearing the bits of this binary value affects the manner in which the tape drive operates, as indicated in Table 4–6.

Table 4–5 indicates the special meanings of each bit in the unit number portion of the magnetic tape minor number. Bits 6 and 7 select the tape density, while bits 4 and 5 represent the unit number, and "x"s represent "don't care":

Table 4–5. Tape Density and Unit Number Bit Settings

Hex Value	7	6	5	4	Selects
c	1	1	x	x	Density = 6250 bpi compressed(HP 7980XC only)
8	1	0	x	x	Density = 6250 bpi (HP 7978 and HP 7980 only)
4	0	1	x	x	Density = 1600 bpi (All mag tapes)
0	0	0	x	x	Density = 800 bpi (HP 7974, opt 800 only)
0	x	x	0	0	Select Unit 0
1	x	x	0	1	Select Unit 1, etc.

Table 4–6 indicates the special meaning each bit has in the volume number of the magnetic tape minor number:

Table 4–6. Magnetic Tape Operation Bit Settings

Bit Order	When Clear (0)	When Set (1)
3	Industry Standard mode	HP-UX 2.0 compatibility mode
2	Immediate report on (ignored by HP 7970/7971)	Immediate report off
1	AT&T-style compatibility mode	Berkeley-style compatibility mode
0	Rewind on close	No rewind on close

Example minor numbers for your tape drive are given in the *Peripheral Installation Guide.*

Naming Conventions for Magnetic Tape

The naming convention described in *mt(7)* of the *HP-UX Reference* is useful for keeping track of how a tape drive minor number is set up. The device `/dev/rmt/0mn` is the first tape mechanism, *m*edium (1600) density, *n*o rewind. The device `/dev/rmt/0h` is the same mechanism configured for *h*igh (6250) density operation, rewind on close (note the absence of *n* suffix).

If you connected an HP 7978 tape drive to select code 14, set the tape drive's bus address to 3, configured `stape` into the kernel, and executed the following commands:

```
mknod /dev/rmt/0mn c 9 0x0e0343
mknod /dev/rmt/0h c 9 0x0e0382
```

you could access the same drive as a 6250 bpi device using the "0h" device and as a 1600 bpi device using the "0mn" name. You could also use the "mt" command to do various positioning operations on the tape without having to provide a device name because `mt` uses the default device `/dev/rmt/0mn`. Since `tar` defaults to `/dev/rmt/0m`, you may also wish to create this file.

If you are on an HP-UX Cluster

The `/dev` directory is a CDF and is context dependent. For this reason you must be aware of the following things:

- If you are creating device files on an HP-UX cluster, you must execute the `mknod` command from the cnode that is connected to the device.
- You should perform your file system backups from the root server. There are two reasons for this:
    - □ Since the file system is connected to the root server, and backups onto any device other than the root server will need to go across the network, the backup will be slower if from any system other than the root server.
    - □ If you shut down your system to back up your system, you will be unable to access the file system from any diskless cnode.

Kernel Driver and Major Number for 9-Track Tape Drives

This section shows the necessary **kernel driver** and **major number** for your peripheral's product number. Magnetic tape drives do not require a block major number so no block-mode major numbers are given.

In addition to the `tape` or `stape` kernel driver as shown in Table 4–7, you must include the kernel driver for the tape drive's HP-IB. The tape drives using the `tape` kernel driver require the 98624 kernel driver. The tape drives using the `stape` kernel driver require either the 98624 or the 98625 kernel driver.

If you don't know what interface your peripheral device is connected to, or can't decide which interface to connect it to, read the information for your peripheral in the *Peripheral Installation Guide*.

Table 4–7. Kernel Drivers and Major Device Numbers

Product Number or Name	Kernel Driver	Character-Type Major Number
HP 7970	tape	5
HP 7971	tape	5
HP 7974	stape	9
HP 7978	stape	9
HP 7979	stape	9
HP 7980	stape	9

For more information on the use of magnetic tape, refer to the "Magnetic Tape" section of the "System Management Concepts" chapter or the *mt(7)* section of the *HP-UX Reference*.

Plotters and Digitizers

To create the character special device file for your plotter or digitizer you must determine the following four parameters for the `mknod` command:

- Device file name.

  You must assign a *unique* special file name to each entry you create. Generally, use `plt` followed by the product number for plotters and `dig` followed by the product number for digitizers. If more than one device with the same product number is present, be certain not to duplicate their special file names. For example, to differentiate between two HP 7580 plotters, name the first one `plt7580` and the second `plt7580.1`.

- Type for the device file.

  Always character for plotters and digitizers.

- Major number.

  Always 21 for plotters and digitizers.

- Minor number.

  The minor number has the format: `0x`*ScBaUV*. The `0x` field indicates that the number is hexadecimal.

  The *Sc* field stands for select code. The select code is the hexadecimal representation of the switch setting on the plotter or digitizer's interface card.

  The *Ba* field stands for bus address. The bus address is the HP-IB address, and is the hexadecimal representation of the switch setting on your plotter or digitizer.

  The final two digits are unused and are both `0`.

In addition to the device file, you must make sure you have the correct kernel driver for your plotter or digitizer. You must have both the `hpib` kernel driver and the 98624 kernel driver.

HP-UX Clusters

The `/dev` directory is a CDF and is context dependent. For this reason you must be aware of the following things:

- If you are creating device files on an HP-UX cluster, you must execute the `mknod` command from the cnode that is connected to the device.
- If you will be adding the plotter or digitizer to a diskless cnode it will not be available from any other cnode.

HP-HIL Devices

HP-HIL devices include the HP Touch Bezel, keyboards, mouse, digitizers, and control knobs. To set up HP-HIL devices, there must be two character device files. There must be one device file, created with a major number 23, if any HP-HIL devices are present. This device file is created with the following mknod command line:

```
mknod /dev/raw_8042 c 23 0x000000
```

In addition, for each HP-HIL device you must create a device file with a major number 24, and a minor number reflecting which HP-HIL device it is. To create this device file use the following `mknod` command line, where *devname* is the unique name of the HP-HIL device file and *n* is the device's place on the HP-HIL loop:

mknod /dev/*devname* c 24 0x0000*n*0

The HP-HIL kernel driver is not an optional kernel driver so you never need to configure it into your kernel.

GPIO Devices

GPIO devices include HP 98622. This is a protocol used mostly for instruments. To set up a GPIO device, you need to create a device file using the following command line, where *Sc* is the select code on the GPIO interface card:

mknod /dev/gpio c 22 0x*Sc*0000

You also must verify (or configure) your kernel contains the `gpio` kernel driver.

Setting Up Data Communication on your System

Data communication means communicating to other people on your system or communicating to people on other computer systems. It also means moving (transferring) files from one system to another.

There are many data communication (datacom) packages on your system, including the following from HP:

- UUCP
- NS-ARPA Services
- NFS
- RJE
- SRM

UUCP

UUCP is a standard UNIX data and file transfer subsystem. You can read about setting up and using UUCP in the Concepts and Tutorials.

NS-ARPA Services

You can purchase Hewlett-Packard's local area network called NS-ARPA Services to communicate with other systems. Refer to the *Installing and Maintaining NS-ARPA Services* manual for information on how to install and set up NS-ARPA Services. You must install the `rfa`, `lla`, `lan01` and `nsdiag` device drivers.

NFS

You can purchase Hewlett-Packard's Network File Services (NFS). Refer to the appropriate manuals that came with the product for installation and setup information. To use this product you will need the `nfs` device driver.

RJE

You can purchase a product called RJE. Refer to the installation manual that comes with the product. You must install the `rje` device driver.

SRM

You can purchase utilities from Hewlett-Packard that allow your HP-UX system to transfer files to and from an SRM (Shared Resource Management). SRM is a hard disk hierarchical file system that you can use with either the Pascal Workstation or the BASIC operating system. Refer to the installation manual that comes with the product. You must install the `srm` device driver.

Kernel Driver and Major Number for Data Communications

This section shows the necessary **kernel driver** and **major number** for your data communication package. Use the `reconfig` program to add the necessary kernel drivers to your operating system (refer to "Creating a New Operating System" in Chapter 6).

Only mass storage devices require a block-mode major number, so there is no block-mode major number for data communication.

Table 4-8. Kernel Drivers and Major Device Numbers for Data Communications

Product Number or Name	Kernel Driver	Character-Mode Major Number
NS-ARPA Services	lla	18
NS-ARPA Services	lan01	19
NFS	nfs	N/A
SRM	srm	13
RJE	rje	15

Setting the Terminal Characteristics

For most effective use of your terminal, HP-UX needs to know the type of terminal or graphics display being used. If no terminal type is provided, the default value is `TERM=hp`. The `tset` command sets terminal characteristics.

The default value of `TERM` is an `hp` terminal. This value works with Hewlett-Packard terminals, but may not allow you to take full advantage of the features of your terminal or graphics display station.

The default local login scripts ask you to enter your terminal type with the prompt:

```
TERM = (hp)
```

If you press [Return], the `TERM` environment variable is set to `hp`. If you enter a different value, the `TERM` environment variable is set to the value you enter.

Selecting a Value for the TERM Variable

HP-UX supports many types of terminals. The `/usr/lib/terminfo` database tells HP-UX how to communicate with each terminal type. When you assign a value to `TERM`, it must equal a value in the `terminfo` database.

For example, in Figure 4–8, the files listed under `usr/lib/terminfo/2` show all the acceptable values for `TERM` that begin with 2.

```
$ ls /usr/lib/terminfo/2
2382        2397a       2621a       2623p       2626-x40    2640a
2392        2500        2621k45     2624        2626A       2640b
2392A       2621        2621nl      2624a       2626P       2644
2392a       2621-48     2621nt      2624p       2626a       2645
2393        2621-ba     2621p       2625        2626p       2647
2393A       2621-fl     2621wl      2626        2627        2647F
2393a       2621-nl     2622        2626-12     2627a       2648
2394        2621-nt     2622a       2626-12-s   2627p       2648A
2394A       2621-pb     2622p       2626-12x40  2628        2648a
2394a       2621-wl     2623        2626-ns     262x        2703
2397        2621A       2623a       2626-s      2640        2709
2397A       2621P
$
```

Figure 4-8. `usr/lib/terminfo/2` **Files**

Table 4-9 outlines the most common terminal and graphics display settings for Hewlett-Packard equipment. When more than one choice is provided, all choices are equivalent.

Table 4-9. Settings for the TERM Environment Variable

If you are using a ..	Set TERM to ..
terminal	the terminal's model number such as 2622, hp2622, or 262x
medium resolution graphics display (512x600 pixels)	3001 or hp3001
high resolution graphics display (1024x768 pixels)	300h or hp300h
HP 98550 display station (1280x1024 pixels)	98550, hp98550, 98550a, or hp98550a
HP 98720 or HP 98721 display station (1280x1024 pixels)	98720, hp98720, 98720a, hp98720a, 98721, hp98721, 98721a, or hp98721a

Setting TERM with the tset Command

The `tset` command is a flexible command that sets the value of TERM and initializes your terminal characteristics. If you always log in using the same terminal type, you may want to change your local login script and eliminate the TERM prompt. In the local script, the following command generates the TERM prompt:

```
eval ' tset -s -Q -m ':?hp' '
```

To customize this command, replace the `?hp` with the desired value of TERM. For example, the following command initializes your terminal as a high-resolution graphics display (300h):

```
eval ' tset -s -Q -m ':300h' '
```

The `tset` command allows you to define conditions for choosing between multiple terminal types. If you use more than one type of terminal (for example, one at work and one at home), you may want to modify your `tset` command to include multiple terminal types.

Adding to /etc/checklist

The `/etc/checklist` file contains a list of file systems and swap devices. A **file system** is an organization of files and directories on a disk. When you installed HP-UX, one file system, the `root` file system, was created. You may create several file systems (one per disk) if necessary. **Swap devices** are secondary storage devices, usually on a hard disk drive, which are reserved for use by the virtual memory management system. File systems, by default, leave room for swapping at the end of the disk.

Add an entry to `/etc/checklist` for each file system you want automatically mounted, checked, or used for swapping.

The format of an entry in `/etc/checklist` is:

special_filename directory type options bckup_freq pass_number comment

where:

special_filename — The block-mode device name used by `fsck`, `mount`, or `swapon`.

directory — The directory where *special_filename* is to be mounted. The *directory* must already exist and must be an absolute path name.

type

A code for the type of device. The possible values for type are: `hfs`, `nfs`, `swap`, and `ignore`.

`hfs` — Use if the file system is an HFS file system. The HFS file system is the standard file system on Series 300 HP-UX.

`nfs` — Use if the *special_filename* is a remote NFS file system. For NFS file systems, the *special_filename* field must be the serving machine name followed by :. The colon is followed by the path on the serving machine of the directory to be served. For example, `server:/mnt`. The fields *pass_number* and *bckup_frequency* are ignored for NFS entries.

`swap` — Use when the *special_filename* will be used as a swapping device. During system initialization, the `/etc/rc` file executes the `swapon -a` command. `swapon` will enable all devices in `/etc/checklist` that are labelled with type `swap`. The fields *directory, pass_number,* and *bckup_frequency* are ignored for all `swap` entries. All `swap` entries are ignored by the `mount` and `fsck` commands.

`ignore` — Use when the entry should not be used by any command. This can be used to mark currently unused entries. All `ignore` entries are ignored by the `mount` and `fsck` commands.

options

If the entry is marked as `hfs` and is to be used by the `mount` command, then this field can contain a list of comma-separated options to the `mount` command.

bckup_frequency — This field is reserved for future enhancements to backup utilities.

pass_number — The `fsck` program uses this field to determine the order file system checks are done when using the `-P` option (as in during bootup).

The `fsck` program will check the `hfs` type entries in the order you specify. The root file system (`/dev/dsk/0s0`) should always be "1" and it should be the only file system set to "1". Any file system labeled "2" will be checked after the root file system. File systems labeled "3" will be checked after the file systems labeled "2". If more than one file system is specified as the same value, then `fsck` deals with all of them in parallel. This shortens the time required for `fsck`. A file system with a *pass_number* of 0 will be ignored by the `fsck` command. If *pass_number* is not present, `fsck` will check each file system sequentially after all eligible file systems with pass numbers have been checked.

comment — This field is preceded by a "#". You can put any comment in this field.

The information stored in the `/etc/checklist` file is useful to the following HP-UX programs:

`mount -a` — During bootup (via `mount -a` in `/etc/rc`), or if you execute the command, `mount -a`, all file systems with a type of `hfs` or `nfs` in `/etc/checklist` will automatically be mounted.

`swapon -a` — During bootup (via `swapon -a` in `/etc/rc`), or if you execute the command, `swapon -a`, all swap devices with a type `swap` in `/etc/checklist` will be enabled.

`fsck` — If you execute `fsck` without providing a list of file systems, all devices in `/etc/checklist` of type `hfs` will be checked.

Note — If additional swap devices are added to the system, they must be configured as discussed in Chapter 6, the section "Configuring Swap Space".

The /etc/checklist file is also useful during file system checking and system shutdown. The shutdown command executes umount -a. The umount -a command attempts to unmount all file systems listed in the /etc/checklist file.

The example /etc/checklist file in Figure 4–9 will cause the following to happen at bootup:

- The fsck program will check both the /dev/dsk/0s0 and the /dev/rdsk/1s0 file systems, and will ignore all entries of type swap, nfs, and ignore, and all entries with *pass_number* of 0.
- All entries with type hfs and nfs will be mounted. In this example, /dev/dsk/1s0 will be mounted on the directory /usr with read/write permission. The remote NFS file system on the server machine will be mounted at the /mnt directory with read/write permission. This is accomplished automatically at bootup, using the mount -a command in the /etc/rc file.
- If the command /etc/swapon -a has been added to the /etc/rc file, then swapping to each device with the swap entry will be enabled during bootup.

```
/dev/dsk/0s0   /      hfs  defaults  1  0  # root disk
/dev/dsk/1s0   /usr   hfs  rw        2  0  #7945
/dev/dsk/1s0   /usr   swap                 #7945 swap
server:/mnt    /mnt   nfs   rw,hard  0  0  #mount from server
```

Figure 4–9. /etc/checklist File

If you are temporarily removing a disk from the system, you should invalidate the entries for that disk by changing the disk's type to ignore.

Note that root is assumed to be a swap device, so you don't need a separate swap entry for the root file system. All other file systems that are also swap devices must have both a file system entry (if you wish it to be mounted automatically) and a swap entry (if you wish it to be automatically enabled for swapping).

Configuring your HP-UX Cluster

Before using the diskless workstation capabilities you must create a clustered environment on the machine that will be your root server. You will use `/etc/reconfig` to create a clustered environment.

Prerequisites and Assumptions

Before creating a cluster environment you must have the following installed:

- Version 6.0 (or later) HP-UX (including the AXE, PE, and NS-ARPA Services/300 products).
- Model 319C+, 330, 350, 360, or 370 (if version 6.2 or later), Model 350 (if version 6.0) to use as a root server. The root server must be equipped with at least 8 Mbytes RAM, and at least a 130 Mbyte disk drive.

The steps given here assume you do not already have a clustered environment. If you do, you should add to the environment, rather than re-create it.

Note If this was an update, and you previously had NS-ARPA Services running on your machine, there are some important decisions you must make about which system and ARPA hostnames to use. Refer to *Installing and Maintaining NS-ARPA Services*.

HP-UX Cluster Information Sheet

You must gather information about the computers that will be on your cluster. This information will help you set up the cluster. The following chart is for the example cluster from the "Cluster Concepts" section of Chapter 2. It shows the cnode names, internet numbers, link level addresses, and select codes for the cnodes. There are blank forms on the next page. Copy and fill out as many as you need for your cluster.

Workstation	ARPA Hostname	Internet Address	Cluster LAN card's Link Level Address and Select Code
root server			/
cnode #1			/
cnode #2			/
cnode #3			/
cnode #4			/
cnode #5			/
cnode #6			/
cnode #7			/
cnode #8			/
cnode #9			/
cnode #10			/
cnode #11			/
cnode #12			/
cnode #13			/
cnode #14			/
cnode #15			/
cnode #16			/
cnode #17			/
cnode #18			/
cnode #19			/
cnode #20			/

Workstation	ARPA Hostname	Internet Address	Cluster LAN card's Link Level Address and Select Code
cnode #21			/
cnode #22			/
cnode #23			/
cnode #24			/
cnode #25			/
cnode #26			/
cnode #27			/
cnode #28			/
cnode #29			/
cnode #30			/
cnode #31			/
cnode #32			/
cnode #33			/
cnode #34			/
cnode #35			/
cnode #36			/
cnode #37			/
cnode #38			/
cnode #39			/
cnode #40			/

Example Cluster Information Sheet

Workstation	ARPA Hostname	Internet Address	Cluster LAN card's Link Level Address and Select Code (SC)
root server	daisy	192.25.204.1	0x080009004a11/SC=21
cnode #1	donald	192.25.204.2	0x0800090044ff/SC=21
cnode #2	dewey	192.25.204.3	0x08000900a63f/SC=21
cnode #3			
cnode #4			
cnode #5			
cnode #6			
cnode #7			
cnode #8			
cnode #9			
cnode #10			

Hostname

This is a 1-8 character name. The name must be unique in the cluster.

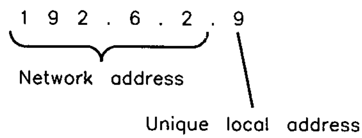
Internet Address

Unique network addresses make it possible for a network to communicate with other networks around the world. If your network has not been assigned a unique network address, contact:

Network Administration Office
Information Networks Division
Hewlett-Packard Company
19420 Homestead Road
Cupertino, California 95014
(408/725-8111)

If your network address is not assigned by the Network Administration Office and you try to link with other networks, you may need to change all the addresses on your network.

When you receive the network address portion of the internet address from the Network Administration Office, it is in decimal dot notation as shown below. Decimal dot notation consists of four address fields separated by periods (.):



The high order three numbers are the **network address.** The lower number is the **local address.** All nodes on a network share the same network address and each node has a unique local address. Once you have a network address, assign unique local addresses for each of your computers. Do not put leading zeros on any of the numbers; a leading zero indicates an octal number, not a decimal number. *Do NOT assign the local address values 0 or 255. These are reserved addresses.*

If you wish to read more about internet addresses refer to the *Installing and Maintaining NS-ARPA Services* manual.

LAN card link level address and Select Code

Each LAN card has a *link level address* associated with it. The link level address is set at HP and cannot be changed. You will need this number while setting up your clustered environment and when adding diskless cnodes.

If you have only one LAN card in the machine, leave this blank for now; you will be instructed to enter the value later.

If you have more than one LAN card on the machine, determine the select code and/or link level address for the cluster's LAN card now. You can get a list of all LAN cards and their select codes by using the `dmesg` command (you must be superuser).

Creating the Clustered Environment

Step 1. Start reconfig.

You must start the `reconfig` program from the root directory (as the root user) while in a quiet state (init state s). First, log in as the root user, then type:

```
cd /
shutdown
```

Wait for the message: "WARNING: YOU ARE SUPERUSER!!", then remount any file systems and start the `reconfig` program by typing:

```
mount -a
reconfig
```

You will see the menu shown in Figure 4–10.

```
*******************************************
          RECONFIG -- MAIN MENU
*******************************************

User Configuration
Workstation Configuration
Cluster Configuration








[        ] [        ] [  NEXT  ] [PREVIOUS]  [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 4–10. Reconfig Main Menu

Step 2. Select "Cluster Configuration"

Using the NEXT or PREVIOUS keys, highlight the `Cluster Configuration` menu item. Press the SELECT softkey. If you have shutdown your system as stated in Step 1 then you will see the screen shown in Figure 4–11. If not, then you will see the following message:

```
No extra processes can be running when setting up a clustered
environment.  Exit reconfig and use the shutdown (1M) command
to shutdown your system.

Press [Return] to return to Main Menu >>
```

If you followed the procedure shown in this step (before starting the `reconfig` program), you will be in init state s and you will not be in a windowed environment. You can type "y" and continue. You will now see the menu shown in Figure 4–11.

If you have not shut down your system, and you are not sure you are in the correct environment, type "n" and begin with Step 1.

If, instead of the above message, you see a menu allowing you to add or remove diskless cnodes, then the file called `/etc/clusterconf` already exists on your system. The `reconfig` program assumes that if `/etc/clusterconf` exists then you already have a clustered environment. To override this you must exit the `reconfig` program and remove the `/etc/clusterconf` file.

```
*******************************************
 RECONFIG -- SET UP A CLUSTER ENVIRONMENT
*******************************************

 Root server's Cluster Node Name:
 LAN Card's Link Level Address:
 NS-ARPA Internet Address:
 Min. # of Cluster Server Processes: 4









 Root Server's Cluster Node Name? >>






[RESTORE] [      ] [      ] [      ]   [ HELP ] [      ] [PREVMENU] [      ]
```

Figure 4–11. Creating a Cluster Menu

Step 3. Type the requested information.

1. Enter your root server's cnode hostname.

   Type the node name. The node name can be 1 to 8 ASCII characters. The name you enter will be used as your root server's HP-UX hostname and cnode hostname.

   If you have only one LAN card (for the cluster LAN) then continue with the Step 3.2 below. If you have two LAN cards, and one was previously set up to communicate with other systems, you must use that ARPA hostname here.

2. Select your cluster LAN card's link level address.

   This is the link level address of your cluster LAN card. You will see one address for each LAN card currently attached to your system. The list will be in the menu shown below:

   link_level_address `(at SC ##)`

   Use the `NEXT` or `PREVIOUS` softkeys to highlight the proper address, then press `SELECT`.

   Write this number onto your checklist.

3. Type your internet address.

   This should be on your checklist.

   If a value is automatically displayed for this field, that value is the internet address associated with the cnode name chosen in Step 3.1. This cnode name and its address already existed in the `/etc/hosts` file. You cannot change this number and should continue with Step 4. If the internet address is associated with the non-cluster LAN card you must follow the instructions in "After Setting Up the Cluster Environment".

4. Type the number of cluster server processes (CSPs) you wish to run.

   The default value is 4. The best number of cluster server processes depends on your particular cluster. Use the default unless you have previously determined a better value. The absolute upperbound is the value of the `ngcsp` operating system parameter. In general, the more cnodes on your cluster, the greater this value should be.

Once all the values are entered, you will be prompted to verify the numbers. Once you have verified the values you will see the following screen:

```
The required Context Dependent Files (CDF's) will
be created, a fully loaded Cluster kernel will be
built for the localroot, the needed NS-ARPA files
will be created or updated, and the /etc/clusterconf
file will be created with the localroot entry.
After this completes, the system will automatically reboot.

Once the process completes, you will be running in a Clustered
Environment and you will be able to call /etc/reconfig to add
additional cluster nodes as needed.

Do you wish to continue (y or n)? >>
```

Type "y" to continue the process. `reconfig` will execute several scripts that will perform the tasks identified in the message shown above. The scripts are in the files called `CDFcustomize` in the `/system/*` directories. The customization takes about 5 minutes.

Look in the `/tmp/reconfig.log` file to see if any errors have occurred. If there are errors you should determine what happened, fix the problem, and re-execute the `reconfig` program.

You are finished configuring your root server to support an HP-UX cluster. Read the remaining topics in this section: they are very important. When you are finished you can add diskless cnodes to your cluster using the procedures given in the section "Adding a Diskless Cluster Node".

What the reconfig command has done to your system

You now have a clustered environment on your root server. The `reconfig` command has done the following:

- created the `/etc/clusterconf` file

  There will be two lines in the `/etc/clusterconf` file. The first line contains your root server's link level address. The second line has some root server node information.

- turned several files into CDFs

  For a list of the files that are now CDFs, along with their subfile names and permissions, refer to the "Concepts" chapter.

- created a fully loaded cluster kernel for the root server

  The kernel is built using `/etc/conf/dfile.maxservr` which is copied to `/etc/conf/dfile+/rootserver-nodename`. During customization, `/hp-ux` is turned into a CDF. Your root server's version of the kernel resides in the file `/hp-ux+/`*rootserver_cnodename*. For example, if your root server's cnode name is `daisy`, the kernel would be in `/hp-ux+/daisy`.

- put an entry for the root server in each of the following files (unless the entry was already there):

`/etc/hosts`	*(root's home directory)*
`/etc/hosts.equiv`	
`$HOME/.rhosts`	
`/etc/XO.hosts`	*(if it exists)*

- *On a system with a previously existing LAN card,* where your cluster LAN card is not at the same select code as an existing `/dev/lan`: `reconfig` created a new device file called `/dev/ieee.cluster`, and edited the `/etc/rc` file to change the line `/etc/rbootd` to `/etc/rbootd /dev/ieee.cluster`.

- moved `/etc/netlinkrc` to `/etc/netlinkrc.stdl`

- moved `/etc/newconfig/netlinkrc.dsk` to `/etc/netlinkrc`

  Any customization done to `netlinkrc` must be redone to the new `netlinkrc`.

If your root server has two LAN cards:

If you have only one LAN card on your root server, you can now set up your diskless cnodes.

If you have problems with the concepts described in this section then refer to the *Installing and Maintaining NS-ARPA Services* manual. The steps described here are specific to the situation of adding a cluster LAN card to a system on an existing LAN, and should be followed, but the section does not describe the concepts.

If your system was on an existing LAN (which is different from your new cluster LAN) then use your system as a gateway to the old LAN. This means that you set up you root server hostname to be the same as your old ARPA and HP-UX hostname. You now must add entries to configure the new cluster LAN card's ARPA hostname.

In the following set of steps, assume the original ARPA hostname (and HP-UX hostname) is `arpahost`, and the new (cluster's) ARPA hostname will be `servrlan`. The `arpahost` internet address is xx.xx.xx.1 and the `servrlan` internet address is xx.xx.xx.2.

Before adding diskless cnodes to your cluster you must complete the following:

- You must add a new entry (for the cluster LAN card) into the following files: `/etc/hosts`, `/etc/hosts.equiv`, and `$HOME/.rhosts`. Using the example above, the files would look like:
  - □ `/etc/hosts`

    ```
    xx.xx.xx.1 arpahost
    xx.xx.xx.2 servrlan
    ```

  - □ `/etc/hosts.equiv`

    ```
    arpahost
    servrlan
    ```

  - □ `/.rhosts`

    ```
    arpahost root
    servrlan root
    ```

- You also must add a route statement for each cnode in the cluster. Route statements are put into the `/etc/netlinkrc` file. Find the commented case statement that looks like the following:

```
#   case $NODENAME in
#         *) # add rootserver routes here
#             ;;
#   esac
```

  To add a route statement for all other diskless cnodes on the cluster LAN, replace the commented line with a valid route command, using the ARPA hostname associated with the root server's cluster LAN. Using the above example, your final case statement would look like:

```
   case $NODENAME in
$ROOTSERVER) # do nothing with the gateway unless you need to
      # add route statements for other systems
          *) /etc/route add default servrlan 1
             ;;
   esac
```

- Also in the `/etc/netlinkrc` file, you must add `ifconfig` statements. For each LAN card in the root server, up to and including the cluster's LAN card, you must have an `ifconfig` statement. Find the following case statement:

```
case $NODENAME in
   *) /etc/ifconfig lan0 'hostname' up
      ;;
 esac
```

  Add additional case statements so the section looks like:

```
Case $NODENAME in
   $ROOTSERVER) /etc/ifconfig lan0 'hostname' up
                /etc/ifconfig lan1 servrlan up
                ;;
            *) /etc/ifconfig lan0 'hostname' up
               ;;
esac
```

The way this works: Your cluster LAN card is associated with a name that is different than your root server's name. The `ifconfig` statements executed for the `$ROOTSERVER` will power up both LAN cards and the `/etc/route` statements executed for all non-`$ROOTSERVER` entries will make requests to the `arpahost`

LAN through the `servrlan` LAN card. The advantage in doing this is that you (the root server) will maintain one name within your cluster as well as outside your cluster. The same solution will work if the gateway is on a diskless cnode.

After Setting Up the Cluster Environment

You now have a working root server. After you have set up your cluster environment, you can add diskless cnodes. If you know that you will have a diskless cnode as a gateway, do the "After Adding a Cnode" before adding any cnodes (including the one that will be the gateway). Then reboot the root server and add cnodes.

If you decide, later, to add a gateway cnode, this step will be done after adding the cnode. However, if you know now that one of your cnodes will be a gateway, doing this step first will prevent you from rebooting multiple cnodes later.

Adding a Diskless Cluster Node

One of the benefits of an HP-UX cluster is the ability to add an additional workstation without adding an additional disk drive and re-installing HP-UX and/or applications. The new workstation is called a diskless cluster node, or cnode. Adding and removing a diskless cnode is done with the `reconfig` program.

Before Adding a Cnode

Before adding a cnode make sure of the following:

- your cnode has Rev B or later boot ROM
- you have a root server set up (refer to the section "Creating a Clustered Environment")
- the LAN card (that will connect to your cluster) on the cnode is one of the two lowest select codes of all your LAN cards (generally Select Code 21 or 22)
- your cnode has at least 3 Mbytes of RAM installed

Procedure

If you have a cnode identified to be a LAN gateway, add that cnode first. When you add a gateway cnode, you must reboot the cluster. If you add the gateway first, there will be fewer systems to reboot.

To add a cnode to your existing cluster follow these steps:

Step 1. Powerup your diskless cnode and get into the boot ROM's attended mode.

To add the diskless cnode to the cluster, you must supply the link level address of the cnode's cluster LAN card. To get the address, powerup the diskless cnode get into the boot ROM's attended mode. (To get into the attended mode, press the space bar immediately after turning on power, and continue to hold it down until you see the word `keyboard` listed on the left side of the console display. This will stop the boot ROM sequence to give you time to copy the address.) You will see a screen similar to the one shown in Figure 4–12.

If you are already running an HP-UX system on the computer, type:

```
shutdown -h 0
```

then wait for the "halted" message and cycle power (and boot in the attended mode).

```
Copyright 1987,
Hewlett-Packard Company.
All Rights Reserved.

BOOTROM Rev. C
Bit Mapped Display
MC68050 Processor
Keyboard
HP-IB
HP98620B
HP98644 at 9
HP98625 at 14
HP98643 at 21, 080009000001
4182016 Bytes






SEARCHING FOR A SYSTEM (RETURN To Pause)
RESET To Power-Up
```

Figure 4–12. Example Boot ROM Display

Each LAN card will show up in the format:

lancard_partnum at *select_code*, *link-level-address*

(for example, HP98643 at 21, 080009000001). Write down the cluster LAN card's link level address on the information sheet you started for your cluster.

Leave the cnode in this state. You will come back later to finish the boot sequence.

Step 2. Start the reconfig program and go to the "Add a cluster node" menu.

On your ROOT SERVER:

Log in as the superuser, root, *on your root server*. Start the `reconfig` program by typing:

```
/etc/reconfig
```

You will see the menu shown in Figure 4–13.

```
*******************************************
           RECONFIG -- MAIN MENU
*******************************************

User Configuration
Workstation Configuration
Cluster Configuration
```

		NEXT	PREVIOUS	HELP	SELECT		QUIT

Figure 4–13. Reconfig Main Menu

Using the NEXT or PREVIOUS keys, highlight the menu item "`Cluster Configuration`". Press the SELECT softkey. You will now see the menu shown in Figure 4–14.

```
*******************************************
     RECONFIG -- CLUSTER CONFIGURATION
*******************************************

Add a diskless cluster node
Remove a diskless cluster node
```

		NEXT	PREVIOUS	HELP	SELECT	PREVMENU	

Figure 4–14. Adding/Removing Cnode Main Menu

Using the NEXT or PREVIOUS keys, highlight the menu item "`Add a diskless cluster node`". Press the SELECT softkey. You will now see the menu shown in Figure 4–15.

```
******************************************
  RECONFIG -- ADD A DISKLESS CLUSTER NODE
*******************************************

Cluster Node Name:
LAN Card's Link Level Address:
NS-ARPA Internet Address:





Cluster Node Name? >>









[RESTORE] [ ] [ ] [ ]   [HELP] [ ] [PREVMENU] [ ]
```

Figure 4–15. Form for Adding a Cluster Node

Step 3. Add the new cnode information to your root server.

1. Type the cnode's name.

   The name must be between 1 and 8 characters (any ASCII characters). This name must be unique to the network.

   If you have an existing network, and you will be using this cnode as a gateway into the existing network, this field must be the same as the existing ARPA hostname and HP-UX system hostname. This is explained in more detail in the section "If you have two LANs on your root server".

2. Type your cnode's LAN card link level address.

   You wrote this number down in Step 1. If not, go back to Step 1 to obtain the cnode's LAN card link level address. Do not type the select code, type just the link level address.

3. Enter the cnode's NS-ARPA internet address.

   You wrote the internet address down on your checklist. It has the same network address portion as the root server's internet address, but a different local address part. The local address part makes this internet address unique within the network.

   If an address is displayed here, it is the address associated with the ARPA hostname entered in step 1 above. You cannot change the address in this case. If this address is associated with the original LAN card, then you must follow the instructions in the "After Adding a Cnode" section.

After you verify that all the cnode information is correct, the `reconfig` program will customize your cluster to add the new cnode. You will know it is finished when you see the message: `New cluster node <name> added to the cluster`. The `reconfig` program does the following:

- adds the new node's entry into the `/etc/clusterconf` file
- modifies several system CDFs to add the new cnode. The system CDFs are listed in the section "System CDFs" in the "Concepts" chapter.
- adds entries into the `/etc/hosts`, `/etc/hosts.equiv`, `/etc/X0.hosts` (if it exists) and `$HOME/.rhosts` files, if the entries are not already there.
- adds an element to `/hp-ux+`

  The kernel matches `/etc/conf/dfile+/cnode-name` which is a copy of `/etc/conf/dfile.cnode`.

Step 4. Exit the reconfig program or add additional cnodes.

Exit the reconfig program by pressing the PREVMENU softkey twice, then pressing the QUIT softkey.

If this diskless cnode will also be the LAN gateway, complete the section "If your diskless cnode will also be a LAN gateway" before continuing with Step 5.

Step 5. Go back to your cnode.

If you left the cnode in attended boot ROM mode in Step 1, you should see an entry on the right side of your screen for the root server. It will look similar to:

```
LAN, 21, daisy
 1H SYSHPUX
 1D SYSDEBUG
 1B SYSBCKUP
```

Choose the correct operating system (for example `1H`). When the cnode boots, you will receive a login prompt. Log in, remembering that you are using the same logins and passwords as on the root server.

If you did not leave the boot ROM in attended mode in Step 1 and your cnode is attached to a disk drive containing a bootable system, then you are probably booted to the system on the disk drive. Log in. If it is an HP-UX system, type:

```
getcontext
```

If you get a message saying "not found", or if the returned string contains the word "standalone" or "localroot", then you are booted to the wrong system. If the returned string contains the word "remoteroot", then you are booted to the correct system.

If you are booted to the wrong HP-UX system, then type:

```
shutdown -h 0
```

when you receive the "halted" message, cycle power on your cnode and boot in attended mode.

If you do not see the new operating system entry, cycle power to restart the boot ROM to see if it now sees the root server. If not, begin with Step 1 and verify all the information.

After Adding a Cnode

You can now add terminals and other peripherals for the new cnode if you wish. Because the `/dev` directory is a CDF, any peripherals you attach to the diskless cnode can be used only locally, by the cnode, and must be added while logged into the cnode to which the peripheral will be connected.

If you wish to use a swap device local to the new diskless cnode, you must follow the instructions in Chapter 6 for reconfiguring the kernel for swap space. In most clusters the diskless cnodes will swap to the root server.

If your diskless cnode will also be a LAN gateway:

If your diskless cnode has multiple LAN cards, and will be the gateway between the HP-UX cluster and other LANs, then you must complete this section.

If you have problems with the concepts described in this section then refer to the *Installing and Maintaining NS-ARPA Services* manual. The steps describe here are specific to the situation of adding a cluster LAN card to a system on an existing LAN, and should be followed, but the section does not describe the concepts.

If your cnode was previously connected to a LAN, and your system will now act as the gateway between your HP-UX cluster and systems on other LANs, the ARPA hostname associated with the cluster's LAN card will be different than the ARPA hostname by which non-cluster computers know you.

As an example, assume your original ARPA hostname is `arpahost`, and the cluster LAN's ARPA hostname will be `cnodelan`. The internet address associated with `arpahost` is xx.xx.xx.3 and the internet address associated with `cnodelan` is xx.xx.xx.4.

Before booting this diskless cnode, you must do the following:

1. You must add a new entry (for the cluster's LAN card) into the following files: `/etc/hosts`, `/etc/hosts.equiv`, and `.rhosts`. Using the example above, the files would look like:

   - `etc/hosts`

     ```
     xx.xx.xx.3 arpahost
     xx.xx.xx.4 cnodelan
     ```

   - `/etc/hosts.equiv`

     ```
     arpahost
     cnodelan
     ```

   - `/.rhosts`

     ```
     arpahost root
     cnodelan root
     ```

2. You must add a route statement on each cnode in the cluster. Route statements are put into the `/etc/netlinkrc` file. Find the commented case statement that looks like the following:

```
#   case $NODENAME in
#               *) # add route commands for specific nodes here
#                  ;;
#   esac
```

   To add a route statement for all other cnodes on the cluster LAN, assuming the example above, you would add lines so your final case statement would look like:

```
case $NODENAME in
     arpahost) # do nothing with the gateway
               ;;
            *) /etc/route add default cnodelan 1
               ;;
esac
```

3. Also in the `/etc/netlinkrc` file, you must add `ifconfig` statements. For each LAN card in your system, up to and including the cluster LAN card, you must have an `ifconfig` statement. Find the following case statement:

```
case $NODENAME in
   *) /etc/ifconfig lan0 'hostname' up
     ;;
esac
```

   Change the case statement so in now looks like:

```
case $NODENAME in
arpahost) /etc/ifconfig lan0 'hostname' up
          /etc/ifconfig lan1 cnodelan up
          ;;
       *) /etc/ifconfig lan0 'hostname' up
         ;;
esac
```

4. Reboot the cluster.

5. Boot the cnode.

The way this works: Your cluster LAN card is associated with a different name than your diskless cnode hostname. The `ifconfig` statements executed for the `arpahost` will power up both cards. The `/etc/route` statements done for all non-`arpahost` entries will make requests to the `arpahost` LAN go through the `cnodelan` LAN card. The advantage in doing this is that you (the diskless cnode) will maintain one name within your cluster as well as outside your cluster.

Removing and Renaming a Cluster Node

This section explains how to remove and/or rename a diskless cnode. You cannot remove the root server, and it is difficult to rename the root server; neither will be discussed.

Removing a Diskless Cluster Node

Step 1. Start the reconfig program and go to the "Add a cluster node" menu.

You must be logged in as the superuser, root, on the root server. Move to the root directory and start the `reconfig` program by typing:

```
cd /
/etc/reconfig
```

You will see the menu shown in Figure 4–16.

```
******************************************
        RECONFIG -- MAIN MENU
******************************************

User Configuration
Workstation Configuration
Cluster Configuration
```

		NEXT	PREVIOUS	HELP	SELECT		QUIT

Figure 4–16. Reconfig Main Menu

Using the NEXT or PREVIOUS keys, highlight the menu item "`Cluster Configuration`". Press the SELECT softkey. You will now see the menu shown in Figure 4–17.

```
*******************************************
     RECONFIG -- CLUSTER CONFIGURATION
*******************************************

Add a diskless cluster node
Remove a diskless cluster node
```

		NEXT	PREVIOUS	HELP	SELECT	PREVMENU	

Figure 4–17. Adding/Removing Cnode Main Menu

Using the NEXT or PREVIOUS keys, highlight the menu item "`Remove a cluster node`". Press the SELECT softkey. You will now see the form shown in Figure 4–18.

```
**********************************************
  RECONFIG -- REMOVE A DISKLESS CLUSTER NODE
**********************************************

Cluster Node Name:                  cnode1
Remove Cluster Node Specific CDF's: no




The cluster nodes currently defined within your cluster are:

cnode1    cnode 2       cnode3


Cluster Node Name? >> cnode1






RESTORE  |        |  NEXT  |PREVIOUS|    HELP  | SELECT |PREVMENU|
```

Figure 4-18. Form for Deleting a Cluster Node

Step 2. Select the node for deletion

Using the NEXT or PREVIOUS keys highlight the name of the cluster node you wish to remove from the cluster (you can type the cluster node name if you prefer). Press the SELECT softkey.

Answer `yes` or `no` to the question `Remove Cluster Node Specific CDF's?`.

If you decide to remove cnode-specific CDFs, the `reconfig` program will check the file system for all CDFs that contain a file with your cnode's name, and will remove those files. If you answer no, these cnode-specific files will stay on your file system. You should answer this "yes" unless you have some specific reason to leave the cnode-specific files on your system. This process will take from one to ten minutes depending on the size of your file system. You will see the following warning:

```
WARNING:  All CDF entries of "name" will be removed.
```

The `reconfig` program asks if you wish to remove additional cnodes. If you wish to remove additional cnodes go to step 2.

If you are finished removing cnodes, press the PREVMENU softkey to return to the previous menu. You may then do additional tasks or exit the `reconfig` program.

Renaming a Diskless Cluster Node

The easiest way to rename a diskless cnode is to remove it and then add it again with a different name. The steps are described below:

1. Using the `reconfig` program, remove the diskless cnode (and all its CDFs). This procedure is described in the previous section "Removing a Diskless Cluster Node".
2. Edit the `/etc/hosts` file. Remove the cnode's entry in `/etc/hosts`.
3. Using the `reconfig` program, add the diskless cnode. This procedure is described in the previous section "Adding a Diskless Cluster Node."

Creating a New File System

If you run out of space on your root file system, you can either remove enough files to gain space or you can add an additional disk and create another file system. You then *mount* this new file system to your existing file system hierarchy and use the disk space to store your files. The `newfs` command is used to create the file system on your disk. Before creating a new file system, you may wish to consider some of the default `newfs` values to decide if they are correct for you.

Many of the parameters for `newfs` are determined from the `/etc/disktab` file. This file contains information on Hewlett-Packard disk drives. It contains many comments that explain how parameters are determined, what the default swap space is for a disk, descriptions of disks, and example mknod commands.

As of the 6.2 release of HP-UX you have the option of creating a file system allowing the standard 14-character file name limit or allowing up to a 255-character file name. There is a new option to the file system creation commands so you can create file systems with the long file names. Refer to the section "Enabling Long File Names" for information on creating and using long file names.

Prerequisites

Before creating a file system on your disk drive or flexible disk you must have completed the following:

- Connected the mass storage device on which the file system will exist to your HP-UX system. Refer to the "Adding Peripheral Devices" section in Chapter 4, and the installation manuals supplied with your Series 300 computer and the mass storage device.
- Created *both* a character and a block device file for the mass storage medium. Use the instructions in the "Adding Peripheral Devices" section of Chapter 4.
- Initialized the media with the media initialization utility `mediainit`. This utility initializes the medium on which the file system will reside. Refer to "Initializing Media" in Chapter 5 or the *mediainit(1)* entry in the *HP-UX Reference*.

Creating the File System

Once you have an initialized disk (refer to the "Prerequisites" section above) you are ready to create the file system. Follow these steps:

1. Turn on the disk drive (if you have not already done so).
2. Log in as the root user (if you have not already done so).
3. Create the file system by using the `newfs` command.

   A file system can be created on the Series 300 HP-UX system using either `newfs` or `mkfs`. `newfs` is a friendly front end to `mkfs`, and is the recommended command to use. The `newfs` syntax is:

   `newfs [-L | -S] [-n] [-v]` [*mkfs-options*] *device_file disk_type*

`-L`	Creates a file system with long file names (up to 255 characters). The default (which can also be specified with the `-S` option) is the standard 14-character file name limit.
`-n`	Prevents the bootstrap programs from being installed.
`-v`	Prints a verbose listing of the `newfs` actions.
mkfs-options	Options to `mkfs` that will override default parameters. The default for many of these options are defined for your disk drive in the `/etc/disktab` file. For details on these refer to the *newfs* or *mkfs* entries in section *1M* of the *HP-UX Reference* or to Table 4–10 later in this section.
device_file	The character device file name associated with the disk drive where you are creating the file system.
disk_type	The type of the disk as specified in the `/etc/disktab` file. `newfs` uses defaults from both `mkfs` and from the `/etc/disktab` file. The `mkfs` defaults are listed in Table 4–10 (the N/A means Not Applicable). The `disktab` file contains disk-specific information for `newfs`. Refer to *disktab(4)* and *newfs(1M)* in the *HP-UX Reference* for more information. If your disk is not in the `/etc/disktab` file, read the comments in `/etc/disktab` for information on creating a new entry.

If you need to change the default swap space size, the minimum amount of free space on your file system, or any other parameter to `newfs/mkfs`, you should determine the values now. Refer to the section "Configuring Swap Space" in Chapter 6 for a description of swap space.

For example, if you have an HP 7946 disk drive associated with the device file `/dev/rdsk/1s0`, and you wish to create a file system using only default values, type in:

```
newfs /dev/rdsk/1s0 hp7946
```

An example of creating a file system specifying the swap space is shown in the section "Configuring Swap Space and the Root Device" in this chapter.

Note The `mkfs` and `newfs` programs will list the alternate superblock locations in a file called `/etc/sbtab`. Get a hardcopy of this file in case your file system becomes corrupted and you need the information; you can use an alternate superblock with `fsck` to repair your file system.

After Creating the File System

Before you can use the file system you have put on your disk you must mount it into your file system hierarchy. The basic steps are:

1. Create a directory on which to mount the new file system.

   `mkdir` */mount_directory*

2. Mount the new file system onto */mount_directory*.

   `mount` */dev/fname /mount_directory*

   where */dev/fname* is the name of the block file associated with the mass storage medium.

   For additional information on mounting refer to the section "Mounting and Unmounting File Systems" in this chapter.

If the newly created file system is intended as a permanent addition, add it as an entry to the /etc/checklist file. This will cause the new file system to be checked and mounted when the system is booted.

Add the new file system to /etc/checklist by inserting the name of the character device file name associated with the file system. Refer to the section "Adding to /etc/checklist" in this chapter for more information.

In addition, if the new file system will be used as an additional swapping device, refer to the section "Configuring Swap Space" in Chapter 6.

Table 4-10. mkfs Defaults

Parameter	/etc/disktab Names	Range	Default	Comments
size	s0	N/A	none	In 1024 blocks. If using `newfs` size will (by default) be taken from `/etc/disktab`. Size is the total disk size minus the swap size in 1024 blocks.
block-size	b0	4K or 8K	MAXBSIZE (8K)	Specified in bytes.
frag-size	f0	1024 to block-size	1024 bytes	Specified in bytes. Must be an even multiple of 1024.
number of tracks per cylinder	nt	greater than 0	16	Taken from `/etc/disktab` if using `newfs`.
number of cylinders per disk	nc	1 to 32	16	
% free space reserved	N/A	0 to 100	10 %	Once this threshold has been crossed only the superuser can continue to write.
revolutions per minute	rpm	N/A	3600	If using `newfs` the parameter is revolutions/minute and the value can be taken from `/etc/disktab`.
number of bytes per inode	N/A	1 to (function of file system size and other parameters)	max (2048, fragment size)	Number of inodes allocated is a function of block size. Maximum is 2048 inodes per cylinder group.

For most installations, the default file system setup is correct. If you have an installation that requires a different file system configuration, use the following guidelines:

- If your system will have many small files, you can decrease the average number of bytes per inode. This will give you more inodes, and enable you to create more (but smaller) files. A larger number of inodes will take more space on your file system.
- If your system will have only a few large files, you can increase the space available for data by increasing the average number of bytes per inode.
- Decreasing the `minfree` parameter enables you to write to an additional percentage of file system space. The lower the percentage, the greater the possibility that your file's blocks will be scattered on the disk. Also, performance decreases as the disk fills up.
- Decreasing the file system size will give you more swap size, which enables you to run larger programs, but decreases the area where you can store files. Some optional application programs for HP-UX require above-average amounts of swap space. Refer to your application's manuals to see if large swap areas are required.
- Increasing the file system size will give you less swap size. If you have a swapping device separate from your file system, this will give you a larger file system, yet allow you to execute large programs.

Enabling Long File Names

To adhere to AT&T System V UNIX, all HP-UX releases prior to 6.2 impose a maximum file name length of 14 characters. The system silently truncates all file names passed in through a system call to 14 characters.

BSD 4.2 UNIX introduced what is commonly called long file names. It allows file names to be up to 255 characters long and uses variable-sized directory entry data structures to allow for efficient disk and memory usage. Since many HP-UX systems are networked to other UNIX systems that support long file names, HP implemented this feature in the 6.2 release.

HP-UX file systems can be configured to support either long or short file names. Short file names are the standard AT&T UNIX 14-character file name limit. Long file names are names up to 255 characters. HP-UX on the Series 300 machines is installed as a short file name system. You can optionally enable long file names on a per-file-system basis. When configured with the short file names, HP-UX file systems are the same as earlier system releases that are not configured to accept long file names.

Main Differences Between Long and Short File Names

In an HP-UX file system using short file names the maximum length of a file name or directory name is 14 characters. The system will accept file names longer than 14 characters but will truncate the file name after the fourteenth character. The system will return an error when a file name longer than 255 characters is passed to a system call.

Also, in an HP-UX file system with short file names, directory entries are always aligned on 32-byte boundaries because an entry always contains 32 bytes of information. In a file system that accepts long file names the directory entries may vary in size and are guaranteed only to be aligned on a 4-byte boundary.

A file system with long file names has a different magic number than a file system with short file names. The magic number is in the superblock of the file system. Refer to "File System Implementation" in Chapter 2 (or in `/usr/include/sys/fs.h`) for a description of the file system superblock.

Long File Names or Short File Names?

If you wish to have the flexibility of 255-character file names, or if you need the long file names for compatibility with other systems, you should change your file system to support long file names.

Do not change to long file names if:

- You need 14-character file names to be compatible with other systems.
- You plan to use applications that read directory file information and do not use portable directory routines (like those described in the *directory(3c)* entry of the *HP-UX Reference*). If these applications assume that directories are an array of fixed-size entries, they will not work with long file names. You must rewrite the application to correct the assumptions about directories using the `directory` routines. You must include `ndir` instead of `dir.h` to use `directory` routines. This way, the applications can properly parse the directory file information required by long file names. There is no way, other than reading the source code, to determine correct usage.

  For more information, refer to the *directory(3c)* pages in the *HP-UX Reference*.
- You plan to use programs that open or read directories directly (with no source code available) that were developed or compiled on releases of HP-UX that do not support long file names.

  To determine this, you need to contact the original supplier of the software to see if it opens or reads directories directly. If this is not possible, the software should be tried on a scratch long file name system.
- Other systems in your organization run versions of UNIX or HP-UX that impose a 14-character limit on file name length. In this environment, you may want uniformity across the systems so that files may be moved to different systems.

All HP-UX commands and utilities work properly with either configuration.

Enabling Long File Names on an Existing File System

Before Converting to Long File Names

Once you have converted your system to long file names, it is not easy to change back to short file names. Because of this, you should find out (before converting) if your applications will perform properly with long file names. To be sure, you can convert a temporary or scratch file system and then run your applications on this test file system.

Converting to Long File Names

If you have an HP-UX file system with short file names, you can use the `/etc/convertfs` utility to convert the file system to allow long file names. Follow these steps:

1. Back up your system before you use the `convertfs` utility.

   The procedures for backing up your file system are in the section "Backing Up Your File System" in Chapter 5.

2. Shut down your system to the system administration state:

   ```
   cd /
   /etc/shutdown
   ```

   Refer to "Shutting Down the System" in Chapter 3 for more information.

3. Execute the `convertfs` utility. It has the following syntax:

   `/etc/convertfs` *[file_system_names]*

   There are two ways to use this command: let it prompt you for the file system names listed in `/etc/checklist` or supply it with a list of file system names.

   a. To let `convertfs` prompt you for the file system names listed in `/etc/checklist`, enter:

      ```
      /etc/convertfs
      ```

You will receive these messages:

```
Warning:  Conversion to long file names is irreversible and certain
programs may not work with long file names.

Converting the file system will cause a system reboot.  The system
should be shutdown into single user state and all non-root file
systems should be unmounted before this utility is run.

Do you wish to continue? [y/n]
```

If you used the shutdown command to get your system to the system administration state (single-user), then answer y.

The program will then ask you if you want to convert all of the normally mounted file systems listed in the `/etc/checklist` file. If you answer no to this prompt, for each file system listed in `/etc/checklist/convertfs`, it will ask if you wish to convert the file system. Respond to the prompt for each file system. HP recommands changing all or none of the file systems.

b. If you wish to specify the names of the file systems to convert, follow the `convertfs` command with the name of the file system you want to convert. For example, to convert `/dev/rdsk/c2d0s10`, enter:

```
/etc/convertfs /dev/rdsk/c2d0s10
```

The `convertfs` utility will convert the named file system without prompting for input.

Note Although the `convertfs` utility allows just one (or a few) file systems to be selected for conversion to long file names, HP recommends that you convert all or none of your normally mounted file systems in `/etc/checklist`. This is to prevent inconsistencies and undesired events that can occur if you mix long and short file names on the same system.

The `convertfs` utility will modify the superblocks and reformat the directories in the file systems you want to convert. After modifying each file system, `convertfs` will execute an `fsck` so that the file system can again be mounted.

If you have converted the root file system, `convertfs` will reboot the system so that the changes made to the file system superblock will not be overwritten by an update of the superblock in the system memory.

After you reboot the system or remount the converted file systems you will be able to use long file names on the converted file systems.

After Converting to Long File Names

Once you have converted your file system to enable long file names, there are a few things to be aware of:

- After the filename conversion is complete, you must recompile all pre-6.2 programs that use the routines listed in *directory(3c)* in the *HP-UX Reference*. This will ensure that these routines will correctly parse directories for a system that supports long file names.
- Once you have converted to long file names, you cannot easily convert back to short file names. If you must convert back to short file names, follow these instructions:
  - Make sure all file names are 14 characters or shorter. Use the `mv` command to change long file names to short file names.
  - Back up the entire file system using the procedure described in "Backing Up Your File System".
  - Recreate the file system with short names using the `-S` option to `newfs` or `mkfs`. If the root file system needs to be converted back to short file names, reinstall it from the installation tape.
  - Recover your files from the backup media.
- The `newfs` program will create new files of the same type as the root file system. If you converted the root file system, then all new file systems you create will (by default) allow long file names. There are two new options to `newfs` and `mkfs`: `-S` for short file names and `-L` for long file names.

Long File Names and Your System

Using a Recovery System with Mixed File Name Lengths

Your recovery system will work with mixed file name lengths. However, as always you must have a current version of recovery system so it understands both types of magic numbers, and so it has the right version of file system commands.

Mounting File Systems with Long File Names

You can mount file systems with long file names onto file systems with short file names, and vice versa. However, mixing file systems may cause your users some confusion. For example, if you copy from a file system with long file names to one with short file names, the file name may be truncated:

```
cp /lfn/supercalefragelistic /sfn/.
```

would yield the file `/sfn/supercalefrage`

This new file has a different name, but also, if a file by that name already existed, this new file would replace the old one.

Writing Programs in a System with Long File Names

Make sure shell scripts and programs do not assume a 14-character limit. Note that scripts and programs making this assumption would work on a system with short file names but could stop working when moved to a system with long file names.

User programs may need to determine if a file system uses long file names. To do so:

1. Create two unique temporary file names. One has 14 characters (file one) and the other has 15 characters (file 2). The first 14 characters of these two files must be the same.
2. Stat each file.
3. Compare both `st_dev` and `st_ino` of the stat structure for the two files. If both fields of both files are the same, the file system in which file 1 and file 2 reside uses short file names. If they are not the same, it is a long file name file system.

Do not assume other file systems (i.e., over NFS) have a maximum length of either 14 or 255. Although most NFS servers support one of these two maximum file name length, there are a few vendors that use a different size.

For example, the following pieces of C code will check file name length:

```
/* WARNING: Although most UNIX (Trade Mark of AT&T) systems support file
     systems whose maximum file name length is either 14 or 255
     characters, there are some venders who support file systems with
     various filename lengths.  If an application is to run in a NFS
     environment, is_truncated() should be used to determine if a
     particular length is allowed in a certain file system.
     is_lfn() and is_truncated() are written without error checking,
     users might want to check return code of system calls.
*/
#include <sys/types.h>
#include <sys/stat.h>
is_lfn()
{
 struct stat stat1, stat2;
 char file1[MAXNAMLEN+1];
 char file2[MAXNAMLEN+1];
 int i;

 sprintf(file1, "abcdefgh.%-.5d", getpid());
 sprintf(file2, "abcdefgh.%-.5da", getpid());
 creat(file1, 600);
 creat(file2, 600);
 stat(file1, &stat1);
 stat(file2, &stat2);
 if ((stat1.st_dev == stat2.st_dev) && (stat1.st_ino == stat2.st_ino))
  i = 0;
 else  i = 1;
 unlink (file1);
 unlink (file2);
 return(i);
}

/* This routine tells caller if the file can be created with that name (without
   being truncated). */

is_truncated(filename)
register char *filename;
{
 struct stat stat1, stat2;
```

```
 char other[MAXNAMLEN];
 int len, i;

 len = strlen(filename) - 1;
 strncpy(other, filename, len);

 creat(filename, 600);
 creat(other, 600);
 stat(filename, &stat1);
 stat(other, &stat2);
 if ((stat1.st_dev == stat2.st_dev) && (stat1.st_ino == stat2.st_ino))
  i = 1;
 else i = 0;
 unlink (filename);
 unlink (other);
 return (i);
}
```

The following show situations where your existing program may not work when your system is converted to long file names:

- Software that opens directories, reads the directory entries directly, and expects the size of directory entries to be a constant.

  This software should be changed to use directory library routines or use `getdirentries` system calls. You should include `ndir.h` instead of `sys/dir.h` when you use the directory library.

  Here are two examples: the first one works only with short file name systems, the second works with both long and short file name systems:

```
/* This routine accept 2 arguments. The first is a directory path and
   the second is a filename which is to be searched in the directory.
   This routine will work in a short filename file system but NOT in a
   NFS environment nor in a long filenames file system. */
#include <sys/dir.h>
find_name(dnamep, fnampe)
register char *fnamep, *dnamep;
{
   register int i, len;
   register int fd;
   struct direct ds;

   fd = open(dnamep, O_RDONLY));
   len = strlen(fnamep);

   while (read(fd, &ds, sizeof(struct direct)) == sizeof(struct direct))
      if (ds.d_namlen == len && !strcmp(ds.d_name, fnamep)) {
    printf("%s is found\n", fnamep);
    close(fd);
    return(0);
      }

   printf("%s is not found\n", fnamep);
   close(fd);
   return(1);
}
```

```
=====================================================================

/* This routine accept 2 arguments. The first is a directory path and
   the second is a filename which is to be search in the directory.
   This routine will work in a NFS environment, long filenames and
   short filenames file systems.*/
#include <ndir.h>
find_name(dnamep, fnamep)
register char *fnamep, *dnamep;
{
 register int i, len;
 DIR *dirp;
 struct direct *dp;

 dirp = opendir(dnamep));
 len = strlen(fnamep);

 while (dp=readdir(dirp)){
    if (dp->d_namlen == len && !strcmp(dp->d_name, fnamep)) {
  printf("%s is found\n", fnamep);
  closedir(dirp);
  return(0);
    }
 }
 printf("%s is not found\n", fnamep);
 closedir(dirp);
 return(1);
}
```

- Software that assumes that the maximum length of file names is 14 characters. For example

```
char filename[14]
```

If you need only a few buffers, use MAXNAMLEN for buffer size. Else you will need to allocate memory dynamically. If you need to allocate memory dynamically you can obtain the file name size from d_namlen in the struct direct. For example:

```
/* This routine stores all filenames in a directory in memory.*/
#include <ndir.h>

store_filenames(dnamep)
char *dnamep;
{
 DIR *dirp;
 struct direct *dp;
 char *cp;

 dirp = opendir(dnamep);
 while (dp = readdir(dirp)) {
  cp = malloc(dp->d_namlen + 1);
  strcpy(cp, dp->d_namlen);
  .
  .
  .
 }
 closedir(dirp);
}
```

- Software that uses `dirsiz` and assumes it is a constant of 14, and that uses it to mean the maximum length of the file name. For example:

  ```
  char filename[DIRSIZ];
  ```

  Use MAXNAMLEN as the maximum length of file names. Use DIRSIZ_CONSTANT as the maximum file length on a short file name system.

  If the DIRSIZ_MACRO compilation flag is turned on, DIRSIZ is a macro instead of a constant of 14. It accepts an argument which is the pointer to a `struct direct`. It returns the actual size of the directory entry in a long file name system. DIRSIZ_MACRO is mainly for porting programs which were originally developed under BSD 4.2 UNIX.

- Software that uses MAXNAMLEN and assumes that MAXNAMLEN equals 14.

  MAXNAMLEN is now 255. Most of the software need only be recompiled, although the memory allocation could be fairly large.

- Software that includes `dir.h` and uses `struct direct`.

  The `struct direct` for short file names is a fixed-size structure while the `struct direct` is variable length for long file names. HP recommends that software should be changed to include `ndir.h` and to use directory library routines.

- Software that assumes there is only one file system magic number.

  The magic number for long file names is different than that for short file names. Software should be changed to allow the new magic number.

In general, all programs and routines that include `sys/dir.h` or use FS_MAGIC need to be reviewed and possibly changed to work with long file names.

Creating System Run-Levels

You may find it useful to create new system run-levels for particular tasks or applications specific to your installation. The material in this section covers some protection issues associated with these capabilities followed by guidelines for creating new system run-levels. The procedure for changing run-levels is in the chapter called "Periodic System Administrator Tasks".

As discussed in the "System Startup and Shutdown" chapter in this manual, the system administrator (or anyone with the `root` user capabilities) may change the system's run-level by executing the `init` command. Also, anyone having write permission to the file `/etc/inittab` can create new run-levels or redefine existing run-levels. Make sure that the permissions on `/etc/init` and `/etc/inittab` are:

```
-r-xr--r--    root    other          /etc/init
-rwxr-xr-x    root    root           /etc/inittab
```

The run-level to which your system automatically boots is called the **initdefault** run-level. As shipped, run-level 2 is the initdefault run-level. For more information refer to the "System Startup and Shutdown" chapter.

Run-level 0 is a special run-level reserved for system installation. Do not run in run-level 0.

Run-level s is a special run-level reserved for system administration tasks. Shutting down the system for system administrator tasks will bring you to run-level s. You should not change to run-level s without using the `shutdown` command (i.e., do not execute `init s`).

If you are on a diskless cnode in a cluster, changing run-levels has no effect on other cnodes in the cluster. If you are on the root server of a cluster, changing run-levels can affect the diskless cnodes. In particular, if you change to run-level s, the server will be unable to respond to the diskless cnodes' requests. This means that the diskless cnodes will panic and halt.

Guidelines for Creating New System Run-Levels

On multi-user systems, it is necessary to make certain, suggested additions to run-level 2 (such as the addition of more `getty` entries to `/etc/inittab`). This was discussed in Chapter 3. You can also create new run-levels if you find it useful. Before creating a new run-level, take the following precautions:

- Make a copy of the original `/etc/inittab` file (using the `cp` command) and save the original version of the file under a different name (such as `/etc/orig_inittab`). In case you corrupt your file, you will still have a copy of a working version of `/etc/inittab`.
- Change the `initdefault` entry in your test version to "s". Note that "s" is not a normal run-level. If you create this test version, you should replace the "s" with "2" after testing is complete. Run-level s is for system maintenance only.

  By changing the `initdefault` entry to "s", you will come up in run-level s when you boot. You can *change* to run-level 2 after booting by executing `init 2`. If your new run-level 2 does not work, you can still boot.

  If you do not have a working state for the `initdefault` state, you may not be able to boot your system. After thoroughly testing your changes, restore the original initdefault value.

To create new run-levels, use one of the HP-UX text editors to make entries in `/etc/inittab`. These entries will define how you want the system to operate in its new run-level. Each one-line entry in `/etc/inittab` should contain:

- a unique two-character id used to identify an entry,
- a list of run-levels to which each entry applies,
- an action to be performed, such as `respawn`,
- the command that will be executed when that run-level is entered.

Refer to *init(1M)* and *inittab(4)* in the *HP-UX Reference* for a more complete description of `inittab`'s run-level entries. Once `/etc/inittab` contains all of the entries you want for the new run-level, save the file and exit the text editor. At this point you are ready to follow the procedures in the chapter called "Periodic System Administrator Tasks" in section "Changing the System's Run-Level".

In a few cases, such as when a newly-created run-level closely matches an existing run-level (that is, the differences between the two are trivial), you can move freely between run-levels as long as entering the new run-level does not kill user or system processes that may have begun in the previous run-level. If the new run-level is not specified in the `rstate` field of the `/etc/inittab` entry for the process's `getty`, the process will be killed.

In addition to possibly killing processes as a result of changing a run- level, you must be aware of other possible side effects as well. Consider the case where a user logs off, then you change run-levels (from run-level 2 to run-level 4). The user will be unable to log back in unless the user's `/etc/getty` entry is defined for both run-level 2 and run-level 4.

Example /etc/inittab File

The following is an example `/etc/inittab` for a system that contains a system console and six terminals. Run-level s is a system administration run-level. The initdefault run-level is run-level 2. Run-level 2 is a multi-user run-level, with a `getty` on every terminal. Run-level 3 is a test run-level, with a `getty` on both the system console and the system administrator's terminal (`/dev/tty01`) and "kill" entries for the other terminals. This run-level could be used by a system administrator who prefers to work from his own terminal rather than from the system console.

```
is:2 :initdefault:
st::sysinit:stty 9600 clocal icanon echo opost onlcr ienqak ixon icrnl ignpa
bl::bootwait:/etc/bcheckrc < /dev/syscon >/dev/syscon 2>&1 #bootlog
bc::bootwait:/etc/brc 1>/dev/syscon 2>&1 #bootrun command
cr::bootwait:/bin/cat /etc/copyright > /dev/syscon
lp::off:/bin/nohup /bin/sleep 999999999 < /dev/lp & stty 9600 < /dev/lp
rc::wait:/etc/rc <dev/syscon >/dev/syscon 2>&1 #run com
pf::powerfail:/etc/powerfail 1>/dev/console 2>&1 #power fail routines
mu:2:wait:/etc/multi_user 0</dev/syscon 1>/dev/syscon 2>&1 #multi_user envir
n1::off:
n2::off:  These are comment lines
n3::off:
co::respawn:/etc/getty console H
01:23:respawn:/etc/getty tty01   H
02:2 :respawn:/etc/getty tty02   H
03:2 :respawn:/etc/getty tty03   H
04:2 :respawn:/etc/getty tty04   3
05:2 :respawn:/etc/getty tty05   M
```

Note The `reconfig` program automatically adds `getty` entries to the `/etc/inittab` file when it is used to add terminals and modems to the system.

Creating and Using a Recovery System

A **recovery system** is a bootable subset of your HP-UX system. It contains only enough of the system to allow you to boot and to help fix your file system. It is used only if your normal HP-UX system cannot boot.

Once your system has been installed, the first thing you should do is make a recovery system. Then, if you can't boot from your root disk because your root disk is too corrupt or because you forgot your root password, your recovery system will be available to boot and repair your file system.

The recovery system is easy to build, and is valuable if you ever need it. A recovery system is built using a shell script, `/etc/mkrs`. You can build a recovery system in multi-user mode; you don't need to have your users log off. Build your recovery system on one 150-foot cartridge tape. If you use a 600-foot cartridge tape it will take much longer to create the recovery system.

Note As of the 6.5 release of HP-UX you cannot create a recovery system on flexible disks. You must use a cartridge tape for your recovery system.

Note If you change the swap space of your system, you MUST create a new recovery system.

Security If you are running a trusted system you must lock the recovery system tape. When you boot from a recovery system you are the superuser in an administrative mode. This means that you have all privileges. You also have no auditing when booted from the recovery system.

If you change the swap space of your system, you must create a new recovery system. The recovery system has a record of swap space addresses. If these addresses change, but are not changed in the recovery system, then when you boot the recovery system, it may overwrite and destroy your root file system.

Creating a Recovery System

Three programs are needed to make a recovery system:

- `/etc/mkrs`—a shell script that creates the device file, mounts the recovery system device, and creates the recovery file system.
- `/etc/mkrs.swap`—a program that determines where your system's swap space is. `/dev/kmem` is a special device file that allows access to the RAM locations occupied by the kernel.
- `/etc/mkrs.devs`—a program that finds the major and minor number of your root device.
- `/etc/mkrs.tool`—the recovery tool

The recovery system has a boot area so you can boot using just the recovery system. The recovery system also has a small file system, containing the following files and directories:

`hp-ux`	A minimal kernel.
`/bin`	The `/bin` directory contains a small subset of HP-UX commands. The actual commands vary depending on your recovery media. Use the `ls` command to list the exact files you have on your recovery system.
`/dev`	The `/dev` directory contains the device files necessary for using the recovery system (block and character device files for the root disk and the recovery drive).
`/disc`	This directory can be used to mount a file system.
`/etc`	The `etc` directory contains the tools and files necessary to fix your root file system: `sbtab`, `fsck`, `init`, `mknod`, `mount`, and `umount`. It also contains small `inittab`, `profile`, and `rc` files, which are necessary for booting.
`/tmp`	This directory is used for temporary file storage.

To create the recovery system:

1. Log in as the superuser, `root`.

   You will be accessing privileged commands, so you must have superuser privileges.

2. Determine whether the device files in `/dev` exist for the device on which you wish to create the recovery system. The `mkrs` program will use the following defaults:

   a. Default for the root device: `/dev/dsk/0s0`, `/dev/root`, or `/dev/hd`. One of these files should exist on your system. If none of them exist, you must either create one of them or supply the name of the device file associated with your root disk as an option on the command line. Only the block device file must exist for the root disk.

   b. Default for the recovery device: `/dev/update.src`, `/dev/rct/c0`, or `/dev/rct`. One of these files should exist on your system. If none of them exist you must either create one of them or supply the name of the device file associated with the recovery device as an option on the command line. The recovery device file can be either block or character type; the other *does not* have to exist.

   Refer to the section "Adding Peripheral Devices" for procedures on how to create device files.

3. Create the recovery system by using the `mkrs` command. `mkrs` has the form:

   `mkrs [-f` *rcdev*`] [-r` *rootdev] [-v] [-m series]*

   See the following section, "Using the mkrs Script", for details on options and defaults.

   This process takes approximately 1-2 hours on a 150-foot cartridge tape, and 4-6 hours on a 600-foot cartridge tape.

   If `mkrs` doesn't exist on your system (you receive a message `file not found`), follow the guidelines in "Installing Optional Software and Updating HP-UX" to install the ACONFIG fileset. `update` will automatically load all filesets dependent on ACONFIG.

4. Boot the recovery system to verify that it works. For this step, you will need to shut down the system. You probably will want to test-boot the recovery system during off hours if you have other users on your system. Follow the steps in the section "Booting the Recovery System".

5. Put the recovery system in a safe place and LOCK IT!

   When you boot using the recovery system, you come up as the root user. This is potentially a serious security problem. It is up to you, the system administrator, to keep this recovery system safe (so you can use it if needed) and out of sight (so unauthorized people do not have access to it).

6. If you loaded the ACONFIG fileset just to create a recovery system, you may wish to recover the disk space used by ACONFIG. To remove the fileset, refer to the instructions in "Removing Optional Products and FileSets" in this chapter.

Using the mkrs Script

`mkrs` has the form:

`mkrs [-f` *rcdev*`] [-r` *rootdev*`] [-v] [-m` *series*`]`

where:

rcdev — is the name of the device file for the cartridge tape drive on which you will create your recovery system. The `mkrs` command will, by default, look for the following device files:

`/dev/update.src`	if it exits as a character device file, else
`/dev/rct/c0`	if it exits as a character device file, else
`/dev/rct`	if it exits as a character device file, else the device file must be specified.

If none of the above defaults exist on your system, you must either create one of them or you must specify the recovery device file using the `-f` option. The recovery device file can be either a block or a character device file.

An error message will result if the user does not use one of the defaults and does not specify a recovery device file name.

rootdev — is the name of the device file for the root device. The `mkrs` command will, by default, look for the following root device files:

`/dev/dsk/0s0`	if it exists as a block device file, else
`/dev/root`	if it exists as a block device file, else
`/dev/hd`	if it exists as a block device file, else the device file must be specified.

If none of the above defaults exist on your system, you must either create one of them or you must specify the root device file using the `-r` option. The root device file must be a block device file. The character device file need not exist. An error message will result if a default root device file does not exist and the user does not specify a root device file name.

series — This value is normally not needed. If `mkrs` cannot determine the type of system you have it will send you an error message. If this happens re-execute `mkrs` using the `-m` option with either 300 or 500. Since you are creating the recovery system for your Series 300 machine, use the value 300.

For example, if your root file system is associated with the device file, `/dev/dsk/0s0`, and you will be creating your recovery system on the cartridge tape drive associated with the device file `/dev/update.src`, you would type the following command:

```
mkrs
```

The `mkrs` process takes about 1-2 hours on a 150-foot cartridge tape, and 4-6 hours on a 600-foot cartridge tape..

Booting and Using the Recovery System

Caution Do not use a pre-6.5 recovery system to fix your 6.5 system. The old version of `fsck` will damage your system.

A recovery system is intended to repair certain damage that might occur on your root file system. However, if the damage is severe or if the damage extends across the entire file system, it may be easier to re-install HP-UX and then restore files from your backup media.

To use your recovery system, you will need to boot, then use the recovery tools to gain access to your root disk. At bootup you will automatically be placed in the recovery system tool's menu. Follow these steps to boot your system.

1. *Power up all system swap devices* (generally, just your root file system's hard disk) and the recovery system device. Your recovery system uses the same swap space as on your normal system, so the swap devices must be powered up.

   When booting from your recovery tape you must either use the same device drive as you used to create the recovery system, or you must use a device with the same address as the one used to create the recovery system. Since the device files are already created on the recovery system with the address information, it must match.

2. Turn on your computer and (immediately) hold down the space bar until you see the word `keyboard` on your screen. This will access the attended mode of the boot ROM. A list of operating systems will come on your screen.

3. Find the system labelled "SYSHPUX" listed under the device you have your recovery system on. Enter its number on the command line. For example, if you have your recovery system on a 7946 cartridge tape drive, and your regular system on a 9133, you might see the following screen. You would select 2H:

```
9133; 1400, 0, 0
1H SYSHPUX
1B SYSBCKUP
7946; 0700, 0, 0
2H SYSHPUX
```

As the recovery system is booting, you may see lines such as:

```
prod#.... .. ignored
```

Don't worry about these messages. Your recovery system is a minimal subset of the kernel and doesn't have (or need) the device drivers the full HP-UX system has.

The booting process takes about ten minutes, after which you will see a message stating that it is doing an `fsck` of the root device and mounting the root device. The total process takes about twenty minutes.

Follow the instructions on the screen. Once you are in the recovery system's menu you can access a help facility. If you choose to recover the system manually (not recommended), refer to the section called "Recovering the System Manually".

Shutting Down the Recovery System

Following is the correct method of halting, or shutting down, your recovery system.

1. Choose the `Exit Recovery System and Reboot root file system` option of the recovery tool.
2. Take the recovery system's cartridge tape out of the drive.

Recovering the System Manually

This section describes the steps to recover your system manually. Use this method only if you have a specific reason to not use the automatic recovery option from the recovery system tool. This section also describes the steps the automatic recovery is taking.

To recover manually, boot the recovery tool and choose the `Work in a shell to perform manual recovery` option.

This section describes how to recover from specific, and often localized, problems on your file system. If none of these problems apply to your situation, or if after correcting these problems you still cannot boot from your root disk, then you probably need to re-install the system and restore from backups.

Once you have booted your recovery system and you see the shell prompt on your display, you can use the following steps to try to recover your root file system. The procedure outlined here makes the following assumptions:

- you cannot boot your regular system; you suspected a problem and used the recovery system to boot
- your root device is called `/dev/real.root` (block device file) and `/dev/rreal.root` (character device file)
- your recovery device has two names: `/dev/root` is the block device file for the recovery device and `/dev/rroot` is the character device file for the recovery device.

Use `ls -l /dev` to determine what device files are actually present on the recovery system.

Root device, in the following procedure, refers to the device that is root under normal circumstances (the hard disk drive associated with your root file system).

Step 1. Check the Critical System Files

To get your root volume to a state from which you can boot, you need to check several files. The critical files, and the appropriate action, for HP-UX are listed below. You should still be booted from the recovery system.

File	*Action*

`/bin/sh`	Copy the version of this file on the recovery system to the root volume, then remove and relink /bin/rsh. The commands are:
	`cp /bin/sh /disc/bin/sh`
	`ln /disc/bin/sh /disc/bin/rsh`

`/etc/init`	Copy the version of this file on the recovery system to the root volume: `cp /etc/init /disc/etc/init`

`etc/inittab`	If `inittab` is corrupted, `init` might fail. To work around this problem, save the `inittab` file (you may want to edit it later), then create a single line `inittab` by typing:
	`mv /disc/etc/inittab /disc/etc/inittab.save`
	`echo "is:s:initdefault:" > /disc/etc/inittab`
	Be very careful to type the second line exactly as shown (including the quotes).

In an HP-UX cluster this file is a CDF. It must be referenced as: /disc/etc/inittab+/*rootserver_name*.

etc/ioctl.syscon

If you have changed the device used as the console, it is possible that /etc/ioctl.syscon is incorrect, or it may have otherwise become corrupted. Work around this problem by removing the file (the next time the system is booted a correct file will automatically be created):

```
unlink /disc/etc/ioctl.syscon
```

In an HP-UX cluster this file is a CDF. It must be referenced as: /disc/etc/ioctl.syscon+/*rootserver_name*.

/dev/console

This file (which is also linked to /dev/syscon and /dev/systty) could be corrupted, resulting in the system not being bootable.

In an HP-UX cluster this file is a CDF. It must be referenced as: /disc/dev+/localroot/console, /disc/dev+/localroot/syscon, and /disc/dev+/localroot/systty.

If, when listing the file, you do not get the following line (note that the group name and the date information is not important):

```
crw--w--w- 3  root other 0 0x000000 Nov 10  10:12  /disc/dev/console
```

then re-create the files by doing the following:

```
# remove the three files, console, syscon, and systty
rm /disc/dev/console /disc/dev/syscon /disc/dev/systty
# recreate the /dev/console using the following
# mknod command
/etc/mknod /disc/dev/console c 0 0x000000
# link the files
ln /disc/dev/console /disc/dev/syscon
ln /disc/dev/console /disc/dev/systty
```

/hp-ux

If the kernel file got corrupted there are two workarounds: boot from a backup kernel or copy hp-ux.min from the recovery system to the root device.

In an HP-UX cluster this file is a CDF. It must be referenced as: /disc/hp-ux+/*rootserver_name*.

If you have a backup kernel (/SYSBCKUP) on your root device and if the backup kernel is not corrupted, you can boot your system from /SYSBCKUP. Reboot your system, hold down the space bar to get into the boot ROM's attended mode, and select the SYSBCKUP option. Once you have rebooted, you can either create a new kernel (using `config`) or copy the backup kernel to /hp-ux.

If there is no usable backup kernel, then you need to boot using /hp-ux.min from the recovery system. To copy hp-ux.min to the root disk, enter:

```
cp /hp-ux.min /disc/hp-ux
```

Now boot from the root disk.

Step 2. Fix Other Problems

If you have rebooted from the root volume, you can now fix other possible problems as described below:

1. If /etc/inittab was corrupted, you can now edit the version that was saved to fix it. Once edited, move it back to /etc/inittab. (Refer to the next section for details on fixing corrupted files.) Until you have tested your newly edited inittab, you should make the default state run-level s. You can switch to other states using `init x`, where x equals the desired run-level, 0-6.
2. If you lost other system files then you should update your system using the `update` command, which requires that at least the following commands be available:

```
/bin/sh              /bin/mkdir
/bin/pwd             /bin/cpio
/usr/bin/lifcp       /usr/bin/tcio
/etc/update          /etc/sysrm
```

   If any of these are missing, you can get them from your recovery tape by mounting the tape and copying the missing command(s) to the root device. You may already have /bin/sh on your system (from Step 3). To get the files onto your system, execute the following sequence:

```
mkdir /disc            only if /disc does not already exist
mount /dev/root /disc
cp /disc/bin/mkdir /disc/bin/pwd /disc/bin/cpio /bin
cp /disc/usr/bin/lifcp /disc/usr/bin/tcio /usr/bin
cp /etc/update /etc/sysrm /etc
```

3. If you think the boot area on your root device has been corrupted, then create a raw device file for root (assuming none exists) and copy the boot area from the recovery system to the root device as follows:

```
dd if=/dev/rfd of=/dev/real.root count=1 bs=8k
```

Also, the commands normally used to create backups are on your recovery tape. These commands are `cpio`, `tcio`, and `tar`. If these commands are missing from your root disk, and you wish to restore from backups, you can mount the recovery tape and copy the commands to the root disk. Refer to the procedure in step 4 for copying commands from the recovery system to the root disk.

Step 4. Return to Recovery Tool

Type `exit` or `^D` to return to the recovery tool, then reboot the system using the `Exit Recovery System and Reboot root file system` option. Remove the recovery system tape.

Mounting and Unmounting File Systems

When HP-UX is installed, only one file system (the root file system) exists. You may create, modify, and delete files from this system. By default (i.e., installation), the HP-UX file system exists on this single disk. It is possible to have other file systems on *different* disks; any other mass storage device supported by Series 300 HP-UX can be used as an additional file system. To accomplish this, the additional file system(s) are attached to either the root file system or other mounted file systems. The process of attaching additional and functionally independent file systems to the root file system is called **mounting** and is achieved with the `mount` command. The process of removing independent file systems from the root file system is called **unmounting** and is achieved with the `umount` command (note the spelling is `umount`, not `unmount`).

You cannot mount any devices that are attached to a diskless cnode. You cannot execute any mounts or unmounts from a diskless cnode. If you wish to mount or unmount a file system you must be logged into the root server. However, NFS mounts *can* take place on a diskless cnode.

Once a block device file exists for the new disk device and a file system has been created on the disk, the file system the device contains can be mounted. Refer to the "Adding Peripheral Devices" section in Chapter 4 for information on creating device files. The mounting operation makes any files on the new (mounted) file system become part of the file system hierarchy. Files can then be created, modified and deleted on this new file system. When you are finished with the files on that file system, it can be unmounted. Unmounting a file system removes its files from the file system hierarchy. More specifically, the association between the mounted file system and the root file system is broken (disconnected). The files themselves are untouched and remain on the mass storage medium; they may be accessed by re-mounting the file system.

In the file system's primary superblock, there is an informational byte called the clean byte. When you create a file system, the clean byte is set to FS_CLEAN. When you mount a file system, the clean byte is set to FS_OK. When you unmount a file system, the clean byte is reset to FS_CLEAN.

If you add your file system to `/etc/checklist`, `/etc/bcheckrc` will check the file system at bootup, and will use the clean byte to determine if the system was properly shut down. If the clean byte is FS_OK (not FS_CLEAN), then it was not unmounted with the `umount` command and could be corrupted. If the clean byte was FS_OK, then `bcheckrc` will run `fsck` to correct corruption before continuing. After `bcheckrc` has checked each file system in `/etc/checklist`, `/etc/rc` mounts them.

For an explanation on how to add entries to the `/etc/checklist` file, refer to the section "Adding to /etc/checklist".

You can also mount files across an NFS network if you have NFS installed. Refer to the *NFS Node Manager's Guide* for more information.

To Mount a File System

You must create a device file (if one does not yet exist) for the mass storage device containing the file system which is to be mounted. The device file must be a block type device file; refer to the "Adding Peripheral Devices" section in Chapter 4 for details.

1. Turn on the disk drive. If the file system is on removable medium (such as a flexible disk), insert the media in the mass storage device at this time. Do not remove the media until it is unmounted.

2. Choose or create a directory in the HP-UX file system that will be used to mount the file system.

   When a file system is mounted, it is attached to a directory in the existing file system. Any files that the directory previously contained appear to be *temporarily* replaced by the file system. Because of the confusion that may result (from the temporary "disappearance" of files in the mount directory), it is considered good practice to use an empty directory created specifically for mounting.

3. Mount the file system using the `mount` command. For example, to mount the drive at `/dev/dsk/3s0` onto the `/direct` directory, type:

   ```
   mount /dev/dsk/3s0 /direct
   ```

The diagrams that follow, for the sake of illustration, show a file system mounted on a directory that does contain files but this is *not* standard practice. Consider a directory called `/direct` that contains two files, `file1` and `file2`. Assume that you want to mount a flexible disk (that contains a hierarchical file system of its own as shown in Figure 4-18) on the `/direct` directory. The process of mounting the file system, modifying files on that file system, and unmounting the file system is shown in the following illustrations.

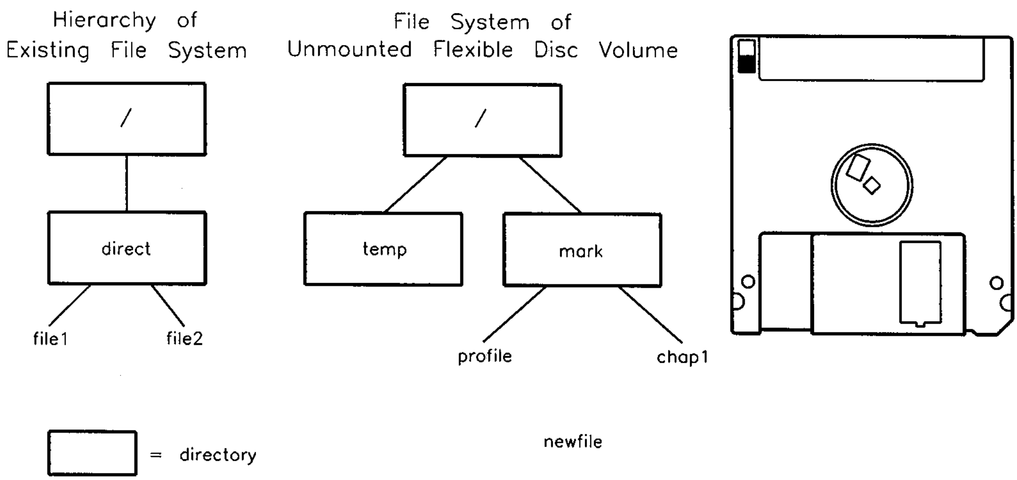


Figure 4–19. File System Hierarchy Before Mounting

Figure 4–19 shows the `/direct` directory on the existing mounted file system. It also shows the file system hierarchy on the as-yet-unmounted flexible disk.

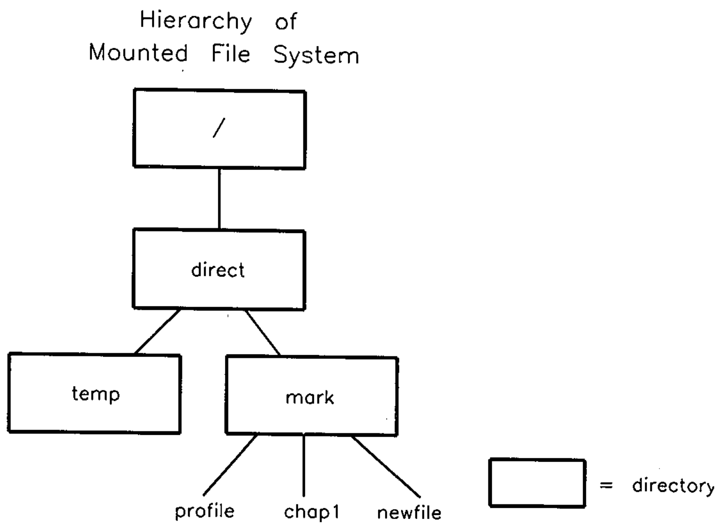


Figure 4–20. File System Hierarchy While File System is Mounted

Figure 4–20 shows the file system hierarchy once the file system is mounted on the `/direct` directory. The file `newfile` has been added to the new file system after it was mounted. Notice that the files `file1` and `file2` previously available in the `/direct` directory are no longer accessible; the files are still there, they just cannot be accessed until the file system is unmounted.

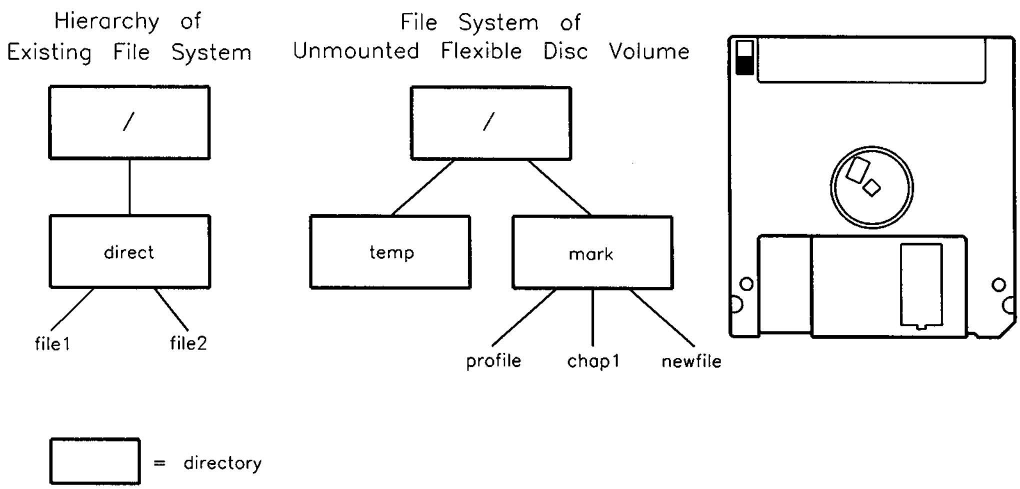


Figure 4–21. File System Hierarchy After Unmounting

Figure 4–21 shows the `/direct` directory on the existing mounted file system after the new file system is unmounted. The `file1` and `file2` files may once again be accessed. The diagram also shows the file system hierarchy on the unmounted file system.

Note Always unmount a file system before removing it from its mass storage device (as in removing the flexible disk). Removing a mounted file system from its mass storage device before unmounting it is likely to corrupt the file system because parts of the file system may be in the system's memory and not yet written out to the mass storage device. If you do not unmount a file system before removing it the clean byte will indicate a need for repair.

To Unmount a File System

Use the following procedure to unmount a file system.

1. Make sure that all files on the file system are closed; no one is accessing any file on the file system (including users on other cnodes if you are part of an HP-UX Cluster). Attempting to unmount a file system that has

open files (including your current working directory) causes the `umount` command to fail without unmounting the file system.

2. Enter the following:

   ```
   /etc/umount fd_name
   ```

   where *fd_name* is the pathname of the block device file of the device associated with the mounted file system. Notice this is `umount` without the `n`, not `unmount`. This command fails if there are open files on the file system you are attempting to unmount.

   For example, if you wish to unmount the file system associated with `/dev/dsk/3s0`, you would type:

   ```
   /etc/umount /dev/dsk/3s0
   ```

3. When the shell prompt is again displayed on your screen, it indicates that the file system is unmounted. If the file system is on a removable medium (such as a flexible disk), the medium can now be removed safely.

You can't unmount a file system that has open files. The following situations are the most frequent cause of open files on a file system:

- Your current working directory on a file system causes an open file on the file system.
- If a program stored on a file system is a shared program and the use count of that program is not zero, the file associated with the program is still open.
- A program on that file system is currently running.

The `fuser` command may be useful in finding processes that have files open. If you are on an HP-UX cluster, use the `cfuser` command.

Mounting/Unmounting File Systems Using /etc/checklist

All file systems in `/etc/checklist` are automatically mounted during bootup via the `mount -a` command in `/etc/rc`.

If you wish to unmount all file systems in `/etc/checklist`, enter the following:

```
umount -a
```

These file systems are automatically unmounted at shutdown.

For more information on adding file systems to `/etc/checklist`, refer to "Adding to /etc/checklist" in this chapter.

Mounting the /usr File System

It is common to place the files in `/usr` on a separate disk, and then mount it (`/users` is another commonly mounted directory). However, many commands are located in the `/usr` directory. These commands must be moved to the new file system before it is mounted.

The following procedure allows the transfer of `/usr` files from the root device to a second disk:

1. Initialize and create a file system on the new disk as described in the section "Creating a New File System" in this chapter. For this example assume the device name of this disk is `/dev/dsk/1s0`.

2. Create a temporary working directory in the root directory, then mount the file system:

   ```
   mkdir /disk1
   mount /dev/dsk/1s0 /disk1
   ```

3. Change your working directory to the directory where you will eventually mount `1s0`:

   ```
   cd /usr
   ```

4. Unmount any file systems mounted below `/usr`. For example, if `/usr/local` is mounted, unmount it now.

5. Copy all the files to the new disk:

   ```
   find . -print | cpio -pvdumx /disk1 | tee /tmp/cpio.log
   ```

   You will see each file listed as it is copied. A copy of the output will also be written to `/tmp/cpio.log`.

   This will take awhile. This would be a good time to read other important parts of this manual, get a cup of coffee, or go to lunch.

6. Verify the copy:

   ```
   dircmp /usr /disk1 | tee /tmp/dircmp.log
   ```

   This also will take awhile.

7. Edit the `/etc/checklist` file, adding the following entry:

```
/dev/dsk/1s0  /usr  hfs  rw  2  0  #/usr mounted
```

8. Unmount 1s0:

```
umount /dev/dsk/1s0
```

9. Shut down the system so there is no file system activity on `/usr`, then remove everything in `/usr`. If you do not remove everything on `/usr` you won't free the disk space on the root device:

```
shutdown
rm -r /usr/*
```

10. Return to your normal run-level.

    If you require users to login when accessing the system, type:

    ```
    init 2
    ```

    If you do not require users to login when accessing the system, type:

    ```
    init 1
    ```

    This will mount 1s0 on `/usr` when `/etc/rc` executes `mount -a`.

Removing Optional Products and Filesets

When you installed your system, files were copied from the installation media onto your root disk. These files were grouped into filesets. A **fileset** is the smallest unit you can move onto your system using the installation or update procedures.

If you are no longer using a fileset, you should remove it so you have more space on your file system. The program, `sysrm` performs the opposite of `update`; it removes optional filesets.

Note You cannot remove filesets that are on remote file systems, such as NFS-mounted file systems.

To remove filesets:

1. Determine which filesets you wish to remove.

   The directory, `/etc/filesets`, contains a complete list of filesets you have installed on your system. This list does not contain filesets that exist on NFS-mounted file systems.

2. Become the root user.
3. Move to the root directory and shut down the system with the default grace period, by typing in:

   ```
   cd /
   shutdown
   ```

4. Type in the following command, where *product* is the name or number of the product to be removed (determined in step 1):

   `/etc/sysrm` *product*

   The fileset(s) you typed in will no longer be available. You can load them back into the system, if needed, by using the update procedure (refer to the section "Installing Optional Software and Updating Your HP-UX Kernel" later in this chapter).

5. Return your system to normal operating mode by typing in:

   ```
   reboot
   ```

The reconfig Command

The `reconfig` program is useful for accomplishing several configuration tasks. The diagram in Figure 4–22 shows a map of the tasks you can accomplish using `reconfig`. At the bottom is the name of the section in this manual where the task is documented. The equivalent non-`reconfig` procedure often requires more knowledge of the system. It also provides more background on what needs to happen and why. The easiest way to perform these tasks, though, is to use `reconfig`.

This section gives an overview of how to use the `reconfig` command. For more information on the `config` command refer to Chapter 6. For more information on the `mkdev` command, refer to the section "Adding Peripheral Devices" in this chapter. For more information on the `mklp` script refer to the *HP-UX Reference* or the `/etc/mklp` script.

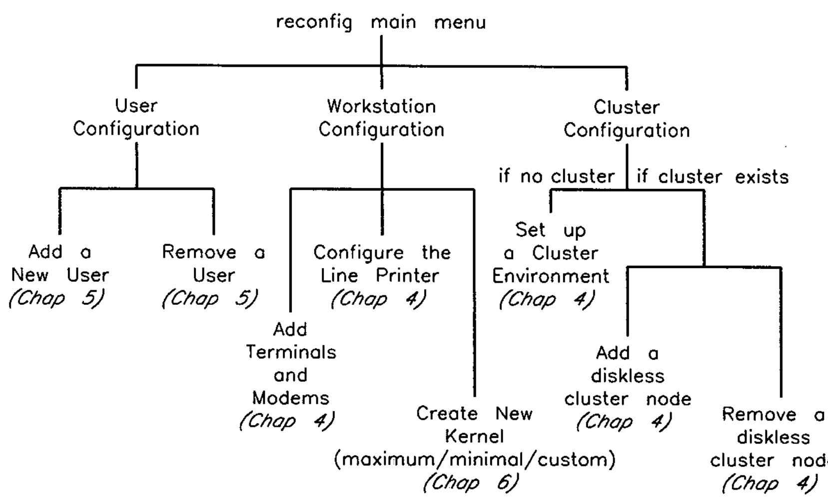


Figure 4–22. Reconfig Tasks

Using reconfig

This sub-section describes how to use `reconfig` in the general case. For specifics on how to do tasks in `reconfig` refer to the section stated in Figure 4–22.

Starting reconfig

There should be only one person running `reconfig` at any time. To have more than one person running `reconfig` could result in corrupted system files.

1. Log in as the superuser, root.
2. Start the `reconfig` program.

   If you have a Hewlett-Packard terminal, type:

   ```
   reconfig
   ```

   You will see the menu shown in Figure 4–23.

If you do not have a Hewlett-Packard terminal, you must enter `reconfig` with the `-m` option:

```
reconfig -m
```

This enters a special mode that does not produce escape-code sequences. This mode of operation is the same as normal operation with the following exceptions:

- Menus offering a choice of actions to perform identify each action with an associated number or letter. To choose an option, press the number or letter (uppercase or lowercase), then press the [Return] key.
- The [QUIT], [RESTORE], [PREVMENU], and [HELP] softkeys are replaced with the letters `q`, `r`, `p`, and `h` respectively. To select one, press the correct letter followed by [Return].
- When prompted for a choice in a multiple-choice field, type the corresponding number or letter or type the choice itself, then press [Return].

The Reconfig Main Menu

Figure 4–23 illustrates the main menu in the `reconfig` program:

```
**************************************************
           RECONFIG -- MAIN MENU
**************************************************

User Configuration
Workstation Configuration
Cluster Configuration









[        ] [        ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 4–23. Reconfig Main Menu

To choose a menu item, press the NEXT and PREVIOUS softkeys to move up and down the list, then press the SELECT softkey to go to the menu associated with the chosen item.

The reconfig Function Keys

There are seven possible function keys that appear on the menus in `reconfig`. Not all function keys appear on every screen: they appear only on the screens where they can be used. Use them as follows:

Reconfig Main Menu

Softkey Label	Definition or Use
[HELP]	Displays information describing your currently available options.
[PREVMENU]	Immediately exists the current menu and returns to the previous menu. No configuration information associated with the current menu is processed, and any modified values are destroyed.
[NEXT] [PREVIOUS]	Moves cursor to next or previous choice on multiple-choice menu. Displays next or previous choice on multiple-choice field. You can also use the [Next] or [Prev] keyboard keys.
[QUIT]	Exits the `reconfig` program. [QUIT] appears only in the main menu.
[RESTORE]	Restores all fields in the current menu to their original (default) values. The original values are the values that were shown on the screen before you changed any values.
[SELECT]	Initiates the configuration action associated with the current cursor position (the highlighted field). For all cases, the [Return] and [Select] keys perform the same function as the [SELECT] softkey.

Many menus also prompt for inputs. Each input on a non-multiple choice field is terminated by pressing the [Return] key. To select the displayed value for the field, press the [Return] key without typing an entry.

Additional Notes on reconfig

This subsection provides a few reminders for `reconfig`.

Help

If you wish to see more information while in the `reconfig` program, but you do not wish to read the documentation, use the [HELP] softkey.

Yes or No Answers

When you see a prompt that requires a yes or no answer, you can enter either `y` or `yes`, or `n` or `no`, followed by [Return]. These responses can be any combination of upper- and lower-case letters. Yes or no answers are required when a prompt ends with: `(y or n)`.

Getting Out of reconfig

Generally, you can exit the `reconfig` program by pressing the [QUIT] softkey from the main menu. If you accidentally started reconfig without the `-m` option, and are using a non-HP terminal, you can exit the program by using the `q` key.

Setting Up Printers and the LP Spooler

On a multiuser system, access to printers requires careful management and control. Many users need to print documents at the same time so there needs to be a way to print each file separately and to determine which file will be printed first.

HP-UX provides a series of commands, collectively referred to as the Line Printer Spooler (LP spooler), which are used to set up and control line printer spooling. Spooling temporarily stores print jobs in a spool directory until they can be printed. This means users can send files to the printer without needing to see if the printer is busy. The system will control how the print jobs are printed (see Figure 4–24). You can customize the LP spooler to spool files to different printers on both local and remote systems. In addition, you can group printers into classes so files are printed on the next available printer. This increases system efficiency.

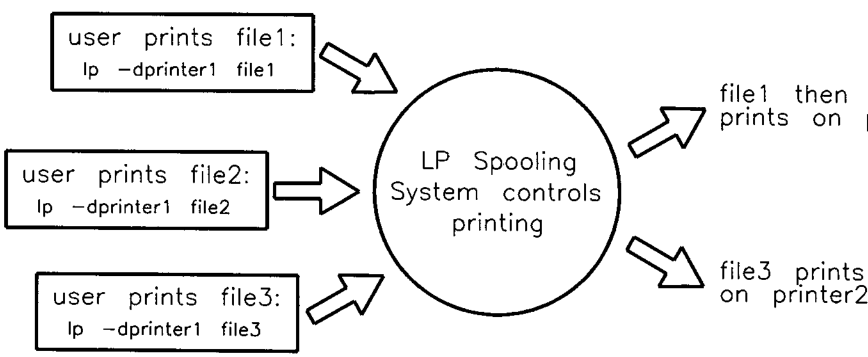


Figure 4–24. Line Printer Spooling

The LP spooler system allows smooth flow of jobs to the printer with minimum administrator intervention. Users submit their print requests to the system (via the `lp` command), then continue working at their terminals. Later they can use a command to check the status of the print job. Users can also stop their files from printing.

On some systems a person is designated to serve as the LP spooler administrator. In this section it is assumed that the LP administrator is the same person as the system administrator. The system administrator (for LP Spooling) is either the superuser (with user ID=0) or the LP spooler administrator (the user `lp`).

Some of the LP commands are available to all users; others, only to the system administrator.

On an HP-UX cluster only the root server can support the LP spooler system. Although you can add local printers to diskless cnodes, they cannot be part of the spooler system.

What Is in This Section

This section explains how to set up and use the LP spooler on your Series 300 HP-UX machine. It consists of the following subsections:

- "LP Spooler Terminology and Overview"—an introduction to the LP Spooler.
- "Installing Printers and the LP Spooler"—step-by-step method to install and start the spooler system.
- "Configuring the LP Spooler for Remote Operation"—details on how to set up remote spooling.
- "General-Purpose LP Spooler Commands"—spooler commands available to all users.
- "System Administrator LP Spooler Commands"—spooler commands available only to the system administrator (the root user).
- "Remote Spooling Commands"—spooler commands available across a network.
- "LP Spooling Directories"—files and directories used in the LP spooler system.
- "LP Spooler Administrator Duties"—tips on monitoring the spooler.
- "How Models Work"—description of what a printer model is.
- "Setting Up a Printer (the Manual Method)"—description of how to create a device file for a printer without using the `reconfig` program.
- "The LP Spooler in an HP-UX Cluster"—information on how the lp spooler system is different in a cluster than on a standalone system.
- "LP Spooler Errors"—information on how to correct common LP spooler errors.

LP Spooler Terminology and Overview

A **request** is a combination of one or more files to be printed and all associated information such as destination, number of copies, and other `lp` options. When `lp` is invoked, it associates a unique ID with the request and passes the request to the **LP scheduler** (invoked by the `lpsched` command). The LP scheduler routes the request to the proper **interface program** to do the actual printing on a device; the program functions as an interface between `lpsched` and printing devices. **Models** of interface programs are supplied with the LP spooler and, in some cases, have options to use specific printer features such as expanded or compressed print. The models can be used as is, modified for your specific needs, or used as templates for creating new interface programs.

The `lp` command directs output to the default destination unless a destination is specified when `lp` is executed. The **default destination** may be set or changed by the system administrator. A **destination** is either a printer or a class. A **class** is a name given to a list of printers. Each class must contain at least one printer although a printer may belong to zero, one, or more classes. If the destination is a specific printer, the output gets handled only by that printer. If the destination is a class, the output gets handled by the first available printer belonging to that class.

A complete LP spooler configuration for a system consists of devices, destinations (printer names and classes), interface programs, and the LP spooler commands in the `usr/bin` and `usr/lib` directories.

The LP spooler distinguishes between logical destinations and physical destinations. Logical destinations are defined using the `lpadmin` command whereas physical destinations are defined using `mknod`—which associates physical devices with device files. A single physical destination may be associated with one or more logical destinations. The `lp` requests are directed to a logical destination as long as it has been set to accept requests (refer to *accept(1M)*). When a corresponding physical destination (a printer) is available and has been enabled (refer to *enable(1)*), the request is transferred to it.

Installing Printers and the LP Spooler

This procedure will show you how to use the `reconfig` program to configure printers into your system, and will install and start the LP spooler system. If you wish to install a printer, but not install and start the LP spooler system, go to the section called "Installing a Printer (the Manual Method)". If you wish to install remote spooling, go to the section called "Configuring the LP Spooler for Remote Operation". If you do not want to use the `reconfig` program to install your printer and spooler system, you can use a program called `/etc/mklp`. If you wish to use `/etc/mklp`, read the `/etc/mklp` file for information: it is heavily commented and you can learn to use it from the comments.

If you are setting up a printer on a diskless cnode of an HP-UX cluster, you must use the manual method to set up the printer.

Printer and Spooler Setup

1. Log in as the superuser, root. If you are on an HP-UX cluster you must log into the root server as the superuser.
2. Start the `reconfig` program by typing:

   ```
   /etc/reconfig
   ```

   The main menu of `reconfig` will appear on your screen (Figure 4–25).

```
**************************************************
          RECONFIG -- MAIN MENU
**************************************************

User Configuration
Workstation Configuration
Diskless Configuration








[        ] [        ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 4–25. Reconfig Main Menu

Using the NEXT and PREVIOUS softkeys, move to the `Workstation Configuration` menu item and press the SELECT softkey. You will see the menu shown in Figure 4–26.

The last 3 lines of the menu will appear only if you have the ACONFIG fileset loaded on your system. You do not need the ACONFIG fileset to configure a printer.

```
**************************************************
        RECONFIG -- WORKSTATION CONFIGURATION
**************************************************

Add terminals and modems
Configure the line printer
Create a minimally loaded operating system
Create a fully loaded operating system
Create a custom operating system
```

		NEXT	PREVIOUS	HELP	SELECT	PREVMENU	

Figure 4–26.

3. Using the NEXT and PREVIOUS softkeys, move the cursor to the `Configure the line printer` menu item and press the SELECT softkey.

   The form for configuring a line printer will appear on your screen (Figure 4–27).

```
************************************************
      RECONFIG -- CONFIGURE A LINE PRINTER
***********************************************

Printer Name:
Model:
Printer Interface:      hpib (non-ciper)
Select Code:
Sub-Address:
Default System Printer: no






Printer Name? >>





[RESTORE] [ ] [ ] [ ]   [HELP] [ ] [PREVMENU] [ ]
```

Figure 4–27.

4. Type the printer name.

   This is the name you will use to access the printer. The printer name can be any unique combination of up to 14 characters and numbers; however, a name describing the printer is normally chosen. For example, if you are adding an HP 2225A Thinkjet printer to your system, any of the following names would be appropriate:

   `thinkjet` `thinkjet1` `hp2225a` `lp1`

   Press the [Return] key. The `reconfig` program will verify the printer name is 14 characters or less, and that it does not contain any spaces.

5. Type the model name.

   You will now see a list similar to the following on the bottom of your screen:

```
The printer models available on your system are:

dumb            hp2225a         hp2631g         hp2686a
hp2934a

   Model? >>

[RESTORE] [       ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [PREVMENU] [       ]
```

Figure 4–28. Choosing Line Printer Model

   By using the NEXT and PREVIOUS softkeys choose one of the models listed, then press the SELECT softkey.

   Each printer model listed has a corresponding description file in the `/usr/spool/lp/model` directory. The `lp` program uses these descriptions to access the special features of the printer. If your printer does not fit into any of these models, use the **dumb** model.

6. Select the printer interface for your printer.

   By using the NEXT and PREVIOUS softkeys choose one of the interfaces listed, then press the SELECT softkey.

   You will see the following information at the bottom of your screen:

```
The possible printer interfaces are listed below:

rs-232          hpib (ciper)          hpib (non-ciper)


 Printer Interface? >> hpib (non-ciper)

[RESTORE] [       ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [PREVMENU] [        ]
```

Figure 4-29. Choosing Line Printer Interfaces

This field allows you to indicate which type of interface the printer supports. The three choices are HP-IB (ciper and non-ciper) and RS-232. This information is needed to guarantee that HP-UX properly communicates with the printer. If you do not know which interface the printer supports, refer to the manual supplied with the printer. You can also use Table 4-4 to choose the proper interface. Find your printer's model number in the first column. If the second column says `printer`, then select `HP-IB (non-ciper)` if your printer is connected to an HP-IB interface card or select `rs-232` if it is not connected to an HP-IB interface card. If the second column says `ciper`, then select `HP-IB (ciper)`.

7. Type the select code, then press the Return key.

   Each interface card installed in your system has a unique address associated with it, which is referred to as its select code. The select code is set with switches located on the interface card. The select code is a number from 0 through 31. Sometimes a sticker displaying the select code is attached to the interface card. If you have used the *Peripheral Installation Guide* to set up your printer, you will have the correct select code on your worksheet.

   If you have not used the *Peripheral Installation Guide* to set up your printer and your interface card does not have a sticker on the back displaying the select code, you will need to shut down your computer, turn the power off, and remove the card to check the select code switches on it.

Caution *Do not* remove the card while your computer is turned on.

When you press the Return key the `reconfig` program will check that the select code you entered is between 0 to 31, but will not verify that the select code is correct.

8. Type the sub-address of your printer, then press the [Return] key.

   The sub-address is a number from 0 through 31. The sub-address depends on the type of printer, RS232 or HP-IB. Use Table 4-10 to help determine the sub-address number.

Table 4-11. Printer Sub-Addresses

Printer Interface	SubAddress
HP-IB	HP-IB address assigned to the printer. Look at the switch settings on the back of the printer or refer to the *Peripheral Installation Guide.*
RS232 with single channel (HP 98626 or HP 98644 card)	0 See the note on RS232 printers.
RS232 with 4-channel MUX (HP 98642 card)	channel number on the card (refer to the *Peripheral Installation Guide*) See the note on RS232 printers.

   When you press the [Return] key the `reconfig` program will check that the sub-address you entered is between 0 to 31, but will not verify that it is correct.

9. Decide if this should be the default system printer.

   Type `yes` or `no`, then [Return] (or just press [Return] to choose the answer shown).

   The **default system printer** is used if you do not specify a particular destination printer in the `lp` command. The default system printer is the printer associated with the device file `/dev/lp`. For examples on using the default printer and non-default printer, refer to the "Commands" sections toward the end of this section.

   You do not need to have a default system printer, but if you don't, then you must specify a destination printer each time you print a file. The system supports only one default printer at a time. Each time a new one is set, the old one is replaced.

10. You will now see the message:

```
Are the above values ok (y or n)? >>
```

If you type n then the form will cycle around again (go back to step 4) except instead of the original default values you will see the values you just entered.

If you type y then you will see the following message:

```
New printer <printer name> added, press [Return] to continue >>
```

where `<printer name>` is the new printer you just added. Go back to the main menu by pressing first the [Return] key and then the [Prev] function key.

Before returning to the previous menu, the `reconfig` program performs the following functions:

- creates a device file for the printer in the `/dev` directory,

  For example, if you named your printer `thinkjet1`, the following file will be created:

  ```
  /dev/thinkjet1
  ```

- adds the new printer name to the line printer spooler mechanism so HP-UX knows you now have a printer.
- links the printer device file to the `/dev/lp` device file if you choose to make this printer the default printer.

You will now see the main menu on your screen.

11. Press the [QUIT] softkey to exit `reconfig` and return to your shell.
12. Configure the operating system so it supports this printer. If you have already configured it to be a fully loaded operating system, or if you have already created a custom operating system containing the correct driver, you can skip this step. Refer to the section below called "Kernel Driver and Major Number for Printers" for help on deciding which kernel driver you need.

Kernel Driver and Major Number for Printers

This section shows the necessary **kernel driver** for your peripheral's product number. Use the `reconfig` program to add kernel drivers to your operating system (refer to "Creating a New Operating System" in Chapter 6).

In addition to the `printer` or `ciper` kernel driver as shown in Table 4-11, you need the kernel driver of the interface card the printer is connected to. If your printer is connected to an HP-IB card, you must have the 98624 kernel driver. If your printer is connected to an RS-232 card, you must have the 98626, 98628, or 98642 kernel driver. If you don't know what interface your printer is connected to, or can't decide which interface to connect it to, read the information for your peripheral in the *Peripheral Installation Guide*.

Printers that are connected to an HP-IB card and are using graphics mode need the `hpib` kernel driver.

Table 4–12. Table 4–12. Kernel Drivers for Printers

Product Number or Name	Kernel Driver
HP 2225, HP 2245, HP 2601, HP 2602A, HP 2631, HP 2671A/G, HP 2673, HP 2686, HP 2932, HP 2933, HP 2934, HP 82905, HP 82906, HP 9876	printer
HP 2563, HP 2565, HP 2566, HP 2567, HP 2608S	ciper

A Note on Adding RS232 Printers

When an RS232 printer, using the `dumb` printer model, is added to the system, the operating system will, by default, talk to that printer at 300 baud. For many people, this speed is inadequate. To increase the baud rate, you must become the superuser and edit the file `/etc/rc` to add a line similar to the following (you can find the exact lines you need in the *Peripheral Installation Guide*):

```
nohup sleep 2000000000 < /dev/PRINTER&
stty 9600 -ienqak ixon -parenb opost onlcr tab3 < /dev/PRINTER
```

where *PRINTER* is the printer name you specified in the form. If you wish to run at a speed other than 9600 baud, then replace the 9600 with the desired baud rate.

Configuring the LP Spooler for Remote Operation

You must configure both the client and the host machine before you can submit remote spooling jobs.

Configuring the Client Machine

To configure the client's LP spooler for remote operation, either follow the steps below, or read these steps, then modify and run the `/etc/mklp` script:

1. Log in as the superuser and shut down the LP scheduler:

   ```
   /usr/lib/lpshut
   ```

2. Refer to `/usr/spool/lp/model` or the description of supported printers in `/etc/mklp` and select the appropriate model shell scripts for the printers you want to add to the system. Make sure the model scripts have a permission mode of 644 and that they are owned by `lp` and group `bin`.

   If you want to modify one of the models for your system needs, copy then edit the model file and then associate the new model with a printer using `lpadmin -i` (the interface option).

3. Execute the `lpadmin` command with the `-p` option to name a printer. Repeat the command for each remote printer you want to configure into the system. The `lpadmin` syntax is:

   `/usr/lib/lpadmin -p`*name* `-v`*dev_name* `-m`*model#* *rem_sp_option*

   where *name* is what users will call the printer, *device* specifies the full path name of the device file of the printer, *model#* is the printer model (must be listed in the `/usr/spool/lp/model` directory), and *rem_sp_option* refers to the options shown in Table 4–13.

Table 4-13. Remote Spooler Options

Option	Description
`-ob3`	Use 3-digit request numbers associated with the printer directory. This is for consistency with BSD systems. The default is to use 4-digit request numbers.
`-oci`*rcancel_path* or `-ocm`*rcancel*	Causes the `cancel` command to use `rcancel_path` to cancel request to remote printers. This option is provided for administrators who wish to write their own remote cancel program. Specify the full path name to ensure that the correct command is used. The model `rcancel` will be used to cancel requests to remote printers. Currently, the only model is `rcmodel`, which causes `cancel` to call `/usr/lib/rcancel`.
`-orm`*machine*	The name of the remote machine is *machine*.
`-orp`*printer*	The printer to use on the remote machine is called *printer*.
`-orc`	Restricts users to cancelling only their own requests. The default is *not* to restrict the `cancel` command.
`-osi`*rlpstat_path* or `-osm`*rlpstat*	Causes `lpstat` to use `rlpstat_path` to obtain the status of requests on remote printers. This option is provided for administrators who wish to write their own LP status program. Specify the full path name to ensure that the correct command is used. The `rlpstat` model will be used to obtain the status of requests to remote printers. Currently, the only model is `rsmodel`, which causes `lpstat` to call `/usr/lib/rlpstat`.

An example of remote spooling setup is:

```
/usr/lib/lpadmin -plp3 -mrmodel -v/dev/null -ocmrcmodel -osmrsmodel \
-ormsystem2 -orplp -ob3
```

It names the local printer (lp3) and its model (rmodel). Since the spooler requires a device name, but the networking software takes care of the device, the device destination is specified as `/dev/null`.

The `-ocm` option specifies the `rcmodel` model to cancel requests to the remote printer. The `-osm` option specifies that the (`rsmodel`) command obtains the status of remote printers. The -orm option specifies that remote spooling be sent to the system name system2. The -orp option gives the remote printer name (lp). The -ob3 option makes the HP-UX and BSD UNIX spooling systems compatible.

4. For each of the printers defined with lpadmin, execute accept and enable to allow requests to reach the printer:

```
/usr/lib/accept lp3
/usr/bin/enable lp3
```

5. Select a printer as the system default. For example, to make the printer lp the default, use the command:

```
/usr/lib/lpadmin -dlp3
```

6. Restart the LP scheduler, then check that the LP spooler's scheduler is running properly:

```
/usr/lib/lpsched            [prints out status information]
lpstat -t
```

7. If the scheduler doesn't start, remove the FIFO and SCHEDLOCK files, then repeat Step 6:

```
rm -f /usr/spool/lp/SCHEDLOCK  /usr/spool/lp/FIFO
```

SCHEDLOCK prevents more than one scheduler from running at a time. FIFO is a filter created by lpsched for LP scheduler communications.

Configuring the Host Machine

You must configure the host machine so it knows it will receive requests from your local machine. There are two methods to do this. Method 1 may be easier to administer since it uses familiar files. Method 2 increases security.

Method 1. Add information in two places:

1. Add the client machine's information to the /etc/hosts.equiv file. For example, if the client's name is prclient, you would add the following line:

```
prclient
```

2. If this is the first client machine you must start the rlpdaemon on the host machine. To do this, add the following lines in the the /etc/rc file, in the lp_start shell function:

```
if [ -x /usr/lib/rlpdaemon ]
then
   /usr/lib/rlpdaemon -L/usr/spool/lp/rlpdaemon.log
   echo remote lp daemon started
fi
```

3. You must now start the remote lp daemon to enable the new information in `/etc/rc`. To do this, type:

   ```
   /usr/lib/rlpdaemon -L/usr/spool/lp/rlpdaemon.log
   ```

   From now on, the `/etc/rc` script will automatically restart the lp daemon each time you reboot.

Method 2. This method increases security on the host machine by allowing clients to access only the printer service on the host machine. Further information about the following files can be found in the "Configuration and Maintenence" section of the *Installing and Maintaining NS-ARPA Services* manual.

Perform the following steps:

1. If this is the first client machine, you must add the following line in the `/etc/inetd.conf` file on the host machine:

   ```
   printer stream tcp nowait root /usr/lib/rlpdaemon rlpdaemon \
   -i -L/usr/spool/lp/rlpdaemon.log
   ```

   And you must start the remote lp daemon by typing:

   ```
   /etc/inetd -c
   ```

   This allows any machine to access the host machine's printer service. To limit the printer service to specific machines you must also complete Step 2.

2. To limit printer service to specific machines, and therefore increase security, you must edit the `/usr/adm/inetd.sec` file. Perform the following steps:

   a. Check to see that the file exists by typing:

      ```
      ll /usr/adm/inetd.sec
      ```

If it doesn't exist, copy it from the /etc/newconfig directory and set the appropriate permissions by typing:

```
cd /usr/adm
cp /etc/newconfig/inetd.sec .
chown root inetd.sec
chgrp other inetd.sec
chmod 444 inetd.sec
```

b. Add entries to the /usr/adm/inetd.sec file by editing it. For example, if you wish to add the printer named prclient, you would add the line:

```
printer allow prclient
```

For further information on the format of entries in this file, read the comments in the file.

General-Purpose LP Spooler Commands

The following is a brief overview of the LP spooler commands available to all users; for further details consult the *HP-UX Reference* manual.

cancel(1) Cancels requests to an LP spooler line printer made with the lp command. The user may address a specific printer or a specific request ID number. Users may cancel their own print requests. The superuser may cancel any request. Refer to the *lp(1)* entry in the *HP-UX Reference.*

disable(1) Disables one or more physical printers such that they will not print lp requests. Refer to the *enable(1)* entry in the *HP-UX Reference.*

enable(1) Activates one or more physical printers to print lp requests.

lp(1) Sends requests to an LP spooler line printer. Requests are files and associated printing information (flags, etc.) sent to the spooler. The lp command returns (to standard output) a unique ID associated with a request.

lpstat(1) Prints current LP spooler status information such as requests, IDs, and scheduler information.

slp(1)	Allows you to set the options for a printer. For example, some printers don't indent properly. The `slp` command has an option to change the indentation.

System Administrator LP Spooler Commands

The following commands are available only to the system administrator (the user `root`) or the LP spooler administrator (the user `lp`). Further details are contained in this section and in the *HP-UX Reference* manual.

accept(1M) Allows `lp` requests to occur on one or more logical destinations where a "destination" is a printer or class of printers.

lpadmin(1M) Configures the LP spooler system by describing printers, classes, and devices. The LP scheduler must *not* be running when most `lpadmin` command options are used.

For example, if you have an HP 2934A that is accessible through the device file /dev/lp, you can use the following command line:

```
/usr/lib/lpadmin -plp -v/dev/lp  -mhp2934a -h
```

where:

`-plp` specifies the printer. The logical destination name is `lp`.

`-v/dev/lp` specifies the full path name of the printer's device file—the physical destination.

`-mhp2934a` specifies a model in the `/usr/spool/lp/model` directory.

`-h` specifies that the printer is "hard-wired".

lpmove(1M) Moves requests queued by the LP scheduler from one destination to another. The LP scheduler must *not* be running when `lpmove` is used. Refer to the *lpsched(1M)* entry in the *HP-UX Reference.*

lpsched(1M) Schedules requests taken by `lp` for spooling to line printers.

lpshut(1M) Shuts down the LP scheduler. Refer to the *lpsched(1M)* entry in the *HP-UX Reference.*

reject(1M) Rejects `lp` requests on one or more logical destinations where a "destination" is a printer or class of printers. Refer to the *accept(1M)* entry in the *HP-UX Reference.*

Remote Spooling Commands

Remote spooling command allow the spooler to send requests back and forth among HP 9000 Series 300, 800, and BSD 4.2 or 4.3 systems. Refer to the section "Configuring the LP Spooler for Remote Operation" for details on how to set up remote spooling.

rcancel(1M) When a remote printing request is canceled, `cancel` calls this command to actually cancel the printing request.

rlpstat(1M) If status is requested using `lpstat` and there are remote printers on the system, `lpstat` calls this command to report on remote spooling status information such as each request, its ID, the username, the size of the request, and scheduling information.

rlp(1M) If printing to a remote printer, the printer model file associated with that printer calls this command to connect to the remote system and sends the spooling request.

rlpdaemon(1M) This daemon program or spool area handler runs on a remote system that receives requests to be printed. It handles remote system communications such as receiving files for printing, returning status to the system that originated the remote spooling request, and cancelling remote spooling requests. To set up remote spooling on your system you can start this program from `inetd` or directly at boot time in the `/etc/rc` file.

LP Spooling Directories

The spooling system involves several directories that contain the commands, the spooler setup, and the temporary storage area where printing requests wait to be printed. The directories are described here and shown in Figure 4-30.

`/usr/spool/lp`
: LP spooler system parent directory. All information about the setup and printing queues is located here.

`/usr/spool/lp/class`
: Printer classes directory. This contains the files that define how printers are grouped.

`/usr/spool/lp/model`
: System-supplied interface programs. This directory contains model shell scripts designed for particular printer models.

`/usr/spool/lp/interface`
: Interface programs in use on your system. This has shell scripts from the file called `/usr/spool/lp/model`. You can modify them for particular printers. If interfacing a printer for which there is no model file, you may need to create an interface program for it.

`/usr/spool/lp/request`
: Destination queues. This is where all `lp` requests are queued. It usually contains a subdirectory for each printer configured on the system.

`/usr/bin`
: Contains user-executable commands such as the LP spooler commands.

`/usr/lib`
: Contains administrator-executable commands such as the LP spooler commands.

`/usr/spool/lp/cmodel`
: Contains system-supplied interface programs, in the form of model shell scripts, for processing remote cancel requests. Currently, the only model is `rcmodel`.

`/usr/spool/lp/smodel`
: Contains system-supplied interface programs, in the form of model shell scripts, for pro-

	cessing remote status requests. Currently, the only model is `rsmodel`.
`/usr/spool/lp/cinterface`	Contains the shell scripts from `/usr/spool/lp/cmodel`.
`/usr/spool/lp/sinterface`	Contains the shell scripts from `/usr/spool/lp/smodel`.

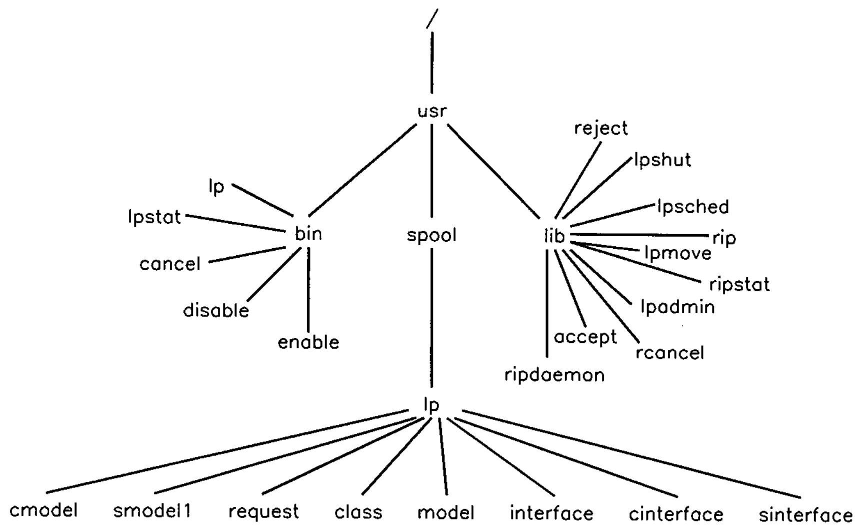


Figure 4-30. LP Spooler Directories and Files

LP Spooler Administrator Duties

There are several activities that you may need to carry out as the system administrator of the LP spooler system:

- determining the current status of the LP spooler system;
- grouping printers into classes;
- removing destinations (printers and classes of printers);
- moving requests to other destinations.

Determining LP Spooler Status

As the LP administrator, you need to monitor the spooling system. The command `lpstat` has options that provide a variety of information about your LP spooler system. Used without any options, `lpstat` prints the status of all requests that you have made to `lp` and the `-t` option gives complete LP spooler status information. For example,

```
lpstat -t
```

results in output similar to:

```
scheduler is running
system default destination: lp
device for lp: /dev/lp
lp accepting requests since Jun 14, 15:37
printer lp now printing lp-165.   enabled since Jun 23 13:31
lp-165          williams        62489    Jul  9  12:53 on lp
lp-166          jones            1374    Jul  9  13:39
```

You can specify any combination of the following options on an `lpstat` command line. You can also allow the command with particular request IDs, in which case `lpstat` provides status information about those requests. The options that you can specify with `lpstat` are:

-a*[list]* Print the request acceptance status (with respect to `lp`) of logical destinations. *List* is a list of intermixed printer names and class names. If you do not specify *list,* the acceptance status of all logical destinations is printed.

-c*[list]* Print class names and their members. *List* is a list of class names. If you do not specify *list,* all classes and their members are printed.

-d Print the system default destination for `lp`.

-o*[list]* Print the status of requests. *List* is a list of intermixed printer names, class names, and request IDs for which you want request status. If you do not specify *list,* `lpstat -o` has the same effect as `lpstat` (with no options).

-p*[list]* Print the status of printers. *List* is a list of logical printer names. If you do not specify *list,* the status of all printers is printed.

-r Print the status of the LP scheduler.

- `-s` — Print a status summary that includes the status of the LP scheduler, the name of the system default destination, a list of class names and their members, and a list of logical printers names and their associated device file names.
- `-t` — Print all status information.
- `-u`*[list]* — Print the status of requests for particular users specified by the login named in *list.* If you do not specify *list,* the status of all users' requests is printed.
- `-v`*[list]* — Print the pathnames of the physical devices associated with the logical printer names specified in *list.* If you do not specify *list,* the names of all of the logical printers and their associated physical devices are printed.
- `-i` — Inhibit the reporting of remote status.

Building Printer Classes

A **class** is a name given to a group of one or more printers. When requests are sent to a class, they are serviced by the first available printer that is a member of that class.

The `-c` option of the `lpadmin` command inserts a printer into a particular class. If the class does not already exist, it is created. If the class does exist, the printer is added. For example, you could associate the lp printer described above to a class named `class1` with:

```
/usr/lib/lpshut                      [to stop the scheduler]
/usr/lib/lpadmin -plp -cclass1       [to restart the scheduler]
/usr/lib/lpsched
```

This creates the class `class1` (unless it already exists) and inserts the printer `lp` into it.

Removing a Printer

LP spooler destinations (printers, classes, or both) are removed with the `lpadmin` command.

If you remove a printer from your system, you need to perform the following steps to stop the lp scheduler:

1. Shutdown the lp scheduler by typing:

   `/usr/lib/lpshut`

2. Remove the printer name from the scheduler by typing:

   `/usr/lib/lpadmin -x`*printer_name*

   where *printer_name* is the name you called the printer when you set it up.

3. Restart the lp scheduler (if you still need it) by typing:

   `/usr/lib/lpsched`

If space is a problem, and this was the only printer of its type on your system, you can remove the printer's kernel driver by creating a new operating system. Refer to the Chapter 6 to decide which kernel driver you can remove.

To remove a printer from a specific class, use `lpadmin`'s `-r` option:

```
/usr/lib/lpadmin  -plp  -rclass1
```

Removing the last remaining member of a class causes the class itself to be deleted. In the example above, since `lp` is the only member of `class1`, the class is deleted.

To remove an entire class of printers, use `lpadmin`'s `-x` option:

```
/usr/lib/lpadmin -xclass1
```

To remove a printer that is not a member of a class, use `lpadmin`'s `-x` option as follows:

```
/usr/lib/lpadmin -xlp
```

Note No printer or class of printers can be removed if it has any pending requests. You can use `lpmove` or `cancel` to move or delete the requests.

Moving Requests

As system administrator you may need to move requests from one destination to another, such as if one printer is down for repairs. Use the `lpmove` command to do this. The general procedure is:

1. Shut down the LP scheduler:

   ```
   /usr/lib/lpshut
   ```

2. Check the status of the new printer with the `lpstat` command.

   The `lpmove` command doesn't check the acceptance status of the new printer when it moves requests, so you should always check. For example, if you will be moving requests to the `lp2` printer, execute:

   ```
   lpstat -alp2
   ```

3. Execute the `lpmove` command.

   The syntax is:

   `lpmove` *from_printer* *to_printer*

   For example, to move all requests for printer `lp1` to printer `lp2`, type:

   ```
   /usr/lib/lpmove lp1 lp2
   ```

   To move the request with the ID `lp1-103` to printer `lp2`, type:

   ```
   /usr/lib/lpmove lp1-103 lp2
   ```

4. Restart the LP spooler system by typing:

   ```
   lpsched
   ```

Note Moving requests from one printer to a dissimilar printer may cause incorrect output. The options specified by the user for the original may be meaningless on the new printer.

How Models Work

Models are shell scripts, C programs, or other executable programs that interface between `lpsched` and devices. Several model scripts are shipped with your system and are located in the `/usr/spool/lp/model` directory. As shipped to you, this directory includes several model scripts. These model scripts must have a permission mode of 644 and be owned by `lp` and group `bin`. Refer to the `/etc/mklp` script for a description of the provided models.

If you want to modify one of the models for your system needs, make a copy of it, modify the copy, and then associate the copy with a printer using `lpadmin` with the `-i` (interface) option. Remember to stop the LP spooler system before using `lpadmin`, and restart it afterward.

Refer to *lpadmin(1M)* for more details on models.

Setting Up a Printer (the Manual Method)

HP-UX supported printers will use either the HP-IB or RS-232 interface. Some HP-IB printers use either AMIGO or CIPER protocol. This allows them to share the HP-IB with other devices. Other HP-IB printers may require a separate interface. RS-232 printers use the same device drivers as terminals. Baud rate, parity, and other RS-232 characteristics must be considered.

All printers on HP-UX are character devices. Within this category of character devices, printers can be used in two different modes: you can communicate with printers using drivers that either interpret (cooked mode) or do not interpret (raw mode) the data.

For a complete list of Series 300 supported printers and their major and minor numbers refer to the *Series 300 Configuration Reference Manual.*

Depending on the interface and protocol of the printers, the internal HP-UX data is prepared for printing. In particular newline characters are expanded to a linefeed-carriage return pair, and tabs are expanded to spaces. These functions may be controlled by the `stty` command (for RS-232 printers), by the device minor number, and/or the device driver in concert with the `slp` command (for HP-IB printers), or by an external filter program.

Volume Numbers for Printer Device Files

For HP-IB devices which use either AMIGO protocol or no protocol, the volume number field of the minor number will be a single-digit hexadecimal value made up of four bits. Table 4–14 describes the bits involved:

Table 4–14. Volume Number with AMIGO Protocol or No Protocol

Bit Order	When Clear (0)	When Set (1)
0	Printer	Plotter or raw mode printer for graphics dumps. This bit overrides all other bits.
1	Normal	No overprint capability
2	Normal	Uppercase only
3	Normal	No page eject on open and close

For HP-IB devices which use CIPER protocol, the volume number field of the minor number will also be a single-digit hexadecimal value made up of four bits. Table 4–15 describes the bits involved:

Table 4–15. Setting the Volume Number for Printers with CIPER Protocol

Bit Order	When Clear (0)	When Set (1)
0	Printer	Plotter or raw mode printer
1	Not applicable	Not applicable
2	Not applicable	Not applicable
3	Normal	No page eject on open and close

Using Printers as Spooled Devices

Printers can be accessed, through the line printer spooler (refer to *lp(1)*), as spooled devices; files are kept in a spool directory until the device is ready to process them. If your printer is set up as a spooled device, you can direct output to it at any time, whether it is busy or not. To set up your printer as a spooled device, follow the instructions in the "Setting Up the LP Spooler" section of this chapter. That section also explains commands which control the LP (line printer) spooler.

Note If you have a system printer, you should always name its corresponding special file `/dev/lp` because some commands use this special file as a default. You can create an individual special file for your favorite printer and give it the pathname `/dev/lp` (there is one created for you during system installation). Alternatively, you can take an existing special file for your favorite printer and create a link to it from the pathname `/dev/lp`.

To create special files for a printer, first determine the printer's location (i.e., interface and bus address). Use the guidelines in the installation guide supplied with your computer to select the most appropriate location. Finally, create the special file for the printer using the `mknod` command. You must assign a *unique* special file name to each entry you create; refer to the *intro(4)* entry in the *HP-UX Reference* for a suggested naming convention.

The LP Spooler in an HP-UX Cluster

All spooler printers must be located on the root server. Therefore only one copy of `lpsched` should be running in the cluster. `lpsched` must be run on the root server. This means that the configuration of the spooling system is the same as a current HP-UX system. All spooled print jobs on all cnodes will have their output spooled in the global file system. `lpsched` will be notified of the new print job and act upon it.

You can have local printers on diskless cnodes, but they cannot be part of the LP spooler system, and cannot be accessed by any other cnodes.

LP Spooler Errors

In the event of errors, check these error messages and perform the suggested corrective actions give below:

Corrupted member reported by lpadmin

Replace contents of `usr/spool/lp/member` with the device file name associated with the line printer (generally `/dev/lp`).

Error message: "lp has disappeared"

Use `lpadmin -xlp` to remove the entry and use `lpadmin` to reinitialize the spooling system.

Error message: "lp: destination 'lp' non-existent"

Check mode, permission, group, ownership, and existence of all `lp| directories and files. You may also need to remove lp` from the member directory and make `/usr/spool/lp/default` a zero-length file.

Error message: "lp: Can't open acceptance status file"

Check ownership of the `lp` directory. It should belong to `lp` and should be in the group `bin`.

Error message: "lpadmin: Can't lock printer status"

Check ownership of the `lp` directory. It should belong to `lp` and should be in the group `bin`.

Error message: "lp: Can't access 'file name'"

The directory does not permit "others" to read. Change the access permission mode of the directory

Error message: "lp: Can't open file"

The file cannot be read by general users. Change the access permission mode of the file.

Error message: "Cannot open /dev/lp"

The printer is off-line, the cable is disconnected, or the printer is powered off. Correct the problem and enable the spooler by issuing the `enable` command.

Installing Optional Software and Updating Your HP-UX Kernel

This section describes the steps necessary to install optional software or update your HP-UX kernel.

In this section, the word **update** has three meanings:

- the HP-UX program used to alter your system (`/etc/update`)
- the procedure described in this section that uses the `/etc/update` process to change your HP-UX system by adding new or different software (updating your system)
- a type of product from Hewlett-Packard (e.g., 6.5 HP-UX Update).

You will use the `/etc/update` process to perform any of the following:

- change from one version of HP-UX to another (e.g., use the update tape to change from version 6.2 HP-UX to version 6.5 HP-UX)
- add optional software or an application to your system (e.g., LAN)
- add an optional partition or fileset to your system (for example, all on-line reference pages for the HP-UX commands included in the HP-UX operating system are in the fileset called AMANUAL, the partition called TEXT).
- upgrade from one user limit to another (e.g., upgrade from a single-user system to a multi-user system).

You should be familiar with the following definitions:

destination directory This is where your software product will be installed. This is usually the root directory (/).

fileset A group of files. A fileset is the smallest loadable piece of HP-UX. One or more filesets makes up a partition.

partition A partition consists of one or more filesets which usually make up either one software product or a well defined software function (for example all on-line reference pages for HP-UX commands are in the "TEXT" partition).

source device	This is the peripheral device (either flexible disk drive or cartridge tape drive) that you will use to install your software product.
tape	In the update procedure, tape means cartridge tape.
flexible disk	In the update procedure, flexible disk means 3 1/2 inch microflexible disk.

Update Process Overview

The /etc/update process reads the contents of the tape or flexible disks from a **Source Device** and stores the information in a **Destination Directory.** The /etc/update process may then reboot the system.

An overview of the entire updating procedure is:

- Read the *Read Me Before* information supplied with your update media.
- Prepare the system for an update.
- Locate and write protect the update software media.
- Load the update tools if you are updating your operating system. Once you've loaded the update tools you can read the update information file, or you can wait until you have finished the update. The update information file is:

  ```
  /etc/newconfig/Update_info/to.6.5
  ```

- Begin the /etc/update process.
- Set or verify the Source Device and Destination Directory.
- Read the source tape or flexible disk.
- Select and load the desired partitions and filesets.
- Exit the update program.
- Check for additional information in the *Installation Notes* or in the directory called /etc/newconfig/Update_info.

Each of the above steps is described in following subsections, starting with the section "Preparing to Modify Your System". First, there are the following subsections describing the `/etc/update` process:

- Updating with non-HP terminals
- No-reboot option in `/etc/update`
- Softkeys in `/etc/update`
- Interactive versus batch processing
- Aborting `/etc/update`

Overview of Updating if You Will Use an HP-UX Cluster

If you currently have an HP-UX Cluster, and you are updating to a new version of HP-UX, simply update your system and continue to use it.

If you currently have a standalone system, and you are updating your system to take advantage of HP-UX cluster capabilities, follow these steps:

1. Prepare the system for an update.
   a. Read the "Cluster Concepts" information in Chapter 2.
   b. Check memory requirements in your *Before Installing* document.
   c. Prepare your system as described in "Preparing to Modify Your System"
2. Update the system.
   a. Insert tape.
   b. Shut down system.
   c. Run GETTOOLS
   d. Run the update program. You must install/update all filesets and partitions on the tape if you wish to run an HP-UX cluster.
   e. Check `update.log`.
3. Set up a cluster environment using the procedures described earlier in this chapter.
4. Add diskless cnodes to your cluster using the procedures described earlier in this chapter.

General Discussion of the Update Process

This section gives a general discussion of how the /etc/update process works.

Updating with Non-HP Terminals

If you do not have an HP (or HP compatible) terminal, use the -m option for /etc/update to enter a special mode that does not produce escape-code sequences. This mode has identical menu screens, but you interact with the program in the following way:

- All the menus and user interaction scroll up the screen.
- Menus offering a choice of actions to perform identify each action with an associated number or letter. To choose an option, press the number or letter (uppercase or lowercase), then press [Return].
- There is no select, so you must mark, then load, partitions and filesets to copy them onto your system.
- Use the "d" option (Display options for a new partition) to leave a fileset menu and return to the main menu.

No Reboot Option in /etc/update

Some optional software and applications may not require an automatic reboot when exiting /etc/update. For these software products you can use the -n (no reboot) option when starting /etc/update. Use this option only if so stated in the installation instructions or *Read-Me* for your software product.

The /etc/update program will verify that the product does not require a reboot when it reads the source tape (Step 6 in the update instructions).

Softkeys in /etc/update

During the update there will be up to six softkeys shown on the bottom of your screen. A **softkey** (or function key) is a key corresponding to one of eight fields shown in reverse-video on the bottom of your screen. Each of the six softkeys defined for /etc/update is described below:

- [MARK] marks a partition or a fileset to be loaded. This key acts as a toggle: pressing the key once will place an asterisk (*) to the left of the highlighted partition or fileset. Pressing the key twice will unmark that partition or fileset. All asterisked partitions and filesets will be loaded when the LOAD softkey is pressed.

- LOAD will cause all marked partitions and filesets to be copied from the source tape or flexible disk to your system.
- NEXT will move the cursor (highlighted item) to the next item on a multiple-choice action menu. The Next keyboard key functions the same as this softkey.
- PREVIOUS will move the cursor (highlighted item) to the previous item (the item listed above the current item) on a multiple-choice action menu. The Prev keyboard key functions the same as this softkey.
- SELECT will execute the currently highlighted option. This softkey functions the same as the corresponding Select keyboard key.
- QUIT when pressed in the main menu will exit the `update` process. When pressed from other menus will exit to the main menu.

Interactive versus Batch Processing in /etc/update

There are two ways to select, then load, the desired partitions and filesets:

Interactive: Using the SELECT softkey you choose a fileset and wait while it loads onto the system. You are then returned to the menu and may choose another fileset. If you SELECT a partition you will go to the fileset menu showing all the filesets in that partition.

This is the recommended method for flexible disk users, and is described in detail in the section "Updating and Installing Optional Software from Flexible Disk".

Batch: Using the MARK softkey you choose the filesets and partitions you wish to load onto your system. When you have marked them, press the LOAD softkey. The update procedure will then load all the marked filesets.

If you attempt to exit `/etc/update` (press the QUIT softkey) while items are still marked, you will receive the following prompt:

```
There are still choices marked for loading.  Do you want to
load them before exiting the update process?
(Enter y or n) >>
```

Type in y (yes) or n (no) depending on what you want.

If you have only one product to update you can simply set up your Source Device and Destination Directory, MARK all partitions and filesets you want loaded, press the QUIT softkey, and type y. The marked partitions and filesets will load, and the `/etc/update` program will terminate without any additional input from you.

This method is recommended when updating from cartridge tape and is described in detail in the section "Updating and Installing Optional Software from Cartridge Tape".

Aborting /etc/update

The only way to abort the `/etc/update` program is to press the QUIT softkey from the main menu. This will initiate a system reboot unless the `-n` option was used.

Preparing to Modify Your System

Updating to a new release of the operating system and installing some optional software products will modify your HP-UX kernel. To minimize any possible file system or kernel damage due to mistakes while running `/etc/update`, you should properly prepare your system.

Unless the installation instructions for your optional software instruct you to ignore the preparatory steps, you should prepare your system using the procedure described below. While still the superuser: copy source device information, shut down the system, back up, and check the file system. In addition, if you are updating to a new release of HP-UX, you must load the update tools from the source tape.

Copy Source Information

The `/etc/update` program defaults to using a device file called `/dev/update.src` for the source device. If this is the correct device file name for your source device then skip this step.

If you do not have a device file called `/dev/update.src`, or if you will be updating from a different device, then you should copy your source device information now. You need to know only the source device's device file name. If you don't know the device file name, then determine its name from the information given in a directory listing. If your device files are under the old naming convention (all device files are in the `/dev` directory), then type:

```
ll /dev/*
```

If your device files are under the new naming convention (device files are grouped in sub-directories), then type:

`ll /dev/rdsk/*`	*if installing from flexible disk*
`ll /dev/rct/*`	*if installing from cartridge tape*

You will see lines of the form:

`cr--r--r--    root    other` *maj 0xSaBaUV* `/dev/rdsk/1s0`

The number that is shown in the above example as **maj** is called the major number. The number after the major number (shown in the above example as *0xScBaUV*) is called the minor number. The minor number is a hexadecimal number. The minor number is comprised of addressing information:

0x This indicates the number is hexadecimal. It will not help you in identifying your device.

Sc The select code. The select code is set on the interface card the device is connected to. This number will probably be 07 or 0e (0e is the hexadecimal equivalent of 14 in base 10). Sometimes your interface card will have a sticker showing the select code. The interface card is on the back of your computer.

Ba The bus address. The bus address is set on the back of the device. This is probably the information you need to identify the source device. Match this hexadecimal number to the setting of the binary switches on the back of your device.

U The unit number. This will probably be 0 unless the device is an integrated device. Integrated means two drives in one unit using only one controller (for example a hard disk drive and a cartridge tape drive). If it is an integrated device with only one controller, the hard disk should be 0, and the cartridge tape drive or flexible disk drive should be 1.

V The volume number. This will probably be 0. It will not help you to identify the device.

Add Your Destination Directory to the /etc/checklist File (Optional)

If you plan to load the software product into the / directory (usually the case), then skip this step.

If you plan to load the software product into a directory other than /, and the directory is on another disk, you must verify (and add) that file system to the `/etc/checklist` file. Refer to the section "Adding to /etc/checklist" in this chapter for more information on `/etc/checklist` entries.

Check Available File System Space

You must check to see that you have enough disk space to perform the update. If you run out of disk space during the update the update program will stop. Everything loaded until then will be on your system and will be usable, but you may be missing some dependencies.

Your application's manual or the *Read-Me* that came with the product should give the necessary disk space. If not, call your local HP Sales and Service office to get the information before continuing.

Use the procedures in "Controlling Disk Use" in Chapter 5 for information on using the `df` command to check for free disk space.

If you run out of space during an update, you should list all the partitions and/or filesets not yet loaded. With this information, you can remove extra files off your system, and can come back to the update process and you will know what you still need to load.

Shut Down Your System

Shutting down your system brings your system to run-level `s`. To be in this run-level, all users other than the root user on the console are logged out.

For example, if you have users on the system, and want to allow them one minute to save their files and log off, you would enter:

```
shutdown 60
```

You will get the message:

```
Do you want to send your own message? (y or n):
Type your message followed by ctrl d
```

You can either type y and enter your own message (followed by a [CTRL] [D]), or type n. If you type n the system will send a default message to all users that are currently logged in.

For more information on shutting down your system, refer to Chapter 3, "System Startup and Shutdown".

Back Up Your File System

If you make a mistake while updating your system, you could possibly corrupt your file system. This could mean you have lost data on your file system. To prevent permanent data loss you should back up your file system before starting the /etc/update program. You should be able to recover all of your data *if you have adequately backed up your system.* See the "Backing Up and Restoring the File System" section of Chapter 5 ("Periodic System Administrator Tasks") for more information on how to back up your system.

Check Your File System

Now that your system is in system administration mode (from the shutdown procedure), check your file system by using the fsck command. For more on how to use the fsck command refer to Chapter 3, the section called "Power Fail or Disk Crash Recovery" and to Appendix A, Volume 2, "Using the fsck Command").

If fsck showed no problems with your file system, and your system is in system administration mode, you are ready to begin the update and optional software installation program.

Updating an HP-UX Cluster

If you are on an HP-UX Cluster, you must be on the root server of the cluster. This means that your media must be on a drive connected to the root server, and that you must execute the `update` command from the root server.

Updating or Installing Optional Software from Cartridge Tape

You should now be the superuser and should have your system prepared for the update (refer to the previous sections).

Note If you use NFS, you should update the NFS server first, then update all the clients.

Security To adhere to the evaluated trusted system configuration you cannot *update* to 6.5, but must *install* to 6.5. All trusted system software will still work correctly, but you will not have as good assurance as if you do a full install.

If your product is on flexible disks, go to the section called "Updating or Installing Optional Software from Flexible Disk".

If your product is on cartridge tape, follow these steps:

Step 1. Locate and Write-Protect the Cartridge Tape

Locate the write-protect mechanism (labeled "SAFE") on the top, rear, left-hand corner of the cartridge tape. *The arrow on the protect screw should point toward the word SAFE.* If it does not, use a coin or screwdriver to turn the protect screw such that the arrow points toward the word SAFE. Place this tape in the cartridge tape drive connected to your system with the SAFE label in the rear left hand corner. Only the BUSY and PROTECTED indicators should now be lit. The drive will begin a cartridge tape conditioning sequence that takes approximately two minutes. *Do not proceed until the busy light remains off.* (See Figure 4–31.)

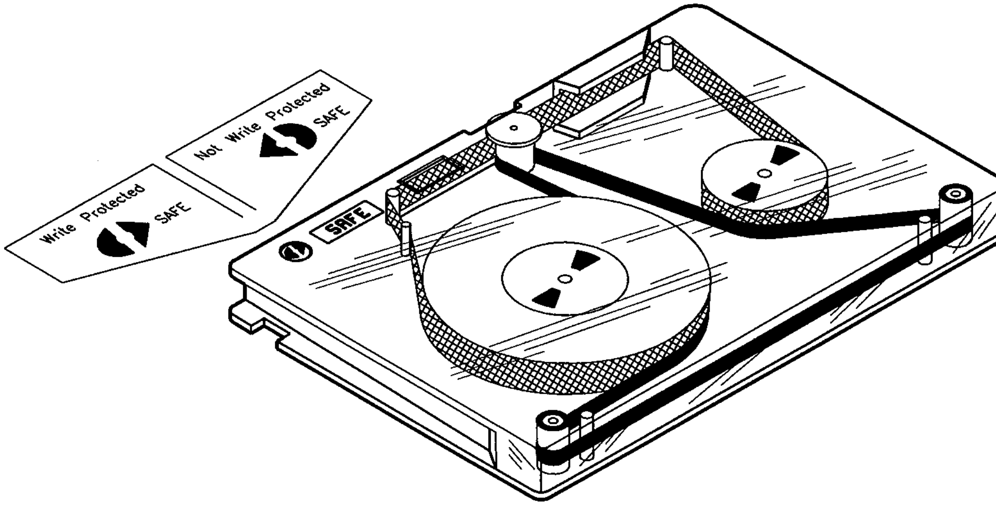


Figure 4–31. Write Protect Mechanism on Cartridge Tape

Step 2. Load the Update Tools

The update tools must be loaded if you are updating HP-UX. If you are adding an optional partition or loading optional software, you do not need to load the update tools; go to Step 3.

If you are updating your operating system, type:

```
lifcp -a /dev/source:GETTOOLS /tmp/gettools
chmod 700 /tmp/gettools
/tmp/gettools /dev/source
rm /tmp/gettools
```

where the device `/dev/`*source* is the device file name assigned to the same cartridge tape drive you just inserted the update media in. Note that this should be the same device file you found in the "Copy Source Information" section of "Preparing to Modify Your System". Executing `/tmp/gettools` causes any new tools related to the update process to be extracted from the media and put into your current file system. This could take from one to several minutes to complete.

Step 3. Start the /etc/update Program

Caution If you have not checked your free disk space to make sure you have enough space for the update, do so now. Not having enough disk space could leave your system unusable. Refer to the preparatory step discussing disk space.

Note If you have a non-HP terminal, you *MUST* execute the update program with the `-m` option (`/etc/update -m`). Also, if you are on an HP-UX Cluster, you must execute the `update` command from the root server.

Start the `/etc/update` program by typing in:

```
/etc/update
```

If you did not use the `-n` option, and forgot to shutdown the system, then `/etc/update` will exit (without rebooting) and give the following error message:

```
FATAL ERROR:  Too may processes running for single user state.
              Execute the shutdown command before performing the update.
```

You should execute the `shutdown` command before re-executing `update`.

If you have file systems in your `/etc/checklist` file that mount across NFS, the update program will now restart NFS and will re-mount the file systems. This will take up to 40 seconds. You will see a message similar to the following (followed by some network startup messages):

```
Restarting networking.
```

Mounting the NFS file systems means that your complete file system is in view for the update program. The update program will update all local files. It will not update remote files.

If `update` abruptly reboots, it is because NFS mounts are specified in `/etc/checklist`, but the system's hostname is not set. An error to that effect is written in `/tmp/update.log`. Be sure the hostname is set before attempting update. NFS mounts will not succeed without a valid hostname.

The screen now clears and shows a menu similar to that shown in Figure 4–32.

```
*****************************************************************
            HP-UX UPDATE UTILITY -- MAIN MENU
*****************************************************************
                     Select Choice

Source Device                          Destination Directory

/dev/update.src                        /
Major Number    = 4                    Major Number    = 0
Select Code     = 7                    Select Code     = 14
Bus Address     = 0                    Bus Address     = 0
Unit Number     = 0                    Unit Number     = 0
Volume Number   = 0                    Volume Number   = 0

DISPLAY contents of a new tape or disk
EXIT update
CHANGE source device
CHANGE destination directory



[        ] [        ] [  NEXT  ] [PREVIOUS]   [        ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 4-32. Main Utility Menu

From this menu you can change the destination directory (described in Step 4), change the source directory (described in Step 5), or display the contents of a new tape or disk (Step 6). If your source device and destination directory are correct, go to Step 6.

Step 4. Change the Destination Directory (Optional)

The destination directory defaults to the / directory on your hard disk. AXE and PE updates must always be loaded into the / destination directory. Some optional software and application programs can be loaded into other destination directories. *Refer to the installation instructions for the optional software.*

The most common reason to change the destination directory is to put the new product onto a mounted volume. This may be because your root volume doesn't have disk space, or just because you wish to organize your file system this way. Assuming the volume is already in your `/etc/checklist` file, you need only specify the directory.

If you wish to change the destination directory, use the NEXT and PREVIOUS softkeys to highlight the menu item `CHANGE destination device`, then press the SELECT softkey. You will now see the menu shown in Figure 4–33 (note that the address information shown here may not match what you see on your screen. The address information will be accurate for your directory.)

```
**************************************************************
        HP-UX UPDATE UTILITY -- CHANGE DIRECTORY MENU
**************************************************************

Current Destination Directory:  /
                      Major Number  = 0
                      Select Code   = 14
                      Bus Address   = 0
                      Unit Number   = 0
                      Volume Number = 0

Enter the ABSOLUTE path of the NEW destination directory.
>>
```

Figure 4–33. Change Destination Directory Menu

Enter the full (absolute) pathname of the new destination directory. If this is a valid directory you will be returned to the main menu shown in Figure 4–32 with the correct addressing information. If it is not a valid directory, you will again see the menu shown in Figure 4–33 and you must enter another directory.

The `/etc/update` program will check to see if you can change destination directories when it reads the contents file of the cartridge tape or flexible disk (Step 6).

Step 5. Change the Source Device (Optional)

The source device is the tape drive you will use to update your system or install optional software. If you do not wish the new software to be loaded from the device associated with `/dev/update.src`, or if each value of the Source Device is `-1`, then you need to change the source device.

The `/etc/update` program always looks for a file called `/dev/update.src`. If the file `/dev/update.src` does not exist on your file system then each value of the Source Device is `-1`. Your last update or installation left the `/dev/update.src` device file in your file system. Assuming you always use the same device for installing and updating your software (usually the case), the existence of this file makes updates much easier. The `/etc/backup` script and the `/etc/mkrs` program (used to create a recovery system) also use `/dev/update.src` as their defaults.

Use the [NEXT] and [PREVIOUS] softkeys to choose the `CHANGE source device` option on the main menu and press [SELECT]. You will see a menu similar to that shown in Figure 4–34.

```
*************************************************************
         HP-UX UPDATE UTILITY -- CHANGE DEVICE MENU
*************************************************************

Current Source Device:  /dev/update.src

                       Major Number  = -1
                       Select Code   = -1
                       Bus Address   = -1
                       Unit Number   = -1
                       Volume Number = -1

Enter the new Source Device file name below.  Include the
ABSOLUTE path for the device file name (eg. /dev/update.src).
If you don't have a device file name then enter [Return]
and you will be asked for the address of your Source Device.

NEW source device file name? >>
```

Figure 4-34. Change Source Device Menu

Either enter the name of the device file associated with your source device, or hit [Return] to enter the correct major number and address information. Each of these is described below.

Use the correct major number for your source device. The major numbers are explained in the section on setting up mass storage media in Chapter 4. Use the major number for a character device. Most of Hewlett-Packard's flexible disk drives and cartridge tape drives are called CS/80 or SS/80 drives and will have a major number of 4.

The address information is the minor number. The minor number has the format:

```
0xScBaUV
```

where 0x indicates the number is in base 16, Sc is the select code set on your interface card (generally 7 or 14), Ba is the bus address (check the address switch on the back of your drive), U is the unit number (0 or 1), and V is the volume (generally 0). You will be prompted for each item; enter the Source Device's select code, bus address, unit and volume numbers. You can enter these numbers either in decimal format, or can enter them in hexadecimal format, starting each entry with 0x.

Once you enter the name of the source device or the new address information, the `/etc/update` program will create the `/dev/update.src` file to match your device file, then return you to an updated menu similar to Figure 4–32. Notice that the new Source Device values are now shown. Check that the values shown on the menu are correct. It is possible that you could make a mistake while converting from hexadecimal to decimal format.

Step 6. Read the Source Tape

If you are sure your Source Device and Destination Directory are correct select the `DISPLAY contents of a new tape or disk` menu item. If you are not sure, the best way to find out is to exit the program and check your `/dev` directory for the correct information.

You will see the screen shown in Figure 4–35.

```
****************************************************
      HP-UX UPDATE UTILITY -- READING FILE MENU
****************************************************



Be sure media for a new partition is in source device.
Wait for the busy light to remain off and hit [Return]>>
```

Figure 4–35. Reading File Menu

Press Return. The `/etc/update` program will now read the update tape to get a list of available options, which takes a couple of minutes. While it is reading, the word `Reading` will appear under the screen heading.

If you see the message:

```
cannot read table of contents on /dev/update.src.
Press [Return] to continue>>
```

Then you have specified the wrong source device. Press Return. You will be returned to the main menu. Continue with Step 5.

The /etc/update program checks two things while reading the contents file:

- If you requested the no reboot option (/etc/update -n), the /etc/update program now checks to see if rebooting is required. If it is, then you will receive the message:

```
This product may not be loaded using the -n option of
update. Please exit and rerun update without the -n option.
Press [Return] to continue >>
```

  When you press the [Return] key the /etc/update program unloads the media from the source device and exits (returning you to your shell). Continue with Step 3. Do not use the -n option.

- Once /etc/update has checked the no reboot option it checks the destination device. If the software must be loaded under the root directory but you changed the destination directory to be something other than root (/), you will get the following message:

```
Current Destination Directory: xxxxx

You can not change the destination directory for this
product.  It must be updated relative to the / directory.
The destination directory has been changed back to /.
Enter [Return] to continue >>
```

  When you press the [Return] key, /etc/update will continue with the menu shown in Figure 4–37. The /etc/update program will use / for the destination directory.

When the update procedure has finished reading the table of contents, you will see either a menu similar to that shown in Figure 4–36, or that *similar* to that shown in Figure 4–37 will appear.

```
                   ****Update Kernel Configuration Notes****

    Update will configure an updated HP-UX kernel for you if you
    load both the "KERNEL" and "ACONFIG" filesets (in partitions
    "SYSCORE" and "MISC_UTILS") from this tape/disk.  This REQUIRES
    a configuration file, /etc/conf/dfile, which accurately reflects
    your current kernel's configuration.

    You will have the proper configuration file if:
       * you have never created a kernel, and use the default
         kernel from the last update
       * you used the reconfig program to create your kernel
       * you created /etc/conf/dfile to configure a kernel

    If you will be updating these filesets (normally the case),
    and you need to verify that /etc/conf/dfile matches your kernel,
    enter "y" below to get a shell.  If /etc/conf/dfile does not exist,
    update will simply install a default kernel.

    Do you want a shell so you can examine your dfile (y/n)?
```

Figure 4-36. Kernel Description File Screen

If you wish to escape to the shell, type y. You will see the following shell prompt:

```
(update sh)$
```

You can now look in your `/etc/conf/dfile` to see what kernel you currently have configured. Use CTRL-D to return to the update program; you will need to verify that you are ready to return to the update program.

If you have the proper configuration file and do not wish to check your dfile, type n.

Continue now with Step 7.

```
Partition "AXE-MU"                                     Media Number "1"
       ***********************************************
                HP-UX UPDATE UTILITY -- MAIN MENU
       ***********************************************
                          Select Choice



SYS_CORE         HP-UX Core system
TEXT             Text Processing and Manual Pages
SYS_TOOLS        Shells, UUCP/Mail and Commands
STARBASE         Starbase Graphics
WINDOW           Windows/9000 Window System
X11WINDOW        X Window System Version 11
PROG_LANGS       Compilers
MISC_UTILS       Misc Commands and Utilities
NLS              Native Language Support

Process ALL partitions
DISPLAY contents of a new tape or disk
EXIT update
CHANGE source device
CHANGE destination directory


[ MARK ] [ LOAD ] [ NEXT ] [PREVIOUS]   [      ] [SELECT ] [      ] [ QUIT ]
```

Figure 4–37. Partition Menu for Update on Cartridge Tape

Step 7. Choose Partitions

Note that you now have two additional softkeys. The softkeys are described in the section "General Discussion on the Update Process". The most efficient way to load from cartridge tape is to use the batch method. The batch method is described here. The interactive method is described in the section "Updating or Installing Optional Software from Flexible Disk".

- If you wish to load all partitions, use the [NEXT] and [PREVIOUS] softkeys to move to the `Process ALL partitions` menu item, and press the [MARK] softkey. This will put an asterisk (*) to the left of each partition.

Press the LOAD softkey. The update program will automatically load all partitions and filesets from this software product. You will be prompted when to change media, and will be returned to the menu in Figure 4–34 when everything is loaded.

It takes approximately one hour to load everything from one tape.

If you have other software products to load, select the `DISPLAY contents of a new tape or disk` menu item. The `/etc/update` program will now prompt you to insert new media so it can read the contents of the new media. Continue with the beginning of Step 10.

- If you wish to process only some partitions, use the NEXT and PREVIOUS softkeys to move to the `Load xxxxxx partition` menu item, and press the MARK softkey.

  When you press the MARK softkey, an asterisk (*) appears to the left of the item. If you wish to cancel a selection, press MARK again and the asterisk disappears. Pressing MARK by itself does not load any software.

  To load software, press the LOAD softkey after marking the necessary menu entries. All items marked are loaded.

Note If you are updating your system for operating system revision, *you must load all SYS_CORE items.* Failure to do so may result in a non-functional system. This means you must re-install HP-UX.

- If you wish to process only a part of a partition, use the NEXT and PREVIOUS softkeys to move to the `LOAD` *xxxxxxx* `partition` item, where *xxxxxxx* is the partition you want. Then press the SELECT softkey to move to the Fileset Menu (see Figure 4–38). Loading filesets is described in Step 8.

```
Partition:  "MISC_UTILS"            Media Number:  "1"
**********************************************************
           HP-UX UPDATE UTILITY -- FILE SET MENU
**********************************************************
                       select choice


ACONFIG  Kernel configuration files
ATRMINFO Non-HP terminal control files


Process ALL file sets
Use the [Quit] softkey to exit the menu
```

MARK	LOAD	NEXT	PREVIOUS		SELECT		QUIT

Figure 4-38.

Step 8. Choose Filesets

You will perform this step only if you wish to choose specific filesets. If you have chosen entire partitions or ALL partitions (Step 7), you do not need to read this step. Continue with Step 9.

The fileset menu is used to load one or more filesets in a partition. The partition is selected from the partition menu. On the new menu, the partition name appears on the upper left segment of the screen.

Use the [MARK] softkey to mark all filesets you wish loaded. If you exit this menu and go back to the Partition Menu, all of your marks made previously are preserved. When you press the [LOAD] softkey while in the Fileset Menu, all the marked menu items will be loaded, including those in other partitions that you previously marked and not yet loaded. Once a fileset is loaded, its mark is removed.

If you mark a subset of the filesets, then return to the Partition Menu, the menu will display a "+" to the left of the partially marked partition.

Step 9. Load the Filesets

When you have marked all the partitions and filesets you wish loaded onto your system, press the `LOAD` softkey from either the partition menu or from the fileset menu.

The `/etc/update` program now loads the filesets from the installation (or update) tape onto your file system.

For each file, you will see a message similar to the following:

```
Loading: /bin/cat
```

For each remote file (if your system has file systems mounted across NFS), you will see a message similar to the following:

```
Skipped Remote File: /usr/bin/lp
```

If a fileset is on another cartridge tape you will be see the message:

```
Fileset x is on media #N of xx, press [RETURN]
to unload media or enter s to skip:
```

Some filesets are *dependent* on other filesets. This means that for functions in one fileset to work, another fileset (the dependent fileset) must also be present. The `/etc/update` program will automatically load all dependencies, even if you did not select or mark them.

If one of these dependent filesets already exists on your system, you will be asked if you want to replace it. This may surprise you since you may not have marked it for loading. However, this is normal. You should answer `yes`.

Step 10. Continue Marking and Loading

Continue marking and loading filesets for all products you wish loaded (optional).

If you marked and loaded all filesets (or partitions), then you will now be at the Main Utility menu shown in Figure 4–32. You can now either exit the `/etc/update` program (Step 11) or you can install another optional software product (Step 6).

If you marked and loaded only selected filesets you will be returned to the Partition menu shown in Figure 4–37. Any partitions that were completely loaded will not be on the menu. If you go to the fileset menu, any filesets that were loaded will not be on the menu. All filesets and partitions that were not completely loaded will still be on the menu. You can now either exit the `/etc/update` program (Step 11), mark and load more partitions and filesets (Step 7), or install another optional software product (Step 6).

Step 11. Leave the Update Program

If you have loaded all the partitions you want, select the `EXIT update` option on the main menu. The program will inform you that it is unloading the media, which will take a few minutes.

After unloading the media, the following will happen:

1. The system builds a new kernel unless you specified the no reboot (`-n`) option.

   You will see output from the kernel creation program.

2. The system reboots unless you specified the no reboot (`-n`) option.

   It reboots with the new kernel built in the previous step. You will first see system shutdown messages, then see system reboot messages. Your old kernel will first be saved in \SYSBCKUP.

Caution Do not cycle power during the reboot or you may corrupt your file system.

---

3. If there are scripts to *customize* your system and SYS_CORE was loaded, the customize scripts run now.

If SYS_CORE was not loaded then the customize scripts were executed as the filesets were loaded.

This step could take 15 to 45 minutes to complete. *DO NOT cycle power.* Wait for the system to complete execution of the scripts.

4. The system reboots again after the customize scripts finish.
5. The system comes up and gives a login prompt.

Take out the cartridge tape and store it in a safe place.

Step 12. Check for Additional Information

Following this reboot process, you should log in and check the following files and directories:

- The `/etc/newconfig/Update_info` directory contains files with software product update information.
- The `/etc/newconfig` directory contains new versions of some files normally put into the `/etc` directory (for example, `rc`, `brc`, `backup`, `backupf`). These files are not replaced by the update procedure or by the customize scripts because you may have edited the original versions.
- Any `/etc/update` error messages will be in the `/tmp/update.log` file.
- The file called `/etc/newconfig/README` contains useful information about files in `/etc/newconfig`.

Follow instructions in the above files, and instructions in the *Installation Notes* or *Read-Me,* if supplied.

Updating or Installing Optional Software from Flexible Disk

You should now be the superuser and should have your system prepared for the update (refer to the previous sections).

If your product is on cartridge tape, go to the section called "Updating or Installing Optional Software from Cartridge Tape".

If your product is on flexible disks, follow these steps:

Step 1. Locate and Write-Protect the Flexible Disk

Locate the write-protect mechanism on the label end, and the back side of the disk. Slide the write protect shutter toward the label end of the disk, so you uncover the write protect hole. (See Figure 4-39.)

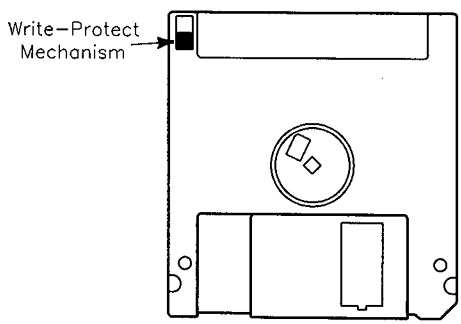


Figure 4-39. Write Protect Mechanism on Flexible Disk

Step 2. Load the Update Tools

The update tools must be loaded if you are updating HP-UX. If you are adding an optional partition or loading optional software, you do not need to load the update tools; go to Step 3.

If you are updating your operating system you must insert the appropriate disk of the SYS_CORE partition and copy the tools. Refer to the *Read Me* document shipped with your media for the correct disk number. Insert the disk and type:

```
lifcp -a /dev/source:GETTOOLS /tmp/gettools
chmod 700 /tmp/gettools
/tmp/gettools /dev/source
rm /tmp/gettools
```

where the device `/dev/`*source* is the device file name assigned to the same flexible disk drive you just inserted the update media in. Note that this should be the same device file you found in the "Copy Source Information" section of "Preparing to Modify Your System". Executing `/tmp/gettools` causes any new tools related to the update process to be extracted from the media and put into your current file system. This could take from one to several minutes to complete.

Step 3. Start the /etc/update Program

Caution If you have not checked your free disk space to make sure you have enough space for the update, do so now. Not having enough disk space could leave your system unusable. Refer to the preparatory step discussing disk space.

Note If you have a non-HP terminal, you *MUST* execute the update program with the `-m` option (`/etc/update -m`). Also, if you are on an HP-UX Cluster, you must execute the `update` command from the root server.

Insert the first disk from the partition or from the software product and start the `/etc/update` program by typing in:

```
/etc/update
```

If you did not use the `-n` option, and forgot to shutdown the system, then `/etc/update` will exit (without rebooting) and give the following error message:

```
FATAL ERROR:  Too may processes running for single user state.
              Execute the shutdown command before performing the update.
```

Execute the `shutdown` program before re-executing `update`.

If you have file systems in your `/etc/checklist` file that mount across NFS, the update program will now restart NFS and will re-mount the file systems. This may take up to 40 seconds. You will see a message similar to the following (along with some networking startup messages):

```
Restarting networking.
```

Mounting the NFS file systems means that your complete file system is in view for the update program. The update program will update all local files. It will not update remote files.

If `update` abruptly reboots, it is because NFS mounts are specified in `/etc/checklist`, but the system's hostname is not set. An error to that effect is written in `/tmp/update.log`. Be sure the hostname is set before attempting update. NFS mounts will not succeed without a valid hostname.

Then the screen will clear and a menu similar to that shown in Figure 4–40 will appear.

```
*****************************************************************
             HP-UX UPDATE UTILITY -- MAIN MENU
*****************************************************************
                        Select Choice

Source Device                         Destination Directory

/dev/update.src                       /
Major Number    = 4                   Major Number    = 0
Select Code     = 7                   Select Code     = 14
Bus Address     = 0                   Bus Address     = 0
Unit Number     = 0                   Unit Number     = 0
Volume Number   = 0                   Volume Number   = 0

DISPLAY contents of a new tape or disk
EXIT update
CHANGE source device
CHANGE destination directory




[        ] [        ] [  NEXT  ] [PREVIOUS]   [        ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 4-40. Main Utility Menu

From this menu you can change the destination directory (described in Step 4), change the source directory (described in Step 5), or display options for a new partition (Step 6). If your source device and destination directory are correct, go to Step 6.

Step 4. Change the Destination Directory (Optional)

The most common reason to change the destination directory is if you wish to put the new product onto a mounted volume. This may be because your root volume doesn't have disk space, or just because you wish to organize your file system this way. Assuming the file system is already in the `/etc/checklist` file, you need only specify the directory.

The destination directory defaults to the / directory on your hard disk. AXE and PE updates must always be loaded into the / destination directory. Some optional software and application programs can be loaded into other destination directories. *Refer to the installation instructions for the optional software.*

If you wish to change the destination directory use the [NEXT] and [PREVIOUS] softkeys to highlight the menu item `CHANGE destination device`, then press the [SELECT] softkey. You will now see the menu shown in Figure 4–41 (note that the address information shown here may not match what you see on your screen. The address information will be accurate for your directory.)

```
*************************************************************
         HP-UX UPDATE UTILITY -- CHANGE DIRECTORY MENU
*************************************************************

Current Destination Directory:  /
                      Major Number  = 0
                      Select Code   = 14
                      Bus Address   = 0
                      Unit Number   = 0
                      Volume Number = 0

Enter the ABSOLUTE path of the NEW destination directory.
>>
```

Figure 4–41. Change Destination Directory Menu

Enter the full (absolute) pathname of the new destination directory. If this is a valid directory you will be returned to the main menu shown in Figure 4–40 with the correct addressing information. If it is not a valid directory you will again see the menu shown in Figure 4–41. You must enter another directory. The `/etc/update` program will check to see if you can change destination directories when it reads the contents file of the flexible disk (Step 6).

Step 5. Change the Source Device (Optional)

The source device is the tape drive or flexible disk drive you will use to update your system or install optional software. If you do not wish the new software to be loaded from the device associated with `/dev/update.src`, or if each value of the Source Device is `-1`, then you need to change the source device.

The `/etc/update` program always looks for a file called `/dev/update.src`. If the file `/dev/update.src` does not exist on your file system then each value of the Source Device is `-1`. Your last update or installation left the `/dev/update.src` device file in your file system. Assuming you always use the same device for installing and updating your software (usually the case), the existence of this file makes updates much easier. The backup program and the program to create a recovery system also use `/dev/update.src` as their defaults.

Use the [NEXT] and [PREVIOUS] softkeys to choose the `CHANGE source device` option on the main menu and press [SELECT]. You will see a menu similar to that shown in Figure 4–42.

```
*************************************************************
          HP-UX UPDATE UTILITY -- CHANGE DEVICE MENU
*************************************************************

Current Source Device:  /dev/update.src

                      Major Number  = -1
                      Select Code   = -1
                      Bus Address   = -1
                      Unit Number   = -1
                      Volume Number = -1

Enter the new Source Device file name below.  Include the
ABSOLUTE path for the device file name (eg. /dev/update.src).
If you don't have a device file name then enter [Return]
and you will be asked for the address of your Source Device.

NEW source device file name? >>
```

Figure 4–42. Change Source Device Menu

Either enter the name of the device file associated with your source device, or hit [Return] to enter the correct major number and address information. Each of these is described below.

Use the correct major number for your source device. The major numbers are explained in the section on setting up mass storage media in Chapter 4. Use the major number for a character device. Most of Hewlett-Packard's flexible disk drives are called CS/80 or SS/80 drives and will have a major number of 4.

The address information is the minor number. The minor number has the format:

```
0xScBaUV
```

where 0x indicates the number is in base 16, Sc is the select code set on your interface card (generally 7 or 14), Ba is the bus address (check the address switch on the back of your drive), U is the unit number (0 or 1), and V is the volume (generally 0). You will be prompted for each item; enter the Source Device's select code, bus address, unit and volume numbers. You can enter these numbers either in decimal format, or you can enter them in hexadecimal format, starting each entry with 0x.

Once you enter the name of the source device or the new address information, the /etc/update program will create the /dev/update.src file to match your device file, then return you to an updated menu similar to Figure 4–40. Notice that the new Source Device values are now shown. Check that the values shown on the menu are correct. It is possible that you could make a mistake while converting from hexadecimal to decimal format.

Continue with Step 6.

Step 6. Read the Source Disk

If you are sure your Source Device and Destination Directory are correct select the DISPLAY contents of a new tape or disk menu item. If you are not sure, the best way to find out is to exit the process and check your /dev directory for the correct information.

You will see the screen shown in Figure 4–43. Continue with Step 7.

```
****************************************************
      HP-UX UPDATE UTILITY -- READING FILE MENU
****************************************************



Be sure media for a new partition is in source device.
Wait for the busy light to remain off and hit [Return]>>
```

Figure 4–43. Reading File Menu

Step 7. Choose Partitions

All flexible disks in a partition are grouped by color-code labels. Refer to the appendix "Partitions and Filesets" for a detailed discussion.

Insert any one of the disks from a partition, and press [Return]. The /etc/update program will now read the contents of the partition from the flexible disk. While it is reading, the word Reading will appear under the screen heading.

If you see the message:

```
cannot read table of contents on /dev/update.src.
Press [Return] to continue>>
```

Then you have specified the wrong source device. Press [Return]. You will be returned to the main menu. Continue with Step 5.

The /etc/update program checks two things while reading the contents file:

- If you requested the no reboot option (/etc/update -n), the /etc/update program now checks to see if rebooting is required. If it is, then you will receive the message:

  ```
  This product may not be loaded using the -n option of
  update. Please exit and rerun update without the -n option.
  Press [Return] to continue >>
  ```

  When you press the [Return] key the /etc/update program unloads the media from the source device and exits (returning you to your shell). Continue with Step 3. Do not use the -n option.

- Once /etc/update has checked the no reboot option it checks the destination device. If the software must be loaded under the root directory but you changed the destination directory to be something other than root (/), you will get the following message:

  ```
  Current Destination Directory: xxxxx

  You can not change the destination directory for this
  product. It must be updated relative to the / directory.
  The destination directory has been changed back to /.
  Enter [Return] to continue >>
  ```

  When you press the [Return] key, /etc/update will continue with the menu shown in Figure 4-45. The /etc/update program will use / for the destination directory will be /.

When the update procedure has finished reading the table of contents, you will see either a menu similar to that shown in Figure 4–44, or that *similar* to that shown in Figure 4–45 will appear.

```
                    ****Update Kernel Configuration Notes****

    Update will configure an updated HP-UX kernel for you if you
    load both the "KERNEL" and "ACONFIG" filesets (in partitions
    "SYSCORE" and "MISC_UTILS") from this tape/disk.  This REQUIRES
    a configuration file, /etc/conf/dfile, which accurately reflects
    your current kernel's configuration.

    You will have the proper configuration file if:
       * you have never created a kernel, and use the default
         kernel from the last update
       * you used the reconfig program to create your kernel
       * you created /etc/conf/dfile to configure a kernel

    If you will be updating these filesets (normally the case),
    and you need to verify that /etc/conf/dfile matches your kernel,
    enter "y" below to get a shell.  If /etc/conf/dfile does not exist,
    update will simply install a default kernel.

    Do you want a shell so you can examine your dfile (y/n)?
```

Figure 4–44. Kernel Description File Screen

If you wish to escape to the shell, type y. You will see the following shell prompt:

```
(update sh)$
```

You can now look in your `/etc/conf/dfile` to see what kernel you currently have configured. Use CTRL-D to return to the update program; you will need to verify that you are ready to return to the update program.

If you have the proper configuration file and do not wish to check your dfile, type n.

```
Partition:  "MISC_UTILS"              Media Number:  "1"
***********************************************************
         HP-UX UPDATE UTILITY -- FILE SET MENU
***********************************************************
                     select choice


ACONFIG  Kernel configuration files
ATRMINFO Non-HP terminal control files







Process ALL file sets
DISPLAY contents of a new tape or disk
EXIT update
CHANGE source device
CHANGE destination directory



 MARK   LOAD   NEXT   PREVIOUS     HELP   SELECT          QUIT
```

Figure 4-45. Fileset Menu for Update on Flexible Disk

Note that you now have two additional softkeys: MARK and LOAD. The softkeys are described in the section "General Discussion on the Update Process". The MARK and LOAD softkeys can be used to process the filesets as a batch. The SELECT softkey can be used to choose, and immediately load, particular filesets interactively. Either method can be used to choose the `Process ALL Filesets` menu item. These are described below.

Note

If you are updating your system for an operating system revision, *you must load the SYS_CORE partition and all the filesets in SYS_CORE.* Failure to do so may result in a non-functional system. This means you must re-install HP-UX.

Continue now with Step 8.

Step 8. Choose Filesets

The fileset menu is used to load one or more filesets in a partition. The partition is a set of flexible disks with the same color label. On the menu, the partition name appears on the upper left segment of the screen.

You have the following choices:

- If you wish to process all filesets in this partition, use the `NEXT` and `PREVIOUS` softkeys to highlight the `Process ALL file sets` menu item. You can then either:
  - press the `MARK` softkey, then press the `LOAD` softkey. This will put an asterisk (*) next to all the filesets, then will copy all the filesets onto your system.
  - press the `SELECT` softkey. This will copy all the filesets onto your system.

  You will be prompted to change media.
- If you wish to process only selected filesets, either:
  - use the `MARK` softkey to mark all filesets you wish loaded. When all desired filesets are marked (they have an asterisk beside them), press the `LOAD` softkey. Once a fileset is loaded, its mark is removed.
  - use the `SELECT` softkey to immediately copy the fileset from the flexible disk to your hard disk.

  You will be prompted when to change media.

  For each file, you will see a message similar to the following:

  ```
  Loading: /bin/cat
  ```

For each remote file (if you have file systems mounted across NFS) you will see a message similar to the following:

```
Skipped Remote File: /usr/bin/lp
```

You will be returned to the menu shown in Figure 4–45. Any filesets that were completely loaded will not be on the menu. You can now either exit the `/etc/update` program (Step 10), mark and load more filesets (beginning of Step 8), or install another partition or optional software product (Step 9).

If a fileset is on another flexible disk, you will be see the message:

```
Fileset x is on media #N of xx, press [RETURN]
to unload media or enter s to skip:
```

Some filesets are *dependent* on other filesets. This means that for functions in one fileset to work, another fileset (the dependent fileset) must also be present. The `/etc/update` program will automatically load all dependencies, even if you did not select or mark them.

Step 9. Continue Processing Partitions and Filesets

Continue processing disks for all partitions and/or products you wish loaded (optional). To load another partition onto your system, insert another disk of different color, use the [NEXT] and [PREVIOUS] softkeys to highlight the `DISPLAY contents of a new tape or disk`, and press the [SELECT] softkey. You will see a menu similar to that in Figure 4–45. Continue with step 8.

When you are finished loading all the partitions, continue with Step 10.

Step 10. Leave the Update Program

If you have loaded all the partitions you want, select the `EXIT update` option. The program will inform you that it is unloading the media.

After unloading the media, the following will happen:

1. The system builds a new kernel unless you specified the no reboot (`-n`) option.

   You will see messages from the program that builds the kernel.

2. The system reboots unless you specified the no reboot (`-n`) option.

It reboots with the new kernel built in the previous step. You will first see system shutdown messages, then see system reboot messages. Your old kernel will first be saved in \SYSBCKUP.

Caution Do not cycle power during the reboot or you may corrupt your file system.

---

3. If there are scripts to *customize* your system and SYS_CORE was loaded, the customize scripts run now.

   If SYS_CORE was not loaded, the customize scripts were executed as the filesets were loaded.

   This step may take from 15 to 45 minutes to complete. *DO NOT cycle power.* Wait for the system to complete execution of the customize scripts.
4. The system reboots again after the customize scripts finish.
5. The system comes up and gives a login prompt.

If `/usr/bin` *is mounted across NFS* then you need to manually update the files you have in the local `/usr/bin`. Some of these files are required for bootup, before the NFS file system is mounted. Therefore you must do the following:

1. If you have not already done so, update the NFS server.

   If the NFS server does not have the updated files, the rest of this procedure will not put the correct files on your file system.
2. Unmount the NFS-mounted file system.

   ```
   umount /usr/bin
   ```
3. Copy the files in `/usr/bin` to a file.

   ```
   cd /
   find /usr/bin -print > /tmp/files_to_get
   ```

   This will list all files required for bootup from `/usr/bin` in a file called `/tmp/files_to_get`.
4. Mount the NFS volumes by typing:

   ```
   mount -t nfs
   ```

5. Copy the files from the remote system into `/tmp` by typing:

```
cp `cat /tmp/files_to_get` /tmp
```

6. Unmount the NFS-mounted file system by typing:

```
umount /usr/bin
```

7. Copy the files from `/tmp` to `/usr/bin`. To do this if you are using `/bin/sh` as a shell, you would type:

```
cd /usr/bin
for file in *
do
   cp /tmp/$file $file
done
```

   Be sure all copies succeed before proceeding to Step 7.

8. (Optional) Clean up `/tmp` by typing:

```
rm `cat /tmp/files_to_get`
```

9. Re-mount `/usr/bin` by typing:

```
mount -t nfs
```

Step 11. Check for Additional Information

Following this reboot process, you should log in and check the following files and directories:

- The `/etc/newconfig/Update_info` directory contains files with software product update information.
- The `/etc/newconfig` directory contains new versions of some files normally put into the `/etc` directory (for example, `rc`, `brc`, `backup`, `backupf`). These files are not replaced by the update procedure or by the customize scripts because you may have edited the original versions.
- Any `/etc/update` error messages will be in the `/tmp/update.log` file.
- The file called `/etc/newconfig/README` contains useful information about files in `/etc/newconfig`.

Follow instructions in the above files, and instructions in the *Installation Notes* or *Read-Me*, if supplied.

Note Do not remove any files from the `/system` directory. This directory contains scripts used to customize your file system.

Step 12. Create a New Recovery System

Follow instructions in the section "Creating and Using a Recovery System" in this chapter.

Caution Do not use a pre-6.5 recovery system on a 6.5 system. The old `fsck` program can damage your system.

Contents

5

Periodic System Administrator Tasks

You have invested a significant amount of time installing HP-UX and creating file systems; it is important to maintain the file system to ensure the integrity of the system for your users. Simple daily checks and procedures, and correcting problems before they become catastrophic, will save you from remaking the entire system. This chapter discusses both periodic system maintenance tasks and other tasks you will need to perform as your system grows and changes.

All the file system maintenance tasks are important. Here is a reminder of what needs to be done:

- System shutdown.

  Every time you shut down the system you should follow the procedure in the "System Startup and Shutdown" chapter. *Do not simply shut off the power!*

- File system consistency check (`fsck`).

  As shipped, `fsck` is run automatically if an improper shutdown is detected at bootup. Whenever you suspect there is a problem with your file system, you should run `fsck` interactively. Refer to Appendix A, "Using the FSCK Command" for a description.

- Check and understand disk usage.

  Unused and large files use space on your file system. You should check, and remove or archive, large and unused files weekly, or whenever you are running low on space. Follow the procedure in this chapter, the section "Controlling Disk Use".

- Recovery system.

  After you install your HP-UX operating system, you should create a recovery system using the `mkrs` command. You need to remake a recovery system if you change your swap size. Follow the procedure in Chapter 4, "Customizing the HP-UX System".

- Back up your system.

  Your system should be backed up periodically to prevent loss of work in case of a catastrophic system failure. Follow the guidelines and procedures in the section "Backing Up and Restoring the File System" in this chapter.

Adding/Removing Users

The material in this section covers only the software configuration aspects of adding and removing a user account to or from the HP-UX system. If you will be adding a terminal for this user, you must also install the terminal and perform some associated configuration tasks before the user can log in to the system; refer to the section "Adding Peripheral Devices" in Chapter 4.

Each user is defined by an entry in `/etc/passwd`. To protect the security of the system, `/etc/passwd` should be owned by `root` (the superuser) and no one, not even the superuser, should have write permission to the file. If you, as the superuser, want to modify `/etc/passwd`, temporarily change the permission using `chmod`, modify `/etc/passwd`, then change the permission of `/etc/passwd` back. As the superuser, you could also use the enforced write command of your editor (for example, `:w!` in `vi`) to override the file protection. Actually, users who are not the superuser should not be allowed to write to *any* of the files contained in the `/etc` directory (or to the directory itself). (Refer to the section, "System Protection and Security", in Chapter 4.)

There are two methods to add a new user:

- Using reconfig

  This is the easiest way to add a new user. You must use this method if you are running a trusted system. Do not use this method if you need to assign specific user IDs or if you wish to create a home directory somewhere other than `/users`.

Security — If you are running a trusted system you must use the `reconfig` program to add new users.

- Using the manual method

  Generally you should use the reconfig method to add users. Use this method if you need to assign specific user IDs or if you wish to create a home directory somewhere other than `/users`.

Adding a New User (Using Reconfig)

Even if your HP-UX system has only a single-user license you probably need several different login names. You might want a different login for each person that accesses the computer or for each task you want to do.

To add a new user, perform the following steps:

1. Log in as the superuser, root.
2. Start the `reconfig` program by entering the command:

   ```
   /etc/reconfig
   ```

   The main menu of `reconfig` will appear on your screen, as shown in Figure 5–1.

```
**************************************************
            RECONFIG -- MAIN MENU
**************************************************

User Configuration
Workstation Configuration
Cluster Configuration

[        ] [        ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 5–1. Reconfig Main Menu

3. Using the NEXT and PREVIOUS softkeys, move the cursor to the `User Configuration` menu item and press the SELECT softkey (the cursor is probably already on this menu item).

   You will see the `User Configuration Menu` on your screen (Figure 5–2).

```
**************************************************
        RECONFIG -- USER CONFIGURATION MENU
**************************************************

Add a new user
Delete a user



|        |        |  NEXT  |PREVIOUS|   |  HELP  | SELECT |PREVMENU|        |
```

Figure 5–2. Reconfig User Configuration Menu

4. Using the NEXT and PREVIOUS softkeys, move the cursor to the `Add a new user` menu item and press the SELECT softkey (the cursor is probably already on this menu item).

   The form to add a new user will appear on your screen (Figure 5–3).

```
*****************************************************
          RECONFIG -- ADD A NEW USER
*****************************************************

User Name:
Group Name:             root
Default Shell:          /bin/sh
Windows at Login:       no
Password:               no




User Name>>


[RESTORE] [ ] [ ] [ ]   [ HELP ] [ ] [PREVMENU] [ ]
```

Figure 5-3. Add a New User Menu

If you wish for more information on the `reconfig` main menu, refer either to the help information in the `reconfig` program (by pressing the [HELP] softkey) or to the section "Using Reconfig" in this chapter.

5. Type the new user's desired login name, then press the [Return] key.

   This is the name your system uses to identify the user. This name will be the login name (name used to identify the user when logging into the system), and will be the name of the HOME directory for the new user. The user name usually represents the persons name or function. The `User Name` might be the person's initials, first name, or last name. If you want a separate login for using an application such as HP PCDS, you might use a user name of `pcds`.

The user name must be limited to 8 characters, the first character must be an alphabetic character, at least one character must be lowercase, and the name must not contain blank spaces. Also, the name you choose must not already exist on the system.

When you press the [Return] key the `reconfig` program will check for errors. If it finds an error you will see a message describing the error, which begins with the word `INVALID`:

6. Choose the Group Name.

   This is the name of the group the new user will belong to. A list of existing groups is displayed on the form. Figure 5–4 shows what you might see on the bottom of your screen.

```
The groups currently defined on your system are:
  root          other          bin            sys
  adm           daemon         mail           rje
  guest         news           users

Group Name >> root

[RESTORE] [       ] [ NEXT ] [PREVIOUS]  [ HELP ] [SELECT] [PREVMENU] [       ]
```

Figure 5–4. Choosing in a Group Name

You can choose one of the existing groups, or you can type a new group name. To choose one of the existing groups use the [NEXT] and [PREVIOUS] softkeys to cycle through the group names on the list. Press the [SELECT] softkey when the correct group name appears.

If you hit the [SELECT] softkey (or the [Return] key) without typing a group name, you select the default (displayed) group name. The default group name is the first group name `reconfig` found in the `/etc/group` file. Note that this may not be the best group name! In the example shown, `root` is the default group. *Your normal user should not belong to the* `root` *group.* Most users should belong to the `users` or `other` group.

If you wish to use a group name that is not on the list, type the group name and press the [Return] key. If you accidentally press the [SELECT] softkey `reconfig` will use the value that was displayed, not the value you typed. Also, when typing the group name, you do not need to erase or type over the parts of the displayed value that are longer than the name you type; only the letters you type will be accepted an the group name. When you finish adding the new user, the `reconfig` program will add the new group to the file `/etc/group`. Once you have created a new group name it will also be displayed on the form.

Using the `reconfig` program you can assign only one group to the user.

When you press the SELECT softkey (or the Return key) the `reconfig` program will check to see that the group name is from one to eight characters and that it begins with an alphabetic character.

7. Choose the new user's `Default Shell`.

   The **shell** is a program that provides an interface between the user and the operating system. The user enters commands and accomplishes work through the shell. Six shells are provided from Hewlett-Packard, but you may use your own program or application.

   A list of existing shells is displayed on the form. Figure 5–5 shows what you might see on the bottom of your screen.

```
The available shells are:

/bin/sh     /bin/ksh     /bin/csh     /bin/pam     /bin/rsh     /bin/rksh


            Default Shell? >> /bin/sh

RESTORE |        | NEXT | PREVIOUS |  HELP | SELECT | PREVMENU |
```

Figure 5–5. Choosing a Shell

   Use the NEXT and PREVIOUS softkeys to scroll to your choice of shells. Press SELECT to make your choice. You can also enter any program not shown here if you wish the user to have access to only the program. You must type it in as a full path name.

8. Choose the windowing system for the new user.

A list of existing window systems is displayed on the form. Figure 5-6 shows what you might see on the bottom of your screen.

```
The choices for the window system to start are:

None  X11Windows WINDOWS/9000


  Windows at Login? >> None

RESTORE |        | NEXT | PREVIOUS |  HELP | SELECT | PREVMENU |
```

Figure 5-6. Choosing a Window

Use the NEXT and PREVIOUS softkeys to scroll to your choice of windows. Press SELECT to make your choice.

If you choose `X11Windows` or `WINDOWS/9000` the window system will start automatically when the user logs in; the user's shell will appear in a window. Your operating system must already be configured for the windowing system you choose. To be configured for X11Windows you must have the `lan` kernel driver in your kernel and you must have the AX11RENV fileset loaded. To be configured for HP Windows/9000 you must have the `ptymas` and `ptyslv` kernel drivers and you must have the AWINDOW fileset loaded. Check the `/etc/filesets` directory to see if you have the proper fileset. Refer to the section called "Creating a New Operating System" for information on adding kernel drivers.

If you choose `none` the user's shell will appear on the screen, but not in a window.

9. Decide if you want this login to use a password. Using a password is one method of securing your computer system and the data it contains.

   This prompt requires a yes or no answer (`y` or `yes`, `n` or `no`, followed by [Return]).

   If you choose `yes` the new user will be forced to enter a password during the first login session.

   If you choose `no` the user is not prompted for a password. Anyone who knows the user's login name will be able to enter the computer system. Any user can override a `no` choice by using the `passwd` command while logged in.

   Pressing the [Return] key without typing in `yes` or `no` selects the value shown.

10. You will now see the message:

    ```
    Are the above values ok (y or n)?>>
    ```

    Check the values. If one or more value is incorrect, then type `n` and [Return]. If you type `n` `reconfig` will start again starting at step 4 showing the values currently set.

    If the values are correct, then type `y` and press the [Return] key.

    The `reconfig` program now performs the following tasks:

    - updates the `/etc/passwd` file with information about the new user.
    - updates the `/etc/group` file with the new user name, and a new group name if you created a new group.
    - creates the HOME directory in `/users`. The HOME directory will have the same name as the USER NAME.
    - puts the default environment files associated with the chosen shell into the new user's HOME directory.

You will now see the message:

```
Next User Name? >>
New user <user_name> added to group <group_name>.
Press [PrevMenu] if there are no more users to add.
```

where `<user_name>` and `<group_name>` are the user and group names you just choose.

11. Exit or continue.

    If you wish to add another user, continue with step 4. If you do not need to add more users, press the PREVMENU softkey. The main menu of `reconfig` will appear on your screen.

12. Set up a terminal for the user if necessary.

    Skip this step if the new user will use an existing remote terminal, or will use the console.

    If the new user will have a separate terminal, go to the entry in `reconfig` for `Add remote terminal support`. See the section called "Setting Up and Removing a Remote Terminal" for information on setting up a terminal.

13. Press the QUIT softkey to exit the `reconfig` program and return to your shell.

Example:

If you want to add one user with the following information:

- Name is Ryan Asp
- Member of group `aspies` (this group does not exist on your system yet)
- Needs X11 Windowing system
- Wants PAM for a shell
- Needs a password

Perform the following steps:

1. Type `reconfig`
2. Because the "User Configuration" option is the first option in the main menu it is already highlighted. Press the [SELECT] softkey; the "User Configuration" menu will appear on your screen.
3. Because the "Add a new user" option is the first option in the main menu it is already highlighted. Press the [SELECT] softkey; the "ADD A NEW USER" form will appear on your screen.
4. Enter the desired `User Name` (for example, `ryan`) and press the [Return] key.
5. Type the new group name, `aspies`, and press the [Return] key.
6. Choose the shell, `/bin/pam` and press the [SELECT] softkey.
7. Choose the `X11Windows` option and press the [SELECT] softkey.
8. Enter "Yes" followed by [Return] to choose the `Password?` option.
9. Enter "Yes" followed by [Return] in response to `Are the above values OK (y or n)?`
10. Press the [PREVMENU] softkey to exit the form.

Adding a New User (the Manual Method)

The easiest way to add and remove users is with the `reconfig` program. The step-by-step method described here is provided for those of you who wish to understand the details of what needs to be done. This method should also be used if you must assign specific user ID numbers; for example, if you need to create the same `password` file for all systems on a network.

1. Log in to the system as the superuser `root`.
2. Create the `/etc/passwd` entry. Details for creating this entry are provided later in this section.
3. Add the user to a group in the `/etc/group` file.

If you are using HP-UX's group access capability, you may want to add the user to a group or change the group ID associated with the user's files. The user's group must exist in `/etc/group` and the user must be made a member of that group before the `chgrp` command can be used to change the group ID associated with the user's files. A user can be in multiple groups. Some systems currently do not support this feature, although they will in the future. For details on these operations, refer to the "Creating Groups/Changing Group Membership" section in this chapter, and to the *chgrp(1)*.and *group(5)* entries in the *HP-UX Reference*.

4. Create a login directory for the user with the `mkdir` command by typing:

   mkdir /users/*user_name*

   where *user_name* is the new user's name and the full path namebreak (`/users/`*user_name*) matches the *login_directory* field of the user's `/etc/passwd` entry. The following section, "Creating the /etc/passwd Entry" describes the fields in `/etc/passwd`.

5. Create the login initialization file for the user (the name of the login file depends on the default shell; for example, `.profile` for the Bourne shell). If the login initialization file exists in a user's login directory, the shell attempts to execute that file at the end of the login process, but before the user receives the first prompt. This file typically contains shell commands and environment variable definitions which customize the user's environment and/or automatically run one or more programs. Examples found in `/etc/d.*` may be customized to fit the general needs of users on your system, then copied to their home directory.

6. Create or customize other default files for the user.

   - If the file `.exrc` exists in the user's login directory, it is used to map terminal characteristics and key definitions for the `vi` editor. An example exists in `/etc/d.exrc`.

   - If the file `.mailrc` exists in the user's login directory, it is used to map mail functions whenever the user reads mail with the `mailx` command.

7. Change the ownership on the new user's directory by typing:

   chown *user_name* **/users/***user_name*

   where *user_name* is the user and `/users/`*user_name* is his login directory.

8. Change the group on the new user's directory by typing:

   `chgrp` *group_name* `/users/`*user_name*

   where *group_name* is the user's group and `/users/`*user_name* is his login directory.

9. Change the ownership on the new user's files by typing:

   `chown` *user_name* `/users/`*user_name*`/.[a-z]*`

   where *user_name* is the user. The specification `/users/`*user_name*`/.[a-z]*` matches all the files in the user's directory that begin with a period, are followed by a lower-case letter, and then anything else.

   Depending on the needs at your installation, consider using the `chmod` command to change the protection mode of the user's login directory and files. A commonly used mode value is 0755 which provides read, write, and execute (search) permission for the file's owner while providing only read and execute (search) permission for all others.

Creating the /etc/passwd Entry

If this is the first time you are following this procedure, make a copy of the original `/etc/passwd` file that was shipped with your system before continuing; this useful precaution takes only a moment. To copy the file, type:

```
cp /etc/passwd /etc/passwd.old
```

where `/etc/passwd.old` will be your unmodified (original) copy of the file.

Next, use the text editor of your choice (such as `ed`, `vi`, or `ex`) to edit the `/etc/passwd` file. Add a line to the file describing the new user. The new line must have the form:

user_name`::`*user_id:group_id:comment:login_directory:command*

where:

:	The colon character (:) is used to delimit the various fields in the entry.
user_name	The user's login name consists of 1 to 8 lowercase letters, at least one number or special character, and any other characters you desire.

::

The double colon represents an empty password field. Passwords and the `passwd` command are discussed later in this section.

user_id

The real user ID is a *unique* integer value that the system uses to identify the user. If the real user ID is 0, then that user has superuser capabilities. As the system was shipped to you, the real user ID 0 is associated with the user `root`. By convention, the values 1 through 99 are reserved for system use. Therefore, pick any unused number greater than 99, but less than 60000, for this field. User IDs greater than 59999 will be invalid.

Note There should be only one entry per real user ID; the user whose real user ID is 0 should be named root.

Security In a trusted system, it is *required* that there is only one entry per real user ID.

---

group_id

The real group ID is an integer value shared by all members of the same group. This entry corresponds with the group entry in `/etc/group`; refer to the "Creating Groups/Changing Group Membership" section in this chapter for details.

comment

The comment is a word or phrase that identifies the user or specifies the reason for the entry. Typically, this field contains the user's full name and other information such as his location or phone number. The comment is printed on the banner page of spooled `lp` jobs.

login_directory

This is the absolute path name of the user's login directory. The login directory becomes the user's working directory when he logs in. The directory need not exist when the entry to `/etc/passwd` is made. However, the directory must exist before the user can log in.

command This field contains the name of a single command to be executed for the user at login; this should be an absolute pathname. Typically, `/bin/sh` (or `/bin/csh` or `/bin/pam`) is placed in this field to invoke the shell (or C shell or PAM) for the user. However, the name of any executable program or command may be placed in this field. The command can be either a compiled program or a shell script but no arguments to the command or script should be supplied. If the command field is left blank, `/bin/sh` is executed by default. When the user logs in, the command listed in this field is executed and control is passed to that program. Once the program terminates, the user is logged out.

Once the new entry in the `/etc/passwd` file is complete, the `pwck` command should be used to verify the format of `/etc/passwd`. When you are satisfied with the contents of `/etc/passwd`, write the modified file to the disk and terminate the editing session.

Setting the New User's Password

The new user does not have a password at this point but may log in without one. Depending on the security needs at your installation and your own inclination, you can:

- Ask the user to create a password for himself.
- Create a password for the user and tell him what it is.
- Force the user to create a password for himself the first time he logs in to the system.

The procedures for the last two choices are supplied below.

Creating a Password for the User

To set a password for the user, first become superuser. Then type:

```
passwd  user_name
```

and respond to the system's prompt for a new password; refer to the *passwd(1)* entry in the *HP-UX Reference* for details. This will set the new user's password to the one you typed.

Forcing the User to Create a Password

If you neither want to create a password for the user nor depend on the user to create a password, you can set a parameter that forces him to create a password the first time he logs in to the system. To accomplish this, you must put a comma and two periods in the password field of the user's `/etc/passwd` entry. The password field has an optional aging field that is separated from the password field by a comma (refer to the *passwd(5)* entry for details). When the aging field contains two periods, it forces the user to create a password. Thus a typical entry for a user without a password might be:

```
john:,..:105:77:J Jackson,production:/users/john:/bin/sh
```

Removing a User (Using Reconfig)

Sometimes it is necessary to remove a user from the system. For example, if someone quits your company you should not allow the person to log into your system anymore. Do not remove system users, such as `root`.

To remove a user follow these steps:

1. Login as the superuser root.
2. Start the `reconfig` program by entering the command:

   ```
   /etc/reconfig
   ```

   The `reconfig` main menu will appear on your screen. This is the same main menu shown in the procedure for how to add a new user.
3. Using the [NEXT] and [PREVIOUS] softkeys, move the cursor to the `User Configuration` menu item and press the [SELECT] softkey.

   You will see the "User Configuration" menu on your screen. This is the same as shown in the procedure for how to add a new user.
4. Using the [NEXT] and [PREVIOUS] softkeys, move the cursor to the `Delete a user` menu item and press the [SELECT] softkey.

   The form to delete a user will appear on your screen (Figure 5-7).

```
************************************************
          RECONFIG -- DELETE A USER
************************************************

User Name:
Delete User's Files:    no






User Name? >>





[RESTORE] [       ] [       ] [       ]   [  HELP  ] [       ] [PREVMENU] [       ]
```

Figure 5-7. Delete a User Menu

5. Type the login name of the user you wish to remove from the system, followed by [Return].
6. Decide if you want to delete the user's files.

   This prompt requires a yes or no answer (`y` or `yes`, `n` or `no`, followed by [Return]).

   If you choose "yes", the user's HOME directory and all files in the HOME directory are removed from the system. If the user has files you (or anyone on the system) need, you should make a copy of these files *before* executing this command.

   If you choose "no" the user's HOME directory and files remain on the system. The password file will not have a name to associate with the user's id number, so the files will appear as being owned by a number, rather than a user. These files (also the home directory, and all subdirectories) should be re-assigned to another user by using the `chown` command or should be removed. The `rmdir` command removes a directory, the `rm` command removes a file.

The next time you add a user, the new user will get the user id from the deleted user. If you have not changed the ownership of the deleted user's files, the new user will automatically inherit the files. Unless you want this to happen, re-assign the files before adding a new user.

In either case, when you delete a user, andy optional entries in an access control list (ACL) associated with that user will be removed.

Pressing the [Return] key without typing in `yes` or `no` selects the value shown.

You will now see the message:

```
Are the above values ok (y or n)?>>
```

7. Check the values. If one or more value is incorrect, then type `n` and [Return]. If you type `n` `reconfig` will start again starting at step 4 showing the values currently set.

   If the values are correct, then type `y` and press the [Return] key.

   The `reconfig` program now performs the following tasks:

   - updates the `/etc/passwd` file to remove the entry for the user.
   - updates the `/etc/group` file to remove the user from the appropriate group.
   - if the user's files are to be removed, then the user's HOME directory and all files in it are removed.

   The `Delete a User` menu is redrawn and you will see the message:

```
Next User Name? >>
User <user_name> deleted.
Press [PrevMenu] if there are no more users to delete.
```

   where <user_name> is the user you just deleted.

8. Exit or continue.

   If you wish to delete another user, continue with step 4. If you do not need to remove more users, press the PREVMENU softkey. The main menu of `reconfig` will appear on your screen.

Removing a User from the System (the Manual Method)

The easiest way to add and remove users is with the `reconfig` program. The step-by-step method described here is provided for those of you who wish to understand the details of what needs to be done.

To remove a user from the system:

1. Remove the user's $HOME directory, subdirectories, and files by typing:

   ```
   find /users/user_name -user user_name -exec rm {} \;
   ```

   If you wish to remove the user's $HOME directory and *all* his other directories and files type:

   ```
   find / -user user_name -exec rm {} \;
   ```

   Note that this *removes all* the user's files and directories, and may include programs and scripts the user owned that other people depend on.

2. Delete his entries from the `/etc/passwd` and `/etc/group` files.

3. If you also wish to remove the terminal associated with that user, delete the terminal's entries from the `/etc/ttytype` and `/etc/inittab` files. Refer to the "Adding Peripheral Devices" section of Chapter 4, and to *inittab(5)* in the *HP-UX Reference* for details.

Suspending a User from the System

If you need to temporarily revoke a user's login privileges you need to modify the `/etc/passwd` file. Replace the password field with an asterisk (*). This will prevent the user from logging in. To allow him to log in again, remove the asterisk and assign a new password. The following example shows an asterisk in the password field:

```
atilla:*:101:5:Atilla the Hun:/users/atilla:/bin/sh
```

Backing Up and Restoring the File System

Background Information

Backing up your system means copying your system's files onto a backup medium, such as cartridge tape. You can use this copy to recover lost data if there is a hardware failure, a system crash, or if you accidentally remove or corrupt a file. Backups can be made on cartridge tape, flexible disk, optical disk, or 9-track tape. This section shows how to choose a backup method, how to perform the backup, and how to recover files from your backup media.

To minimize the chance of loss, store your backups at a different location from the main file system. "Data safes", specially designed air-tight, water-proof containers for mass storage media, are available from many computer accessory manufacturers. If a file or the entire file system is lost or destroyed, you can recover by restoring the latest version of your system backup.

If you are on an HP-UX cluster you must run the backup on the system with the backup device. For performance reasons, you should back up an HP-UX cluster from the root server, rather than from a cnode.

Backup Strategies and Trade-offs

The method, frequency, and extent of the backup operation depends on how much you use your system and how much data you feel you can afford to lose.

Archive Verses Incremental Backups. A complete backup is often called an **archive backup.** Restoring the file system from a full archive backup consists of restoring files from the most recent backup. While relatively expensive in terms of media, system resources, and the time required to make full daily backups, the time and effort spent recovering the system is minimal.

An **incremental backup** contains only files that have changed since the last archival backup. Incremental backups almost always require less time and less backup media than archive backups.

Hewlett-Packard recommends the following backup schedule:

- full archival backup weekly or biweekly
  - store the archival backup at least 2 weeks
  - take one archive backup per month and store it in a permanent archive
- daily incremental backup

  You should continue to make incremental backups until:
  - one or two weeks have passed since the last archive backup (if you are maintaining an archival schedule);
  - the size of the backups becomes unwieldy (for example, larger than one tape); or
  - you feel it is necessary to create a new archive backup for any reason.

Suppose, for example, that you make a complete backup of the file system on Monday and make incremental backups on Tuesday and Wednesday. Each incremental backup contains only those files that have changed since Monday. Further assume that, on Thursday, the file system is destroyed. The file system may be reconstructed by first restoring Monday's archival backup of the file system and then restoring the files from Wednesday's incremental backup. Note that the file system is now restored to the end of Wednesday's incremental backup. All work not on a backup has been lost.

Backups and the System Clock. One of the characteristics of incremental backups is that they depend heavily on the system clock. Both the current time and date as well as the time and date associated with the file being used as a reference point for the backup (such as `/etc/archivedate`) have to be reasonably accurate to insure useful incremental backups. Always check, and if necessary reset, the clock (using the `date` command) if the system has been powered down for any reason or if a check of the clock shows any appreciable amount of inaccuracy.

Choosing a Backup Method

There are four major methods of backing up your system:

- Using one of the backup scripts
- Using the `fbackup` command
- Making a file copy of your system
- Backing up selected files using the `cpio` command

There are tradeoffs to using any of these methods. Some of the tradeoffs are shown in the following table:

Table 5-1. Backup Method Tradeoffs

Method	Tradeoffs	Prerequisites
Backup Script	■ Easy to set up and change backup options ■ Uses the cpio command which is not very backup-friendly ■ Shipped with previous releases, so already familiar to existing customers ■ Difficult to access individual files for recovery.	Use `backupf` on 3 1/2 inch flexible disks, use `backup` on cartridge tape.

Table 5-1. Backup Method Tradeoffs Continued

Method	Tradeoffs	Prerequisites
Fbackup command	■ Currently the recommended method ■ Required method if you wish to preserve optional ACL entries. ■ Easier to recover files than from the backup scripts if backed up to a 9-track tape ■ Greater flexibility in backup strategy ■ Can specify multiple devices ■ Generates an index of files ■ Faster throughput ■ Allows you to back up active file systems	Pipe to `tcio` when backing up to cartridge tapes
Optical Autochanger file copy	■ No script or other software tool to help with this method: it is entirely a manual method. ■ Once on the disk, you can easily and quickly access any file.	Must have the optical disk drive and autochanger.
cpio/tcio command	■ Use for backing up individual files, rather than for backing up the full file system.	

Security If you are running a trusted system you must use the `fbackup` program for backups, and `frecover` for recovering files. This is the only backup and recovery method that preserves Access Control List (ACL) information.

Backup Media Considerations

You need to decide what media to use when backing up your system. The media available on the Series 300 HP-UX system, in order of ease of backing up, are:

1. 9-track tape

   Nine-track magnetic tape is the most expensive backup media, but if you invest in a magnetic tape drive that writes 6250 bpi (bits per inch), you will be able to store more data per tape than with any other option. (A 2400-foot magnetic tape written at 6250 bpi will hold approximately 140 Mbytes of data.) 800 or 1600 bpi 9-track tapes hold less per tape than cartridge tapes.

2. cartridge tape

   Cartridge tape is the most recommended backup media because it holds a lot of data (so you don't need to change media as often during a backup), it is easy to store, and it is easy to retrieve information from.

   If you are backing up onto cartridge tape, and you have an auto tape changer, make sure you modify your backup script to take advantage of this. Refer to the section called "Do I Need to Modify the Backup Script?".

3. optical disks

   Recovery from optical disks is the easiest, since you can simply mount the disk and copy files. It is also a fast backup media.

4. flexible disks

   Backing up onto flexible disk may be desirable from an initial system cost viewpoint: the flexible disk drive costs less than the cartridge tape drive. However, it is much more difficult to do a file system backup onto flexible disk. This is because flexible disks do not hold much information (only 770 Kbytes). This means that you will need to stay with your system throughout the backup to change disks. It also means that you must keep track of many disks, keep them in order, and label them consecutively.

Since so many disks are required for a flexible disk backup, you might consider backing up only part of your system during regular backups. *You should always have one current complete archival backup including your kernel and all other system files.* Back up the file systems or directories that change regularly (for example the `/users` directory). When backing up part of the file system you must change to the directory you wish to back up (for example, the `/users` directory), instead of changing to the root directory (`/`).

If you are backing up onto flexible disk you *must* have all the disks initialized before beginning your backup. The backup process will not work on uninitialized flexible disks. In addition, you cannot suspend the backup process to initialize more flexible disks; if you run out of disks, you must restart the backup process from the beginning.

Since you must install HP-UX from cartridge tape you probably already have a cartridge tape drive. However, since *all* UNIX systems can talk to 9-track magnetic tape, you might perform a backup onto magnetic tape simply so other systems can read the tape (that is, for file transfer).

A cartridge tape holds 16 Mbytes on a 150 foot tape or 67.5 Mbytes on a 600 foot tape. A high density, double-sided 3 1/2 floppy only holds 0.75 Mbytes. If you fill a 150 foot tape, it would take 22 floppies. If you fill a 600 foot tape, it would take approximately 80 floppies.

You must choose a device file that is associated with the chosen backup drive. The program provided for backing up your system uses, as a default, the device you used during installation or your last update. This device file name is `/dev/update.src`. If you do not want to use this device, write down the name of the device file you will be using. When you execute the back up command it will ask for the device file name.

If you are on an HP-UX cluster you must run the backup from the system with the backup device. For performance reasons you should back up the system on the root server.

Backing Up your File System Using the Backup Scripts

Note If you have just updated your system from an older version of HP-UX you will need to move the new `backup` or `backupf` script from the `/etc/newconfig` directory to the `/etc` directory.

Background Information on the Backup Script

There are several things to think about when backing up your system, particularly the first time.

- What backup script should I use?
- Where is the backup script?
- Do I need to modify the backup script?
- Where will errors be logged?

Which Backup Script Should I Use?. There are two backup scripts on your system: `backup` and `backupf`. Use `backup` for backing up your system onto cartridge tape (or 9-track magnetic tape, with modifications to the `backup` script). Use `backupf` for backing up your system onto flexible disks.

Where Is the Backup Script?. On a newly *installed* system these scripts are in the `/etc` directory.

On a newly *updated* system (for example, you just updated your system from 6.2 to 6.5), the new version of these scripts are in the `/etc/newconfig` directory. Move the scripts to the `/etc` directory. If you have made any changes to your old version of the backup script you may wish to implement the changes in the new version. The scripts have changed for the 6.5 release of HP-UX. If you use an older version of the script you should follow the instructions in the previous revision of the manual.

Do I Need to Modify the Backup Script?. In general you do not need to modify the backup script. In a few cases you *must* modify the script, and in several other cases you may *want* to modify the script. This manual will not discuss the optional modifications.

You *must* modify the script if:

- You are backing up to 9-track magnetic tape.

  To prepare to backup, you must edit the `/etc/backup` script and replace these lines:

  ```
  cpio -ocx       |
  tcio -o $dest
  ```

  with this line:

  ```
  cpio -ocBx > $dest
  ```

- You are using a cartridge tape drive with an autochanger.

  If you will use an autochanger cartridge tape drive (such as the HP 35401) you must edit the `/etc/backup` script and add the `-l` and `-n` options to the `tcio` command. for example, if you wish to use up to four tapes, starting with tape one, your new `tcio` command line will look like this:

  ```
  cpio -ocx       |
  tcio -o -l 1 -n 4  $dest
  ```

  The autochanger must be in selective mode for the autochanger option to work properly. For additional details on the autochanger options, refer to the *tcio* entry in section 1 of the *HP-UX Reference*.

- You are backing up an HP-UX cluster onto flexible disks.

  If you are backing up an HP-UX cluster onto flexible disks, you must modify the backup script so it backs up all CDFs. To do so, add `-hidden` to the arguments in the `find` command. For example:

  ```
  find / -newer $archivedate -hidden -print   |
  cpio -ocvx > /dev/rdsk/0s0
  ```

  Because the file system for an HP-UX cluster is typically very large, Hewlett-Packard does not recommend backing up to flexible disks. A complete backup requires many (over 100) flexible disks. Using 1 or 2 cartridge tapes is more reliable.

Where Will Errors Be Logged?. The `backup` and `backupf` scripts will write (log) the following information to the file `/etc/backuplog`: the start and finish times of the backup, the number of blocks copied, and any error messages that may have occurred during the backup. Information and messages written to this log file are appended onto the end of the file.

What Time of Day Should I Perform My Backup?. To make sure all files are correctly backed up, your system should be shut down to run-level s for backups. This is to prevent any programs or users from accessing and modifying files during the backup. If files are modified during backup, file consistency cannot be maintained.

The easiest way to perform a backup is to executed at night, using the `cron` procedure to schedule it automatically. The section called "Performing Backups Automatically" discusses this.

If you are backing up onto flexible disk you must execute your backups interactively since you need to switch flexible disks. This means you must either come in sometime when all other users are off the system, or you must get all other users off during the day.

Performing the System Backup with the Backup Script

The following procedure will back up your system onto cartridge tape (or magnetic tape if you have modified the `/etc/backup` script to write to a magnetic tape device). If you wish to back up to flexible disk you must use the `/etc/backupf` command rather than the `/etc/backup` command.

1. Log in as the superuser `root`. If you are not the superuser you will have problems copying files that you do not own or have permission to access.
2. Change to the root directory by typing:

   ```
   cd /
   ```

3. Execute the shutdown command. For example, if you wish to shut down the system without waiting for users to log off, you would type:

   ```
   shutdown 0
   ```

   For more information on the shutdown procedure, refer to the section "Shutting Down the System" in Chapter 3.

4. Mount all file systems you wish to back up.

   The `shutdown` command stopped all processes and unmounted all file systems in the `/etc/checklist` file. If you wish to back up information on those file systems you must now re-mount them by typing:

```
mount -a
```

This creates the /etc/mnttab file and mounts all file systems in /etc/checklist.

5. Insert the backup media.

   If you are backing up onto cartridge tape with the autochanger tape unit, it must be in the selective mode to use the autochanger options from tcio.

6. Type the backup command line.

   a. If you are doing an **archival** backup, type:

   ```
   /etc/backup -archive
   ```

   (/etc/backupf for backing up to flexible disk).

   b. If you are doing an **incremental** backup, type:

   ```
   /etc/backup
   ```

   (/etc/backupf for backing up to flexible disk).

7. Verify or type in the correct destination device path name.

   You will get a message similar to:

   ```
   backing up to /dev/update.src
   enter new device name to change the backup destination.

   this will timeout in 1 minute
   ```

   The device file, /dev/update.src, is the file from which you installed or updated your system. This is the device file associated with the source media on your install or update. If you are using that device for your backup, either press [Return] or let the program time out.

   If you wish to back up to some other device, type in the device file name (make sure it is the character special device rather than the block special device).

   If you don't know, you can exit the backup procedure by typing ^D.

   The backup procedure will now check the file name. It will verify that the device file is the character special file, and that it already exists. If it doesn't exist, you will get the following message:

```
Warning: you may be backing up to a file
```

If you get this message, exit the backup procedure (by entering [CTRL] [D]) and check the device file name.

During a backup to flexible disk, if the backupf command runs out of room on the flexible disk it prompts you to insert a new medium with the following message:

```
errno: 20, Can't write output
If you want to go on, type device/file name when ready
```

Insert a new medium, enter the device file name, and press the [Return] key. The backup process will then continue. If you type the wrong device file name, or simply press the [Return] key, cpio will abort.

During a backup to cartridge tape, if the end of the cartridge tape is reached during the backup process, it prompts you to insert a new medium with the following message:

```
tcio: to continue, type new device name when ready,
return implies same device
```

When this occurs, change the backup medium and press the [Return] key. The backup process will then continue.

If you have a cartridge tape autochanger you do not need to respond to this message.

If you get the following message:

```
Optional acl entries for <file_name> are not backed up.
```

then some of your files have Access Control List (ACL) information. This script does not back up ACL information—you must use the fbackup command if you wish to back up ACL information. For more details on ACLs, refer to Chapter 8, the section on Access Control Lists.

8. When the backup has finished, remove the backup media.
9. Label the backup media with the media number (e.g. 1 of 2, 2 of 2), the date, and the type of backup (archive or incremental) and store it in a secure place.
10. Unmount your file systems. *This must be done before you change directories or start processes.* To unmount your file systems, enter:

```
umount -a
```

11. Check the file system by typing in:

    `fsck -p` *[filesystem_name]*

    For more information on `fsck` refer to "Using the FSCK Command" in Appendix A in Volume 2 of this manual.

12. Return to your normal run-level. You can either reboot or you can use the `init` command:

    a. to reboot to the normal run-level, enter:

    ```
    reboot
    ```

    b. To use the `init` command, enter:

    ```
    rm /etc/rcflag
    init 2
    ```

    Note that if you do not require users to log in, your normal run-level will be 1, not 2.

13. Examine the information and messages sent to the `/etc/backuplog` file to determine if any errors occurred during the backup process.

Backing Up the File System Using the Fbackup Command

The `fbackup` command is a versatile and powerful backup utility. It provides for both full and incremental backups. You can assign various level numbers (from 0 to 9) to different types of backups. This allows you to implement as complex a backup strategy as required for your needs. For example, you can assign a full backup - level 0, a weekly backup - level 3, and a daily backup - level 8. `fbackup` attempts to back up live file systems (that is, those with active files). It will try to back up an active file up to five times (by default) or as many times as specified using the `-c` option with the `fbackup` command. To specify a configuration file see the description of `fbackup` in the *HP-UX Reference*.

The syntax of the command, showing commonly used options, is as follows:

```
                      ┌ -f device_2 ┐
                      │ -0-9        │
fbackup -f device     │ -u          │
                      │ -H          │
                      │ -g graph    │
                      └ -c config   ┘
```

Option	Description
`-f` *device*	Specifies the name of an output device file. Multiple files can be specified to increase the amount of media that you can use for backup. So, if you have more than one magnetic tape drive, you can specify more than one drive and load tapes into all drives specified in the command. The command backs up onto the first driver specified first, then the second, etc., then returns to the first and continues cyclically (provided that the tape was changed - otherwise it stops and waits).
`-0-9`	Indicates the backup level (a single-digit number) where level 0 is a full backup and higher numbers are incremental backup levels. To determine what to back up, `fbackup` looks in the file called `/usr/adm/fbackupfiles/dates`. All files modified since the last date stored in this ilfe for the same file system at lesser levels will be backed up. If no date is determined by the level, the beginning of time is assumed; thus, the option 0 causes the entire file system to be backed up.
`-u`	Updates `/usr/adm/fbackupfiles/dates` so it contains the backup level, when the backup occurred, and the graphfile (see `-g` option for definition).
`-H`	Search hidden subdirectories (context-dependent files or CDFs). Normally, only the CDF element matching the current context is backed up, without expanding the path name to show the actual element. For more information on CDFs, refer to the "Concepts" chapter of this manual.
-g *graph*	refers to a text file called *graph* that contains a list of files included or excluded from the backup. If a directory is included, it is expanded to back up the entire subtree. Graph file entries have the form: `i` *pathname_to_be_included* or `e` *pathname_to_be_excluded* A sample graph file is shown later, in the section called "Sample Graph File".

Option	Description

-c *config*	Specifies a configuration file called *config* that specifies values for the following parameters (only commonly used ones shown here):
	■ name of a file to be executed when a fatal error occurs, for example:
	`error /user/adm/fbackupfiles/error`
	■ number of times to try to back up an active file (default is 5). For example:
	`maxretries 5`
	■ number of file reader processes to use (default is 2, maximum is 6). For example:
	`readerprocesses 2`

Sample Graph File

You can ease backup procedures by creating a graph file and specifying it in the `fbackup` command line. For consistency and ease of identification, call this file: `/usr/adm/fbackupfiles/graphfile`. Since the basic directory structure of your system should remain the same, use the same graph file for all backups.

As an example, suppose you had the directory structure shown in Figure 5–8.

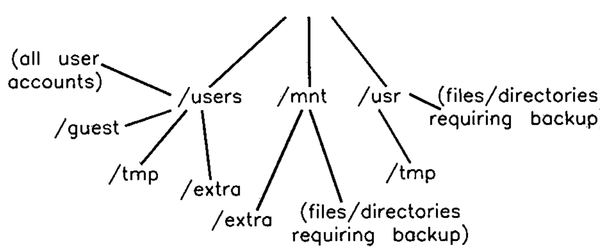


Figure 5–8. Example Directory Structure

If you wish to recursively back up all of `/usr` except `/usr/tmp`, all of `/mnt` except `/mnt/extra`, and `/users` except for `/users/guest`, `/users/tmp`, and `/users/extra`, you would have the following graph file entries:

```
i /usr
e /usr/tmp
i /mnt
e /mnt/extra
```

```
i /users
e /users/guest
e /users/tmp
e /users/extra
```

Examples

The following examples show how to back up all directories, subdirectories, and files implied recursively in the text file /usr/adm/fbackupfiles/graphfile to cartridge tape at /dev/update.src using the fbackup command. The /usr/adm/fbackupfiles/config file specifies a shell script for the system to execute (/usr/adm/fbackupfiles/error) if an error is encountered during backup.

To do a full backup, type in the following (all on one line):

```
/etc/fbackup -u0f - -g /usr/adm/fbackupfiles/graphfile -c
/usr/adm/fbackupfiles/config | tcio -o /dev/update.src
```

To do a weekly bckup, type in the following once each week (all on one line):

```
/etc/fbackup -u3f - -g /usr/adm/fbackupfiles/graphfile -c
/usr/adm/fbackupfiles/config | tcio -o /dev/update.src
```

To do a daily backup, type in once each day (all on one line):

```
/etc/fbackup -u5f - -g /usr/adm/fbackupfiles/graphfile -c
/usr/adm/fbackupfiles/config | tcio -o /dev/update.src
```

Backing Up the File System Using the Optical Disk Autochanger

HP Series 6300 Model 20GB/A Optical Autochanger

The following procedure will back up your system onto the Model 20GB/A autochanger.

1. Log in as the superuser.
2. Determine the size (in blocks) of the files in the directory you will be backing up. Execute the disk usage command by typing:

   ```
   du -sr /
   ```

   In the case of a full system backup, the source directory is the root directory, /.

Record the number of blocks that are returned.

3. Prepare an optical disk.

   Select a disk and disk surface on which you will write the backup files.

   Insert the disk into the slot of your choice. Refer to the operation instructions of the optical autochanger.

   If the disk surface you intend to use is not initialized, start at Step a), initialization. If the disk surface you intend to use is initialized, start at Step b).

Caution The following procedure destroys any data on the surface of the disk you select in Step a).

---

   a. Initialize the surface.

      In the following command `slotsf`—
      slot = disk slot number (location) in the autochanger
      sf = disk surface that will be initialized—"a" or "b" (labeled on the optical disk cartridge as "A" and "B").

      Type:

      ```
      mediainit /dev/rac/slotsf
      ```

      For example, initializing the "a" surface on a disk in slot 3, the command would be:

      ```
      mediainit /dev/rac/3a
      ```

   b. Create a new file system by typing:

      ```
      newfs /dev/rac/slotsf hpS6300.650A_noswap
      ```

      The *disk_type* parameter of `hpS6300.650A_noswap` is the *disk_type* name found in the `/etc/disktab` file. Both the standalone optical disk drive and the autochanger use the same *disk-type* parameter. The optical autochanger should not be used for swap space so the `_noswap` option is used.

4. Mount the disk surface onto the file system.

Note The term *destination_directory* in the following steps means that you are to use the full path name.

Create a directory on which to mount the surface by typing:

```
mkdir /destination_directory
```

Assign the access permissions desired by typing: (full read/write access is shown)

```
chmod 777 /destination_directory
```

Mount the surface read/write by typing:

```
mount /dev/ac/slotsf /destination_directory
```

5. Determine how many blocks are available on the optical disk surface by typing:

```
df -t /destination_directory
```

Record the number of blocks. Use the number under the "blocks" heading, rather than the number under the "total blocks" heading.

Compare the number of blocks to be backed up (source size) in step 2 with this number (destination size). If the source size is smaller than the destination size, the backup can be done on one surface. Otherwise, the backup will have to be split up onto two or more surfaces. If the backup needs more than one surface, other optical disk surfaces must be initialized and then, as needed in the sequence, be mounted onto the system to receive the additional segment(s).

To split up the source directory into sizes that will fit onto a surface use the command:

```
du -r /
```

This gives a list of all the subdirectories and their sizes. With this information you can split up the source directory into surface-size chunks.

6. Copy the files. This step is repeated for as many surfaces as are required to do the backup. Use the following command.

```
find /  -print |cpio -pdlmvx destination_directory
```

Because the optical disk surface is mounted for reading and writing at this point, data could be accidentally overwritten. To prevent this, remount the surface "read only" by typing:

```
umount /destination_directory
mount /dev/ac/slotsf /destination_directory -r
```

See Step 3a for explanation of "slotsf".

This surface is now mounted "read only" and users may recover the files without your assistance.

HP Series 6300 Model 650/A Optical Disk Drive

The following procedure will back up your system onto a Model 650/A Optical Disk Drive.

1. Log in as the superuser.
2. Determine the size (in blocks) of the files in the directory you will be backing up. To do this, execute the disk usage command by typing:

   ```
   du -sr /
   ```

   In the case of a full system backup, the source directory is the root directory, /.

   Record the number of blocks that are returned.
3. Prepare the optical disk.

   Insert the disk into the disk drive.

   If the disk surface you intend to use is not initialized, start at Step a), initialization. If the disk surface you intend to use is initialized, start at Step b).

Caution The following procedure destroys any data on the surface of the disk you select in Step a).

---

   a. Initialize the surface by typing:

   ```
   mediainit /dev/rdsk/mo
   ```

b. Create a new file system by typing:

```
newfs /dev/rdsk mo
```

4. Mount the disk surface in the file system.

Note The term *destination_directory* in the following steps means that you are to use the full path name.

Create a directory on which to mount the surface by typing:

mkdir /*destination_directory*

To mount the surface by typing:

mount /dev/rdsk /*destination_directory*

5. Determine how many blocks are available on the optical disk surface by typing:

df -t /*destination_directory*

Record the number of blocks. Use the number under the "blocks" heading, rather than the number under the "total blocks" heading.

Compare the number of blocks to be backed up (source size) in step 2 with this number (destination size). If the source size is smaller than the destination size, the backup can be done on one surface. Otherwise, the backup will have to be split up onto two or more surfaces. If the backup needs more than one surface, other optical disk surfaces must be initialized and then, as needed in the sequence, be mounted onto the system to receive the additional segment(s).

To split up the source directory into sizes that will fit onto a surface use the command:

```
du -r /
```

This gives a list of all the subdirectories and their sizes. With this information you can split up the source directory into surface-size chunks.

6. Copy the files. This step is repeated for as many surfaces as are required to do the backup. Use the following command.

```
find source_directory -print |cpio -pdlmvx /destination_directory
```

Since the optical disk surface is mounted for reading and writing at this point, data could be accidentally overwritten. To prevent this, remount the surface "read only" by typing:

```
umount /destination_directory
mount /dev/rdsk /destination_directory -r
```

This surface is now mounted "read only" and users may recover the files without your assistance.

Backing Up Selected Files onto Cartridge Tape

Cartridge tapes are 1/4 -inch tapes enclosed in plastic cartridge boxes. Assuming you installed your system from cartridge tape, you have a character special device file called `dev/update.src`. This file is associated with the cartridge tape you used to install HP-UX. If you prefer to use a different device file that is associated with your cartridge tape drive, replace `/dev/update.src` (in the following examples) with the appropriate character special device file name.

In all the following examples, *cpio will overwrite everything currently on your cartridge tape with the new information.* Make sure you do not have any information on the tape that you want to save.

If you are backing up selected files, do the following:

1. Go to the directory containing the files you wish to back up:

   ```
   cd directory_name
   ```

2. Use the `tcio` and `cpio` commands as follows:

   a. To back up all files and subdirectories from the current directory, type:

      ```
      find . -print | cpio -ocx | tcio -o  /dev/update.src
      ```

   b. To back up all files in your current directory, type:

      ```
      ls | cpio -ocx | tcio -o  /dev/update.src
      ```

   c. To back up selected files in your current directory and subdirectories, type:

      ```
      ls file_name dir_name/file_name|cpio -ocx|tcio -o /dev/update.src
      ```

The file names should be separated by blank spaces. If your files are in different directories, you can specify relative path names for your file names.

Backing Up Selected Files onto Flexible Disk or Magnetic Tape

Magnetic tape (also known as 9-track tape) is used in reel-to-reel tape drives (as opposed to cartridge tape).

During system installation or update, a character special device file called break `dev/update.src` was created. This device file is associated with the flexible disk you used for installing HP-UX. If you created a special device file for your drive with some other name, or if you are backing up to magnetic tape, replace `/dev/update.src` (in the following examples) with the appropriate character special device file name.

In all the following examples, *cpio will overwrite everything currently on your cartridge tape with the new information.* Make sure you do not have any information on the tape that you want to save.

If you are backing up selected files, do the following:

1. Go to the directory containing the files you wish to back up:

   `cd` *directory_name*

2. Use the `cpio` command as follows:

   a. To back up all files and subdirectories from the current directory, type:

      ```
      find . -print | cpio -ocBx > /dev/update.src
      ```

   b. To back up all files in your current directory (but no subdirectories), type:

      ```
      ls | cpio -ocBx > /dev/update.src
      ```

   c. To back up selected files in your current directory and subdirectories, type:

      `ls` *file_name dir_name*`\file_name | cpio -ocBx>/dev/update.src`

      The file names should be separated by blank spaces. If your files are in different directories, you can specify relative path names for your file names.

Backing Up Selected File Onto Rewritable Optical Disks

If you are backing up selected files, do the following:

- Prepare the optical disk (if necessary).

  Use the procedures in "HP Series 6300 Model 20GB/A Optical Autochanger" or "HP Series 6300 Model 650/A Optical Disk Drive" (as applicable) in the subsection "Backing Up the File System Using the Optical Disk Autochanger" to prepare a disk and mount it under the desired directory.

- Go to the directory containing the files you wish to back up:

  ```
  cd /directory name
  ```

- Use the cpio command as follows:

  To back up all files and subdirectories from the current directory, type:

  ```
  find . -print | cpio -pdlmvx /dev/rac/destination_directory
  ```

  To back up all files in your current directory (but no subdirectories) type:

  ```
  ls | cpio . -pdlmvx > /dev/rac/destination_directory
  ```

  To back up selected files in your current directory and subdirectories, type:

  ```
  ls file_name dir_name/file_name | cpio -pdlmvx /dev/rac/destination_directory
  ```

  The file names should be separated by blank spaces. If your files are in different directories, you can specify relative path names for your file names.

Restoring the System

Restoring the system means using your backups to put lost files back on the system.

General Information about Restoring Your System

There are three scenarios where you might need to restore your system:

- Someone on the system lost a file (perhaps accidentally deleted it) and needs a copy.

- You can boot the system, but the file system is destroyed to the point of not being able to access files and programs.
- Your file system is so badly damaged you can't even boot the system.

Each of these scenarios are described below.

Note that the system cannot write to a file that is marked as shared and being executed with `exec`. (Shared files are discussed in the "Memory Management" section of the chapter called "System Management Concepts". Refer, also, to the *chatr(1)* entry in the *HP-UX Reference*). Thus, files used during a system restoration, particularly `/bin/cpio`, should not be shared files. If they are marked as shared, they may not be recovered from the backup.

If you are part of an HP-UX cluster, do not restore device files from a pre-6.5 archive onto a 6.5 HP-UX system. If you must do this, read Chapter 2, the section on "Cluster Concepts", the subsection called "Cnode-Specific Device Files".

Restoring Selected Files

When you need to restore selected files onto your HP-UX system, follow the recovery procedures documented in the section pertaining to your backup method.

Restoring When You Can Still Boot the System

If you need to restore the system due to major problems, and the system is still functioning, log in as the superuser `root` and run the `shutdown` and `fsck` commands as referenced previously in the section "Performing a System Backup". In many cases, the `fsck` command can repair even serious problems in the file system. Often, lost files will show up in the `/lost+found` directory after running `fsck`.

Note Any programs or files that are being run or used on the system while restoring the system will not be updated. For example, if the `root` user runs the shell, `/bin/sh`, then `/bin/sh` will not be restored from the tape. You must copy `/bin/sh` to someplace else.

Note

Do not restore system device files such as `/dev/console` and the backup device or your system will hang and you will need to proceed with the method shown in "Restoring the File System When You Can't Boot the System".

To restore the entire file system, follow the procedures shown in the method appropriate to your backup method.

Restoring When You Can't Boot the System

If the entire file system is destroyed or if the system is in such poor shape that the `cpio` command will not function properly, then one of two options is available.

Note

If your file system is destroyed, you should take the time to understand the circumstances which caused the problem so you can prevent having to repeat this procedure.

Option One

If you set up a recovery system after your initial installation (using the procedures described in "Creating and Using a Recovery System" in Chapter 4), you can boot your system and rebuild your file system from your backup recovery system. The recovery system is a minimal functional HP-UX system on cartridge tape. You cannot create a recovery system on flexible disks.

Option Two

If you have not created a recovery system, your system must be re-installed from the original distribution medium. Follow the instructions in the installation manual to re-install the system. If you have updated your system since you installed, you will also have to do one (or more) updates before you begin restoring files.

Once you have re-installed the original system and it is operating properly, use the restore command appropriate to your backup method to copy (restore) the most recent archive and incremental backup(s) from the backup device to the system's root device.

Restoring from the Backup Script Method

1. Log in as either the superuser, `root`, or as the file's owner.
2. Write protect the tape or flexible disk on which your backup is stored.
3. Change directories. You must reside in the correct parent directory before restoring files. The correct parent directory is the same as the directory you were in when you created the backup. If you are restoring from an archive or incremental backup, you must be in the root (/) directory. If you are restoring files from a selected file backup, you must be in the directory where you performed the backup.
4. Place the backup medium in the mass storage device. If you are using a cartridge tape as the backup medium, *wait for the cartridge tape drive's conditioning sequence to complete* (that is, wait for the busy light to go off after putting the tape in the drive) before continuing with this process.

   If you wish to restore a single file (or several files) and you are restoring from a multiple-cartridge tape or flexible disk backup, you can list the files on the media with the `-t` option to `cpio`.
5. Once the backup medium is ready, enter one of the following command forms to copy the files:

   For flexible disk or 9-track magnetic tape, type:

`cpio -iBcdmux` *[patterns] special_file*	*to recover specific files*
`cpio -iBcdmuvx` *special_file*	*to recover the whole file system*

   For cartridge tape, type:

`tcio -i` *special_file* `| cpio -icdmux` *[patterns]*	*to recover specific files*
`tcio -i` *special_file* `| cpio -icdmuvx`	*to recover the whole file system*

   Note that *special_file* is the name of the character special device file associated with the backup device (usually `/dev/update.src`, `/dev/rmt/0m`, or `/dev/rct/c0`). You can use and optional parameter (*[patterns]*) to specify which files to recover. List the files (separated by a blank space) where you see *[patterns]*. If you wish to recover all the files, do not specify a pattern.

Patterns may include wildcard characters like * and ? if they appear inside apostrophes. for example the pattern '*/mail/*' will restore all the files in any directory called `mail`.

If you know which media the file is on (you used the `-t` option in step 4), you can insert the correct media and use the `R` option in `cpio`. The `R` option allows you to resync the headers so you do not need to read through all the media.

Restoring From the Fbackup Command Method

The `frecover` command options are similar to those for `fbackup`. You can also define a graph file to indicate which files should be recovered, and a config file to customize the behavior of the command. For example, the following command recovers all files from cartridge tape specified in `/usr/adm/fbackupfiles/graphfile` and executes a shell script listed in `/usr/adm/fbackupfiles/config` if the recovery process encounters an error (note that the following command must be typed on one line):

```
tcio -i /dev/update.src | /etc/frecover -x -g
/usr/adm/fbackupfiles/graphfile -c /usr/adm/fbackupfiles/config
```

Restoring From Optical Disks

Mount the disk and copy files using `cp` or `cpio`.

Performing Backups Automatically

This section assumes you have read the rest of the "Backing Up and Restoring Your File System" section, and understand terminology and backup strategies. If you are not familiar with the HP-UX backup commands and strategies, go back and read the preceding information.

Incremental backups can be performed automatically by using the `crontab` command to schedule the backup (where it gets executed by the `cron` "clock daemon"). Refer to the *cron(1M)* and *crontab(1)* entries in the *HP-UX Reference.*

Note Although you can perform backups automatically with any backup media, it makes no sense to do it with flexible disks. You must use many flexible disks to backup your system, and it therefore becomes an interactive session. Hewlett-Packard suggests you perform backups automatically only if backing up to cartridge tape or 9-track magnetic tape.

Automatic backups make sense only if you are using cartridge tapes (either limited to one tape or you must have an autochanger unit) or to 9-track magnetic tapes (limited to one reel).

The following is an example of automating the backups in a system using the `backup` script. To add automatic backups to your system, perform the following:

1. Login as the superuser, `root`.
2. Locate your existing cron file.

   If you don't know what your existing cron file is named, you can create a copy of it by executing:

   `crontab -l >` *new_cron_file*

3. Add a line to your existing cron information file to automatically start the backup procedure.

   For example, if you wish to automatically perform incremental (verses the full archival) backup at 11:55 p.m., you would add the following line:

   ```
   55 23 * * 1-5 /etc/backup
   ```

4. Use the `crontab` command with the file's name as an argument. For example, if your cron information file is called `croninfo`, enter:

   ```
   crontab croninfo
   ```

   This creates a file, `/usr/spool/cron/crontabs/root` (which overwrites the existing `/usr/spool/cron/crontabs/root` file).

When running backups with `cron`, there is no way of knowing if the system is inactive without logging users off. Choose a time when you think no one will be working. Send a message to all users requesting them to log off. The `backup` script notifies all current users when the backup begins.

If you run backups from `cron` as described above, perform the following steps:

1. Assign the `backup` script's error output device to a special file associated with a printer (default is `/dev/lp`).
2. Every morning after a backup:
    a. Examine the information and messages listed to the printer during the previous night's backup process.
    b. Remove, label and store the backup medium.
3. Every evening before a backup:
    a. Be certain that the printer associated with the `backup` script's error output device is on-line.
    b. Install a blank backup medium.

Changing a Password

The "Adding/Removing Users" section in this chapter discusses creating passwords for users. This section discusses how a user or the system administrator may change passwords.

Any regular user on the system may change his own (but no one else's) password by typing:

```
passwd
```

The `passwd` command prompts for the existing (old) password before allowing you to continue. Once the correct old password is entered, the command prompts for a new password. Enter at least six (6) characters and/or digits of which at least two are alphabetic and one is a digit or special character. Note that control characters like those generated by [Back space] are accepted but sometimes difficult to remember. The password is not echoed on the screen (for security purposes). The command then prompts you to re-enter the password to confirm it. Do so and, if the two entries match, the program accepts the new password. If the two entries do not match, you will be prompted to enter it twice again. It takes approximately 15 seconds for the system to install the password.

Users will occasionally create passwords for themselves which they cannot remember. Once forgotten, the user cannot log in to the system and will probably come to you, the system administrator, for help. Because only the encrypted form of the password exists in `/etc/passwd`, even you cannot determine the user's password, hence you must assign a new one. To do so, become superuser and type:

`passwd` *user_name*

where *user_name* is the user's login name. You will be prompted for the user's new password. Only the superuser may use this method for changing a password.

If the root password is lost or forgotten, no one can log in as the superuser. If you ever forget your superuser password, the only way to gain access is to use the recovery system (refer to "Creating and Using a Recovery System" in Chapter 4). If you have not made a recovery system, you will have to re-install your system. *Note that a complete re-installation of the system will destroy all the files on the disk.*

Changing the System's Run-Level

A **run-level** is a state in your system where a specific set of processes are allowed to run. This set of processes is defined in the `/etc/inittab` file for each run-level. You can define up to six run-levels (1-6).

Run levels `s` and 2 are predefined: `s` is the system administrator mode and 2 is the normal operating mode (users must provide user name to gain access). Most systems do not need to define additional run levels or use any run levels other than `s` and 2. This section is provided for those installations that have additional needs.

To create new run-levels, make entries in `/etc/inittab` that define how you want the system to operate in its new run-level. For example, identify the run-level entry by a run-level number (in the range 1 to 6), identify the command you want executed for each run-level entry and list any flags that are to be considered. Once `/etc/inittab` contains all of the entries you want for the new run-level, be certain to warn all users to log off before you change run-levels. Changing run-levels while users are logged on will terminate their login processes in the middle of execution unless the `getty` for their terminal is defined for the new run-level. Refer to Chapter 4, the section "Creating System Run-Levels" for more information on `respawn` and on creating run-levels.

If you are in an HP-UX cluster there are special considerations for changing run-levels. System run-levels are machine dependent since `/etc/inittab` is a CDF. The nodes in a cluster can be set to different run-levels. In general, changing the run-level of a system has no effect on the other systems in the cluster. If you are on the root server, however, changing system run-levels may affect the entire cluster.

Entering Run-Level s (The System Administration Mode)

Many of the system maintenance tasks you perform as system administrator require the system to be in run-level s. In this run-level, the only access to the system is through the system console by the user `root`, and the only processes running on the system will be the shell on the system console, background daemon processes started by `/etc/rc`, and processes that you invoke. This means that commands requiring an inactive system (such as `fsck`) should be run in run-level s.

When taking the system from any numbered run-level (run-levels 1 through 6) to the single-user run-level (run-level s), use the `shutdown` command instead of `init s` to change the system's run-level. The `shutdown` command kills all non-essential daemon processes and brings the system safely to run-level s.

To enter run-level s with a grace period of 30 seconds, type in:

```
shutdown 30
```

This will automatically warn all users that they have 30 seconds to log off, `sync` the system, kill all processes, and safely bring the system to run-level s.

Changing Run-levels

The following is a general procedure for changing the system from one numbered run-level to another. You must be logged in as the superuser to change the system's run-level.

1. Warn all logged in users before you change run-levels. Changing to another run-level while users are logged on will kill (terminate) their processes if the run-level you are moving to does not contain explicit `rstate` entries in `/etc/inittab` for their `getty`. Use the `write` or `wall` commands to communicate with the users. Note that the `wall` (write all) command immediately sends your message to the terminal of each user on the system.

   If each `getty` entry has the new run-level in its `rstate` field, or if the `rstate` field is empty (implies all numbered run-levels), you don't need to ask them to log off; their processes will not be killed (unless your new run-level is run-level s).

2. Next, change to the desired run-level by typing:

   `/etc/init` *new_run-level*

   where *new_run-level* is the number of the run-level you wish to enter.

The `cron` process, and other processes initiated by `/etc/rc`, are initiated only the first time the system enters a numbered run-level (run-levels 0 through 6) after a boot or a shutdown.

Run-Level Review

When you start up your system, you will be in run-level 2. This means that at startup your system executed the `inittab` commands associated with run-level 2.

You can move freely between run-levels as long as entering the new run-level does not kill user or system processes that may have begun in the previous run-level. Two examples of possible problems are discussed below:

- If you change to a different run-level, the processes corresponding to entries in `inittab` that do not explicitly include the new run-level will automatically be terminated. For example, assume you are changing from run-level 2 to run-level 3. If the `getty` entry for the console does not have the action `respawn` for `rstate` 3, entering run-level 3 from run-level 2 will cause the console to die!
- A user logs off prior to a change in run-level. You then change run-levels from run-level 2 to run-level 4. When the user attempts to log in, he cannot, unless his `/etc/getty` entry contains both run-level 2 and run-level 4.

When the system enters the new run-level, its actions are similar to the actions described in the "System Startup Functions" section of Chapter 3, except that the commands executed are those identified by the new run-level number.

Creating Groups/Changing Group Membership

Group is a UNIX concept allowing one or more users the same access to files. Being a member of a group enables you to access the same files other group members can access. This is useful for project files where everyone in the project needs access to the files, but no one else on the system does.

A group is defined by a single line in the `/etc/group` file. Each entry in the file consists of four fields, separated by colons. To create a group, edit the `/etc/group` file and make an entry for the group. The general form of the entry for a group is:

```
group_name:password:group_id:member1, member2, ... , memberN
```

where the fields have the following meaning:

group_name — This field contains the name of the group.

password — This field is used by the `newgrp` command. Placing an asterisk (*) in the `password` field prevents non-group members from switching to this group.

group_id — The *group ID* is the unique integer ID shared by all group members.

member1, member2, ... , memberN — This list is composed of the user name of each group member; user names are separated by commas.

To alter a group's membership, simply modify the membership field for the group entry in the `/etc/group` file. When you are satisfied with the group definition, write the modified file to disk and terminate the editing session.

A user can normally belong to only one group at a time.

If your system has a file called `/etc/logingroup`, users can belong to multiple groups simultaneously. Since `/etc/logingroup` is usually linked to `/etc/group`, the fields are the same as for `/etc/group`, but only the group ID and the list of users is significant. At login, each user belongs to all groups which list that user name as a member. Each user may be associated with a maximum of 20 groups in `/etc/logingroup`; there is no limit for the number of groups the user can belong to in `/etc/group`.

If your HP-UX system is only using the /etc/group file, you can change to a different group with the newgrp command. This command will change your effective group ID to that of another group. For example, to change to the group called proj1, you would type:

```
newgrp proj1
```

If you are not a member of the group, then you receive the message:

```
Sorry
```

If the group does not exist, you receive the message:

```
Unknown group
```

If you have the file /etc/logingroup you will not need to explicitly change groups, since you will simultaneously belong to all your groups.

At login, each user is placed in the group specified by his group ID entry in the /etc/passwd file.

Communicating with System Users

The following are ways to communicate with your users via the HP-UX system:

- The /usr/news directory and the news command provide a way to get brief announcements to the system users. If you are on an HP-UX cluster, placing files in the /usr/news directory will make the news item available to all systems.
- More pressing items (such as announcing an upcoming archival backup) can be entered in the message-of-the-day file, /etc/motd. Keep these messages short and current so users will read them. If you are on an HP-UX cluster, placing text in /etc/motd will make the message available to all systems. As shipped, /etc/motd will not be a CDF, thus the message of the day will be seen by all systems.
- Longer messages or even major documents intended for specific users are best sent with the mail command.
- To write to users who are already logged in, use the write command (for specific users) or the wall command (for all users). The write command is intended for user-to-user dialog, while wall is intended for system-wide announcements. Note that if a user has executed the mesg command with the -n (no) option, write permission to that user's terminal is denied and the write command will not work.

  On an HP-UX cluster, the write command will write to a user on the system from which the write command is executed. A cluster-wide version of write is not available for clusters.
- When the wall (write all) command is run by the superuser, any user protections are overridden; the command immediately sends its message to every user's terminal, regardless of the tasks they are performing. Thus, if you are logged on as the superuser, avoid using wall unless it concerns a pressing matter such as an impending system shutdown; consider a user's irritation at receiving an unimportant message while he is editing a file.

  On a diskless cluster, the wall command will write to all users on the system from which the wall command was executed. In addition, there is a cwall command to broadcast the message to all users in the cluster.

As shipped, the files /etc/profile and /etc/csh.login contain entries to automatically notify each user at login of news, message of the day, and mail.

Controlling Disk Use

As system administrator, you should keep track of the amount of disk space available to users and the distribution of free disk space across file systems. The following guidelines will help you evaluate your disk use and identify future disk needs:

- Execute the du command regularly (weekly or bi-weekly) and keep the output in an accessible file for later comparison. This method lets you spot users who are rapidly increasing their disk usage. For example:

  ```
  du -s /users/*
  ```

- To list the directories and files, with the largest first, type:

  ```
  du -a / | sort -nr
  ```

- Use the `find` command to locate large or inactive files. For example, the following entry records in the file `aging_files` the names of files neither written nor accessed in the last 90 days:

  ```
  find / -mtime +90 -atime +90 -print > aging_files
  ```

  The following example lists all files larger than 1/2 Mbyte:

  ```
  find / -size +1000 -print
  ```

- Use the `df` command to list the amount of free disk space on a volume.

  The `df` command returns the number of free blocks left on the file system available for the ordinary user (not superuser); it does not count the number of blocks reserved by `minfree`. If you wish to see a more detailed report of disk usage, type:

  ```
  df -t
  ```

  An example of output from `df -t` is:

  ```
  /       (/dev/dsk/0s0   ):     93918 blocks        135875 i-nodes
                                715904 total blocks  163840 total i-nodes
                                550394 used  blocks   27965 used  i-nodes
                                    10 percent minfree
  ```

  In the example, there are 93918 blocks (512-byte blocks) available to the ordinary user, and 165510 512-byte blocks (715904 − 550394 = 165510) available to the superuser.

If you have reached the minfree threshold (refer to "Creating a New File System" in Chapter 4), only the superuser can continue to allocate blocks.

Some HP-UX commands automatically write to files to monitor system use and for general house-keeping. These files are called logging files. If not periodically checked and cleared, these files simply continue to grow.

Here are some of the typical logging files[1] :

/etc/wtmp	A binary file that maintains a history of logins, logouts and date changes (if it exists).
/etc/btmp	A binary file that contains a history of failed login attempts. This file will not be created by your system; it will be used only if it exists.
/usr/adm/sulog	An ASCII file that contains the usage history of the su command.
/usr/lib/cron/log	An ASCII file that contains the history of actions by cron.
/usr/adm/shutdownlog	An ASCII file that contains a log of shutdowns. (If it exists)
/etc/backuplog	An ASCII file that contains a log of backups. (If it exists)
/usr/spool/lp/log	An ASCII file that contains a log of lp spooler requests.
/usr/spool/uucp/*LOG*	ASCII files that contains information of uucp traffic.

These files should be cleaned on a daily or weekly basis. The only way to avoid logging is to link log files to /dev/null. This action is not recommended.

In addition to logging files, there are several other places where unnecessary files may accumulate. To find these files, check the following directories:

- the /tmp and /usr/tmp directories (temporary working files),
- the /usr/preserve directory (editor files saved after a system crash),

[1] /etc/wtmp, /etc/btmp, /usr/adm/sulog, and /usr/adm/shutdownlog are CDFs if you are on an HP-UX cluster. Thus, pruning these files requires pruning all the elements of a CDF, not just the one file accessible from a given system. The /usr/adm directory is a CDF, not each file in the directory.

- the `lost+found` directory (files found by `fsck` without a parent directory).
- the `/usr/spool/uucppublic` directory (files sent to your system via uucp).
- the `/usr/spool/lp/request` directories (files waiting to be printed).

Initializing Media

Media is where you store data: hard disk, optical disk, flexible disk, cartridge tape, or 9-track tape. Before a disk or tape can be used for the first time, it must be **initialized** (or formatted) for use with your disk or tape drive and computer. This initialization is performed with the `mediainit` command. The main function of *mediainit* is to check the mass storage media for defects (areas where information cannot be stored). The `mediainit` command also assigns an interleave factor (refer to the discussion below on interleave).

Another part of initialization is to set up media directory information. This is performed by either `lifinit` or `newfs`. Depending on your intended use for the media, you may not need to do either of these.

What Needs to be Initialized?

The initialization commands you need to use depend on your intended use for the media. Your media can be used for the following:

- as a file system

  Hard disks are commonly used for file systems and swap space. Flexible disks can also be used for file systems. Hewlett-Packard does not recommend using cartridge tapes for file systems because of wear and tear of tape drive in doing random access.

- as a LIF volume for file storage or file transfer

  LIF volumes are commonly used to transfer files between HP-UX and other Hewlett-Packard operating systems, such as BASIC and the Pascal Workstation. LIF volumes can also be used to store files for HP-UX use.

- as unformatted media for file storage or file transfer (for example, for backups)

  Cartridge tapes are normally used for backing up the system. Flexible disks may also be used. Refer to the section "Backing Up and Restoring the File System" in this chapter for more information. General information on `cpio`, `tcio`, and `tar` can be found in the *HP-UX Reference* and in the section "Transferring Files" later in this chapter.

Note You *must* initialize all flexible disks and optical disks (using *mediainit*) before you use them. In addition, Hewlett-Packard recommends you initialize hard disks and cartridge tapes before using.

If no interleave factor is specified, `mediainit` assigns a default interleave factor. This works well with most media. Check the owner's manual that comes with your media's drive. It may suggest an interleave factor.

Prerequisites

Before using the `mediainit` command:

- set up and connect the disk or tape drive
- create the device files for your drive
- log in as the root user

Using mediainit

You must initialize your flexible disks, cartridge tapes, and hard disks before you can put files on them. This section describes the steps necessary to initialize your media. For a general discussion on media initialization, refer to the beginning of this section, "Initializing Media".

Note Make sure you correctly identify the desired device. Do not initialize a mounted file system since initialization erases all data on a media.

Initialize the media by typing in the `mediainit` command using the character special device file's `pathname` (for example, use `/dev/rdsk/1s0` rather than `/dev/dsk/1s0`):

`mediainit` *[options] pathname*

Refer to the *mediainit* entry in section 1 of the *HP-UX Reference* for a list of the available options.

The choice of the interleave factor can have a substantial impact on disk performance (both hard disk and flexible disk). As a general rule the interleave factor for flexible disks should be 2, and the interleave factor for hard disks should be 1. Refer to the `/etc/disktab` file to get the correct interleave factor for your media.

If you change your system configuration (such as moving the system disk from the built-in HP-IB to the HP 98625A/B or Internal High Speed disk interface), it *may* be necessary to reinitialize the disk to preserve optional performance. This is because the optimal interleave factor may change.

If you get the message:

```
mediainit: this type of device unsupported
```

you probably have used the block special file instead of the character special file.

Here are examples of two common uses of mediainit:

- If you are initializing a cartridge tape (associated with the character special device file `/dev/rct/c0`), type:

  ```
  mediainit /dev/rct/c0
  ```

- If you are initializing a flexible disk (associated with the character special device file `/dev/rdsk/1s0`), you must specify an interleave factor of 2 using the `-i` option as shown:

  ```
  mediainit -i 2 /dev/rdsk/1s0
  ```

Remember, it is usually necessary to initialize media only once.

To use the media as a mounted part of an HP-UX file system, you must now create a file system using either the `newfs` or `mkfs` command. Refer to the section "Creating a New File System" in Chapter 4.

To use the media to store LIF format files, or to transfer files to the Series 300 BASIC or Pascal Workstation operating systems, refer to the following section called "Initializing Media to LIF Format".

To use the media to perform file system backups, or to transfer files between HP-UX machines, you do not need to do any further initialization on your media. Refer to the section "Backing Up and Restoring the File System" or to the section "Transferring Files". Both sections are in this chapter.

Initializing Media to LIF Format

If you are going to make a LIF[2] copy of files on a flexible disk, you can use the `lif` commands covered in the *HP-UX Reference*. These commands allow you to make copies of LIF ASCII files on media initialized using any HP Series 200, 300 or 500 workstation-based operating system, whether it be Pascal or BASIC.

Is the Media Already in LIF Format?

To determine whether or not your disk is initialized in LIF format, place the disk in your disk drive and type the following (assuming your disk is at device file `/dev/rdsk/1s0`):

```
lifls /dev/rdsk/1s0
```

If the disk has been initialized and contains files, the `lifls` command causes a list of files similar to the following to be displayed:

```
PLOT_DEMO   PRINT_DEMO  GINPUT_DEMO   HELLO_DEMO   HELLO_TEST
```

If the directory is empty, the command will echo a blank line.

If the disk is not initialized the following appears in your display:

```
lifls : Can't list /dev/rdsk/1s0; not a LIF volume
```

Creating a LIF Volume

You can create a LIF volume with the `lifinit` command. If your media has never been initialized, it must be initialized using `mediainit` before `lifinit` can be used. The LIF volume can be created either directly on the disk or in the HP-UX file system.

---

2 LIF stands for Logical Interchange Format. It is an HP standard format used for directories and files on mass storage devices (such as flexible and hard disks). It allows you to use the same media on the various machines that use LIF, such as using the same disks with several 5 1/4 and 3 1/2 inch flexible disk drives made by Hewlett-Packard.

`lifinit` writes a LIF volume header on a volume or file, and has the following form:

```
lifinit [-vnnn] [-dnnn] [-nVOL_NAME] FILE_NAME
```

The options may appear in any order. Definitions of the options are:

-v *nnn* — sets the volume size to *nnn* bytes. If *nnn* is not a multiple of 256, it is rounded down to the next such multiple. This number should be the capacity of the media you wish to copy files to (for example, 270 336 for 5 1/4 inch flexible disks). If you do not specify a volume size it will default to 256 Kbytes for HP-UX regular files and to the capacity of the disk for device files.

-d*nnn* — sets the directory size to *nnn* file entries. If *nnn* is not a multiple of 8, it is rounded up to the next multiple. If you do not specify a directory size, the default size is calculated based on the volume size. Refer to the *lifinit(1)* entry in the *HP-UX Reference* for details.

-n*VOL_NAME* — sets the volume name to be *VOL_NAME*. If the `-n` option is omitted, the volume name is set to the last component of the path name specified by *FILE NAME*. If *VOL_NAME* is longer than 6 characters, it is truncated to 6 characters. *VOL_NAME must be all uppercase* (this is a LIF standard).

FILE_NAME — The name of the file on the LIF volume. *FILE_NAME must be all uppercase* (this is a LIF standard).

For example, to write a LIF volume header to the flexible disk associated with the `/dev/rdsk/1s0` device file, type:

```
lifinit /dev/rdsk/1s0
```

To write a LIF volume header (named WORK) to an HP-UX file (named TMP) that will be copied to a flexible disk, type the following:

```
lifinit -v270336 -d240 -nWORK TMP
```

where the volume size is the total number of bytes contained on a 5 1/4 inch flexible disk. Note again that the LIF standard dictates that both the LIF volume name (`-nVOL`) and the LIF file name must be in uppercase.

Setting the System Clock

You must make sure the system clock always has the correct time and date because a number of commands use the clock to accomplish their tasks.

Occasionally, the system clock needs to be set or reset. There is no need to reset the system clock on your Series 300 just because you have powered down—Series 300 computers have a battery which keeps the clock current.

If you are on an HP-UX cluster, setting the system clock is the same as with other HP-UX systems. However, the system clock is cluster-wide. Diskless cnodes' clocks are synchronized with the root server's clock as they join the cluster, and remain synchronized. Thus, changing the clock on one system will affect all systems in the cluster.

Setting the Time Zone

Only the superuser can change the system clock. Before attempting to complete the procedures which follow, be sure to log in as the superuser `root`.

The time zone environment variable (TZ) must be set correctly before setting the current time and date in the environment. Typically, TZ's value is set with a variable declaration (as shown below) in three possible files. The three files are: `/etc/csh.login`, `/etc/rc`, and `/etc/profile`. TZ can also be set from an application program with the `tzset` library routine.

The format for setting the time zone environment variable follows. A description of each field follows:

```
TZ=XXXHYYY
```

where:

`XXX` is an alphabetic abbreviation of the standard time zone, usually three letters in length.

`H` represents the difference between standard local time and Greenwich Mean Time, in hours. Fraction hours are indicated in minutes (for example, 3:30 for Newfoundland). Negative hours are allowed (for example, -9:30 for South Australia).

YYY is an alphabetic abbreviation of the daylight time zone for your area, usually three letters in length and may be deleted if Daylight Savings Time is not observed in your geographic area.

Insert or modify the lines in `/etc/rc` and `/etc/profile`:

```
TZ=XXXHYYY
export TZ
```

Insert or modify the line in `/etc/csh.login`:

```
setenv TZ XXXHYYY
```

For example:

- In Eastern time zone, use `TZ=EST5EDT`
- In Central time zone, use `TZ=CST6CDT`
- In Arizona, where Daylight Savings Time is not observed, use `TZ=MST7`

For more information on setting the time zone environment variable, refer to TZ under the *environ(3)* entry in the *HP-UX Reference* manual.

Note that CST6CDT now has two different means (since the United States makes the transition to Daylight Savings time on the first Sunday in April and Canada will make the same transition on the last Sunday in April). The `/etc/tztab` file has this information but requires CST6CDT#Canada to distinguish the two.

Setting the Time and Date

Once the time zone environment variable is correctly set, you must terminate the `cron` process, if it is running, and execute the `date` command. If you are on an HP-UX cluster, setting the time on one cnode will also set the time on every other cnode. To set the time and date, do the following:

1. Kill the `cron` process.

   If you are on an HP-UX cluster, you must terminate `cron` on each cnode. To terminate `cron`, first locate the `cron` process information by typing:

   ```
   ps -ef
   ```

   This will identify the Process ID (PID) for `cron`. To determine all `cron` processes on all cnodes in an HP-UX cluster, type:

   ```
   cps -ef
   ```

Then terminate `cron` by typing:

```
kill pid
```

where *pid* is the process ID associated with `cron`.

2. Set the correct time and date (using the `date` command) by typing an entry of the form:

```
date MMddhhmm{yy}
```

where:

- MM is a two-digit integer representing the month. For example, 03 represents March.
- dd is a two-digit integer representing the day of the month. For example, 02 represents the second day of the month.
- hh is a two-digit integer specifying the current hour in terms of a twenty-four hour clock. For example, 03 specifies 3:00 am and 14 specifies 2:00 pm.
- mm is a two-digit integer specifying the number of minutes past the stated hour. For example, 04 specifies four minutes past the hour.
- {yy} is an optional two-digit integer specifying the last two digits of the current year; this parameter may be omitted if the year is already correct. For example, 87 specifies 1987 as the current year.

When `date` is executed it echoes the time and date on your screen.

3. Restart `cron` if you terminated it in step 1.

To restart `cron`, type:

```
/etc/cron
```

Possible Problems When Changing the System Clock

The `make` program (refer to the *make(1)* entry in the *HP-UX Reference*) is quite sensitive to a file's time and date information and to the current value of the system clock. While setting the clock forward will not effect `make`, *setting the clock backward by even a small amount may cause* `make` *to exhibit extremely* *bizarre behavior.* Avoid setting times earlier than the current system clock's value.

As mentioned in the "Backing Up and Restoring the File System" section in this chapter, the process of making incremental backups depends heavily on the correctness of the date. This is because incremental backups are always made in relation to a dated file. This is yet another reason to keep the date correct on your system.

Note Altering the system clock may cause unexpected results for routines scheduled by `cron`.

Altering the system clock may also cause some unexpected results for routines scheduled by `cron`; refer to the *cron(1M)* entry in the *HP-UX Reference.* When setting time back, `cron` doesn't run until the clock "catches up" to the point from which it is set back. For example, if you set the clock back from 8:00 to 7:30 (which is *not* advised), `cron` will not begin executing until the clock again reads 8:00. If you are setting the clock ahead, `cron` attempts to "catch up" by immediately executing all routines scheduled to run between the old time and the new time. For example, if you set the clock ahead from 9:00 to 10:00, `cron` immediately executes all routines scheduled to run between 9:00 and 10:00.

Transferring Files

There are several methods of transferring files on HP-UX. The intent of this section is to tell you what is available, not to be an in-depth tutorial. Each section gives an overview of the command or utility and directs you to the source of more detailed information for each file transfer method.

The methods you can use to transfer files are:

- raw file I/O
  - `cpio`
  - `tcio` (used only with `cpio` or `tar`)
  - `tar`
  - `dd`
- LIF Utilities
- UUCP
  - `uucp`
  - `uuto`
- Local Area Network
- Virtual Terminal

Each of these commands and utilities is described below. In addition, there is a section on moving full directories from one system to another.

cpio

This command is used to transfer files between your hard disk, 9-track magnetic tape, and flexible disks. It is also used with `tcio` to transfer files to/from cartridge tape.

`cpio` accepts a list of files from standard input (that is, through pipes or redirection) and generates an archive to standard output (again, through pipes or redirection).

For example, if you want to store the file named `prog.c` from the current directory to your flexible disk at `/dev/rdsk/1s0`, you would type:

```
ls prog.c | cpio -ocBx > /dev/rdsk/1s0
```

To list the files on the flexible disk archive, type:

```
cpio -itcv < /dev/rdsk/1s0
```

To restore the file prog.c from the flexible disk into the current directory, type:

```
cpio -iBcdmuivx prog.c < /dev/rdsk/1s0
```

More examples of this command are given in the section "Backing Up and Restoring the File System" in this chapter, and in the entry for *cpio(1)* in the *HP-UX Reference*.

tcio

The command, tcio, is used primarily as a filter between archiving programs and a cartridge tape. The two main archiving programs on HP-UX are tar and cpio. The tcio command enables immediate report (which enables streaming) for cartridge tape drives that support immediate report (HP 9144 drives). It also allows buffering of data into large blocks (up to 64 Kbytes); this reduces wear and tear on older devices that don't support streaming. For more information on immediate report and tape streaming refer to the section "Magnetic Tape" in the "System Management Concepts" chapter.

The tcio program also provides utility functions (-u option) such as unloading tapes, writing tape marks at specific blocks (by default, it writes a tape mark to block 0 to prevent accidental image restores), and performing software-driven verification. It also provides the user with a software interface to autochanger units, allowing the user to specify which of a set of tapes to read, write, or load.

For example, if you want to store the file named prog.c from the current directory to your cartridge tape at /dev/rct/1s0, you would type:

```
ls prog.c | cpio -ocx |tcio -o  /dev/rct/1s0
```

To list the files on the cartridge tape archive, type:

```
tcio -i /dev/rct/1s0 | cpio -itcv
```

To restore the file prog.c from the cartridge tape into the current directory, type:

```
tcio -i /dev/rct/1s0 | cpio -icdmuvx prog.c
```

More examples of this command are given in the section "Backing Up and Restoring the File System" in this chapter, and in the *tcio(1)* entry in the *HP-UX Reference*.

tar

Use `tar` to transfer files between HP-UX and other UNIX operating systems, since `tar` is standard on all UNIX systems. The `tar` command (*t*ape *ar*chive), like `cpio`, understands and keeps intact the hierarchical directory structure.

For example, to archive all your users' directories onto a cartridge tape at `/dev/rct/1s0`, you would type the following:

```
tar cvf /dev/rct/1s0 /users
```

The `c` option is used to create the archive. The `f` option directs `tar` to create the archive in the filename that follows (`/dev/rct/1s0` in this example). This example also uses the verbose (*v*) option. This stores the files on tape with an absolute pathname. Note that there is no way to restore an absolute pathname to a relative pathname.

If you want to look at what was put on the tape, you can use the `t` option (table of contents):

```
tar tf /dev/rct/1s0
```

If you wish to restore the files stored in the previous example, use the `x` option (extract):

```
tar xf /dev/rct/1s0
```

If you wish to restore only the file named `/users/jme/work.backup`, type:

```
tar xf /dev/rct/1s0 /users/jme/work.backup
```

Since the files were stored with an absolute pathname, they will be restored with the same absolute pathname.

To archive a single file named `archive-it`, from the current directory, type:

```
tar cvf /dev/rct/1s0 archive-it
```

To restore the single file named `archive-it`, to the current directory:

```
tar xf /dev/rct/1s0
```

For more information on the `tar` command refer to the *tar(1)* entry in the *HP-UX Reference*.

dd

The command, `dd`, can be used to copy a file from one place to another. It is primarily used for 9-track tapes. It is different from `cp` in that it can be used to make some conversions on the file, transfer the file between different media, or reblock the file.

An example of using `dd` is when you need to do an image copy from one hard disk to another. When doing an image copy, make sure you have properly initialized the destination disk so no information is written onto bad tracks.

For more information on the `dd` command refer to the *dd(1)* entry in the *HP-UX Reference*.

LIF Utilities

The LIF utilities are commands that enable you to read and write files in Hewlett-Packard's LIF format. LIF is a format that HP-UX, the BASIC operating system, and the Pascal Workstation all understand (as well as many other Hewlett-Packard operating systems).

Getting a set of HP-UX files into a format that is readable by a language workstation requires three HP-UX commands: `lifinit`, `lifcp`, and `cat`. These commands are used in the following three steps:

1. Create a LIF volume with `lifinit` (refer to the section "Initializing Media to LIF Format" in this chapter).
2. Use `lifcp` to write the files to this volume.
3. Use `cat` to write the LIF volume to the initialized LIF media (usually a 5 1/4 inch flexible disk).

Copying HP-UX Files to LIF Volumes

Once you have created a LIF volume, you can write files to the volume using the `lifcp` command. The `lifcp` command has the following format:

`lifcp` *hpux_file VOL_NAME*:*FILE_NAME*

where:

hpux_file is a file in your HP-UX directory.

VOL_NAME is the name you gave to the LIF volume when you created it.

FILE_NAME is the name that file is given on the LIF volume.

LIF files must be in the uppercase characters.

For example, if you wish to transfer the HP-UX file called `testing.p` (located in the `/users/engel/progs` directory) to the LIF volume called `VOL1` in the current directory, you would use the following command line:

```
lifcp /users/engel/progs/testing.p VOL1:TESTING
```

You will receive an error message if there is not enough room left on the LIF volume for the entire file. You must then create another LIF volume and copy the file to it.

Moving the LIF Volume to Flexible Disk

After you have copied your files to the LIF volume file, use the HP-UX `cat` command to write the LIF volume to the flexible disk. Do this with the following steps:

1. Insert the flexible disk into the disk drive.
2. List the contents of the disk by executing the following command:

   lifls /dev/*file_name*

   where *file_name* is the name of the device file associated with the disk drive holding your flexible disk (for example, `/dev/rdsk/1s0`).

   Listing the contents of the disk helps determine if there are any files that should be saved. If you need to save a file which is on the disk read ahead to the section called "Moving LIF Files onto HP-UX".
3. Copy the LIF volume file to the disk using the HP-UX command `cat`. *Using the* `cat` *command to copy files to a flexible disk overwrites everything on the disk.* If you think you need to save any files on the flexible disk, go back to step 2.

   If your LIF volume name on HP-UX is called `VOL1` and the flexible disk you wish to copy it to is associated with the device file `/dev/rdsk/1s0`, you would enter the following command:

   ```
   cat VOL1 > /dev/rdsk/1s0
   ```

4. Once the LIF volume has been copied to the flexible disk, remove the LIF volume from your current directory by using the HP-UX `rm` (remove) command. Type the following:

```
rm VOL1
```

Adding Files to a LIF Volume

Assume that you already have a 5 1/4 inch flexible disk in LIF ASCII format and you now want to write an additional file to the disk leaving the current contents of the disk intact. To do this, you use the following command (each italicized parameter is described below):

`lifcp` *hpux_file* `/dev/`*file_name*`:`*FILE_NAME*

hpux_file — is the HP-UX file you want to copy to the LIF disk.

file_name — is the name of the device file associated with the flexible disk drive (for example, `rfd`).

FILE_NAME — is the uppercase file name given to the *hpux_file* stored on your LIF disk.

Before copying a file to the disk, it is a good idea to determine how much storage space you have on the disk. To do this, type:

`lifls -l /dev/`*file_name*

where *file_name* is the name of the device file associated with the flexible disk drive. This gives you a listing of the files on the disk and how much room they are consuming.

Moving LIF Files onto HP-UX

There are some occasions where it becomes necessary to copy LIF files from a flexible disk to the HP-UX file system. An example of this would be when you receive the latest revision of a program on a flexible disk and need to transfer it to the hard disk on your system. To move a LIF file from a flexible disk into your HP-UX directory you must:

1. Place the flexible disk in the disk drive.
2. List the files on the flexible disk using the `lifls` command, as previously illustrated. This helps you to verify that the file is on the disk.

3. Copy the file *FILE_NAME* from the disk into the HP-UX file *hpux_file* using the following command line:

   `lifcp /dev/`*dev_file:FILE_NAME hpux_file*

   where *dev_file* is the special device file name associated with the disk drive holding the LIF file.

   For example, if you wish to copy the file called `TESTING` from the disk in the disk drive associated with the `/dev/rdsk/1s0` file, to a file called `testing.p`, type in the following command line:

   ```
   lifcp /dev/rdsk/1s0:TESTING testing.p
   ```

UUCP

UUCP is the traditional UNIX file transfer method. Most UNIX-like operating systems will be able to send or receive files via UUCP. UUCP can be set up as either direct connect through a MUX port, or as a modem connect through a dial-up modem. The cu program is a part of the UUCP facility that enables you to log in and talk to non-UNIX systems.

The format for sending a file via UUCP is:

`uucp` *localfile remote!remotefile*

where:

localfile is the name of the file to be transferred from the local system.

remote is the name of the remote system (refer to the *hostname(1)* entry in the *HP-UX Reference*.

remotefile is the name of the file to be created on the remote system.

A typical example of sending a file via UUCP is the following:

```
uucp /users/joe/myfile system_4!~/save_for_joe
```

In this example, `/users/joe/myfile` is the local file which will be sent to the remote system. The remote system is named `system_4`. The ~ expands to the UUCP public directory: `/user/spool/uucppublic`. As a result of this command, the file will be saved in `/user/spool/uucppublic/save_for_joe`.

You can also use the `uuto` command to send files to a specified destination, and the `uupick` command to "pick" disposition of the files sent by `uuto`.

Refer to the "UUCP" chapter in *Device I/O and User Interface* volume in the *Concepts and Tutorials* series of manuals.

Local Area Network

Local Area Network (LAN) is one of several products from Hewlett-Packard. It consists of both networking hardware and networking software. The products include NS-ARPA Services and NFS. File transfer over LAN is much faster than over UUCP.

Refer to the *Using ARPA Services, Installing and Maintaining NS-ARPA Services,* and NFS manuals for more information on setup and use.

Virtual Terminal

Virtual Terminal (VT) is a program that allows you to access other HP 9000 computers without logging off your current computer. This helps to transfer files and to access information on other HP 9000 computers.

Refer to the "VT" section in *Misc. Tools* volume of the *Concepts and Tutorials* series of manuals.

Moving Directories

If you need to move an entire directory to a new machine you can use any of the archiving commands. The example here moves the directory */users/jme* onto another machine's directory */users/wwm* via flexible disk.

On the old machine, put the formatted flexible disk into the drive and type:

```
cd /users/jme
find . -print -hidden | cpio -ocBx > /dev/rdsk/1s0
```

The `-hidden` option is used to find all files in CDFs in HP-UX clusters.

On the new machine put the floppy from the old machine into the drive and type:

```
cd /users/wwm
cpio -iBcdmuvx < /dev/rdsk/1s0
```

You could also move the `cpio` file across LAN or UUCP, rather than using the flexible disk.

Contents

6

Kernel Customization

When you add products (such as new peripheral types and some new software) to your HP-UX system, you may need to re-create your kernel so HP-UX is able to communicate with the new product. This includes new types of peripherals, and some new software. When you re-create your kernel (or operating system) to better suit your needs, you are configuring (or customizing) your kernel.

This chapter helps you decide what values to choose when configuring your kernel, and also tells you how to reconfigure your kernel.

In this chapter the terms **kernel** and **operating system** are used interchangeably.

Introduction

Your system, as shipped to you, has many system performance tuning features built in. However, each application has unique system requirements and may require specialized tuning. Your application's manual may provide tuning tips. Understanding the memory management on the Series 300 computers will allow you to make some tuning decisions. For example, using the `chatr` command to make certain commands demand loaded may improve performance. Refer to the section "Memory Management" in Chapter 2.

Each time a kernel is created it replaces the existing kernel. This means that each version of the kernel is built from scratch; there is no way to add to your existing kernel.

Series 300 HP-UX allows you to configure various attributes of your kernel. These include:

- choosing the kernel drivers

  When you configure kernel drivers, you enable your kernel to "talk" to peripheral devices, interface cards, and certain software applications.

- using non-default swap space

  Swap space is space on your file system disk that the operating system needs when executing processes. Although your file system was created with swap space (during HP-UX installation), you may require additional swap space to run your system. You can configure additional swap space if required.

- choosing operating system parameters

  As shipped, operating system parameters are set to default values. These values are correct for most systems. However, for a special application, you may need to change selected operating system parameters to make the application more efficient.

The `config` command allows you to customize your operating system to meet any of the above requirements. If you are changing only the kernel drivers, you may use the `reconfig` command. These two configuration programs overlap in functionality (see in Figure 6–1).

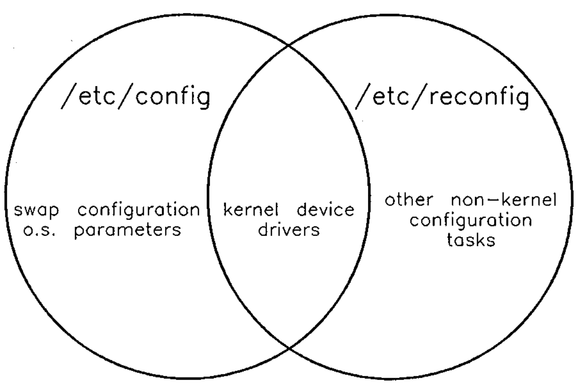


Figure 6–1. Reconfig and Config Functionality Overlap

Using the config Command

This section gives an overview of the steps involved in configuring your kernel using the `config` command. The values needed are discussed in the other sections of this chapter.

Files Required for the config Process

Programs and files used for configuring your kernel are:

- `/etc/config`—the configuration program,
- `/etc/master`—a file of lookup tables used by `/etc/config`,
- `/etc/conf/*.a`—libraries containing the kernel code,
- `/etc/conf/h/*.h`—header files,
- `/etc/conf/machine/*.h`—header files,
- `/etc/conf/dfile.*`—sample configuration description files.
- `/etc/conf/dfile`—the configuration description file for your kernel. By HP-UX convention, this file reflects what is currently configured into your kernel (`/hp-ux`).

If the above files and commands are not on your system, they can be loaded by using the procedures in "Installing Optional Software and Updating your System". Load the ACONFIG fileset from your installation media.

Running config

To use `config`, follow these steps:

1. Log in as the superuser, root. If you are on an HP-UX cluster, log in on the machine for which you are creating a new kernel. You can log in either physically on the cnode or using the `rlogin` command.
2. Change to the `/etc/conf` directory and create a configuration description file (generally called a dfile).

   The configuration description file is a file containing all device drivers you need in your kernel, plus swap space configuration (if not the default values), and operating system parameter configuration (if not the default values). You can use one of the supplied sample configuration description files as a starting point. They are in the `/etc/conf` directory.

By HP-UX convention the dfile for your current kernel is the file called `/etc/conf/dfile`. The software does not ensure that your current kernel is described by `/etc/conf/dfile`, but since `/etc/update` and `/etc/reconfig` use this convention, HP recommends you also use this convention.

In an HP-UX cluster, `/etc/conf/dfile` is a CDF. In an HP-UX cluster you *must* use the file named `/etc/conf/dfile` as your configuration description file; the software will not work properly if you do not follow this convention. Since you have logged in on the appropriate cnode (in Step 1) you will automatically access the correct version of `/etc/conf/dfile`.

Go to the sections on configuring device drivers, I/O cards, swap, and operating system parameters to decide what values need to go into the configuration description file. For kernel device drivers, go to the section "Configuring Device Drivers and I/O Cards". For swap space go to the section "Configuring Swap Space". For operating system parameters go to the section "Configuring System Parameters".

3. Go to the root directory and shut down the system.

   The following commands will put you into the root directory and will shut down your system immediately (i.e. it will put your system into the system administrator mode).

   ```
   cd /
   shutdown 0
   ```

   For more information on the `shutdown` command refer to the section "Shutting Down the System" in Chapter 3.

4. Back up the existing kernel by typing in (*Do not execute this step if you are on a diskless cnode in an HP-UX cluster*):

   ```
   cp /hp-ux /SYSBCKUP
   ```

   In an HP-UX cluster, `/hp-ux` is a CDF with an element for each node in the cluster. `/SYSBCKUP` is a file (not a CDF) that must be usable by the root server. A kernel built for a diskless cnode will probably not work for the root server. Therefore do not overwrite the backup kernel with a new kernel created for a diskless cnode.

5. Change to the `/etc/conf` directory by typing:

   ```
   cd /etc/conf
   ```

   This process creates your new kernel, so *you must be in a directory other than the root directory or you will overwrite the current kernel prematurely.*

   You may actually use any directory other than `/`. Since all files required by the `config` program are under `/etc/conf`, including sample configuration description files, this is the recommended directory.

Note Do not execute `config` in the root directory (/) or you will overwrite your kernel prematurely.

---

6. Execute `config` on your configuration description file. Refer to the *config* page in section *1M* of the *HP-UX Reference* for available options.

   ```
   /etc/config dfile
   ```

   Executing `config` creates `conf.c` and `config.mk`. The `conf.c` file is a C program that will build your kernel. The `config.mk` file is a makefile.

   You may wish to check to see that you have these files by typing `ls`. You should see your configuration description file, `conf.c`, and `config.mk`, in addition to the files shipped with your system.

7. Create the new hp-ux kernel (the file hp-ux) in the current directory by executing:

```
make -f config.mk
```

As it is executing, the config.mk makefile will list the portions of the kernel it is building. For example:

```
/etc/conf/libmin.a
   cs80.o
   mux.o
   muxs.o
   :
/etc/conf/libdevelop.a
   amigo.o
   ciper.o
   :
```

8. Move the new kernel to the / directory.

```
cp hp-ux /hp-ux
```

9. Reboot the system by typing:

```
reboot
```

If the newly configured kernel won't boot, use the boot ROM's attended mode to access the backup kernel saved in Step 4. To use the attended mode, press the space bar during bootup. This halts the automatic boot mechanism and allows you to choose the operating system to load. Select the option called SYSBCKUP rather than the new kernel SYSHPUX.

If you are in an HP-UX cluster on a diskless cnode, do not boot using the backup kernel. Since the backup kernel applies to the root server (refer to Step 4), you must instead create a new kernel from the root server, explicitly moving to the hidden directory in the CDF and working on the appropriate files.

If you have added a second swap device you must enable it before the system will use it. To enable the second device, execute:

swapon *second_swapdevice*

To enable this swap disk permanently, add an entry for the new swap device to the /etc/checklistfile. The swapon -a command in the /etc/rc file will enable all swap devices in /etc/checklist each time you boot your system. Refer to

"Adding to the Checklist File" for more information on this. Enabling your swap device is described in more detail in the section "Configuring Swap Space". If you are adding local swap space to a diskless cnode in an HP-UX cluster, refer to the section "Local and Remote Swapping in an HP-UX Cluster" later in this chapter.

Configuring Device Drivers and I/O Cards

This area of kernel customization enables you to configure the kernel device drivers you need for your system. The kernel device drivers bring in kernel code that allows the following to work:

- hardware drivers (for disk drives, printers, and tape drives)
- cards in the backplane (for HP-IB, RS-232, and datacomm cards)
- software drivers (pseudo drivers) (for Windows/9000, LAN, and HP-UX clusters). Pseudo drivers don't talk to any hardware device.

This section describes the information to put into your configuration description file. To actually reconfigure your kernel with device drivers, you can use either of the following procedures:

- the `config` command, as described in "Using the config Command"
- the `reconfig` program, as described in the section "Configuring Device Drivers and I/O Cards Using Reconfig"

Note *You MUST configure all the kernel drivers you need each time you create a new kernel.* If you do not include all the kernel hardware drivers you need, you will be unable to communicate with the peripherals associated with those drivers. Peripherals are the hardware associated with the files in `/dev`.

Sample Configuration Description Files

Several sample configuration description files came with your system to show how different device drivers and I/O cards can be combined to form different kernels. *These files should not be changed or removed.* The sample files are in `/etc/conf` and are shown in the table on the next page.

File Name in /etc/conf	Description	Used by reconfig
dfile	By HP-UX convention, this file contains the description of your current kernel.	
dfile.full	This file will configure an HP-UX operating system with all the device drivers supported in this release, minus LAN, NFS, and the cluster drivers, but no pseudo drivers other than pty*.	
dfile.full.lan	dfile.full plus the LAN driver.	Default (and fully loaded) O.S. for a standalone system
dfile.min	This file contains one possible minimal (but still useful) operating system configuration.	Minimal O.S. on standalone system
dfile.maxservr	This file contains all the drivers contained in dfile.full.lan, plus the cluster drivers and operating system parameters required for a cluster root server. This is the recommended dfile to use for your cluster's root server.	Default (and fully loaded) O.S. for root server
dfile.minservr	This file contains all the drivers contained in dfile.min, plus the cluster drivers and operating system parameters required for a cluster root server.	Minimal O.S. for root server
dfile.cnode	This file contains a minimum kernel configuration for a diskless cnode on an HP-UX cluster.	Default (and minimal) O.S. for diskless cnode
dfile.cnodemax	This file is the same as dfile.maxservr except all root server-specific information is replaced with diskless cnode-specific information.	Fully loaded O.S. for diskless cnode
dfile.cnodemin	This file is the minimum kernel configuration for a diskless cnode. This file will not work on many diskless cnodes; it is included only as a sample for a possible bare minimum kernel.	

If you create any configuration description files you should put them in the /etc/conf directory. Also, the reconfig program will leave the following files:

- dfile.new and hp-ux.new—if you created a new operating system using reconfig, but did not reboot the system, these files will be left in /etc/conf.
- dfile and dfile.BCKUP—If you created a new operating system using reconfig and rebooted the system, /etc/conf/dfile will hold your current kernel configuration. If /etc/conf/dfile already existed, reconfig renamed it /etc/conf/dfile.BCKUP so that it corresponds to /SYSBCKUP.

Determining the Required Device Drivers

In determining the device drivers for the hardware you have, consult the `alias` portion of the `/etc/master` file. To do this:

1. Make a list of the model numbers or names of all your peripheral devices, I/O cards, and software (LAN, X11 Windows, Windows/9000, RJE, SRM).
2. Look up the part number in the `/etc/master` file and add the corresponding driver name to your configuration description file. For example, if you have an HP 7945 disk, type the following command:

   ```
   grep 7945 /etc/master
   ```

   You will see the following entry:

   ```
   7945        cs80
   ```

   You would add "cs80" to your configuration description file.

   You need only one entry of a device driver. For example, even if you have five CS/80 devices, you need only one `cs80` entry in your configuration description file.

For an example of how your configuration description file should look, list one of the sample files sent with your system.

Table 6–1 shows all the possible kernel drivers you can configure into your system.

After you have created a new kernel with `config` you must create device files for your new peripherals if they don't already exist. You can execute `config -a` to get mknod templates for most device files you need to create. Device files enable you to communicate with the device. Refer to the section "Adding Peripheral Devices" in Chapter 4 for information on creating device files.

Table 6-1. Optional Kernel Drivers

Driver Name	Used For:
cs80	most mass storage devices (included in all sample configuration files)
scsi	SCSI direct access storage devices
amigo	Amigo mass storage devices
ciper	Ciper printers
printer	Non-ciper printers
hpib	Plotters; also include for Device I/O Library (DIL)
tape	9-Track magnetic tape drives
stape	9-Track streaming tape drives
ac	Optical autochanger
dos	HP 98686 DOS Coprocessor driver
vme	HP 98646 VME card
vme2	HP 98577A VME expander
nfs	Support for NFS networking
lla lan01	Support for NS-ARPA networking (formerly the ieee802 and ethernet drivers)
rfa	RFA server
dskless	Diskless code pseudo-driver
98624	Internal disk controller (always included by `config` – other drivers depend on it)
98625	Hi-speed HPIB disk controller
98628	RS-232 datacomm card
98642	RS-232 4-channel MUX card
98626	HP 98626 or HP 98644 RS-232 serial interface
gpio	GPIO card; also include for Device I/O Library (DIL)
srm	Shared Resource Management (SRM)
rje	Remote Job Execution (RJE)
ptymas ptyslv	pseudo terminal drivers (required for HP Windows/9000, Xwindows, and other software). Included in all sample configuration files.

Configuring Swap Space

This section describes how to determine the swap space information to put into your configuration description file. To reconfigure your kernel, use this information with the procedure described in the section "Using the Config Command".

Swap space is a contiguous area on the secondary storage, usually a hard disk drive, reserved for use by the Series 300's virtual memory management system. The virtual memory management system moves processes and shared memory objects to this space when there is a lack of available memory. Refer to Chapter 2, the section "Memory Management", for more information on how swap space is allocated to a process.

Although you can use the optical disk drive for swap space in a temporary situatin or in an emergency, it is not recommended that you put swap space on the optical disk in the general case. Do not put swap space on the surfaces of the disks in the optical autochanger.

Swap space is separate from the file system; it is a portion of your file system disk reserved for swapping and was created when you installed your system. Series 300 computers support both single and multiple swap devices. The multiple swap device mechanism allows the swap space to be present on several disk drives for ease of expanding the swap space. Also, if your applications require much more memory than you have installed, multiple swap devices may increase throughput.

The space for the entire image of every existing segment is allocated on the swap space, therefore, swap space must be large enough to hold all segments of all existing processes. If there is not enough swap space the system will either return an error (such as ENOMEM) for some system calls, or it will kill the user process and send you the message:

`Sorry, pid` *pidnum* `was` *message*

Pidnum is the id number of the process that was killed. *Message* is one of seven possible messages, describing when and why the process was killed.

Note the following about swap space:

- The amount of RAM has little relationship to the amount of swap space you need.

Every process you start requires swap space. The swap space is allocated at process creation and is always required, regardless of RAM.

- You cannot use *all* your swap space.

  Your swap space is generally avaliable to your processes. However, just as fragmentation can occur in your file system, it can also occur in your swap space. Fragmentation and overhead uses some of the available swap space.

- You cannot free the swap space until your process has completed.

  Using the `free` command from your program frees memory only for your process to use later. It does not release swap space for general use. The only way to free swap space for other processes is to end your process.

If you need more swap space, you can add more swap devices or you can rebuild the file system and reserve more swap space on your existing swap device.

Some hints on "gaining" additional swap space are:

- add another disk dedicated to swap.
- Change the mix of running tasks to reduce virtual memory activity.
- Change the way the application grabs memory. For example, if a very large process sleeps to let other processes run, it still holds its swap space. If you, instead, have it exit, it would free the swap space for other processes.

If you are on an HP-UX cluster, read the section in this chapter called "Local and Remote Swapping in an HP-UX Cluster".

In the default file system you will have one disk, which functions as your swap device. However, the kernel will check to see if the swap device has been changed. If you wish to use non-default values for the swap devices you must put it in your configuration description file. The information in the configuration description file is used to create the `conf.c` file.

The kernel initialization procedures, executed immediately after your operating system is loaded during bootup, will use the following rules to determine specifications for the swap device:

1. If the swap device is specified in the configuration file, then use the values from `conf.c`.

2. If the swap device has not been specified in the configuration file, the swapping device will be assigned to the same device as the root device.

Overview of Creating More Swap Space on an Additional Disk

Creating more swap space is a fairly detailed process. The steps below give an overview of what needs to be done. Each step is described in more detail later.

1. Initialize the media.
2. Determine the amount of swap space required.
3. Create the file system on the new disk.
4. Add the appropriate entries to your configuration description file. This should be the file called: `/etc/conf/dfile`.
5. Edit the `/etc/checklist` file if you wish to automatically enable the swap space each time you boot the system.
6. Execute the `config` and `make` commands.
7. Enable the new swap space if you did not add it to the `/etc/checklist` file.

Note If you see the warning message:

`swap: rmap ovflo, lost`<*blocknumber,blocknumber*>

read the information following this note.

Depending on the amount of additional swap space on the additional swap device, you may see the following warning message:

`swap: rmap ovflo, lost`<*blocknumber,blocknumber*>>

This warning message indicates that the kernel data structure is not sufficient to manage the additional swap space with the current `dmmax` value. You need to reconfigure your kernel with the next higher value of `dmmax`, `dmtext`, and `dmshm` as tabulated and explained in Table D-1 in Appendix D (Volume 2).

Step 1: Initialize the Media

If your disk has not yet been initialized (e.g., it is a new disk) you must initialize it now. As an example, for an HP 7946 disk drive associated with the device file `/dev/rdsk/1s0`, you might use the following command:

```
mediainit -i1 /dev/rdsk/1s0
```

For additional information on initializing disks, refer to Chapter 5, "Initializing Media", or the *mediainit* entry in the *1* section of the *HP-UX Reference.*

Step 2: Determine the Amount of Swap Space Needed

You must have enough swap space to hold your process or the process can't execute. You create a file system with the `newfs` command. Unless you use the `-s` option for `newfs`, a default amount of swap space is created. The default size depends on the disk drive. The default parameters are in the `/etc/disktab` file. A file system (with swap space) was created during HP-UX installation.

The swap space you determine should be a multiple of the `dmmax` operating system parameter. This parameter is described in Table D-1 in Appendix D (Volume 2). If you have not changed this parameter, then use multiples of 1/2 Mbyte. If you do not use a multiple of `dmmax` some swap space will be wasted.

If you need to change the amount of swap space, refer to your application's manual to see if it describes swap space requirements. For example, if you are running HP Windows/9000, look in the *HP Windows/9000 User's Manual,* the "Window Limitation" appendix, the "Configuring Swap Space" section.

If you are the root server of an HP-UX cluster you must determine the swap requirements of each cnode that will swap to the root server. After you know what each cnode requires, add all the requirements for a total required swap area for your root server.

There are two methods you can use to calculate your swap space requirements. They are described in the next two sections.

Estimate Your Swap Space Based on Your Largest Application

Estimate the amount of swap space you need by adding the space required by your largest application to the default amount of swap space. Use the following form to determine the swap space. Remember, 1 Kbyte = 1024 bytes:

Determine the swap space (in Kbytes) required by your largest application (look in the manual supplied with your application or ask the manufacturer). If you will be running several applications concurrently (such as HP Windows/9000 and HP PCDS), you should add their swap space requirements together.	____________
Add the current amount of swap space from the messages on your display at boot up, divided by 2. (Current swap space ÷ 2) At powerup the size and location of the swap space on each swap device is displayed in 512-byte blocks. "start = xxxxxx" indicates the swap space's starting disk block number and "size = xxxxxx" indicates size of swap space. You can usually see the bootup message by executing the `dmesg` command.	____________
TOTAL swap space needed (in Kbytes)	____________

Calculate Your Swap Space

Calculate your swap space requirements using the following formula and the worksheet later in this section.

If you run large processes, or if you have many users on your computer, you may need to use the formula given below. It is difficult to determine some of the numbers to use in the formula. Use this formula only if the above procedure didn't work. Make sure you count each process (using the process ID) rather than each program name.

Executing the `size` command on the executable file will give you a starting point for the amount of space it requires (refer to the *size* entry in section *1* of the *HP-UX Reference*). The `size` program will return the process's code size, static initialized data size, and static uninitialized data size. To get the total static data size, add the static initialized and static uninitialized data sizes. There is no simple way to determine the dynamic data size unless you know your program's run-time logic.

To compute the swap space, use the formula:

Swap space = A + B + C + D + E + F + G

where:

A = SUM(all shared code sizes) for all running processes as shown by `ps -el`. Don't count the `pageout` daemon or swapper processes. Count shared code only once. In an HP-UX cluster, count shared code only once per cnode. The `file` command tells if a file contains shared text ("pure" means shared). Refer to the *file* entry in section *1* of the *HP-UX Reference* for more details.

B = SUM(all data sizes) for all processes. If the process does not use shared code, then add its code size (refer to "A") into its data size. It is difficult to determine the exact swap space required by the dynamic data segment. However if you are familiar with the logic of the application program you may be able to tell how much data space the program may expand via `sbrk` or `malloc`. Refer to the *sbrk* entry in section *2*, and the *malloc* entry in section *3*, of the *HP-UX Reference* for more information.

C = SUM(all stack sizes) for all processes. It is difficult to determine the exact stack size of a given process, especially if the process uses recursion. For non-recursive processes use the value of `dmmin` (refer to Table D-1 in Appendix D).

D = SUM(sticky code sizes) for all code files, with the sticky bit set, that have been executed since bootup, but are not currently being used by any process. If the sticky code is currently being used it is already included in A.

You can determine if a program is sticky by executing `ls -l`; all sticky programs have a "t" in their permissions.

E = SUM(all existing shared memory segments' sizes) for all shared memory segments created by users via `shmget`. HP Windows/9000 uses about 4 1/2 Mbytes of shared memory.

F = Size of the scratch area used by `exec`. Default size of the scratch area is 256 Kbytes.

G = Fragmentation and overhead. Use 10% of total swap space for fragmentation and overhead.

Because data and stack segments can grow dynamically, the system uses an algorithm which allocates extra swap space for these segments in anticipation of growth. The system allocates swap space in **units.** The units are controlled by the configurable parameters `dmmin` and `dmmax` as shown in Table D-1 in Appendix D (Volume 2). The first unit allocated is `dmmin` 1-Kbyte blocks. Each successive unit is twice as large as the previous one, until `dmmax` is reached. From this point, all units are `dmmax` × 1Kbyte blocks in size. This algorithm results in a wide divergence of efficiency, depending on the application programs which are run. Because of this, there is no easy way to estimate fragmentation, but it may be computed for known environments. Besides the user process size, some other process-related information is also kept in the swap space.

Calculating Swap Space

A

For shared code, fill out the code space needed by the process. Count shared code only once. In an HP-UX cluster, count shared code once per cnode.

Process ID	Code size
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -

B

For each shared process listed for A, fill out the data space needed by the process. For each non-shared process, add the process's code size to its data size and enter the amount (i.e. total from executing `size`).

Process ID	Data size (minimum data space = dmin; default=16)
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -

C

For each process, fill out the stack space needed by the process.

Process ID	Stack size (minimum stack space=dmin; default=16
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -

D

For each sticky file that has been executed since power-up, but not currently used, fill out the code space.

Program name	Code size
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -
- - - - -	- - - - - -

E

For each shared memory segment give the shared memory segment size. Note that if you are running HP Windows/9000 you must add at least 4 1/2 Mbytes (use `ipcs -ma`).

Shared Memory ID	Segment size
-----	------
-----	------
-----	------
-----	------
-----	------

F

Scratch area size (default is 256 Kbytes): ____________________

G

Fragmentation and overhead. Compute as outlined in the previous discussion, or use an estimate. A suggested estimate is 2 Mbytes.

____

Total Amount of Swap Space Needed

A + B + D + C + D + F + G = Total swap space

Step 3: Create the File System on the New Disk

If this disk will be used entirely for swapping, skip this step and continue with Step 4. If the disk will contain both files and swap area you must create a new file system. For example, if you wish to put the default file system and swap space on an HP 7946 disk drive associated with the device file `/dev/rdsk/1s0`, type:

```
newfs /dev/rdsk/1s0 hp7946
```

For more information on creating file systems, refer to Chapter 4, the section "Creating a New File System", or to the *newfs* entry in section *1M* of the *Configuration Reference Manual.*

Step 4: Add the Appropriate Entries into Dfile

The swap space on your root device (the disk drive that holds the root file system) is called the default swap space. If your only swap space is on the root device you do not need to reconfigure your kernel. However, if you wish to swap to any device other than the root device, you must add "swap" entries (for each swap device) to `/etc/conf/dfile` and reconfigure your kernel.

You can specify swap devices (the secondary storage devices holding swap space) by putting a line in your configuration description file (`/etc/conf/dfile`) that looks like:

`swap` *devname address swplo [nswap]*

where:

devname — Swap device's driver name (e.g., cs80).

If you don't know the driver name, look up the product number in the `/etc/master` file's alias table.

address — Minor device number in hexadecimal, without the preceding 0x.

Refer to the section "Adding Peripheral Devices" in Chapter 4 for a discussion on minor device numbers.

swplo — Swap area's location in decimal.

If (swplo = -1), then the size of the swap area is read from the superblock, and the swap space is put on the disk following the file system. This is the recommended value. If (swplo = 0), then allocate the entire disk to the swap space; don't reserve any area before the swap area. This destroys any existing file system on that disk.

nswap Size of swap area in number of 1 Kbyte units. This is treated as an upper bound for swap size. By default, the space from the end of the file system to the end of the disk is used for the swap area. If you want to specify this swap area size exactly, use *nswap*.

For example, suppose you have two CS80 swap devices, both using the default swap space created when you created a file system. Disk 1 is your root device; enable it with the `swap auto` entry. Disk 2 is at Select Code 14, Bus Address 1. You would put the following two lines into `/etc/conf/dfile`:

```
swap auto   /* disk 1 */
swap cs80 E0100 -1   /* disk 2 */
```

Additional examples are shown in the section "Possible Swap Setups" later in this chapter.

Step 5: Edit the /etc/checklist file

If you want the system to automatically access the new swap device, add it to the `/etc/checklist` file. If not, skip to Step 6.

With the new entry in `/etc/checklist`, the `swapon -a` entry in `/etc/rc` will enable the device each time you boot. Refer to the *swapon* entry in section *1* of the *HP-UX Reference* for more details. For example, in Figure 6–2-a and Figure 6–2-b, if the second disk corresponded to the device file `/dev/dsk/2s0` and had a minor number of 0x0e0100, the second device could be enabled by adding a `/dev/dsk/2s0` swap entry to `/etc/checklist`. Once you have enabled a device for swapping, it can not be disabled without halting the system.

Step 6: Execute the Config Command

For a step-by-step description of how to execute the `config` command refer to the secton "Using the config Command" earlier in this chapter.

Step 7: Enable the New Swap Space if You Skipped Step 5

If you added the new swap space to the `/etc/checklist` file, do not do this step.

The kernel will automatically turn on the first swap device specified. If you want to use the other swap devices you have configured, you must use the `swapon` command after you boot the new kernel.

Possible Swap Setups

Swap space can be on the same disk as your file system, on a separate disk, or both (see Figure 6–2). If you need more than the default swap space, the best swap configurations are Figure 6–2-a and Figure 6–2-b. An explanation of each configuration shown in Figure 6–2 follows.

Note If you rebuild your root file system, you must have another bootable system. The procedure to rebuild your file system is described later in this section.

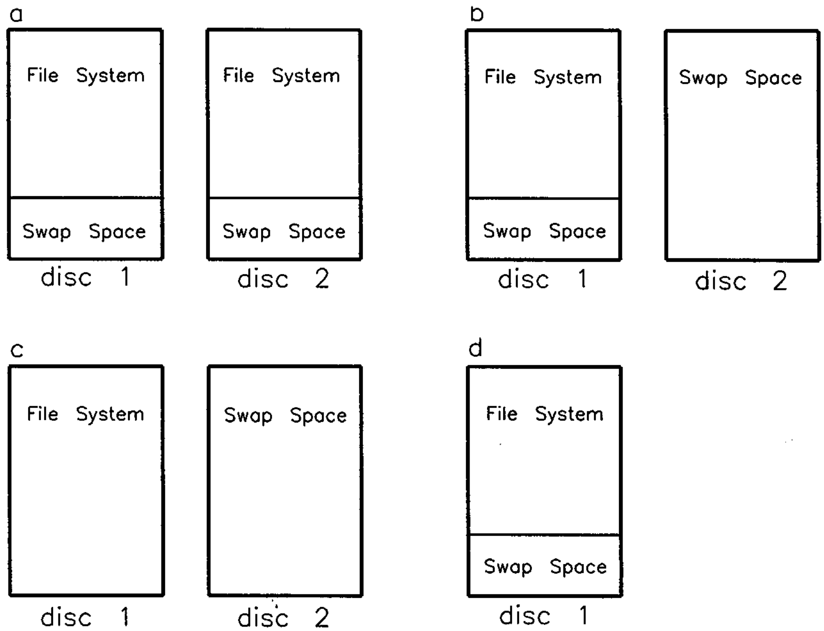


Figure 6–2. Swap Space

Possible Swap Space Configurations

Possible configurations for swap space [1] are:

- Figure 6–2-a shows two file system disks, each with the default swap space.

  On each disk, a file system was created using `newfs`. The entries in `/etc/conf/dfile` would look like:

  ```
  swap auto  /* disk 1 */
  swap cs80 E0100 -1  /* disk 2 */
  ```

- Figure 6–2-b shows the root disk with a the default swap space and file system size, and a second disk dedicated to swapping.

  The entry in `/etc/conf/dfile` for Figure 6–2-b would look like:

  ```
  swap auto  /* disk 1 */
  swap cs80 E0100  0  /* disk 2 */
  ```

[1] All the examples assume disk 1 is the root device and disk 2 is at select code 14, bus address 1.

- Figure 6–2-c shows one disk with no default swap area after the file system, and a second disk dedicated to swapping. This configuration is difficult to create. The installation process automatically leaves some swap space on the root disk. If you wish to create this configuration, you must remake your file system on one disk (leaving no swap space) and allocate all the space on a second disk for swap space. When you remake your file system, you will use the entire disk for the file system, then put the following line in `/etc/conf/dfile`:

```
swap cs80 E0100 0 /* disk 2 */
```

- Figure 6–2-d shows the default configuration.

  Use `newfs` to create your file system (refer to "Creating a New File System" in Chapter 4).

You can also decrease the file system size (and therefore increase your swap space size) by remaking your file system. This is described in the next section.

How to Remake Your File System for More Swap Space

Decreasing your file system's size increases your swap size. This option would look similar to Figure 6–2-d, but the swap space would be larger—less file system space.

The default swap space is determined from the values in `/etc/disktab`: swap space = disk size - s0×DEV_BSIZE, where *s0* is the size of the file system in 1 Kbyte blocks. Disk size can be found in `/etc/disktab` under the `hpxxxx_noswap` entry as `s0`. If there is no swap space, disk space is equal to the file system size.

Note If you increase the size of the swap space on your file system disk without re-making the file system, you may lose parts of your file system and the system will probably crash.

To rebuild your file system you can either use another root disk or you can re-install your system. To rebuild your file system using another root disk, follow these steps:

1. Determine how big you wish your swap space to be using the guidelines in this chapter.

2. From your required swap size, determine the size of your file system. All the sizes below are in Kbytes.

   file system size = disk size − swap size

   For example, if you need 20 480 blocks of swap space on your HP 7945 disk, you would allocate 36 352 blocks for your file system:

   file system size = 56 832−20 480 = 36 352 blocks

3. Install a second hard disk on your system (refer to the section "Adding Peripheral Devices" in Chapter 4) and initialize it (refer to the section "Initializing Media" in Chapter 5).

4. Create another root file system on a second hard disk with the `newfs` command using the *file_system_size* you determined in step 2. You can either specify the new file system size using the `-s` option to `newfs` or by changing the values in the `/etc/disktab` file:

   a. Using the `-s` option.

      In the example in step 2, you would make your file system using the command line:

      ```
      newfs -s 36352 /dev/rhd2 hp7945
      ```

      Refer to the section "Creating A New File System" in Chapter 4 for specific details on the file system creation procedure.

   b. Using the `/etc/disktab` file.

      To change the swap space, change `s0` and `nc` in your `/etc/disktab` file so that: s0 = ns×nt×nc, where

ns	is the number of 1024 byte sectors per track
nt	is the number of tracks per cylinder
nc	is the number of cylinders in the file system

      The *ns, nt,* and *nc* values are in the `/etc/disktab` file. Again, change only `s0` and `nc`.

5. Mount the new file system.

   mount *device_file* **/disc**

6. Copy root file system.

```
cd /
find . `ls | grep -v disc` -print | cpio -pdum /disc
```

Re-installing HP-UX to Increase Swap Space

If you have only one hard disk, and must increase swap space, you must re-install HP-UX, specifying a larger than default swap size. You cannot specify a disk to be all swap, or all file system, during the installation process.

If you choose to re-install HP-UX, you must perform the following steps:

1. Determine how big you wish your swap space to be using the guidelines in this chapter.
2. Back up the entire file system using the procedures in Chapter 5, the section "Backing Up and Restoring the File System".
3. Re-install HP-UX, this time specifying the larger swap size in the "Disk Parameters Menu". Refer to the *HP-UX Installation Manual* for instructions on how to install HP-UX.
4. Restore your files from the backup created in step 2.

Configuring Operating System Parameters

This section describes the operating system parameter information to put into your configuration description file (`/etc/conf/dfile`). To actually reconfigure your kernel, use the procedure described in the section "Using the Config Command".

HP-UX allows you to configure certain operating system related parameters. These operating system parameters determine how your operating system manages memory, limits table sizes, and determines other operating system limits. HP-UX operating system parameters should not be modified unless you fully understand the ramifications of doing so.

There are two types of operating system parameters: operating system parameters and System V IPC code capability. Each parameter is described in detail in Appendix D (Volume 2), "Operating System Parameters". Read the entry for the operating system parameter thoroughly before you attempt to change the parameter.

The format for specifying operating system parameters in the configuration description file (`/etc/conf/dfile`) is:

parameter [decimal-number or formula]

where

parameter is one of the operating system parameters described in Appendix D (Volume 2).

number is either a decimal number or an octal number in C format.

formula is an arithmetic expression made up of numeric constants (in C syntax) and previously specified tunable parameters. There cannot be any forward references, i.e. you cannot use names of parameters that are defined later in the dfile.

HP-UX Operating System Parameters

The operating system parameters that you can configure are:

- *accounting code parameters*—used by system accounting. Parameters: `timeslice`, `acctsuspend`, `acctresume`.
- *time information*—used to determine the time from Greenwich Mean Time and differences due to daylight savings time. Parameters: `dst`, `timezone`.
- *parity errors*—selects action for a RAM parity error. Parameter: `parity_option`.
- *limiter for system resource allocation*—used to calculate values of other global operating system parameters. Parameter: `maxusers`.
- *file system parameters*—number of open files, number of open inodes in the system, number of file system buffer headers, number of file locks, maximum number of shared text descriptors, and file size limit for processes. Parameters: `nfile`, `ninode`, `nbuf`, `nflocks`, `ntext`, `filesizelimit`.
- *process maximums*—the maximum number of processes per user and per system. Parameter: `maxuprc`, `nproc`.
- *maximum number of kernel timeouts*—maximum number of timeouts which can be scheduled by the kernel at a time. Parameter: `ncallout`.
- *user process size limits*—maximum data, stack, and text size. Refer to the description in the section "User Process Size Limits" below. Parameters: `maxdsiz`, `maxssiz`, `maxtsiz`.
- *memory parameters*—memory guaranteed to be available for the virtual memory system and/or system overhead at any time and size of area used by `argdev`. Parameter: `unlockable_mem`, `argdevnblk`, `dos_mem_byte`.
- *pseudo-teletypes*—maximum number of pseudo-teletypes. Parameter: `npty`.
- *number of DIL open device files*—Parameter: `ndilbuffers`.
- *floating point accelerator capability*—Parameter: `fpa`.

- *HP-UX cluster parameters*—configures arrays, swap and other HP-UX cluster structures. Parameters: dskless_node, using_array_size, server_node, serving_array_size, maxswapchunks, minswapchunks, dskless_cbufs, dskless_mbufs, ngcsp, num_cnodes, dskless_fsbufs,selftest_period.
- *total number of ITE text lines*—Parameter: scroll_lines.
- *LAN parameters*—Parameter: num_lan_cards, netmeminit, netmemmax, netmemthresh.

User Process Size Limits

Each of the operating system parameters is described in Appendix D (Volume 2), "Operating System Parameters". The operating system parameters associated with process size interact with each other and need to be explained as a group.

Your process's address space consists of text space, data space, stack space, and possibly some shared memory segments. Figure 2-10 (Chapter 2) shows how the parameters, dmmin, dmmax, dmshm, dmtext, maxtsiz, maxdsiz, maxssiz, shmmaxaddr, and shmbrk control your process's space.

For example, maxssiz limits the stack size and will stop infinite recursive program. Remember that the total process size is limited on a Model 310 to 16 Mbytes, and on other Series 300 computers to 4 Gbytes, regardless of the size of these parameters.

If you change these parameters, make sure they can still work together. Table D-1 at the end of Appendix D (Volume 2) relates these parameters. Also refer to Chapter 2, the subsection "Logical Address Space Management", for a description of user process space.

System V IPC Code

System V InterProcess Communication (IPC) code is included in your HP-UX kernel by default. It is not needed to run HP-UX, but one of your applications may use the System V IPC functions (for example HP Windows/9000). Leave the System V IPC code in your kernel unless you are sure you don't need it.

The format for specifying System V IPC code in config is the same as for other operating system parameters:

parameter (decimal-number or *formula)*

where *parameter* is one of the System V IPC operating system parameters described in Appendix D (Volume 2).

There are 3 parts to System V IPC code: messages (mesg), semaphores (sema), and shared memory (shmem). Each of these parts will be included in your kernel unless you specifically exclude them (i.e.mesg=0, sema=0, shmem=0).

For each of the System V IPC parts, there are several associated tunable parameters. If the part is set to 0, you cannot change any of its associated parameters. If the part is set to 1 (include in kernel), you may change any of its associated parameters.

The tunable parameters associated with the three parts of System V IPC code are:

- *mesg*—kernel code used for System V IPC messages. Parameters: `msgmap`, `msgmax`, `msgmnb`, `msgmni`, `msgseg`, `msgssz`, `msgtql`.
- *sema*—kernel code used for System V IPC semaphores. Parameters: `semaem`,`semmap`, `semmni`, `semmns`, `semmnu`, `semume`, `semvmx`.
- *shmem*—kernel code used for System V IPC shared memory. Parameters: `shmall`, `shmbrk`, `shmmax`, `shmmaxaddr`, `shmmin`, `shmmni`, `shmseg`.

Configuring Device Drivers and I/O Cards Using reconfig

You can create an operating system that is big or small depending on which optional drivers you include in your kernel. A small kernel will give you better performance than a large kernel.

Note Each optional driver requires main memory space. If you have a small amount of main memory, then add only the kernel drivers you need for your system.

The optional drivers you can choose from are shown in Table 6–1, shown earlier in this chapter.

You can use either the `reconfig` program or the `config` program to create a new operating system. Using the `reconfig` program is easier than using the `config` program. However, if you are configuring additional swap space or operating system parameters into your kernel, you must first edit the configuration description fille (usually `/etc/conf/dfile`), then use the custom operating system interface in `reconfig`.

You have three choices when you create an operating system, or kernel, from the `reconfig` program. Each choice is described in detail in the following sections. You can create:

- the minimum operating system (containing a minimal, yet functional, kernel)
- the fully loaded operating system (containing all kernel device drivers applicable to your type of system)
- a customized operating system

Creating the Minimum Operating System

A minimum operating system contains all kernel drivers that are typically needed on a small system, including those for HP Windows/9000. The LAN driver needed for XWindows is included in all minimum kernels except those for a Cluster root server or cnode. The `reconfig` program creates the minimal operating system described in the configuration description files shipped with your system. The configuration description file used depends on what type of system you are on: standalone, cluster root server, or cluster diskless cnode.

Type of System	**Dfile Used**
standalone	`/etc/conf/dfile.min`
cluster root server	`/etc/conf/dfile.minservr`
diskless cnode	`/etc/conf/dfile.cnode`

To create the minimal operating system, perform the following steps:

1. Shut down your system and login as the superuser root.
2. Start the `reconfig` program by typing:

   ```
   /etc/reconfig
   ```

   The main menu of `reconfig` will appear on your screen (Figure 6–3).

```
**************************************************
            RECONFIG -- MAIN MENU
**************************************************

User Configuration
Workstation Configuration
Cluster Configuration









[        ] [        ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 6–3. Reconfig Main Menu

Using the [NEXT] and [PREVIOUS] keys, move the cursor to the `Workstation Configuration` menu item and press the [SELECT] softkey. You will see the screen shown in Figure 6–4.

```
************************************************
    RECONFIG -- WORKSTATION CONFIGURATION MENU
************************************************

Adding terminals and modems
Configure the line printer spooler
Create a minimally loaded operating system
Create a fully loaded operating system
Create a custom operating system









[        ] [        ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [PREVMENU] [        ]
```

Figure 6–4. Workstation Configuration Menu

If you do not see the final three items then you do not have the ACONFIG fileset loaded onto your system. This must be on your system to change your operating system. Exit the `reconfig` program by first pressing the [PREVMENU] key to get to the main menu, then press the [QUIT] softkey to exit `reconfig`. Then use the procedure in "Installing Optional Software and Updating HP-UX" to load the ACONFIG fileset.

3. Using the [NEXT] and [PREVIOUS] keys, move the cursor to the `Create a minimally loaded operating system` menu item and press the [SELECT] softkey. You will see the following message on your screen:

```
Please wait while the new operating system
is created (this may take several minutes) ...

This will not overwrite your current operating
system until the next time you reboot your system.
```

This will not overwrite your current operating system until you specifically request it (see step 4 for instructions). The new operating system is written into the file called `/etc/conf/hp-ux.new`. Associated with this new operating system is a configuration description file, called `/etc/conf/dfile.new`. It lists all the kernel drivers in your new operating system.

You will see the workstation menu for `reconfig` on your screen again.

Note The new operating system is not used until you reboot. If you intended to create the minimal operating system because you needed to increase your available main memory, this will not happen until you reboot. Step 4 explains how to automatically reboot.

---

4. You should now see an additional item in the workstation menu of `reconfig`. This item is called `Reboot system using new kernel`.
   - If you select the reboot option, the old kernel will be overwritten[2] and your new kernel will contain the minimal operating system listed above. HP-UX will now reboot.
   - If you do not want to reboot, press [Prev] to return to the main menu, then press the [QUIT] softkey. You will be asked again if you wish to reboot. Type `no`, then press the [Return] key. Messages on your screen will tell you how to reboot HP-UX later. You will now be returned to your shell.

---

[2] On a standalone system or the root server of an HP-UX cluster, the old kernel is actually moved to `/SYSBCKUP`. This way, if your new kernel doesn't work, you can reboot with your old kernel using the `SYSBCKUP` operating system from attended mode of the boot ROM. If you are creating a kernel for a diskless cnode of an HP-UX cluster, the `SYSBCKUP` file is not, and should not be, replaced.

Creating a Fully Loaded Operating System

With the maximum operating system, more main memory is used than with a smaller operating system. This may cause your larger applications to run slower. If you have a small main memory and your system does not require all available kernel drivers then you should create a minimum operating system or a custom operating system.

The maximum operating system depends on the type of system you are running: standalone, cluster root server, or cluster diskless cnode. The maximum operating system is created from sample configuration description files shipped with your system. The maximum operating system is configured as follows:

Type of System	Dfile Used
standalone	`/etc/conf/dfile.full.lan`
cluster root server	`/etc/conf/dfile.maxservr`
diskless cnode	`/etc/conf/dfile.cnodemax`

To create the maximum operating system, perform the following steps:

1. Shut down your system and log in as the superuser root.
2. Start the `reconfig` program by typing:

   ```
   /etc/reconfig
   ```

   The main menu of `reconfig` will appear on your screen (Figure 6–5).

```
**************************************************
            RECONFIG -- MAIN MENU
**************************************************

User Configuration
Workstation Configuration
Cluster Configuration









[        ] [        ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 6–5. Reconfig Main Menu

Using the NEXT and PREVIOUS keys, move the cursor to the `Workstation Configuration` menu item and press the SELECT softkey. You will see the screen shown in Figure 6–6.

```
**************************************************
    RECONFIG -- WORKSTATION CONFIGURATION MENU
**************************************************

Add terminals and modems
Configure the line printer spooler
Create a minimally loaded operating system
Create a fully loaded operating system
Create a custom operating system







[        ] [        ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [PREVMENU] [        ]
```

Figure 6-6. Rconfig Workstation Configuration Menu

If you do not see the final three items then you do not have the ACONFIG fileset loaded onto your system. This must be on your system to change your operating system. Exit the `reconfig` program by first pressing the PREVMENU softkey to return to the main menu, then by pressing the QUIT softkey to exit `reconfig`. Then use the procedure in "Installing Optional Software and Updating HP-UX" to load the ACONFIG fileset.

3. Using the NEXT and PREVIOUS softkeys, move the cursor to the `Create a fully loaded operating system` menu item and press the SELECT softkey. You will see the following message on your screen:

```
Please wait while the new operating system
is created (this may take several minutes) ...

This will not overwrite your current operating
system until the next time you reboot your system.
```

This will not overwrite your current operating system until you specifically request it (see step 4 for instructions). The new operating system is in a file called `/etc/conf/hp-ux.new`. Associated with this new operating system is a configuration description file called `/etc/conf/dfile.new`. It lists all the kernel drivers in your new operating system.

You will see the workstation menu for `reconfig` on your screen again.

Note The new operating system is not used until you reboot. Step 4 explains how to automatically reboot.

---

4. You should now see an additional item in the main menu of `reconfig`. This item is called `Reboot system using new kernel`.
   - If you select the reboot option, the old kernel will be overwritten[3] and your new kernel will contain the maximum operating system (an operating system that contains all the drivers HP-UX supports). HP-UX will now reboot.
   - If you do not want to reboot, press [Prev] to return to the main menu, then press the [QUIT] softkey. You will be asked again if you wish to reboot. Type `no`, then press [Return]. Messages on your screen will tell you how to reboot HP-UX later. You will now be returned to your shell.

Creating a Custom Operating System

A custom operating system contains only the parts (drivers) you choose to include. The `reconfig` program will use any existing configuration description file as a starting point when you create a custom kernel. All swap, swap size, and operating system parameter entries will be used in building the new kernel, though these entries cannot be modified or deleted (or new entries added) via `reconfig`.

---

3 On a standalone system or the root server of an HP-UX cluster, the old kernel is actually moved to `/SYSBCKUP`. This way, if your new kernel doesn't work, you can reboot with your old kernel using the `SYSBCKUP` operating system from attended mode of the boot ROM. If you are creating a kernel for a diskless cnode of an HP-UX cluster, the `SYSBCKUP` file is not, and should not be, replaced.

1. Decide what parts of the operating system you need to include by following the instructions at the beginning of this section.
2. Shut down your system and log in as the superuser root.
3. Start the `reconfig` program by typing:

   ```
   /etc/reconfig
   ```

   The main menu of `reconfig` will appear on your screen (Figure 6-7).

```
**************************************************
            RECONFIG -- MAIN MENU
**************************************************

User Configuration
Workstation Configuration
Cluster Configuration









[        ] [        ] [  NEXT  ] [PREVIOUS]  [  HELP  ] [ SELECT ] [        ] [  QUIT  ]
```

Figure 6–7. Reconfig Main Menu

Using the [NEXT] and [PREVIOUS] keys, move the cursor to the `Workstation Configuration` menu item and press the [SELECT] softkey. You will see the screen shown in Figure 6–8.

```
*************************************************
    RECONFIG -- WORKSTATION CONFIGURATION MENU
*************************************************

Adding terminals and modems
Configure the line printer spooler
Create a minimally loaded operating system
Create a fully loaded operating system
Create a custom operating system









[        ] [        ] [  NEXT  ] [PREVIOUS]   [  HELP  ] [ SELECT ] [PREVMENU] [        ]
```

Figure 6-8. Workstation Configuration Menu

If you do not see the final three items then you do not have the ACONFIG fileset loaded onto your system. This must be on your system to change your operating system. Exit the `reconfig` program by first pressing the [PREVMENU] softkey to return to the main menu, then by pressing the [QUIT] softkey to exit `reconfig`. Then use the procedure in "Installing Optional Software and Updating HP-UX" to load the ACONFIG fileset.

4. Using the [NEXT] and [PREVIOUS] softkeys move the cursor to the `Create a custom operating system` menu item and press the [SELECT] softkey. You will see a screen *similar* to that shown in Figure 6-9.

   This screen shows the sample configuration description files shipped with your system that you can use as a basis for your custom operating system.

```
************************************************
          RECONFIG -- CREATE A CUSTOM OS
************************************************

Sample Dfile    Description of the kernel created

dfile  Current kernel (/hp-ux)
dfile.BCKUP Existing backup (/SYSBCKUP)
dfile.min Minimum standalone
dfile.full Maximum Workstation, no networking (default)
dfile.full.lan Maximum Workstation with networking



Enter the name of the dfile? >> dfile
```

Figure 6-9. Custom Operating System Menu

5. Choose the kernel description file.

   Enter the name of the kernel description file to use as a base for your custom operating system. By HP-UX convention, your current kernel is described in the default: `dfile`. Once you have chosen the kernel description file, you will see a screen *similar* to that shown in Figure 6-10.

```
**************************************************
        RECONFIG -- DEVICES IN CUSTOM OS
**************************************************

Are the following TYPES OF DEVICES connected to your system?

CS/80 Storage Devices            yes
Plotters:                        yes
Ciper Printers:                  no
Non-Ciper Printers:              yes
Amigo Storage Devices:           no
9-Track Tapes:                   no
Streaming 9-Track Tapes:         no
Optical Autochangers:            no



Are the above values ok (y or n)? >>



[RESTORE] [ ] [ ]  [ ]  [HELP] [ ] [PREVMENU] [ ]
```

Figure 6–10. Devices in Custom Operating System Menu

6. Verify the values for devices in the custom operating system.

   This form allows you to indicate which types of devices should be supported by your custom operating system.

   If the defaults are correct, then type y and [Return] and continue with Step 7.

   If you type n you will see a series of forms, each corresponding to one of the device types. Each form will list Hewlett-Packard model numbers. If you have any of the part numbers connected to your system, respond yes to the prompt for that device type, else respond no. Pressing just [Return] causes the displayed value to be used. Model numbers are shown on the front of most devices.

   The forms will look *similar* Figure 6–11.

```
Below is a list of supported PLOTTERS.
xxxxx     xxxxx     xxxxx

       Are any of these connected to your system (y or n)? >> Yes


[RESTORE] [       ] [       ] [       ]   [HELP] [       ] [PREVMENU] [       ]
```

Figure 6-11. Choosing Kernel Device

After cycling through all of this series of forms you will return to the screen shown in Figure 6–10 to recheck the values. Once the values are correct, answer y, and you will see the menu shown in Figure 6–12.

7. Verify the values for interface cards in the custom operating system. The screen looks *similar* to that shown in Figure 6–12.

```
******************************************************
       RECONFIG -- INTERFACE CARDS IN CUSTOM OS
******************************************************

Following INTERFACE CARDS used on your system?

98625 (hi speed disk):          yes
98265 (SCSI disk):              no
98628 (datacomm):               no
98642 (4-channel mux):          yes
98626 (rs-232):                 yes
98644 (rs-232):                 yes
98622 (gpio):                   no
98286 (dos co-processor)        no
98646 (BCD vme):                no
98577 (vme2):                   no


Are the above values ok (y or n)? >>


RESTORE                          HELP           PREVMENU
```

Figure 6–12. Interface Cards Menu

This form allows you to indicate which optional interface cards you have on your system.

If the defaults are correct, then type y and [Return], and continue with Step 9.

If you type n you will see a series of questions, each corresponding to one of the interface cards. If you have that interface card in your system, respond yes to the prompt for that interface card, else respond no. Pressing just [Return] causes the displayed value to be used. The product numbers are shown on the front plate of the interface cards. The interface cards are in the back of your computer.

If you have a Model 350 you must include the 98625 and 98626 drivers. The Model 350 has these interfaces built-in.

The questions will look similar to that shown in Figure 6–13.

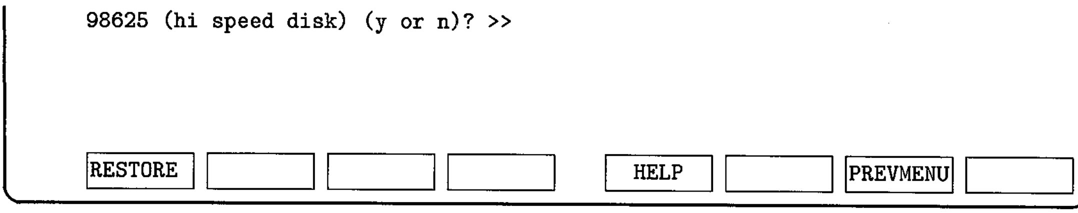


Figure 6–13. Choosing Interface Card Kernel Device Drivers

After cycling through all of this series of forms you will return to the screen labeled "RECONFIG—INTERFACE CARDS IN CUSTOM OS" to re-check the values.

8. Verify the values for optional software products in the custom operating system. The screen looks similar to that shown in Figure 6–14.

```
*****************************************************
      RECONFIG -- OPTIONAL PRODUCTS IN CUSTOM OS
*****************************************************

Are the following OPTIONAL PRODUCTS included in your system?

LAN (ieee 802):          yes
LAN (ethernet):          yes
SRM:                     yes
RJE:                     yes
HP-Windows/9000:         yes
X Version 11 Windows:    yes
Diskless:                no
RFA:                     no
NFS:                     no




Are the above values ok (y or n)? >>

[RESTORE] [ ] [ ] [ ]   [ HELP ] [ ] [PREVMENU] [ ]
```

Figure 6-14. Optional Software Products Menu

This form allows you to indicate which optional software products you have on your system.

Note that before including kernel drivers for optional software into your operating system you must have the optional software product loaded onto your system. The basic system does not include the files required to build an operating system containing optional software kernel drivers. If you wish to include these, you must press the [PREVMENU], then the [QUIT] softkey to exit `reconfig` and use the procedure in "Installing Optional Software and Updating HP-UX" to load the optional software product.

If the defaults are correct, then type y and [Return], and continue with Step 10.

If you type `n` you will see a series of questions, each corresponding to one of the optional software products. If you have that optional software product on your system, respond `yes` to the prompt for that software product, or else respond `no`.

Pressing just [Return] causes the displayed value to be used.

The questions will look similar to that shown in Figure 6–15.

```
LAN (ieee 802) (y or n)? >>



RESTORE                          HELP                PREVMENU
```

Figure 6–15. Choosing Optional Software Products

After cycling through all of this series of forms you will return to the screen labeled "RECONFIG—OPTIONAL PRODUCTS IN CUSTOM OS" to re-check the values.

9. After all the forms have appeared and you have verified them, `reconfig` will create your new operating system. You will see the following message on your screen:

```
Please wait while the new operating system
is created (this may take several minutes) ...

This will not overwrite your current operating
system until the next time you reboot your system.
```

This will not overwrite your current operating system until you specifically request it (see Step 11 for instructions). The new operating system is in a file called `/etc/conf/hp-ux.new`. Associated with this new operating system is a configuration description file called `/etc/conf/dfile.new`. It lists all the kernel drivers in your new operating system.

You will see the workstation menu for `reconfig` on your screen.

Note The new operating system is not used until you reboot. This means that new drivers cannot be accessed until you reboot. Step 11 explains how to automatically reboot.

10. You should now see an additional item in the main menu of `reconfig`. This item is called `Reboot system using new kernel`.

    - If you select the reboot option, the old kernel is overwritten [4] and your new kernel supports each of the items you chose in the previous screens. HP-UX will now reboot.
    - If you do not want to reboot, press [PREVMENU] to return to the main menu, then press the [QUIT] softkey. You will be asked again if you wish to reboot. Type `no`, then press [Return]. Messages on your screen will tell you how to reboot HP-UX later. You will now be returned to your shell.

[4] On a standalone system or the root server of an HP-UX cluster, the old kernel is actually moved to `/SYSBCKUP`. This way, if your new kernel doesn't work, you can reboot with your old kernel using the `SYSBCKUP` operating system from attended mode of the boot ROM. If you are creating a kernel for a diskless cnode of an HP-UX cluster, the `SYSBCKUP` file is not, and should not be, replaced.

Creating a New Kernel in an HP-UX Cluster

The kernel, /hp-ux, is a CDF. It was created such that each cnode has its own /hp-ux file. For example, if you had a cluster with three cnodes, daisy, donald, and dewey, the /hp-ux CDF would have the structure shown in Figure 6-17.

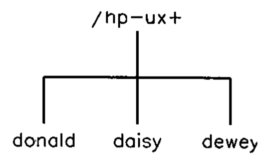


Figure 6–16. The Kernel as a CDF

When creating a new kernel you should log into the appropriate cnode (either physically log into the cnode or use an `rlogin` across the network). This will ensure you get the correct version of /hp-ux. *Do not overwrite the backup kernel (*/SYSBCKUP*).* /SYSBCKUP is *not* a CDF.

There are several configurable operating system parameters for HP-UX clusters. They are fully documented in Appendix D (Volume 2). Table 6–2 lists the new parameters and shows the default values. Suggested settings for these parameters are already in the sample kernel description files in `/etc/conf`: `dfile.cnodemin`, `dfile.maxservr`, `dfile.cnode`, `dfile.cnodemax`, and `dfile.minservr`.

The default for these parameters is the suggested setting for a standalone system. Defaults that are shown as `o/#` have a default of 0 for standalone systems, and a default of # if the diskless code is configured into the kernel. These #s are defined in the section following the table.

Table 6–2. HP-UX Cluster O.S. Parameters

Cluster Parameters	Default	Diskless Cnode	Root Server
dskless_node	0	1	0
using_array_size	0/#1	#1	#1
server_node	0	0	1
serving_array_size	0/#2	#2	#2
maxswapchunks	512	512	512
minswapchunks	4	4	4
dskless_cbufs	0/#3	#3	#3
dskless_mbufs	0/#4	#4	#4
ngcsp	#5	#5	#5
num_cnodes	0	0	5 if 0-5 cnodes 10 if 6-10 cnodes 15 if 11-15 cnodes 20 if 16-20 cnodes
selftest_period	0/120	120	120
dskless_fsbufs	0/#6	#6	#6

Defaults for the HP-UX Cluster O.S. Parameters Table

#1 using_array_size is set to nproc if the dskless code is configured. Using default values for all parameters, this value is 84.

#2 serving_array_size is set to (server_node * numcnodes * maxusers) + (2 * maxusers) if the dskless code is configured. Using default values for all parameters, this value is 16.

#3 dskless_cbufs is set to dskless_mbufs * 2 if the dskless code is configured. Using default values for all parameters, this value is 12.

#4 dskless_mbufs is set to serving_array_size + (2*using_array_size)/32) +1 if the dskless code is configured. Using default values for all parameters, this value is 6.

#5 ngcsp is set to 8*num_cnodes. Using default values for all parameters, this value is 0.

#6 dskless_fsbufs is set to serving_array_size if the dskless code is configured. Using default values for all parameters, this value is 16.

Local and Remote Swapping in an HP-UX Cluster

The root server's swap space is treated the same as swap space for a standalone system. Processes on the root server will be swapped to a physically connected, or local, swap device. All other cnodes in the cluster can either swap locally or swap to the root server. A cnode can swap to only one location (although there may be several swap devices at that location). To have local swapping, `/etc/clusterconf` must reflect it in the <SWAP CNODE> field, `/etc/checklist` must be created for the diskless node, and `/hp-ux` must be a kernel configured specifically for the cnode.

The four steps for setting up local swap are:

1. Connect the disk drive.

   If you have not already connected the disk drive that will be your swap device, do so now. You will need to create a block device file for the disk drive. Make sure you are logged in as root on the local cnode so the device file is created with the correct context. Refer to the chapter called "Customizing your HP-UX System" for details.

   You do not need to create a file system on the disk if it will be used only as swap space. You do need to initialize it using the `mediainit` command.

2. If you have more than one swap device, edit /etc/checklist

   If you have only one local swap device, skip this step: the system will automatically swap to the one device.

   You must add an entry to your cnode's `/etc/checklist` file if you have more than one local swap device. `/etc/checklist` is a CDF. The entry should reflect the new swap disk.

   This entry will be used to automatically enable the swap device when you issue the `/etc/swapon` command in your `/etc/rc` script file.

3. Edit /etc/clusterconf

   Field 5, the swap server field, must specify the cnode's ID number. If field 5 is 1 (root server cnode id), or any number other than the local cnode's ID number, the cnode will swap to the root server. This field was set to 1 when you added the cnode using the `reconfig` program.

For example, if your root server is daisy, cnode 2 (donald) is swapping to the root device, and cnode 3 (dewey) is swapping locally, you might have the following entries in your /etc/clusterconf file:

```
0800090039dd: # clustercast addresses.  Do not remove.
0800090039dd:1:daisy:r:1:8
080009000565:2:donald:c:1:0
08000900297c:3:dewey:c:3:0
```

4. Create a new kernel

   The kernel, /hp-ux, is a CDF.

   The sample description file, /etc/conf/dfile.cnode, contains the recommended minimum kernel configuration for a diskless cnode. You must make a copy of this and then edit that copy as follows:

   a. cd /etc/conf

   b. Delete the * in front of cs80 or scsi, depending on which type of disk drive you are swapping to. If cs80, delete the * in front of the 98625 if this disk is connected to this type of interface.

   c. Add the swap entry just before the line * SWAP CONFIGURATION.

      If your new disk is dedicated to swapping, the swap entry should be (assuming the disk is located at select code 14, bus address 0):

      ```
      swap cs80 E0000 0
      ```

      If your disk contains an HP-UX system to boot as standalone, and you wish to use the swap part of the file system as a local swap device, then your swap entry should be (assuming the disk is located at select code 14, bus address 0):

      ```
      swap cs80 E0000 -1
      ```

   You now need to create a new kernel with the /etc/config command or reconfig command. Follow the instructions earlier in this chapter, the section called "Using config". *Do NOT back up the existing kernel* by moving it to /SYSBCKUP. Although /hp-ux is a CDF, SYSBCKUP, the backup kernel, is not. If you back up your kernel into /SYSBCKUP you will overwrite your root server's backup kernel.

Glossary

The following terms and definitions are defined as they apply to the Series 300 HP-UX operating system.

address

In the context of peripheral devices, a set of values which specify the location of an I/O device to the computer. The address is composed of up to four elements: select code, bus address, unit number and volume number. You can read about addresses in the *Peripheral Installation Guide* and in "Adding/Moving Peripherals" in Chapter 5 of this manual.

access

An interaction between a subject and an object that results in the flow of information from one to the other.

access control mechanism (ACM)

An algorithm and data structure that supports access control decisions. It mediates decisions about whether specific subjects can access specific system objects in specific ways. An ACM might be implicit (contextual),e.g. "only user ID 0 can access files in /etc". It might be explicit, as in a list of objects and their access rights. An explicit ACM might be distributed (data stored with protected objects), as in the UNIX file permissions scheme, or it might be centralized, as in a file describing access rights for all protected system objects.

ARPA hostname

Needed for ARPA Services. A name consisting of one field containing any printable character except spaces, newlines, or the pound sign (#). Assigned in `/etc/hosts`.

attribute

In an HP-UX cluster, the attributes are characteristics of the cnodes, such as the cnode name or processor type. The attributes are collectively referred to as the cnode's `context`.

audit trail

A set of records that collectively provide documentary evidence of processing, used to aid in tracing from original transactions forward to related records and reports, and/or backwards from records and reports to their component source transactions.

block

The fundamental unit of information HP-UX uses for access and storage allocation on a mass storage medium. Block size on an HFS file system can be either 4 Kbytes or 8 Kbytes, and is set at file system creation. The default block size is 8 Kbytes. However, to present a more uniform interface to the user, most system calls and utilities use "block" to mean 512 bytes, independent of the actual block size of the medium.

block mode

Buffered I/O. With buffered I/O data is held in the buffer cache, then transferred one block at a time. Block size for buffered I/O is not the same as block size on the file system. Block size for buffered I/O is defined as BLKDEV_IOSIZE in `/usr/include/sys/param.h`. Compare with **raw mode.**

boot or boot-up

The process of loading, initializing and running an operating system.

boot area

The first 8 Kbytes of the disk that are reserved for system use. This area contains the LIF volume header, the directory that defines the contents of the volume, and the bootstrapping program.

boot ROM

A program residing in ROM (Read Only Memory) that executes each time the computer is powered-up. The function of the boot ROM is to run tests on the computer's hardware, find some devices accessible through the computer and then load either a specified operating system or the first operating system found according to a specific search algorithm. The bootstrap program uses the boot ROM's mass storage or LAN drivers to load and pass control to the kernel. When the kernel gains control, it completes the job of bringing up the HP-UX operating system.

Depending on your boot ROM version, the boot ROM displays may differ slightly from those shown in this manual; any differences between boot ROM versions are noted in this manual when the topic in question is discussed. The boot ROM identifies its version when power is applied to the computer.

bus address

Part of an address used for devices, especially devices on an HP-IB (HP Interface Bus); a number determined by the switch setting on a peripheral which allows the computer to distinguish between two devices connected to the same interface. A bus address is sometimes called a "device address", and no two devices on the same HP-IB can have the same bus address.

bytes per inode

This specifies the number of data bytes (amount of user file space) per inode slot. The number of inodes is calculated as a function of the file system size. The default value is 2048.

certification

The technical evaluation of a system's security features, made as part of and in support of the approval/accreditation process, that establishes the extent to which a particular computer system's design and implementation meet a set of specified security requirements.

CDF

Context Dependent File. The mechanism used by HP-UX clusters to share a path name between different cnodes in a cluster. A CDF consists of a hidden directory and one or more subfiles.

client

A process that is requesting some service from another process. In this manual, client is sometimes used as another name for a diskless cnode.

cluster

An HP-UX cluster is one or more workstations linked together with a local area network (LAN), but consisting of only one file system.

cluster node (cnode)

A computer in a cluster.

cluster server process (CSP)

A special kernel process that runs on a machine in a cluster to satisfy requests from other nodes in the cluster. There are two kinds of CSP, the limited CSP (LCSP) and the general CSP (GCSP).

cnode

*C*luster *node*. A computer in a cluster.

connect session

This denotes the period of time in which a user is connected to the system. It starts when the user logs in and finishes when the user logs out.

context

An ASCII string made up of a number of attributes that ordinarily determines which subfile (if any) is accessed from a CDF. Each workstation on a cluster has an associated context which is set at boot time. The context attributes are: cnode name; processor type; floating point hardware; file system location; and the string "default".

cron

This process wakes up every minute to execute commands at specified dates and times, according to instructions in files contained in the directory `/usr/spool/cron/crontabs`. See the *cron(1M)* and *crontab(1)* entries in the *HP-UX Reference* for more details.

CS/80

A family of mass storage devices that communicate with a computer via the CS/80 (Command Set '80) or SS/80 (Sub Set '80) command set. In HP-UX your file system can be on either a SCSI drive or a CS/80 drive.

cylinder

One or more vertical collections of tracks in a disk pack (refer to Figure Glossary–1 later in this section). Disk acceses within a cylinder do not need a seek.

cylinder group

One or more consecutive cylinders (refer to Figure Glossary–1 later in this section). Each cylinder group contains a superblock, inodes, cylinder group information, and data blocks.

cylinder group information

A data structure located in the cylinder group that contains information about the cylinder group such as numbers of available inodes, data blocks, and fragments, and bitmaps to free space in the cylinder group.

DASS

Direct Access Secondary Storage. A new area of the mass storage business structure that falls between online magnetic disk primary storage and offline magnetic tape secondary storage.

destination device

The mass storage device on which HP-UX is to be installed or updated. The destination device must be a hard disk drive.

Discretionary Access Control (DAC)

A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject (unless restrained by mandatory access control).

diskless cnode

A cnode that does not have a local file system. Its file system resides, remotely, on the root server.

diskless workstation

The same as a diskless cnode.

driver number

Same as major number.

disk

The term used for a collection of recording platters contained in a single disk drive. Disk is synonomous with disk pack.

domain

The set of objects that a subject has the ability to access. A set of (object, rights) pairs. Each pair specifies an object and some subset of the operations that can be performed on it. A right in this context means permission to perform one of the operations.

etc/rc

This is the system initialization shell script. The actions that it performs depend on the state in which it is invoked. To automatically start System Accounting whenever the system is switched to multi-user mode, a command must be added to `rc`. See the chapter "System Boot and Login" in this manual, and *rc(1M)* in the *HP-UX Reference* for more details on the use of `rc`.

etc/shutdown

A shell script that has the primary function of terminating all currently running processes in an orderly and cautious manner. See *shutdown(1M)* for details on this shell.

exchange

An exchange is when an optical autochanger replaces one disk surface in the optical drive with another.

file

A discrete collection of information described by an inode and residing on a mass storage medium.

file types

Several file types are recognized by HP-UX. The file type is established at the time of the file's creation. The types are:

- Regular files - Contains a stream of bytes. Characters can be either ASCII or non-ASCII. This is generally the type of file a user considers to be a file: object code, text files, nroff files, etc.
- Directory - HP-UX treats directories like regular files, with the exception that writing directly to directories is not allowed. Directories contain information about other files.
- Block special files - Device files that buffer the I/O. Reads and writes to block devices are done in block mode.
- Character special files - Device files that do not buffer the I/O. Reads and writes to character devices are in raw mode.
- Network special files - contain the address of another system.
- Pipes - A temporary file used with command pipelines. When you use a pipeline, HP-UX creates a temporary buffer to store

information between the two commands. This buffer is a file, and is called a pipe.

- FIFO - A named pipe. A FIFO (First In/First Out) has a directory entry and allows processes to send data back and forth.
- symbolic link - A type of file that indirectly refers a path name.

file system
: The organization of files and directories on a hard disk. The HFS file system is an implementation of the HP-UX directory structure.

fragment
: A piece of a block. This is the smallest unit of information HFS will read or write. The lower limit of a fragment is DEV_BSIZE (defined in `/usr/include/sys/param.h`). Fragment size is set at file system creation.

free space threshold
: Specifies minimum *percentage* of free disk space allowed. Once the file system capacity reaches this threshold, only the superuser is allowed to allocate disk blocks. The default is 10%; if it is less, file system performance degrades. The free space threshold is set when you create a new file system.

HFS file system
: High performance File System. This is the file system implemented on your Series 300 HP-UX system. Other models of Hewlett-Packard computers running HP-UX may have a different file system.

hidden directory
: A directory used to implement a CDF. It is called hidden because it is normally treated and seen as a file. It can be accessed as a directory only by appending the special character "+" to its name.

HP-UX directory structure
: The hierarchical grouping of directories and files on HP-UX.

HP-UX system hostname
: Name you assign to your system (from a line on `/etc/rc`). It is used for UUCP, mail, and other programs.

inode

A data structure containing information about a file such as file type, pointers to data, owner, group, and protection information.

internet address

An address used by NS-ARPA Services. It consists of two parts: a network number and a host number. Nodes on the same LAN will have the same network number and distinct host numbers.

ITE

The Internal Terminal Emulator program which allows a bit-mapped display to function as a standard computer terminal.

kernel

The core of the HP-UX operating system. The kernel is the compiled code responsible for managing the computer's resources; it performs such functions as allocating memory and scheduling programs for execution. The kernel resides in RAM (Random Access Memory) whenever HP-UX is running.

LIF

Logical Interchange Format. LIF is Hewlett-Packard's standard file format, used for transferring files between Hewlett-Packard systems. Since LIF is a standard, files with LIF format can easily be transferred between different Hewlett-Packard computers (refer to the *LIF(4)* entries in the *HP-UX Reference*).

link level address

A unique 12-digit hexadecimal number which is part of every LAN card. This number appears on the LAN card hardware, on the boot ROM screen, and can be obtained using the `landiag` program.

login

The process of a user gaining access to HP-UX. This process consists of entering a valid user name and its associated password (if one exists).

magneto-optical

Magneto-optical is a form of rewritable optical technology.

major number

An index into a table in the kernel. It is needed to use peripheral devices. The major number for a particular device can be found in the tables in "Adding/Moving Peripheral Devices" in the "Customizing" chapter.

mechanical picker or mechanical changer

The part of the optical autochanger that moves optical disks from slot to drive and vise versa.

media

In terms of optical products, this is the optical disk that holds the data. The term includes the plastic cartridge that houses the optical disk.

minor number

A hexidecimal number made up of a select code and other information specific to the peripheral device you are setting up. Refer to "Adding/Moving Peripheral Devices" in the "Customizing" chapter for information on minor numbers.

MO

See **magneto-optical**

multi-user run-level

A run-level of HP-UX when terminals in addition to the system console allow communication between the system and its users. The multi-user run-level (not to be confused with a multi-user system) is run-level 2 as shipped.

MUX

MUX is an abbreviation for Asynchronous Multiplexer. The HP 98642 four channel MUX is available for the Series 300. Each channel is an RS-232C port which is normally associated with a `/dev/ttyXX` file.

named object

Objects which have names, are visible at the TCB interface, and are shared among users.

NCSC

National Computer Security Center, the government agency that wrote the guidelines for trusted systems.

node

A computer on a network.

NS_ARPA Services

A combined networking product providing both NS and ARPA services. The NS_ARPA Services networking product enables your Series 300 to transfer files to and from remote hosts, log into remote hosts, execute commands on remote hosts, and send mail to and receive mail from remote hosts on the network.

NS nodename

Needed for Network Services and the `rlb` daemon. A name consisting of three fields separated by periods, i.e. node.domain.organization. Each field can contain up to 16 alphanumeric, case insensitive characters.

objects

A passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, networks nodes, etc.

optical autochanger

A rewritable optical mass storage peripheral which includes the mechanics to move optical disks in and out of drive(s), the drive(s), media and controller electronics.

password

A private character string that is used to authenticate an identity.

path name

A series of directory names separated by / characters, and ending in a directory name or a file name.

primary storage

Typically consists of fixed hard disk(s), used for fast, random-access applications. The primary storage devices are used as online system disks.

process

A process is the environment in which a program (or command) executes. It includes the program's code, data, status of open files, and value of variables. It is completely characterized by a single current execution point and address space. For example, whenever you execute an HP-UX command, you are creating a process; whenever you log in, you create a process. For additional information on processes, read the chapter "Concepts."

raw mode

Unbuffered I/O. Data is transferred directly between the device and the user program requesting the I/O, rather than going through the file system buffer cache. Compare with **block mode.**

rewritable optical

An optical disk technology which can be repeatedly written.

resource

Anything used or consumed while performing a function. The categories of resources are: time, information, objects (information containers), or processors (the ability to use information). Specific examples are CPU time; terminal connect time; amount of directly-addressable memory; disk space; number of I/O requests per minute.

root

Root refers to the highest level directory (root directory or /).

root server

The cnode in an HP-UX cluster with the local file system, capable of supporting other diskless workstations.

secondary storage

The storage device(s), typically tape drives, used to back up and archive data stored on the system disks (primary storage). Secondary storage is also used to log transaction, interchange data and distribute software. Secondary storage devices always use removable media.

security policy

The set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information.

select code

Part of an address used for devices; a number determined by switch settings on the interface card. The select code determines the interface card's location in the processor address space. Each interface card is in turn connected to a peripheral. Multiple peripherals connected to the same interface card share the same select code.

sensitive information

Information that, as determined by a competent authority, must be protected because its unauthorized disclosure, alteration, loss, or destruction will at least cause perceivable damage to someone or something.

shell

A program that interfaces between the user and the operating system. HP-supported shells are:

- `/bin/sh`
- `/bin/csh`
- `/bin/ksh`
- `/bin/rsh`
- `/bin/rksh`
- `/bin/pam`

source device

The mass storage device from which HP-UX is installed. The source device must be a cartridge tape drive or flexible disk drive.

special file

Often called a device file, this is a file associated with an I/O device. Special files are read and written just like ordinary files, but requests to read or write result in activation of the associated device. These files normally reside in the `/dev` directory.

standalone

A machine which is not part of an HP-UX cluster.

subfile

Part of a CDF in an HP-UX cluster. The subfiles are under the hidden directory, and are named for one of the system's context attributes.

subject

An active entity, generally in the form of a person, process, or device that causes information to flow among objects or changes the system state. Technically a process/domain pair.

superblock

A data structure containing global information about the file system such as file system size, disk information, and cylinder group parameters. The superblock is created at the same time as the file system and is replicated into each cylinder group.

surface

In terms of optical disks, this is one of the disk sides—surface 1 or 2.

system administrator run-level

A run-level of HP-UX when the system console provides the only communication mechanism between the system and its users. Init state s is the system administrator run-level.

system console

A keyboard and display (or terminal) given a unique status by HP-UX and associated with the special device file `/dev/console`. All boot ROM error messages (messages sent prior to loading HP-UX), HP-UX system error messages, and certain system status messages are sent to the system console. Under certain conditions (for example, the single-user state), the system console provides the only mechanism for communicating with HP-UX.

TCSEC

Trusted Computer Systems Evaluation Criteria, also known as the "Orange Book". This is the book where evaluation criteria for trusted systems is documented.

track

One of several concentric circles on the surface of a disk upon which data is recorded (refer to Figure Glossary–1).

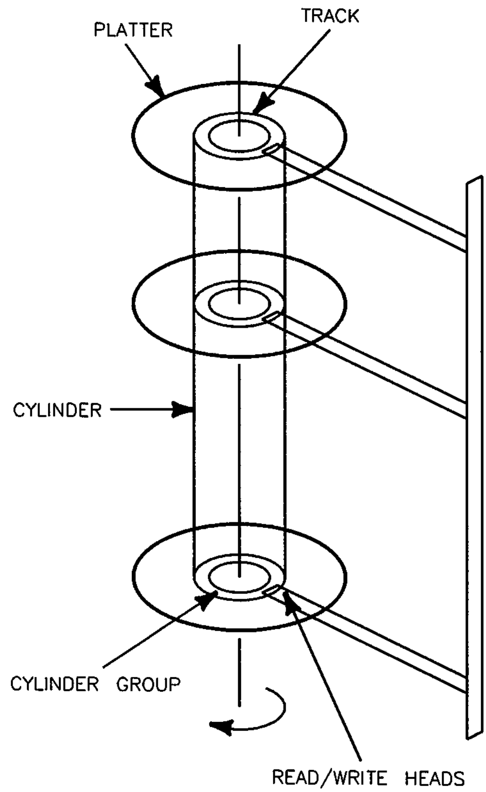


Figure Glossary–1. Track, Cylinder, and Cylinder Group on a Disk

trap door

A hidden software or hardware mechanism that permits system protection mechanisms to be circumvented. It is activated in some non-apparent manner (e.g., special "random" key sequence at a terminal).

Trojan horse

A computer program with an apparently or actually useful function that contains additional (hidden) functions that surreptitiously exploit the legitimate authorizations of the invoking process to the detriment of security. For example, making a "blind copy" of a sensitive file for the creator of the Trojan Horse.

trusted computer system

A system that employs sufficient hardware and software integrity measures to allow its use for processing simultaneously a range of sensitive or classified information.

trusted computing base (TCB)

The totality of protection mechanisms within a computer system—including hardware, firmware, and software—the combination of which is responsible for enforcing a security policy. A TCB consists of one or more components that together enforce a unified security policy over a product or system. The ability of a TCB to correctly enforce a security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (e.g., a user's clearance) related to the security policy.

trusted process

A process that is restricted and is only run by a user with appropriate privilege. For HP-UX 6.5 a trusted process is a process with effective user ID of 0 (superuser).

trusted software

The software portion of a Trusted Computing Base.

unit number

Part of an address used for devices; a number whose meaning is software- and device-dependent but which is often used to specify a particular disk drive in a device with a multi-drive controller. When referring to single-controller integrated disk/tape or disk/flexible disk drive, a unit is used to distinguish between disk and cartridge tape drives or hard disk and flexible disk drive.

The unit number also selects a single partition on the 913x series.

user

Any person who interacts directly with a computer system.

UUCP

Unix-to-Unix File Copy Package, mainly used for mail transport

verification

In a trusted system, the process of comparing two levels of system specification for proper correspondence (e.g., security policy model with top-level specification, TLS with source code, or source code with object code). This process may or may not be automated.

volume number

Part of an address used for devices; a number whose meaning is software- and device-dependent but which is often used to specify a particular volume on a multi-volume disk drive. The volume number is also used to inform the device driver of special handling semantics (such as printer drivers skipping over perforations).

Index

A

B

C

D

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

HEWLETT
PACKARD

HP Part Number
98594-90060

Microfiche No. 98594-99060
Printed in U.S.A. E1288

98594-90660
For Internal Use Only