HP 9000 Series 200/300 Computers

HEWLETT PACKARD

HP-UX System Administrator Manual

HP-UX
HP-UX
HP-UX
HP-UX
HP-UX
HP-UX

HP-UX System Administrator Manual
for HP 9000 200/300 Computers

Manual Reorder No. 97033-90049

Hewlett-Packard Company

3404 East Harmony Road, Fort Collins, Colorado 80525

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. A vertical bar in the margin indicates the changes on each 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 1985...Edition 1

Table of Contents

Chapter 3: Concepts

Chapter 4: System Boot and Login

Chapter 6: System Accounting

Appendix A: Using the FSCK Command

Appendix B: System Accounting Files

Appendix C: Partitions and File Sets

Appendix D: System Parameters

Getting Started 1

Welcome

This manual is written for you, the Series 200 or Series 300[1] HP-UX system administrator. Although some familiarity with computers is assumed, this manual will serve people with varying levels of expertise. If you are already a UNIX[2] expert, you will find much here that is familiar but may well encounter something new. The HP-UX Operating System is composed primarily of Bell Laboratories' System V.2 UNIX. 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 Bell'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 when something goes wrong. Additionally, the system administrator should become an HP-UX "guru", the local expert to whom other HP-UX users go for help.

[1] Unless specifically noted, all information in this manual is correct for both the Series 200 and the Series 300 HP-UX system.
[2] UNIX is a trademark of AT&T Bell Laboratories.

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 which 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: Installing HP-UX

This chapter 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.

Chapter 3: Concepts

This chapter discusses processes, IDs, the super-user, block and character input/output, the file system, file protection, and memory management.

Chapter 4: System Boot and Login

Your system performs many tasks between the time power is applied to the computer running HP-UX and the time a user has logged in (gained access) to the system. This chapter examines what happens and offers guidelines for modifying various files that affect this process. It also describes some configuration files and suggests how to use them to customize your system.

Chapter 5: The System Administrator's Toolbox

Arranged alphabetically by task, this chapter contains instructions for accomplishing tasks the system administrator generally performs.

Chapter 6: System Accounting

As system administrator you may want to periodically evaluate how well your Series 200/300 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.

Conventions Used in this Manual

Naming Conventions

The following naming conventions are used throughout this manual.

- Italics indicate files and HP-UX commands, system calls, subroutines, etc. found in the *HP-UX Reference* manual as well as titles of manuals. Italics are also used for symbolic items either typed by the user or displayed by the system as discussed below. Examples include */etc/profile*, *date(1)*, *getpid(2)* and *tty(4)*. The parenthetic number shown for commands, system calls and other items found in the *HP-UX Reference* is a convention used and described in that manual.

- The first time a file is mentioned, the complete pathname is given; subsequent references to that file contain only the final file name unless there is some chance of ambiguity.

- Boldface is used when a word is first defined (as **flebnee**) and for general emphasis (**do not touch**).

- Computer font indicates a literal either typed by the user or displayed by the system. Keys are shown enclosed in an oval. A typical example is:

  `fsck /dev/7912` [Return]

  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* [Return]

  In this case you would type in your own *device_file_name*.

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

- Unless otherwise stated, all references such as "see the *login(1)* entry for more details" refer to entries in the *HP-UX Reference* manual. 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.

Keyboard Conventions

While installing the HP-UX system, you need to know about two keys: [Back space] and [Return]. The [Back space] key will let you fix typing mistakes made when entering information. Press [Back space] to "back up" over a mis-typed letter.

Most information typed at the system console must be followed by the [Return] key. All Hewlett-Packard terminals supported by HP-UX have a [Return] key.

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 200/300 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 *AXE User's Guide* contains a wealth of information for both the average user and the system administrator. It explains how to use PAM, Windows/9000, *vi*, and applications. It also describes a second (easier) method to perform many system administration tasks.
- 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 200/300 HP-UX Operating System. Use this manual when looking for complete specifications of a given command, special file, etc.
- The seven volume set of *HP-UX Concepts and Tutorials* contains information on a broad range of HP-UX topics and tools. Several sections may be of particular interest to you: *Serial Network Communications*, *The Bourne Shell*, and *vi*.
- Manuals provided with optional products, such as Local Area Network (LAN), provide information specific to the product.
- *Series 300 Configuration Reference Guide* provides information on all products available to your Series 200/300 computer.
- *Peripheral Installation Guide* describes how to set switches and connect peripherals to your Series 200/300 computer.

In addition to the above manuals, your documentation package includes a tutorial, *A User Guide to the UNIX System*.

Single-user vs. Multi-user Systems

The Series 200/300 HP-UX Operating System is supplied as either a single-user or multi-user system (with a 16 user license). The difference between the two types of systems is implied by their names — a single-user system can only be used by one person at a time; a multi-user system, by many at a time.

If you have a single-user system, many of the multi-user topics covered in this manual will still benefit you. Consider the discussion on how to set up and configure the LP Spooler (used to control line printer spooling). While this topic may be more critical in multi-user systems (where the demand on system resources is generally higher), using the LP Spooler on a single-user system can increase the performance and flexibility of that system.

If you have a single-user system and a procedure or task seems irrelevant, you probably can ignore it. Don't reject multi-user topics too quickly though — they often contain information useful for single-user systems.

If you have a multi-user system, but have not chosen system security your system will act like a single-user system—you will not be able to use remote terminals, so only one person can be logged in. See "After Installing HP-UX" in Chapter 2 for a discussion on choosing system security.

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 modify 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 discs. The operating system is installed on a Command Set '80 (CS/80) or Subset '80 (SS/80) hard disc drive. As system administrator, you are responsible for installing HP-UX. Instructions for accomplishing this task are provided in chapter 2, "Installing HP-UX".

Configuring HP-UX

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 5, "The System Administrator's Toolbox", or the instructions in chapter 10 of the *Application Execution Environment Users 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 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, "The System Administrator's Toolbox", or the instructions in chapter 10 of the *Application Execution Environment Users Manual*.

Adding and Moving Peripheral Devices

Another of your responsibilities is to add/move peripherals (printers, terminals, mass storage devices, etc.) to the HP-UX system as they are required. A list of peripherals supported by the Series 200/300 HP-UX system can be found in the *HP 9000 Series 300 Configuration Reference Manual* supplied with your system. Directions for installing the peripherals can be found in chapter 5, "The System Administrator's Toolbox", or the instructions in chapter 10 of the *Application Execution Environment Users 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, "The System Administrator's Toolbox".

Updating the HP-UX System

You will receive software updates by purchasing HP support services that provide periodic updates. These updates modify 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 5, "The System Administrator's Toolbox".

Backing Up and Restoring the System and Creating a Recovery 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, when the system becomes sufficiently corrupt, file system errors increase rapidly.

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

As system administrator, you should make a **backup** - a copy of the HP-UX Operating System, file system and programming languages - and a **recovery system** - a bootable subset of HP-UX.

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.

Depending on your system usage, consider backing up the system on a daily basis. Generally, base the frequency of your system backups on the answer to the question "How much data can I afford to lose?". Instructions for using the supplied *backup* command is given in chapter 5, "The System Administrator's Toolbox". Instructions for using the *mkrs* command are given in chapter 5, "The System Administrator's Toolbox".

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(1M)* 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, repairs) the file system. You should, in addition, check the file system whenever you observe unexpected system behavior. Continuing to use a corrupt file system only further corrupts the system. Instructions for verifying and repairing the file system are located in the "Using the FSCK Command" appendix. Also, see the *fsck(1M)* entry in the *HP-UX Reference* manual.

Assisting Other Users

Since you carry the title "System Administrator", users may come to you for help with the system. You should plan on allocating 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 HP.

Providing a "Back-up" Administrator

At least one other person should be trained as the system administrator to handle your responsibilities in the event of your absence.

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. Chapter 4 ("System Boot and Login") and chapter 5 ("The System Administrator's Toolbox") show you how to create programs that automatically perform tasks at specific times. By scheduling programs using system routines, you can automatically back up the system or initiate communications between your system and another HP-UX system (using the *uucp* utilities).

Notes

User Survey

Name ____________________ Location ____________________

Phone ____________________

Location where you will be using the system. ____________________

____

User Category

☐	Engineers and Managers	(run existing application program; enter data, 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 device
- ☐ Other ______________________________

Are there other users with whom you want to share programs or data? ☐ yes ☐ no

If so, list them. ____

____

Will you be generating or using large amounts of data? __________________________

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

____

____

What long term data storage does your application require? ______________________

____

____

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

____

Which programs are interactive, which will be run in a background mode? Can any programs be run overnight? ____

____

Installing HP-UX 2

You received your HP-UX system on several flexible discs or on two cartridge tape. Installing HP-UX involves inserting the discs or tapes and loading the contents onto your hard disc. After installation, the hard disc contains the information needed to boot HP-UX when you turn power on to your computer. This chapter provides a step-by-step procedure for installing the HP-UX Operating System on HP 9000 Series 200 and 300 computers.

You must read and fill out the checklist items before installing HP-UX. This will save you time on installation, and may prevent you from needing to re-install HP-UX. Each item on the checklist tells you where to look for the information needed to fill out the checklist item.

Installation Overview

For your convenience, here is an overview of the major steps involved in the software installation of the HP-UX Operating System. You should already have your hardware installed, using the guidelines in *Peripheral Installation Guide*.

1. Fill out the pre-installation checklist on the following page.
2. Load the first of the HP-UX installation flexible discs or tapes.
3. Follow and respond to the installation utility menus.
4. Make the necessary changes to your HP-UX system using the *reconfig* program.
5. Store your tapes or flexible discs in a safe place.
6. Follow the guidelines in the "After Installing HP-UX" section

Pre-Installation Checklist

Fill out this checklist before you begin the installation procedure. Information from this checklist will be needed during installation.

File Sets and Partitions

File sets and partitions are explained in Appendix C. You **must** install the filesets from the "SYS_CORE" partition. Other filesets are optional. Go through the list of filesets in Appendix C and decide which optional filesets you wish to load.

The required filesets are already filled in.

Partition:	SYS_CORE	Partition:	____________
Filesets:	98515	Filesets:	____________
	ACORE		____________
	ACORE2		____________
	ACORE3		____________
	ACORE4		____________
	ACORE5		____________
Partition:	____________	Partition:	____________
Filesets:	____________	Filesets:	____________
	____________		____________
	____________		____________
	____________		____________
	____________		____________
	____________		____________

Swap Space Required

Swap space is an area that will be created on your hard disc (destination disc) when your file system is created (during the installation process). You must determine your swap space requirements now; if you require more than the default amount of swap space, and don't request it during the installation process, you will need to re-install HP-UX. Swap space is discussed in Chapter 3, the section "Memory Management", and in Chapter 5, the section "Configuring HP-UX". "Configuring HP-UX" discusses how to determine the amount of swap space you need.

Swap space required (in Kbytes): ____________

Hardware information.

Does your system have a 98620B DMA card? ___YES___

Will your destination (root) disc be connected to an HP98625A high-speed HP-IB interface card? ___YES___

Will you be installing from cartridge tape or from flexible disc? ___cartridge___

Will you be installing from an integrated device? i.e. is the flexible disc or cartridge tape drive part of the hard disc drive? ___NO___

If you are NOT installing from an integrated device, you must fill in (for the destination device) the following address information (If you set up your drives using the guidelines in the *Peripheral Installation Guide*, you can fill in the information from your worksheet:

- Select Code ___14___
- Bus Address ___0___
- Unit Number ____________
- Volume Number ____________

Major Number is always 0.

Before Installing HP-UX

You should already have your computer, hard disc drive, installation tape/flexible disc drive, and all interface and memory cards installed and tested. Follow the guidelines in the *Peripheral Installation Guide* to install your hardware if you have not already done so.

Your Destination Disc

The hard disc drive you will install HP-UX onto is called the **destination disc** in this chapter. If you have information on your destination disc that you do not want destroyed, you must save the information now. The installation process will overwrite everything on your destination disc.

If you have never used this disc, you do not need to worry about this caution.

Set up your System Console

The **system console** is a keyboard and monitor (or terminal) 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.

See Chapter 4, "System Boot and Login" for a description of the system console.

Installing HP-UX from Flexible Discs

NOTE

The whole installation process can take anywhere from one to three hours depending upon the size of the disk being initialized and the number of products you decide to load.

1. Turn your computer off before you load the tape or disc.
2. Turn on the hard disc drive and the drive you will use for the installation.
3. Insert the first installation flexible disc.

   Find the discs from the SYS_CORE partition (the discs with no label color (grey discs)).

   Locate the write-protect mechanism; see Figure 2-1 to locate the write-protect mechanism. For the installation process to be successful, your system **must** be able to write to the first two discs. The write protect mechanism is located on the back of the disc. Move the protect mechanism away from the label end of the disc, as shown in Figure 2-1.

   Hold the flexible disc with the label up and towards you and insert the first installation disc into the left side of the disc drive and push until it clicks into place.

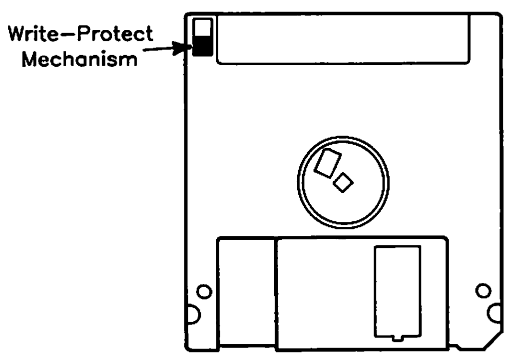


Figure 2-1: Flexible Disc Write Protect Mechanism

4. After the busy light on your flexible disc drive shuts off, turn on your computer and system console display.

NOTE

If you have any bootable system currently on your hard disc, you will need to enter the boot ROM's attended mode by pressing the space bar as soon as you turn your computer on (see discussion below).

If you are installing HP-UX to a disc with a bootable system on it, you need to immediately hold the space bar down during the boot ROM power-up sequence, and continue holding it down until the word "keyboard" appears on the left side of your screen. This stops the boot ROM from automatically booting an operating system if one exists on your hard disc.

You will see a series of messages on your console **similar** to those shown in Figure 2-2.

```
Copyright 1985,                                   :HP9122REM, 704, 0, 0
Hewlett-Packard Company.                             1H  SYSHPUX
All Rights Reserved.                                 1B  SYSBCKUP


BOOTROM REV.A
Bit Mapped Display
MC68020 Processor
MC68881 Coprocessor
Keyboard
HP-IB
HP98644 at 9
HP98625 at 14
2096992 Bytes




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

Figure 2-2: First Installation Messages

5. Choose the correct operating system.

   On the right side of your screen, you will see one or more HP product numbers, followed by a list of operating systems found on the disc drive associated with the product number. Choose the HP-UX operating system (`SYSHPUX`) corresponding to the product number of the drive that contains the installation disc. In the example shown in Figure 2-2, using an HP9122 disc drive for installation, type [1] [H] [Return].

   It will take a couple of minutes for the installation program to be loaded. Follow the installation utility menus as instructed in the following steps.

   The message

   ```
   Booting /hp-ux
   ```

   appears on the screen and the system starts loading the HP-UX operating system off the flexible disc. The HP-UX operating system now displays messages pertaining to the system configuration, how much memory is left, and the system copyrights.

The HP-UX operating system loads some more information, then the copyright information will disappear and a new menu will appear (shown in Figure 2-3) to ask you whether you wish to begin or end the installation procedure.

```
*****************************************************
       HP-UX INSTALLATION UTILITY -- MAIN MENU
*****************************************************



   Choice         Description

     b            BEGIN installation process
     e            EXIT installation process


Please enter choice (b or e), hit [Return] >>
```

Figure 2-3: Main Installation Menu

6. Begin the installation process.

NOTE

The installation procedure will destroy all the files on the destination hard disc, if any exist.

- If this is the first time your are installing HP-UX on this hard disc, or you are sure you don't want files that currently exist on the disc, enter [b] [Return]. A screen similar to that shown in Figure 2-4 will appear. Continue with step 7.
- If you have files on your destination disc (i.e. your destination disc has information on it) and need to save any files, type [e] [Return] to exit the installation process. If you wish to save any of the files currently on your disc, exit the installation procedure and reboot your old system, backing up any files you want to save.

```
**************************************************
      HP-UX INSTALLATION UTILITY -- DEVICE MENU
**************************************************


Source Device                      Destination Device


Major number      4                Major Number          0
Select code      14                Select code          14
Bus address       0                Bus address           0
Unit number       1                Unit number           0
Volume number     0                Volume number         0


Choice           Description

[Return]         CONTINUE installation process
   e             EXIT installation process
   d             change DESTINATION device


Please enter choice (e or d), hit [Return] >>
```

Figure 2-4: Device Menu

7. Verify the destination device.

   The screen you now see (similar to Figure 2-4) shows you the source address (installation drive) and default destination address (root disc drive). The major numbers for both the source and destination devices will be correct.

   The source address will be correct, but you must check the destination address (select code, bus address, unit number, and volume number). If your source device (installation drive) and destination device (root disc drive) are integrated (both housed in the same box, product numbers HP9153A or HP9133D/H/L), these values will be correct. If your source and destination devices are not integrated, you will need to change the destination device address. **You should have the address written on your pre-installation checklist.**

   You may perform any of the following tasks now:

   - If you wish to continue with the installation process, using the device parameters as shown, press [Return] (be careful not to press in several returns). The message

   ```
   Checking values...
   ```

   will be displayed.

   Go to step 8.

   - If you wish to exit the installation process, type [e] and [Return].
   - If you wish to change the address for the destination device, type [d] and [Return]. You will be prompted for the select code, bus address, unit number, and volume number. These are discussed in your *Peripheral Installation Guide*, and should be written on your pre-installation checklist.

   To change a value, type in the new value followed by [Return]; if you just hit [Return] the old value will be retained.

   When you are finished changing the values, the message `Making device file` will be displayed on the screen.

   You will be returned to the screen shown in Figure 2-4. To continue the installation process, hit [Return].

8. After you hit Return (from step 7 above), the message

```
Checking values...
```

will be displayed. *Install* will verify that the information you gave is correct if you have only one hard disc connected. If you have more than one hard disc, you must verify these numbers yourself.

The system will check your destination device, and depending on what kind of device you have, you may or may not get the following message(s):

```
Will your root disc be on a 98625A (disc HP-IB interface card)? [y/n] >>
```

```
Does your system have a 98620B (DMA) card? [y/n] >>
```

You should be able to answer these questions from the pre-installation checklist. When you have answered these questions, the installation will continue.

The disc parameters menu (Figure 2-5) will appear on your console.

9. Verify the values displayed on the disc parameters menu.

You normally should not change these numbers. However, some applications may require non-default hard disc parameters.

Swap space is the most common parameter to change. If your applications require more than the default amount of swap space, it is important that you change swap space now. **If you do not have enough swap space, you may need to re-install HP-UX later.** You should have your swap space requirements filled out on the pre-installation checklist.

```
****************************************************
 HP-UX INSTALLATION UTILITY -- DISC PARAMETERS MENU
****************************************************


Swap Size:        10192     Interleave Factor:         1
Block Size:        8192     Free Space Threshold:     10
Fragment Size:     1024     Rpms of the disc:       3600
                            Density- bytes / inode:  2048


Choice              Description


[Return]            CONTINUE installation process
   e                EXIT installation process
   b                change disc BLOCK size
   d                change DENSITY (bytes per inode)
   f                change disc FRAGMENT size
   i                change disc INTERLEAVE factor
   r                change disc RPMS
   s                change disc SWAP size
   t                change disc free space THRESHOLD size


Use the values shown above unless you need to change them for tuning.
Enter choice followed by [Return] or hit [Return] to continue >>
```

Figure 2-5: Disc Parameters Menu

If you do not wish to change any of the hard disc parameters, type Return to continue.

If you wish to change the any of the values, type the letter corresponding to the choice and Return. Enter the new value followed by a Return to enter the new value. This will display a new message at the bottom of the screen:

```
Please enter choice (e,b,d,f,i,r,s, or t), hit [Return] >>
```

If you want to change any other parameters, enter the appropriate choice.

When you have changed the parameters you want, go to step 10.

10. Press [Return] to continue with the installation.

You should now see a menu similar to the menu shown in Figure 2-6. This menu is shown so you can verify the correct address for the destination device. If the information is correct, go to step 11. If any information is incorrect, enter [e] and [Return] to exit the installation procedure.

```
**************************************************
     HP-UX INSTALLATION UTILITY -- INSTALL MENU
**************************************************


               DESTINATION DEVICE


           Major  Number              0
           Select Code               14
           Bus    Address             0
           Unit   Number              0
           Volume Number              0



Choice         Description


   b           BEGIN Initialization and Installation
   e           EXIT Installation Process



Please enter choice (b or e), hit [Return] >>
```

Figure 2-6: Installation Device Menu

11. Press [b] and [Return] to begin the initialization of the destination disc and the installation of HP-UX.

The following message and prompt will appear at the bottom of your screen:

```
WARNING !!!!! Proceeding will DESTROY the  contents of unit xxx, volume xxx
of the Destination Device. Do you want to proceed? [y/n] >>
```

CAUTION

Do not switch off power to the computer or hard disc drive during disc initialization. Terminating the initialization process in this fashion may seriously corrupt the disc medium.

The **only** way to install HP-UX is to type in [y] and [Return]. **If you think that any of the parameters you entered on previous menus are wrong, or if you do not wish to erase any information on the destination disc**, type [n] and [Return]. Answering n will exit the installation procedure.

If you entered [y], disc initialization begins. The menu, as shown in Figure 2-7, informs you the destination disc is being initialized. The source drive will also be accessed during the initialization process. The media initialization process sets up your hard disc for normal use.

Disc initialization can take from four minutes to fifty minutes depending on the size of your disc. See Table 2.1 at the end of this chapter for approximate initialization times. In addition to initializing the disc, the installation process will create a file system and load important files to the hard disc. **The total time before you must interact with the installation process again is between 30 minutes and 3 hours.**

You will see a trace of the files being copied, similar to the display shown in Figure 2-7.

If you decide to abort the installation process, type [e] and [Return].

```
***************************************************
   HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
***************************************************


Initializing:  /dev/rhd
This will take between fifteen minutes and three hours

Making file system on device:  /dev/hd
Mounting device:   /dev/hd
Making directory:  /dev
Making device file: /dev/console
Making device file: /dev/mem
Making device file: /dev/kmem
Making device file: /dev/null
Making device file: /dev/swap
Making device file: /dev/tty
Making directory:  /etc
Making directory:  /disc
Making directory:  /usr
Making directory:  /usr/bin
Copying file: /hp-ux
Copying file: /etc/copyinit
Copying file: /etc/sbtab
```

Figure 2-7: Installation Trace #1

After the trace shown in Figure 2-7, the trace will disappear, and the following two lines will appear:

```
Copying file: /etc/mkfs
Copying file: /etc/mediainit
```

HP-UX then automatically reboots (this takes about 2 minutes), then prompts you for new media:

```
Please insert media "2" from partition: "SYS_CORE"
when BUSY light remains off, hit [Return] >>
```

Disc 2 must not be write-protected. Insert the second flexible disc from SYS_CORE. You will see message:

```
...verifying...
```

If you loaded the wrong flexible disc, you will be prompted to load the correct disc.

The installation process will now load files from the second disc; you will see the execution trace similar to Figure 2-8.

```
*****************************************************
   HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
*****************************************************


Mounting device: /dev/src
Making directory: /bin
Making directory: /usr
Making directory: /usr/bin
Making directory: /etc
Making directory: /tmp
Making directory: /etc/filesets
Copying file: /disc/bin/cpio
Copying file: /disc/bin/mkdir
Copying file: /disc/bin/sh
Copying file: /disc/bin/pwd
Copying file: /disc/usr/bin/lifcp
Copying file: /disc/usr/bin/tcio
Copying file: /disc/etc/init2
Copying file: /disc/etc/update
Copying file: /disc/etc/sysrm
```

Figure 2-8: Installation Trace #2

After loading the tools, the screen will clear, and the following menu will appear on your screen:

```
*****************************************************
   HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
*****************************************************

setting up device files...
reading contents file...
```

The program then loads all the required files and their file sets and displays a screen similar to the screen shown in Figure 2-9. The "XXXXXXX" refers to the fileset number or fileset name.

install will prompt you to insert the rest of the core (required) discs. As you are prompted to insert discs, put the discs in the flexible disc drive. For each disc, the installation procedure will wait until you insert the disc, and will verify that you have inserted the correct flexible disc before continuing.

Each time you insert a disc, wait for 5 to 10 minutes while the files are copied from the flexible disc to the hard (destination) disc. These discs contain files needed for HP-UX.

```
***************************************************
    HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
***************************************************



Loading files from "XXXXXXX"
      .
      .
      .
Loading: etc/init
      .
      .
      .
```

Figure 2-9: Execution Trace #3

NOTE

You now have the core (required) files loaded on your system.

---

12. Exit the installation process or load more filesets.

If you wish to have only the core HP-UX system loaded, hit [e] and [Return] to exit the installation procedure.

If you wish to install optional filesets or products, hit [d] and [Return]. You will see the following line added to Figure 2-10:

```
Please insert disc
```

Decide which disc contains the filesets you want to load (from your pre-installation checklist or by looking in Appendix C). Insert the disc containing the filesets. You will now see the menu in Figure 2-11.

```
********************************************************
     HP-UX INSTALLATION UTILITY -- FILESET  MENU
********************************************************

Choice          Description


e               EXIT install process
d               DISPLAY installation options for a new fileset

Please enter choice (letter or number), hit [Return] >>
```

Figure 2-10: Fileset Menu #1

The menu shown in Figure 2-11 shows the filesets you can load from the disc you have chosen. You can now choose to load all the filesets on the disc, the individual filesets you want, or exit.

```
********************************************************
     HP-UX INSTALLATION UTILITY -- FILESET  MENU
********************************************************

Choice          Description


e               EXIT install process
d               DISPLAY installation options for a new fileset
a               Process ALL filesets listed below


1               fileset_1
2               fileset_2
3               fileset_3
4               fileset_4

Please enter choice (letter or number), hit [Return] >>
```

Figure 2-11: Fileset Menu

NOTE

The time it takes to complete this process varies; it takes about 15 minutes per flexible disc to copy optional partitions and filesets to your hard disc.

- If you wish to install all the filesets on this menu, press [a] and [Return].
- If you wish to install only certain filesets, enter the fileset number and hit [Return]. You will be returned to this menu when the fileset is loaded.
- If you choose to "DISPLAY installation options", press [d] and [Return].

  You will get a message: `insert media from a new partition`. Choose any flexible disc of the color of partition you wish to install, and press [Return]. The installation procedure will read the table of contents, and return you to a menu similar to the menu in Figure 2-11.
- If you wish to exit this menu, press [e] and [Return].

  If this is the first time you see this menu, and you decide to exit without loading anything, you will have only the core (required) files loaded on the system.

13. After you finish installing both the mandatory operating system (end of Step 11) and the optional pieces you want (end of step 12) your screen will clear and the screen shown in Figure 2-12 will appear:

```
****************************************************
   HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
****************************************************



unloading media...
When Busy light remains off, hit [Return] >>

cleaning up ...
```

Figure 2-12: Screen for Exiting Fileset Menu

The program will unload the media for you. Press [Return] when the light on the flexible disc drive remains off, and remove the media. **Remember to store all installation media in a safe place.**

14. The system will reboot.

    If your file system closed cleanly, a message indicating this appears. Otherwise, the *fsck(1M)* runs automatically and cleans up your file system. This may cause the system to reboot. This situation usually arises when the *shutdown(1M)* command was not used to shut down the system, or when a power failure occurs.

15. The program, *reconfig*, will start up. You should go through some of the reconfig items now, especially the "Security" item. For a description of *reconfig*, read the subsection, "Reconfiguring Your System" in "After Installing HP-UX" in this chapter.

    If you wish to exit *reconfig*, press the **Exit** softkey.

16. Follow the steps in the next section, "After Installing HP-UX".

Installing HP-UX from Cartridge Tape

NOTE

The whole installation process can take anywhere from one to three hours depending upon the size of the disk being initialized and the number of products you decide to load.

NOTE

If you are installing from a cartridge tape drive with an auto-changer (for example the HP 35401) follow the steps in step 3 of the installation procedure.

You are now ready to install HP-UX.

3. Insert the installation tape.

1. Turn your computer off before you load the tape.
2. Turn on the hard disc drive and the drive you will use for the installation.
3. Insert the installation tape.

   Locate the write-protect mechanism; see Figure 2-13 to locate the write-protect mechanism. For the installation process to be successful, your system **must** be able to write to the media.

   The mechanism is on the top, rear, left-hand corner of the cartridge tape and is labelled "SAFE". The arrow on the protect screw should point away from the word SAFE. If it does not, use a coin or screwdriver to turn the protect screw such that the arrow points away from the word SAFE.

   If you are installing HP-UX from an auto-changer tape drive, you do not need to manually change tapes when the first installation tape has finished unloading. To use the auto-changer, perform the following: turn the power to your tape drive off; put the tape drive in sequential mode (set the switch in back); place the first installation tape in slot 1, and the second installation tape in slot 2, of the magazine; put the magazine in the tape drive door; turn the power to your tape drive on.

   To insert a cartridge tape into a tape drive that is not an auto-changer, hold the tape with the SAFE label in the rear left hand corner and insert it into the tape drive door and push until it clicks into place. The BUSY indicator should now be lit. The tape drive will begin a cartridge tape conditioning sequence that takes approximately two minutes.

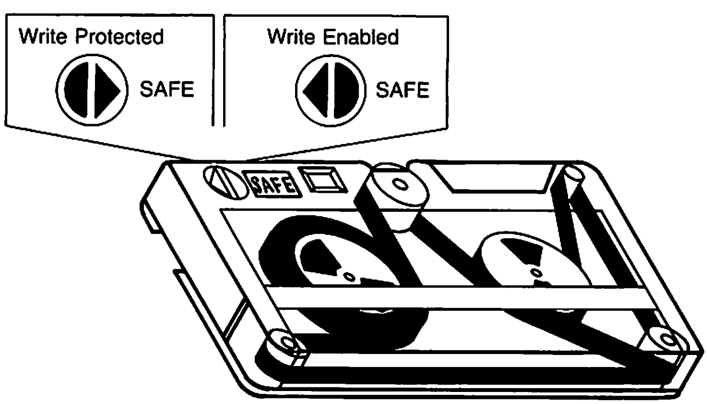


Figure 2-13: Tape Write Protect Mechanism

4. After the busy light on your tape drive shuts off, turn on your computer and system console display.

NOTE

If you have a bootable system currently on your hard disc, you will need to enter the boot ROM's attended mode by pressing the space bar as soon as you turn your computer on (see discussion below).

If you have a bootable system currently on your disc, you need to immediately hold the space bar down during the boot ROM power-up sequence, and continue holding it down until the word "keyboard" appears on the left side of your screen. This stops the boot ROM from automatically booting an operating system if one exists on your hard disc.

You will see a series of messages on your console **similar** to those shown in Figure 2-14.

```
Copyright 1985,                                   :HP9144TAPE, 704, 0, 0
Hewlett-Packard Company.                             1H  SYSHPUX
All Rights Reserved.                                 1B  SYSBCKUP


BOOTROM REV.A
Bit Mapped Display
MC68020 Processor
MC68881 Coprocessor
Keyboard
HP-IB
HP98644 at 9
HP98625 at 14
2096992 Bytes



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

Figure 2-14: First Installation Messages

5. Choose the correct operating system.

   On the right side of your screen, you will see one or more HP product numbers, followed by a list of operating systems found on the tape drive associated with the product number. Choose the HP-UX operating system (SYSHPUX) corresponding to the product number of the tape drive that contains the installation media. In the example shown in Figure 2-14, using an HP9144 tape drive for installation, type [1] [H] [Return].

It will take a couple of minutes for the installation program to be loaded. Follow the installation utility menus as instructed in the following steps.

The message

```
Booting /hp-ux
```

appears on the screen and the system starts loading the HP-UX operating system off the installation tape. The HP-UX operating system now displays messages pertaining to the system configuration, how much memory is left, and the system copyrights.

The HP-UX operating system loads some more information, then the copyright information will disappear and a new menu will appear (shown in Figure 2-15) to ask you whether you wish to begin or end the installation procedure.

```
*****************************************************
      HP-UX INSTALLATION UTILITY -- MAIN MENU
*****************************************************



   Choice         Description

     b            BEGIN installation process
     e            EXIT installation process


Please enter choice (b or e), hit [Return] >>
```

Figure 2-15: Main Installation Menu

6. Begin the installation process.

NOTE

The installation procedure will destroy all the files on a currently installed disc.

- If this is the first time your are installing HP-UX on this hard disc, or if you are sure you want to overwrite the contents of the hard disc, enter [b] [Return]. A screen similar to that shown in Figure 2-16 will appear. Continue with step 7.
- If you have files on your destination disc and need to save the files, type [e] [Return] to exit the installation process. If you wish to save any of the files currently on your hard disc, exit the installation procedure and reboot your old system, backing up any files you want to save.

```
*****************************************************
       HP-UX INSTALLATION UTILITY -- DEVICE MENU
*****************************************************


Source Device                        Destination Device


Major number       4                 Major Number         0
Select code       14                 Select code         14
Bus address        0                 Bus address          0
Unit number        1                 Unit number          0
Volume number      0                 Volume number        0


Choice           Description

[Return]         CONTINUE installation process
    e            EXIT installation process
    d            change DESTINATION device


Please enter choice (e or d), hit [Return] >>
```

Figure 2-16: Device Menu

7. Verify the destination device.

   The screen you now see (similar to Figure 2-16) shows you the source address (installation drive) and default destination address (root disc drive). The major numbers will be correct for both the source and destination.

   The source address will be correct, but you must check the destination address (select code, bus address, unit number, and volume number). If your source device (installation drive) and destination device (root disc drive) are integrated (both housed in the same box, product number HP7908, HP7911, HP7912, HP7914, HP7942 or HP7946), these values will be correct. If your source and destination devices are not integrated, you will need to change the destination device address. **You should have the address written on your pre-installation checklist.**

   You may perform any of the following tasks now:

   - If you wish to continue with the installation process, using the device parameters as shown, press [Return] (be careful not to press in several returns). The message

     `Checking values...`

     will be displayed.

     Go to step 8.

   - If you wish to exit the installation process, type [e] and [Return].

   - If you wish to change the address for the destination device, type [d] and [Return]. You will be prompted for the select code, bus address, unit number, and volume number. These are discussed in your *Peripheral Installation Guide*, and should be written on your pre-installation checklist.

     To change a value, type in the new value followed by [Return]; otherwise the old value will be retained.

     When you are finished changing the values, the message `Making device file` will be displayed on the screen.

     You will be returned to the screen shown in Figure 2-16. To continue the installation process, press [Return].

8. After you press [Return] (from step 7 above), the message

```
Checking values...
```

will be displayed. *Install* will verify that the information you gave is correct if you have only one hard disc connected. If you have more than one hard disc, you must verify these numbers yourself.

The system will check your destination device, and depending on what kind of device you have, you may or may not get the following message(s):

```
Will your root disc be on a 98625A/B (disc HP-IB interface card)? [y/n] >>

Does your system have a 98620B (DMA) card? [y/n] >>
```

You should be able to answer these questions from the pre-installation checklist. When you have answered these questions, the installation will continue.

The disc parameters menu (Figure 2-17) will appear on your console.

9. Verify the disc parameters.

You normally should not change these numbers. However, some applications may require non-default disc parameters.

Swap space is the most common parameter to change. If your applications require more than the default amount of swap space, it is important that you change swap space now. **If you do not have enough swap space, you may need to re-install HP-UX later.** You should have your swap space requirements filled out on the pre-installation checklist.

```
 ****************************************************
  HP-UX INSTALLATION UTILITY -- DISC PARAMETERS MENU
 ****************************************************


 Swap Size:         10192    Interleave Factor:         1
 Block Size:         8192    Free Space Threshold:     10
 Fragment Size:      1024    Rpms of the disc:       3600
                             Density- bytes / inode: 2048


 Choice                Description


 [Return]              CONTINUE installation process
    e                  EXIT installation process
    b                  change disc BLOCK size
    d                  change DENSITY (bytes per inode)
    f                  change disc FRAGMENT size
    i                  change disc INTERLEAVE factor
    r                  change disc RPMS
    s                  change disc SWAP size
    t                  change disc free space THRESHOLD size


Use the values shown above unless you need to change them for tuning.
Enter choice followed by [Return] or hit [Return] to continue >>
```

Figure 2-17: Disc Parameters Menu

If you do not wish to change any of the disc parameters, type [Return] to continue.

If you wish to change the any of the values, type the letter corresponding to the choice and [Return]. Enter the new value followed by a [Return] to enter the new value. This will display a new message at the bottom of the screen:

```
Please enter choice (e,b,d,f,i,r,s, or t), hit Return] >>
```

If you want to change any other parameters, enter the appropriate choice.

When you have changed the parameters you want, go to step 10.

10. Press [Return] to continue with the installation.

    You should now see a menu similar to the menu shown in Figure 2-18. This menu is shown so you can verify the correct address for the destination device. If the information is correct, go to step 11. If any information is incorrect, enter [e] and [Return] to exit the installation procedure.

```
*****************************************************
     HP-UX INSTALLATION UTILITY -- INSTALL MENU
*****************************************************


               DESTINATION DEVICE


           Major  Number              0
           Select Code               14
           Bus    Address             0
           Unit   Number              0
           Volume Number              0



Choice         Description


   b           BEGIN Initialization and Installation
   e           EXIT Installation Process



Please enter choice (b or e), hit [Return] >>
```

Figure 2-18: Installation Device Menu

11. Press [b] and [Return] to begin the initialization of the destination disc and the installation of HP-UX.

The following message and prompt will appear at the bottom of your screen:

```
WARNING !!!!! Proceeding will DESTROY the  contents of unit xxx, volume xxx
of the Destination Device. Do you want to proceed? [y/n] >>
```

CAUTION

Do not switch off power to the computer or hard disc drive during disc initialization. Terminating the initialization process in this fashion may seriously corrupt the disc medium.

The **only** way to install HP-UX is to type in [y] and [Return]. **If you think that any of the parameters you entered on previous menus are wrong, or if you do not wish to erase any information on the destination disc**, type [n] and [Return]. Answering n will exit the installation procedure.

If you entered [y], disc initialization begins. The menu, as shown in Figure 2-19, informs you the destination disc is being initialized. The source drive will also be accessed during the initialization process. The media initialization process sets up your hard disc for normal use.

This process can take from four minutes to fifty minutes depending on the size of your disc. See Table 2.1 at the end of this chapter for approximate initialization times. In addition to initializing your hard disc, the installation process creates a file system and copies some important files from your installation tape to your hard disc. **The total time before you will need to interact with the installation process again will be between 30 minutes and 3 hours.**

You will see a trace of the files being copied, similar to the display shown in Figure 2-19.

```
**************************************************
   HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
**************************************************


Initializing:  /dev/rhd
This will take between fifteen minutes and three hours

Making file system on device:  /dev/hd
Mounting device:   /dev/hd
Making directory:  /dev
Making device file: /dev/console
Making device file: /dev/mem
Making device file: /dev/kmem
Making device file: /dev/null
Making device file: /dev/swap
Making device file: /dev/tty
Making directory:  /etc
Making directory:  /disc
Making directory:  /usr
Making directory:  /usr/bin
Copying file: /hp-ux
Copying file: /etc/copyinit
Copying file: /etc/sbtab
```

Figure 2-19: Installation Trace #1

After the trace shown in Figure 2-19, the trace will disappear, and the following two lines will appear:

```
Copying file: /etc/mkfs
Copying file: /etc/mediainit
```

HP-UX then automatically reboots (this takes about 2 minutes), loads more files; you will see the execution trace similar to Figure 2-20.

```
**************************************************
   HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
**************************************************


Mounting device: /dev/src
Making directory: /bin
Making directory: /usr
Making directory: /usr/bin
Making directory: /etc
Making directory: /tmp
Making directory: /etc/filesets
Copying file: /disc/bin/cpio
Copying file: /disc/bin/mkdir
Copying file: /disc/bin/sh
Copying file: /disc/bin/pwd
Copying file: /disc/usr/bin/lifcp
Copying file: /disc/usr/bin/tcio
Copying file: /disc/etc/init2
Copying file: /disc/etc/update
Copying file: /disc/etc/sysrm
```

Figure 2-20: Installation Trace #2

After loading the tools, the screen will clear, and the following menu will appear on your screen:

```
**************************************************
   HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
**************************************************


setting up device files...
reading contents file...
```

The program then loads all the required files and their file sets and displays a screen similar to the screen shown in Figure 2-21. The "XXXXXXX" refers to the fileset number or fileset name.

It will take from 30 minutes to one hour to load the mandatory file system.

```
**************************************************
    HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
**************************************************



Loading files from "XXXXXXX"
      .
      .
      .
Loading: etc/init
      .
      .
      .
```

Figure 2-21: Execution Trace #3

NOTE

You now have the core system installed.

---

12. Load optional filesets or products.

    You will now see the menu shown in Figure 2-22. From this menu, either exit (if you want only the core system) or choose which optional partition you want to load.

```
*********************************************************
     HP-UX INSTALLATION UTILITY -- PARTITION MENU
*********************************************************

Choice          Description


e               EXIT install process
d               DISPLAY installation options for a new partition
a               Process ALL partitions listed below


1               MISC_UTILS
2               SYS_TOOLS
3               PROG_LANGS
4               TEXT

Please enter choice (letter or number), hit [Return] >>
```

Figure 2-22: Partition Menu

The options are:

- If you wish to load all partitions and filesets hit [a] and [Return]. After everything has been loaded, you will return to the menu shown in Figure 2-22, with only the e and d choices (described below).
- If you wish to load individual partitions, enter the partition number and hit [Return]. You will see a fileset menu similar to the menu in Figure 2-23. Go to step 13.
- If you wish to install another product, such as LAN, press [d] and [Return].

  You will be prompted to insert the tape you wish to load. Insert the tape, and after the busy light remains off, press [Return]. You will now see a fileset menu similar to the menu in Figure 2-23. Go to step 13.
- If you wish to exit this menu, hit [e] and [Return]. You will see the screen shown in Figure 2-24. If this was the first time you saw the partition menu, you will have only the core (required) files loaded on the system.

13. Select your filesets.

You can now choose to load all the filesets onto the hard disc, the individual filesets you want, or exit.

```
********************************************************
     HP-UX INSTALLATION UTILITY -- FILESET  MENU
********************************************************

Choice          Description


e               EXIT install process
d               DISPLAY installation options for a new fileset
a               Process ALL filesets listed below


1               partition_1
2               partition_2
3               partition_3
4               partition_4

Please enter choice (letter or number), hit [Return] >>
```

Figure 2-23: Fileset Menu

The time it takes to complete this process varies depending on whether you load one or all of the products.

- If you wish to install all the filesets on this menu, hit [a] and [Return].
- If you wish to install only certain filesets, enter the fileset number and hit [Return]. You will be returned to this menu when the fileset is loaded.
- If you choose to "DISPLAY installation options", hit [d] and [Return]. You will be returned to the menu in Figure 2-22. Continue with step 12.
- If you wish to exit this menu, hit [e] and [Return].

14. After you finish installing the mandatory operating system (step 11) and the optional pieces you want (steps 12 and 13) your screen will clear and the screen shown in Figure 2-24 will appear:

```
**************************************************
   HP-UX INSTALLATION UTILITY -- EXECUTION TRACE
**************************************************



unloading media...
When Busy light remains off, hit [Return] >>

cleaning up ...
```

Figure 2-24: Screen for Exiting Fileset or Partition Menu

The program will unload the tape for you. Hit [Return] when the light on the source drive remains off, and remove the tape. **Remember to store the installation tape in a safe place.**

15. The system will reboot.

    If your file system closed cleanly, a message indicating this appears. Otherwise, the *fsck(1M)* runs automatically and cleans up your file system. This may cause the system to reboot. This situation usually arises when the *shutdown(1M)* command was not used to shut down the system, or when power

16. The program, *reconfig*, will start up. You should go through some of the reconfig items now, especially the "Security" item. For a description of *reconfig*, read the subsection, "Reconfiguring Your System" in "After Installing HP-UX" in this chapter.

    If you wish to exit *reconfig*, press the **Exit** softkey.

17. Follow the steps in the next section, "After Installing HP-UX".

After Installing HP-UX

NOTE

Now that your system is installed, you have the ability to modify and customize the system as described in the next section. If you have purchased software support services from Hewlett-Packard, you can make only limited changes to certain files without voiding your agreement. Consult your support plan or an HP System Engineer regarding intended changes.

Now that your HP-UX system has been installed, you should look through your documentation and perform the following tasks to protect your system:

1. reconfigure your system,
2. create a recovery system,
3. set minimum protections,
4. customize your system,
5. backup your system.

Reconfiguring Your System

After the installation program is finished, the program called *reconfig* will automatically begin.

The *reconfig* program lets you upgrade and customize your system easily. The program lets you:

- add a new user,

  See the section called "Adding and Removing a User" in Chapter 10 of the *Application Execution Environment User's Manual*. You can also add a new user following the procedure in the "System Administrator's Toolbox" chapter in this manual.

- delete a user,

  See the section called "Adding and Removing a User" in Chapter 10 of the *Application Execution Environment User's Manual*. You can also remove a user following the procedure in the "System Administrator's Toolbox" chapter in this manual.

- add a remote terminal (secure systems only),

  See the section called "Adding a Remote Terminal" in Chapter 10 of the *Application Execution Environment User's Manual*. You can also add a remote terminal following the procedure in the "System Administrator's Toolbox" chapter in this manual ("Adding/Moving Peripherals").

- set up a printer,

  See the section called "Setting Up and Removing a Printer" in Chapter 10 of the *Application Execution Environment User's Manual*. You can also set up a printer following the procedure in the "System Administrator's Toolbox" chapter in this manual ("Adding/Moving Peripherals").

- create a new HP-UX operating system,

  You will see this menu item only if you loaded the ACONFIG fileset. See the section called "Creating a New Operating System" in Chapter 10 of the *Application Execution Environment User's Manual*. You can also create a new operating system following the procedure in the "System Administrator's Toolbox" chapter in this manual ("Configuring HP-UX").

- modify system security.

  See the description below.

Security?
A secure system forces people to login to your computer. With security on, no one will be able to simply turn on the power to access or change your information.

You must respond "YES" or "NO" to this question.

YES If you request a secure system, the only way to gain access to HP-UX is to complete a login request; the system must know who you are before it will let you execute any programs.

A secure system has four default logins:

- "window"— when you log in you will get the *pam* shell inside a window.

You must have windows installed before you log in as "window".

- "nowindow"— when you log in you will get the *pam* shell with no window.
- "root"— when you log in you will get the *Bourne shell* with no window.

  Be careful when you log in as "root". The "root" user is privileged. This allows the "root" user to execute critical system commands not accessible by regular users, possibly leading to inadvertent damage to the HP-UX system.
- "guest"— when you log in you will get a *restricted* shell with no window.

  The "guest" login should be used by people who should not have access to the whole HP-UX system. Anyone using HP-UX regularly should have their own login.

None of these four logins has a password assigned. If you want your system to be truely secure, you must assign a password to `root`, `window`, and `nowindow`. **Your system is not secure unless root has a password.** You don't need to assign a password to "guest" since its login shell is restricted. Follow the guidelines in Chapter 5, the section "Changing your Password", to assign passwords to these three logins.

NOTE

As shipped to you, the "root" user has no password. **You must give the "root" user a password to secure your system.**

NO No security means the system does not need to identify you before you have access. When you turn on your computer, you will be in the *pam* shell with you have not installed Windows/9000).

If you do not have a secure system, then each time you need to perform privilaged commands (commands associated with the system administration duties in Chapter 5), you must become the "root" user by typing in:

`su root` [Return]

You cannot use remote terminals if you do not have a secure system. You can access HP-UX only through your system console.

After You Exit reconfig

You will be prompted to confirm and, if not correct, set the date. 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 as shown below:

`MMddhhmm{yy}` [Return]

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. For example, 84 specifies 1984 as the current year. This parameter may be omitted if the year is already correct.

When *date* is executed it echoes the time and date on your screen. If the time and date are not correct, repeat the above procedure. Note that you must be the super-user to change the date. After typing in the date, the system again asks you to confirm the date. If you do not have a secure system and you type [y], the system will come up with PAM in a window (if you have HP Windows/9000 installed) or just come up with PAM on your screen (if you have not installed HP Windows/9000). If you have a secure system, the system logs you off and provides a `login:` prompt.

Creating a Recovery System

A recovery system will allow you to boot up if your root disc is corrupted.

NOTE

It is very important that you create a recovery system. In case your root disc is corrupted, a recovery system may prevent you from having to re-install HP-UX and possibly losing all the files you have created.

For information on creating a recovery system, read the section, "Creating a Recovery System", in the "System Administrator's Toolbox" chapter .

Setting Minimum Protections for a "Secure" System

You need to set protections only if you chose "YES" in the "Security?" item of *reconfig*. The default is "no security". If you do not have a "secure" system, skip to the section called "Customizing Your System".

Some of the system's protections are not set up when you first install your HP-UX system. The most important of these is that the user `root` has no password. This means that anyone with access to the system console (or any system terminal for that matter) can **login** as `root` (also called the "super-user"). (If you are unfamiliar with the login process, read "Session 1" in *A User Guide to the UNIX System.*) The super-user can execute critical system commands not accessible by regular users. By definition, the super-user is potentially dangerous to the system.

NOTE

Do not execute any commands — except those specified in this section — while logged in as the super-user (the user with the user name, `root`) until you are very familiar with the system. Otherwise, you may inadvertently damage the operating system. While getting familiar with the system, log in with a user name other than root.

Setting the Password for Root

After installation, you are automatically logged into the system. The last line displayed on the screen should be similar to:

```
TERM = (hp2622)
```

If you are on a Series 300, enter `hp` if you have a low or medium resolution display, or 9837 if you have a high resolution display. If you are on a Series 200, enter your monitor's product number.

Hit a [Return] to receive the super-user prompt. The system will respond with:

```
WARNING: YOU ARE SUPERUSER!!

#
```

To protect your system, assign a password to the root user by typing:

`passwd root` [Return]

The system will prompt you for a password. Enter at least 7 characters and/or digits, one of which must be a digit, of your choosing and press [Return]. Note that control characters like those generated by [Back space] are accepted but sometimes difficult to remember. The password you enter is not displayed on the console. The system 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 a password twice again. The user `root` will now have to enter that password to login to the system.

Write down the password you assigned (in a secure place); if it is lost or forgotten, no one can log in as the super-user. If the super-user password is lost, the system will have to be re-installed (which will destroy any existing files on the system). If this happens, call your local HP sales and service office for assistance.

An Unattended System Console

One other protection item deserves mention: depending on the perceived need for security at your installation, use discretion about leaving a system console unattended while logged in as the `root` user as this defeats the password protection. Remember, any user logged in with the name `root` (user ID = 0) can execute **any** HP-UX command — a situation possibly hazardous to the integrity of your system.

Customize Your System

At this point you can customize your system by making several changes (such as setting the system clock).

1. Edit */etc/csh.login*, */etc/rc*, and */etc/profile* to set the correct date information in the environment. `XXX` and `YYY` are three letter representations of the standard and daylight time zones for your area and `H` represents the difference between standard local time and Greenwich Mean Time, in hours. (`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                    : PST8PDT
   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, see the section "Setting the System Clock" in "The System Administrator's Toolbox".

2. Edit */etc/ttytype* and change the samples (e.g., 9836 console) to reflect the true terminal types attached to your system.

3. Using the discussion in Chapter 4, "System Boot and Login", customize your system's files.

Backup Your System

You should make a backup of the entire system after you have customized your system. There are two major ways to accomplish this and both are explained in the "Backing Up and Restoring the File System" section of Chapter 5.

Installation Information

Table 2.1: Approximate Disc Initialization Times

Disc	Size	Initialization Time
HP 7907A	20.5 fixed	9 minutes
HP 7908P	16.6 Mbytes	9 minutes
HP 7911P	28.1 Mbytes	4 minutes
HP 7912P/R	65.6 Mbytes	10 minutes
HP 7914P/R	132.1 Mbytes	14 minutes
HP7933H/7935H	404.5 Mbytes	49 minutes
HP 7941/7942A	24 Mbytes	7 minutes
HP 7945A/7946A	55.5 Mbytes	15 minutes
HP 9133D/9134D	16 Mbytes	19 minutes
HP 9133H/9134H	20 Mbytes	26 minutes
HP 9133L/9134L	44 Mbytes	48 minutes
HP 9153A/9154A	10 Mbytes	7 minutes

Notes

Concepts 3

This section discusses several essential concepts needed to manage an HP-UX system. It is not necessary to initially understand all of these concepts in depth, however you should at least be familiar with the terms.

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 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.

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* 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 deallocated.

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. Decendants 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 ([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 a 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.

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

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 super-user (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 protects 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 the process belongs to. 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*.

The Super-User

The term super-user describes the system users whose effective user ID equals 0. Users with effective user ID equal to 0 are provided with special capabilities by HP-UX (hence the name "super-user"). Many commands and system calls can only be accessed by a super-user. Other commands and system calls provide additional features that can only be accessed by a super-user. A super-user 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 super-user commands and some system calls (those used heavily by the system administrator) require the user's name to be "root" and his real user ID to be zero. You should maintain a super-user 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. To prevent other users from accessing super-user 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 super-user. 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 super-user 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. See 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 200/300 uses a file system called the High Performance File System (HFS). *A User Guide to the UNIX System*, 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 detail on how to create your file system, see the section "Creating a File System" in chapter 5 of this manual.

The files of the HFS file system are stored on a formatted mass storage medium, usually a disc. 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.

Disc Layout

Each hard disc drive used for the file system begins with an 8 Kbyte volume header area. The rest of the disc 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 3-1). If you have a hard disc that supports "hard" partitions, such as the HP9133H, then you can address each partition separately (the volume bit in the minor address indicates the partition).

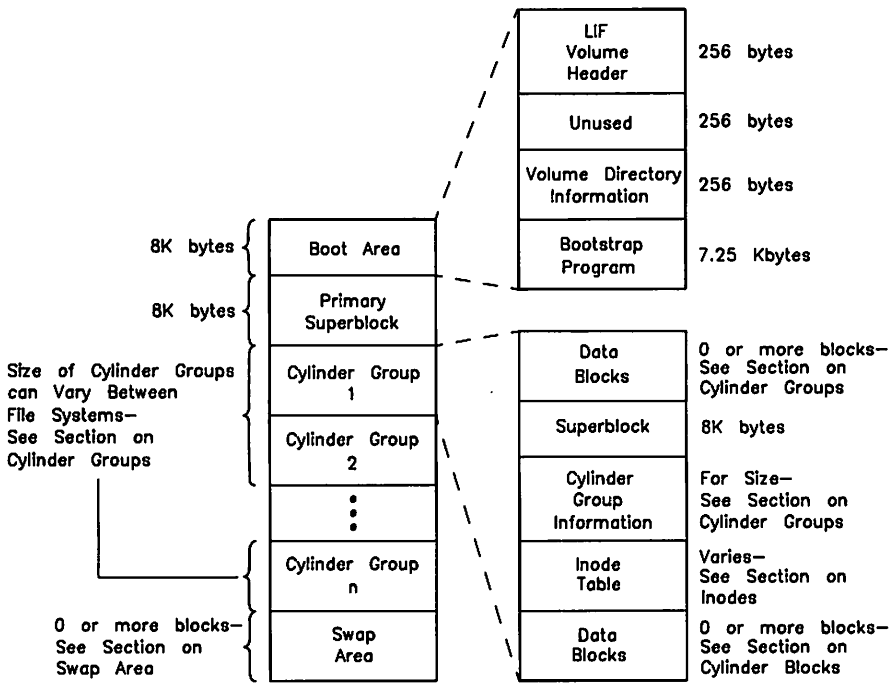


Figure 3-1: File System Layout on Disc

The Boot Area

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

After media initialization, the file, /etc/BOOT, needs to be put on your disc if you intend to boot from that disc. If you are creating your file system with *mkfs*, the boot area must be explicitly created using *dd*. If you are creating your file system with *newfs*, the boot area will automatically be created. See the section "Creating a File System" in the "Toolbox" chapter for more information on creating your file system.

Each boot disc must have a **volume header** in the boot area to identify the volume format. On the Series 200/300 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*. SYSDEBUG does not correspond to any file on your system; it is a Hewlett-Packard internal development program. SYSBCKUP corresponds to */SYSBCKUP*, which is used as a backup kernel (see "System Administrator's Toolbox" for the procedure to back up the kernel). All three are assumed by HP-UX to be object files.

The last 7¼ Kbytes on the boot area contain the **bootstrap program**. The boot ROM loads and passes control to the bootstrap program 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.

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, cylinder group information, and inode table is located in each cylinder group at varying offsets 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 disc characteristics. The primary superblock also keeps track of file system update information in its **summary information** area.

8 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".

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 3-2). 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. See */usr/include/sys/inode.h* for more details on the inode structure.

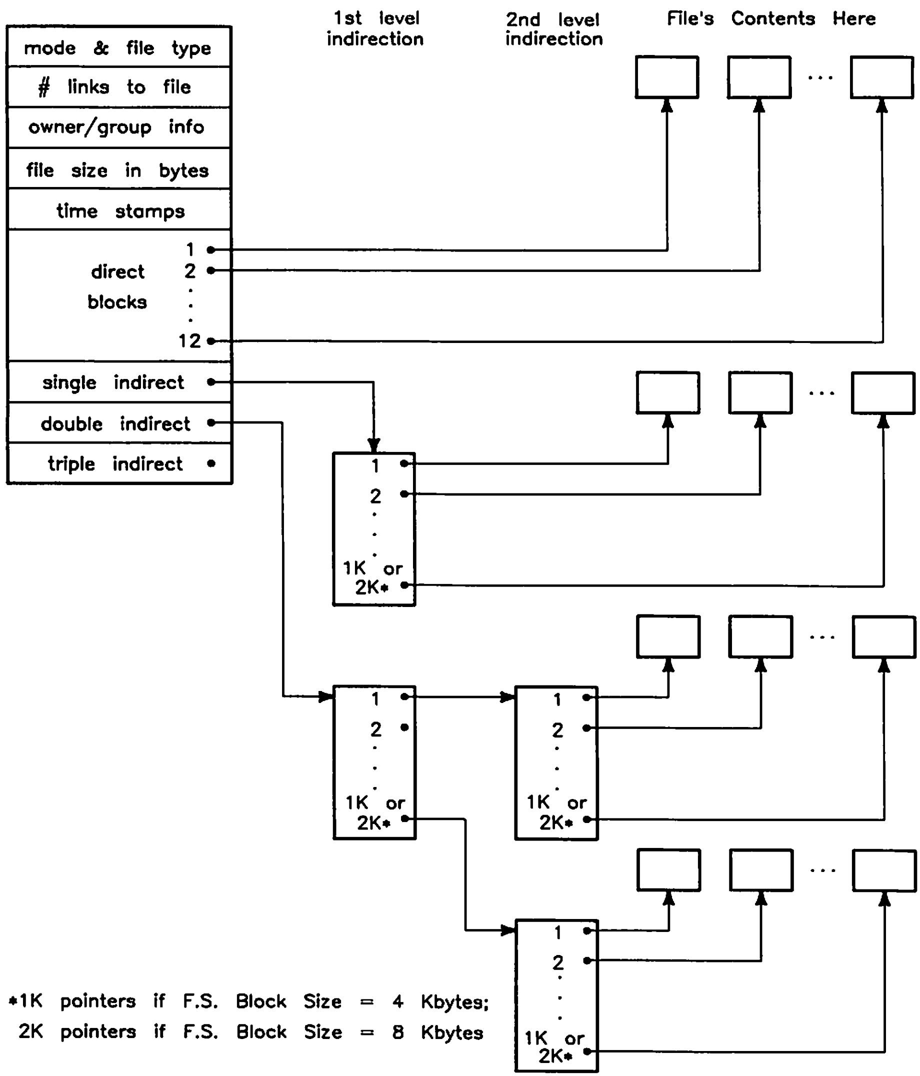


Figure 3-2: Regular File Mapping Scheme and the Inode Structure

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 × *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* × (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 by the size of your disc drive ($fs\_bsize \times (12 + num\_ip + num\_ip^2 + num\_ip^3)$). Your disc drive probably doesn't have this much space and files cannot cross disc drive boundaries.

Inode pointers hold the address of a fragment. The address can be interpreted as referencing a whole block or as referencing 1 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 (1 or more fragments) will be allocated only at the end of a file, so if there are 3 pointers to data, pointers 1 and 2 will point to full blocks, but pointer 3 may point to a partial block.

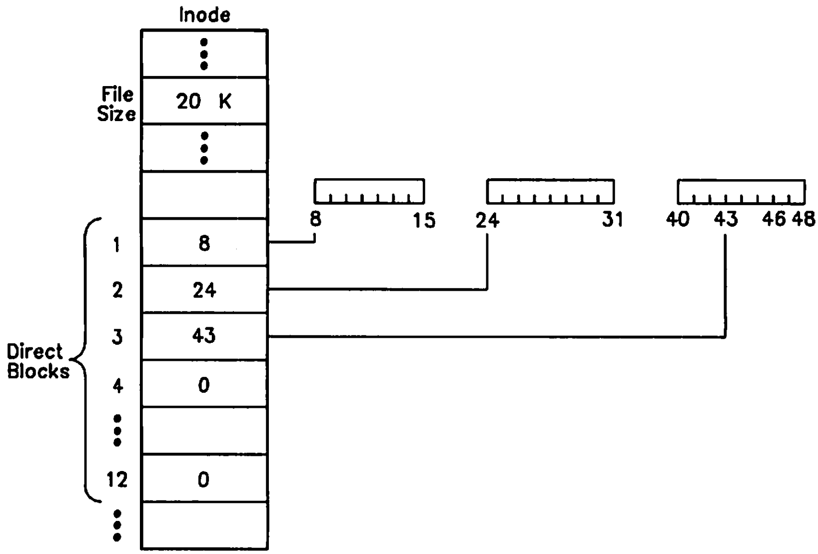


Figure 3-3: Inode Addressing Example

Figure 3-3 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.

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 than what is shown in Figure 3-2. The differences are:

- FIFO and Pipes

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

- Character or block special 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 (see */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 (see "Creating a File System" 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 super-block, cylinder group information, and inode table contain the blocks used to store data for regular files, directories, pipes, and FIFOs (see Figure 3-1). 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 disc 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 DEV_BSIZE (defined in */usr/include/sys/param.h*) and on the upper end by the block size, and must be a multiple of DEV_BSIZE. Fragment size is specified at file system creation.

Allocation of Disc Space

Free space availability is determined from a bitmap associated with each cylinder group. The bitmap 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 Figure 3-4.

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

Figure 3-4: Example Free Block Bitmap in an 8192/1024 File System

The free fragments in this example are fragment numbers 14-21 and 24-31, indicated by *1*s in the bitmap. The allocated fragments are fragment numbers 0-13 and 22-23, indicated by *0*s in the bitmap. 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 3 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 and the free block bitmap is updated.

A certain percentage of free space must always be available in the file system to localize a file's blocks (accessing a file is quicker if the whole file is localized). This 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%. 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 super-user can continue to allocate blocks. When the percentage of free space drops below the threshold, system throughput usually drops because the file system can no longer localize the blocks for a file.

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.

At the global level, the decision of which cylinder group to put a file or directory in is made. 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 in it.

Global policy specifies that once the file size reaches MAXBPG, HFS will allocate blocks from another cylinder group (MAXBPG is defined in */usr/include/sys/fs.h*). 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 asked for
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. 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 disc update is handled asynchronously from the buffer write. The data, along with the inode information reflecting the change, is written to the disc sometime later unless the file was opened in the synchronous mode (see *open(2) O_SYNCIO)*. The process is allowed to continue even though the data has not yet been written to the disc. If the system is halted without writing the incore information to disc, the file system on the disc 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 disc 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, 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, the open file is closed, an *fsync(2)* is issued for the file, or *O_SYNCIO* is set for the file. If a file is opened with the O_SYNCIO flag set, all writes are immediate.
cylinder group information	The cylinder group information is updated whenever a *sync* is executed, or the system needs a buffer, and the cylinder group is written.

A file system inconsistency can also occur if you execute *fsck* 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. **Do not *fsck* mounted file systems**, and always reboot the system after altering the root device with *fsck*.

Maintaining Your File System

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.

The "System Administrator's Toolbox" documents the correct procedures for maintaining your file system. *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 Administrator's Toolbox". **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. Read "Appendix A: Using the FSCK Command" for a description.

- *sync* the system.

  sync writes information in the system I/O buffers to disc. This helps prevent corruption of the file system. A program, called *syncer*, does automatic periodic syncing for you.

- Check and understand disc 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 the "System Administrator's Toolbox".

- 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 the "System Administrator's Toolbox".

- Back up your system.

  Your system should be backed up. Follow the guidelines and procedures in the "System Administrator's Toolbox".

Corruption of the File System

A file system can become corrupt in a variety of ways. The most common ways 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 200 or Series 300 computer system and discs 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, as the system administrator, to make 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 disc pack, or as blatant as a non-functional disc controller.

Detection and Correction of Corruption

A quiescent (no programs are running—check with *ps*—and accounting is turned off) file system may be checked for structural integrity by executing *fsck*. The *fsck* command checks on the 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 multipass nature of the *fsck* program. Init run-level s is the only safe state to check file systems.

When an inconsistency is discovered, *fsck* reports the inconsistency. See "Appendix A: Using the FSCK Command" for an explanation of the actions *fsck* takes in response to these inconsistencies with different options.

SuperBlock Consistency

The summary information associated with the super-block 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 super-block and its associated parts are most often corrupted when the computer is halted and the last command involving output to the file system was not a *reboot* or *shutdown* command.

The super-block 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. The addresses of all copies of the superblock were reported during file system creation.

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. Each inode can be checked for the following inconsistencies:

- format and type
- link count
- duplicate blocks
- bad blocks
- inode size
- block count

Format and Type

Inodes may be one of the following types:

- regular file
- directory
- block special
- character special
- network special
- fifo

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.

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. This size can be checked for inconsistencies, e.g., directory sizes that are not a multiple of thirty-two characters, or the number of blocks actually used not matching 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 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 block, and is done iteratively.

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 directories 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 super block 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. See 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 200/300 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 used the Bell System III file system (BFS), and the Series 500 HP-UX system uses Structured Directory Format (SDF).

To transfer files between your systems, use the following guidelines:

UNIX to UNIX file transfer:

- *cpio*
- *tar*
- *uucp*
- LIF utilities (HP-UX to HP-UX only)

  These include *lifcp, lifinit.*
- mounted file system (between S200/300 HP-UX, version 5.x only)
- LAN
- SRM

To transfer between Basic or Pascal and HP-UX on Hewlett-Packard machines, use:

- LIF utilities
- terminal emulator running on the workstation, uploading or downloading files.

File Protection

When each file in the file system is created, it is assigned a set of file protection bits, called the mode of the file. The mode determines 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 mode of a file is initially determined by *umask*, or by parameters passed to *creat*, *mknod*, or *mkdir*, when the file is created. The mode may be changed with the *chmod* command. The mode of the file is represented as the binary form of four octal digits as shown in Figure 3-5. 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).

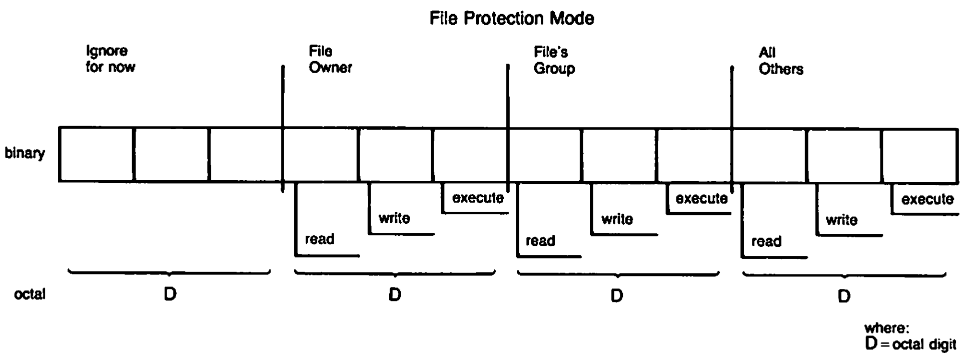


Figure 3-5: File Protection Mode

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 has a mode, 754 (octal). Octal 754 is equivalent to 111 101 100 binary. From the diagram in Figure 3-6, you can see that this grants the owner of the file read, write, and execute permission. It grants read and execute permission to all users who are members of the file's group (except the file owner). That is, 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, may read and execute the file. It grants read permission to all other system users. The *ll(1)* command represents this mode as `rwxr-xr--`.

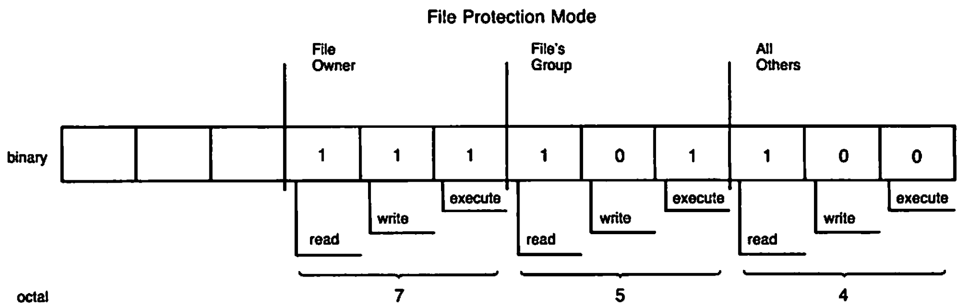


Figure 3-6: File Protection Mode

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

`chmod` *who operation permission*

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 others, enter: `chmod o-w` *filename*. To make a file executable for everyone, enter: `chmod +x` *filename*.

Protecting Directories

Directories, like all files in the HP-UX file system, have a mode. The format of a directory's mode is identical to the mode 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 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

In the section discussing effective user and group IDs, you found that 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 mode digits (previously discussed). This 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. See the section which follows on file locking, or *lockf(2)* in the *HP-UX Reference*.

For example, suppose that the file mode is 6754. Octal 6754 is equivalent to 110 111 101 100 binary. The meaning of the mode is shown in Figure 3-7 and Table 3-1.

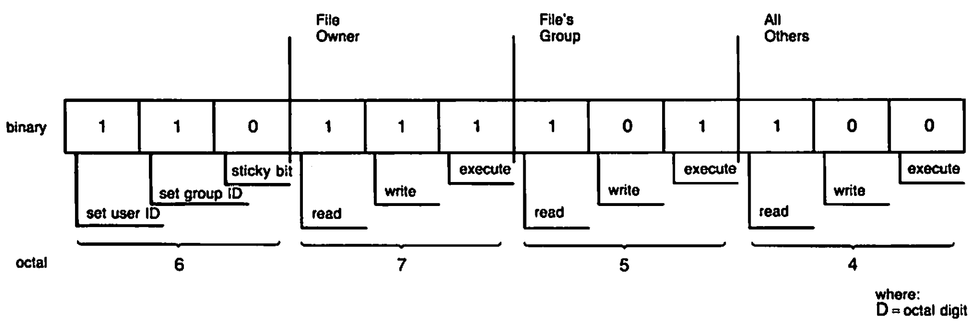
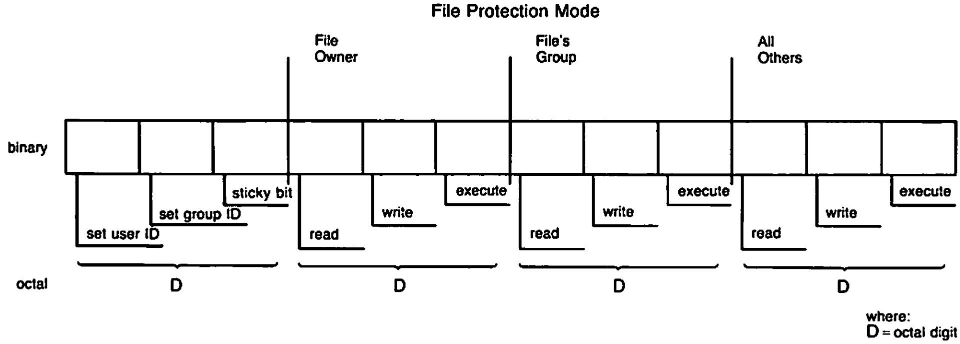


Figure 3-7: File Protection, Example 2.

Table 3-1: File Protection

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

File Protection Mode



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 Memory Management Concepts 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 only be removed by changing the file's mode such that the sticky bit is no longer set.

Only the super-user 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 disc 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 200/300, file locking is done with the *lockf(2)* system call, and it handles two modes of functionality. **Advisory locks** are placed on disc 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. 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 by turning the set-group-id bit on (*chmod g+s*). For example if we opened a file which normally has its file access mode set to 644, an *ll* of the file would look something like:

```
-rw-r--r--    1 George   LubHood        0 May  7 16:11 FubNibWitz
```

To turn enforcement mode on, we would turn the set-group-id bit on (resulting in 2644). This could be done from the shell with the *chmod(1)* command, or from a program with the *chmod(2)* system call. A long listing would show:

```
-rw-r-Sr--    1 George   LubHood        0 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 went to all the trouble of adding to begin with.

WARNING

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 command which is locked under enforcement mode, the shell will go into a sleep state until the command is released. This provides a means for one script to control execution of another, separate script. Care should be exercised when doing this, because as just noted a system deadlock is possible.

Locking Activities

As stated earlier, all file locking is controlled with the *lockf* system call. 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 and 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, see the *lockf(2)* section of the *HP-UX Reference* manual.

The File System Buffer Cache

Program code and the data which it uses must be transferred from disc 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 size of each buffer and the number of buffers in the cache is determined when you power up your system, and is based upon the amount of available RAM (see 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.

The File System Buffer Cache: Benefits vs. Cost
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 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 peformance 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 size buffer alter the default size by changing system parameters (see "Configuring Your Operating System" in the "System Administrator's Toolbox").

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 to store data on. Even with the advent of larger capacity hard disc 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 (½ inch) And beyond this, magtape is also the most interchangeable media between different hardware and operating systems.

Hewlett-Packard also manufactures a series of ¼-inch data cartridge tapes which are used for the installation and updates of HP-UX on the Series 200/300. The cartridge tapes can also be used for inexpensive backups. These data cartridges, model HP 88140, have most of the benefits of 9-track magtape but are cheaper and easier to handle. See chapter 2, *Installing HP-UX*, for a list of CS/80 devices which have a built in data cartridge tape drive.

With the 5.x release of HP-UX on the Series 200/300, there are new drivers intended to optimize I/O throughput to the HP 7971, HP 7974 and HP 7978 magtape drives. This discussion will help you effectively and efficiently use the HP 797x series of magnetic tape drive.

Magtape 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. 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 supports both 1600 bpi and (optionally) 800 bpi, and HP 7978 magtape drive supports a density of 1600 and 6250 bpi. For more specifications see the *HP 9000 Series 300 Configuration Reference Manual.*

The higher the density, the more information can be stored on a tape. On a 2400 foot tape, an HP7974 at 800 bpi can only store 22 Mbytes of data, at 1600 bpi the HP7974 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) 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 ½-inch gap of unrecorded tape, known as the **inter-record gap** or IRG. The next record follows the gap. When the tape drive reads the tape if either the frame parity or the CRC is incorrect, 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 3-8). Logical EOT is two consecutive file marks. Physical EOT is a foil mark about 25 feet from the end of the reel. 2.x and 5.x drivers handle physical EOT differently. See the discussion on 2.x and 5.x drivers.

Note that the distance between the EOT detector and the read/write head may vary between 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 is large; for large records this variation is small.

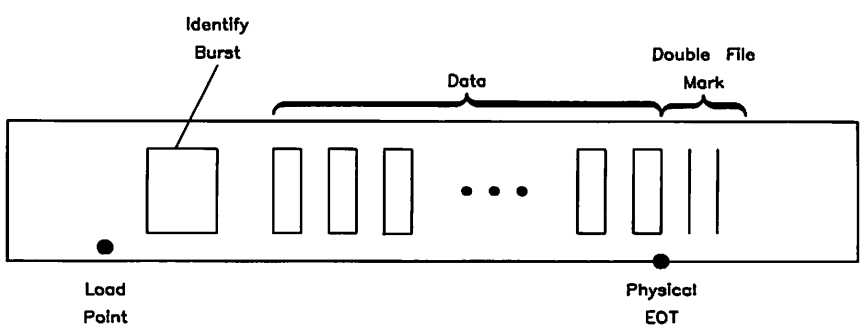


Figure 3-8: Magtape Format

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 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 ½-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 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.

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:

7971	1600 bpi—32 Kbytes
7974	1600 bpi—16 Kbytes 800 bpi—16 Kbytes
7978A	1600 bpi—16 Kbytes 6250 bpi—16 Kbytes
7978B	1600 bpi—32 Kbytes 6250 bpi—60 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 8 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 re-tries

Tape, in its usage for long-term archive and data interchange, is somewhat more prone to error than discs. 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 re-try 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" ½ 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 200/300 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 and HP 7978 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 preceeding 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 is much slower.

Immediate Response
To help your computer send data fast enough to permit the drive to stream, the HP 7974 and HP 7978 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 your Series 200/300 don't collect their data from the disc 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.

Version 2.x Drivers
The version 2.x and 5.x drivers treat records written across or beyond the physical end of tape mark differently. The Series 200 2.x HP 7970 device driver reports an error on read or write if a record crosses physical EOT. When writing on multiple reels, the 2.x driver will finish writing the record, but since writing that record generated an error the application (e.g. *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 2.x drivers will receive a read error, and will not use the phantom record.

Version 5.x Drivers

For the 5.x revision of the Series 200/300 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 2.x and 5.x drivers. For multiple-reel magtape archives, the consequence of this change is that you can no longer interchange tapes between 2.x and 5.x drivers without setting a compatibility mode bit (see "Backward Compatibility" below). Without compatibility mode, the "phantom" record of the 2.x multiple-reel archive will be read from both tapes on drives using immediate response. In particular, *cpio(1)* from one version will not correctly read multiple tapes created from the other version.

Backward Compatibility

The 5.x version HP 7970, HP7974, and HP 7978 drivers support a non-default 2.x compatibility mode which the user may select by setting the third least significant bit in the device file minor unit number (i.e. 0x000008). In this mode these drivers can read and write tapes with 2.x end-of-tape semantics. The only time you need to set the compatibility mode bit is when you are reading 2.x-written tapes with a 5.x driver, or when you are writing a tape with a 5.x driver to be read by a 2.x driver. When the compatibility mode bit is set, the HP 7974 and HP 7978 will have a slower writing rate. 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, 2.x driver, then you have the "phantom" record. The **only** way you could have a phantom record is if you wrote the tape using an HP 7970 driver, version 2.x. If you delete this phantom record, you will no longer need to run your 5.x driver in compatibility mode.

NOTE

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

To delete the phantom record (assuming you have only one file that crosses over more than one tape), 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
```

Memory Management

Overview

Series 200/300 computers use **demand page virtual memory management** to allow the user process's **logical address space** to be larger than the actual physical memory. The series 200 and series 300 Model 310, equipped with the Motorola 68010, supports a logical address space of 16 Megabytes. The series 300 model 320 uses the Motorola 68020 processor and allows the logical address space to be as large as 4 Gigabytes.

The demand page virtual memory management subsystem manages 3 types of resources: the logical address space, the physical memory, and the swap space.

This chapter 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. If you, the system administrator, understand the demand page virtual memory management, you will know when and where to look for further information for changing the default settings. The system parameters' default settings for demand page virtual memory management will support a broad range of users' applications and HP-UX utility programs.

A few major features of Hewlett-Packard's demand page virtual memory management on the Series 200/300 computers are:

- Both series use an HP proprietary Memory Management Unit (MMU) to provide protection against illegal accesses in a multitasking environment. The MMU also supports mapping between logical address space and physical memory at the page level. This means than protection and sharing is possible at the granularity of a page.
- Series 200/300 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 to free memory.

Besides providing the fundamental support for virtual memory, both series have shared memory for high bandwidth interprocess communication (for example *shmget(2)*, *shmat(2)*, and *shmctl(2)*); device mapping for mapping physical addresses into logical address space allowing direct access to I/O devices (for example *iomap(4)* and *graphics(4)*); and process locking for locking all or part of the user process space for real time application needs (for example *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 200/300 computer consists of three logical segments (see Figure 3-9):

- the code segment,
- the data segment,
- the stack segment.

The series 200/300 computer supports 3 execution formats:

- the normal format where the text segment is neither write-protected nor shared (*-N* option of *ld*),
- the shared format where the text 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 text 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 on the Series 300 computer and 2 Kbytes on the Series 200 computer. **Demand loading** means individual pages are brought into memory 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(2)* and *malloc(3)*). 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 3 configurable system parameters:

maxtsiz	Limits the size of the code segment. The default value is 16 Mbytes.
maxdsiz	Limits the size of the data segment. The default value is 16 Mbytes.
maxssiz	Limits the size of the stack segment. The default value is 2 Mbytes.

These parameters are configurable using the *config(1M)* command (see "Configuring HP-UX" in Chapter 5).

Shared memory segments can be attached (placed) between the data and the stack segment using *shmat(2)*. 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(1M)*.

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(2)* or *shmat(2)* system calls, or stack growth, were to cause the segments to overlap, the user would either receive an error (such as ENOMEM) or the user process will be killed.

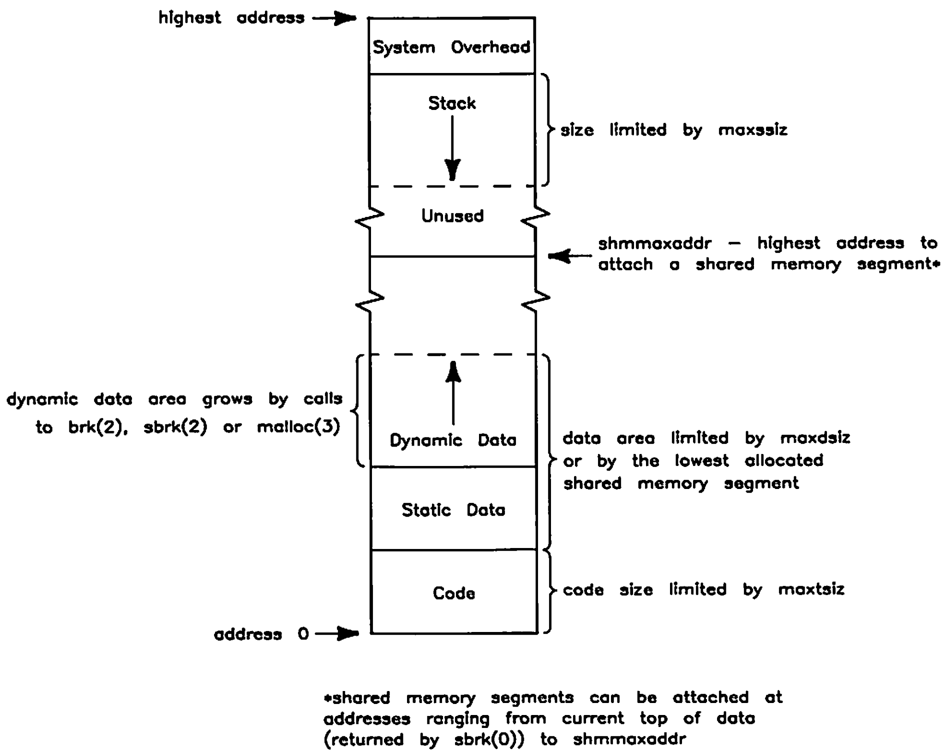


Figure 3-9: User Process Logical Address Space

Physical Memory Utilization

The maximum amount of physical memory you can install on your series 200/300 computer is seven and one half Megabytes. In general a single-user Series 200/300 HP-UX system will perform well with 2 Mbytes of physical memory. 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(1M)* 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 HP-UX" in Chapter 5.

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, such as networking when the *npowerup(1M)* command is executed, or by user processes using *plock(2)* or *shmctl(2)* 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(1M)* 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" 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 up 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(1)* command with *-l* option.

Swap Space Management

Swap space is a contiguous area on the secondary storage, usually a disc drive, reserved for use by demand page virtual memory management. 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 200/300 computers support both single and multiple swap devices. The multiple swap device mechanism allows the swap space to be present on several disc drives expanding the swap space. Also, if your applications require above average number of I/O operations, multiple swap devices may increase throughput.

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 disc 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 in 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 text, data, stack and shared memory segments to be as large as 20 Megabytes. These settings are bigger than necessary for series 200 and series 300 model 310, but since they do not actually consume any extra resource they are set to be the same for all series 300 and 200 systems.

If these default settings are too small for your application program, you can change them using *config(1M)* and Table D.1 in Appendix D. For example, if an application requires a data segment that is greater than 20 Mbytes and less than 40 Mbytes, change the *dmmin, dmmax, dmshm, dmtext,* and *maxdsiz* to be:

```
dmmin 32
dmmmax  4096
dmshm 4096
dmtext 4096
maxdsiz  size of the data segment
```

See the "configuring HP-UX" section in Chapter 5 for more detail.

NOTE

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 entire image of every existing segment is allocated on the swap space; therefore swap space must be large enough 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 HP-UX" in Chapter 5 to create more swap space.

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. See section "Determining the amount of swap space needed" in the "Configuring HP-UX" section of Chapter 5 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 disc) like any other user process segments. See *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 (i.e., not freed for other uses).

To release shared, "sticky" code, a super-user must change the protection/access bits on the file such that the sticky bit is no longer set. If this is done while the text is not being used, its swap space and associated data structure is released.

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 (see the *ntext* entry in Appendix D).

When shared code has its sticky bit set, response time improves because bringing the code in from the swap area is 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.

Demand Load

Programs often contain routines and code which are rarely accessed. For example, error handling routines can comprise 80% 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 disc.

You can set demand loading at link time with the *-q* option of *ld(1)*, or changed in the executable file with the *-q* option of *chatr(1)*.

Demand Load: Benefits vs. Cost

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 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

The supplied tutorial text discusses re-directing program output to a file and program input from a file. 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 special 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, simply 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*. Typically, block devices are disc mass storage devices. However, a disc's built-in cartridge tape drive (available with HP7946, HP7911, HP7912 or HP7914 disc drives) is occasionally used as block devices, but not recommended. Character devices include any device which is not a block device, including printers, plotters, terminals, magnetic tape drives, and paper tape punches/readers. Disc mass storage devices are used as both character devices and as block devices. Network devices are described in the *Network Servvices 9000 LAN User's Guide.*

Drivers

The *mknod* command creates a device file from a driver number, an address (defined next) 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. Use the *dfile* you created when configuring your system to get a list of drivers installed in your system (see "Configuring HP-UX" in the "System Administrator's Toolbox").

Address

An address is a set of values which specify the location of an I/O device to the computer. The address is composed of up to four hexadecimal fields; the select code field, the bus address field, the unit field, and the volume field.

An address is specified in a packed field that has the form:

`0xScAdUV`

`0x` is a two character field specifying that the following field is a packed hexadecimal field.

`Sc` is a two digit, hexadecimal value specifying the select code.

`Ad` is a two digit, hexadecimal value specifying the bus address or port number.

`U` is a single digit, hexadecimal value specifying the unit number.

`V` is a single digit, hexadecimal value specifying the volume number.

The select code is determined by the switch settings on the device's interface card; refer to the *Peripheral Installation Guide*.

The bus address allows the computer to distinguish between two devices connected via the same HP-IB interface. Refer to the manual supplied with the peripheral to determine if the device has an address, and the method in which that address may be changed.

The unit number is used only for disc drives with a multi-disc controller, and is used to specify a specific disc drive. The unit number should be 0 if you only have one drive, or if the address is for other devices.

The volume number specifies a volume on a multi-volume drive.

Many drivers assign different meanings to the second, third, and fourth field of the address. For a precise definition of these fields, refer to the discussion entitled "Adding/Moving Peripheral Devices" in this manual.

Notes

System Boot and Login 4

From the time you switch on power to the computer until you have successfully logged in, many tasks are automatically performed 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 step-by-step description of the computer's activities from power-up through successful completion of the *login* routine. Throughout this tour, you will learn about features of HP-UX that can ease your role as system administrator. Later material in this chapter shows you how to modify these features for your specific needs.

The terminology defined at the beginning of Chapter 2 ("Installing HP-UX") is used throughout this chapter; you may want to review those terms before continuing with this material.

System Boot

When more than one operating system is present on the system's mass storage devices, both the location of the operating systems and the type of media on which they are stored determine which operating system is loaded.

What the Boot ROM Does

The boot ROM is a general purpose piece of software residing in Read Only Memory that was specifically developed to support a wide variety of present and future Hewlett-Packard operating systems. Because different operating systems use different aspects of the boot ROM and, because you are primarily concerned with how HP-UX works with the boot ROM, the following description of the boot ROM's operation focuses on its use with HP-UX.

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

1. checks for and tests some interface cards and internal peripherals (see 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,

   See the section "Description of a System Console", later in this chapter.
3. polls all supported mass storage devices connected to the computer to search for an operating system to boot,
4. loads the first "bootable" operating system found,
5. passes control to that operating system.

The following shows a typical display of the boot ROM's operation. **The display varies depending on the version of the boot ROM.**

```
Copyright 1985,                                       :HP7946, 1400, 0, 0
Hewlett-Packard Company.                              1H SYSHPUX
All Rights Reserved.                                  1D SYSDEBUG
                                                      1B SYSBCKUP
BOOTROM Rev. A
Bit Mapped Display
MC68010 Processor
Keyboard
HP-IB
HP98620B
HP98644 at 9
HP98625 at 14
2096978 Bytes





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

Once the `SEARCHING FOR A SYSTEM` message appears on the system console, the screen will clear and a series of messages appear, including the following information:

Console information
Processor information
Interface cards
real mem =
buffer information
root device information
swap device information
avail mem =
unlockable mem =
copyright information

On the media where your bootable system resides (hard disc or recovery system) there is a boot area. The boot area has a file, called */hp-ux*, that is your operating system. SYSHPUX points to */hp-ux*.

If you have not modified your system since you installed it, SYSDEBUG will also point to */hp-ux*. If you have modified your system, SYSBCKUP should point to the previous version of */hp-ux*. You have probably renamed it */BACKUP*.

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.

Description of a System Console

The **system console** is a keyboard and monitor (or terminal) 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:

- 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.
- An "internal" bit-mapped display is present. This is true if a Model 310 built-in video output, HP 98542A, HP 98543A, HP 98544A or HP 98545A board is present. It is also true if an HP 98700H display station with its display interface card (HP 98287A) configured for "internal" control space is present. Note that if the HP 98700H display station's display interface card (HP 98287A) is configured for "external" control, it is never chosen ("external" bit-mapped displays have a specific select code address and "internal" bit-mapped displays **do not**).
- An HP 98546A compatibility video interface is present.
- 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 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 boot ROM tolerates this, HP-UX will not.

---

1 The HP 98628A Datacomm Interface Card with its "remote bit" set is not supported as a remote console by the Revision A 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 ITE (terminal) associated with the HP 98628A card.

2 On the HP 98626A Serial Interface board the remote bit is set by cutting a jumper as described in the installation manual supplied with your computer. On the HP 98628A, HP 98642A and HP 98646A interface cards the remote bit is set by setting a switch on the board as described in each board's installation manual.

Additional Considerations

This section contains information on bit settings for the HP 98626A RS-232C Serial Interface Card's Line Control Switch Pack.

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 follows:

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:

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 manual supplied with that card.

The Boot ROM's Search Sequence

The boot ROM has two modes of 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 boot the first operating system in its search sequence.

Unattended Mode

You should use the unattended mode of booting either 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 prioratized list of devices. The first system found on one of these devices will be booted. If no system is found, the list will be searched again until a system is found. This means that discs not found at power-up can 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 before you turn on your computer.

The search sequence that the boot ROM uses to find an operating system follows these priorities:

1. Internal mini-floppy, Unit 0 (i.e., Drive 0) (Model 236 only).
2. External discs at Select Codes 0 thru 31, Bus Address 0, Unit 0, Volume 0.
3. Shared Resource Manager at Select Code 21, Volume 8.
4. HP98259 (Bubble Memory) at Select Code 30.
5. HP98255 (EPROM card) on Unit 0.
6. ROM systems (for ROM-based operating systems).
7. Internal mini-floppy, Units 1 thru N (Models 236 only).
8. Other external discs at Select Codes 0 thru 31, Bus Addresses 0 thru 7, Units 0 thru 15, Volumes 0 thru 7 (not Bus Address 0, Unit 0, Volume 0).
9. Other Shared Resource Managers at Select Codes 0 thru 31 (not Select Code 21, Volume 8).
10. HP98259 (Bubble Memory) on Select Code 0 thru 29.
11. Remaining HP98255 (EPROM cards).

According to the above priorities, external devices are searched in order of select code; a system at select code 4 would be found before a system at select code 5. Also, multiple units at the same select code and bus address are searched before moving to the next ascending select code or bus address. Thus, the HP-UX system on the root mass storage device at select code 14, bus address 0, unit 0, is found and loaded before any of the HP-UX "utilities" systems at select code 14, bus address 0, unit 1.

To summarize, the search levels for HP-UX devices, in ascending order are:

1. device class (internal discs, external discs)
2. select code
3. bus address (also called device address)
4. unit number
5. volume number

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. 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 name, listed under a mass storage device name and address. For example, if you have both an HP-UX kernel and an HP-UX backup kernel 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
```

You select an HP-UX operating system by typing the 1 to 3 digit string associated with the desired system (in this example, press [1] [H] or [1] [B]).

After Booting

After your system is booted, it performs a series of checks and starts system processes.

The following sections discuss the system processes that go on after booting.

The Root File System

Once the HP-UX Operating System has been found and loaded successfully, HP-UX searches 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 (i.e., 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.

The root file system can exist on any mass storage medium supported by Series 200/300 HP-UX that has at least 16 Mbytes (for HP-UX) of storage (though this minimum size leaves little storage space for significant development work).

The Init Process

After finding the root file system, HP-UX sets up its first process, */etc/init*. */etc/init* becomes process one (1) and has no parent.

After booting, the process *init* reads the configuration file */etc/inittab*. Each line in the file */etc/inittab* describes some activity for the system to take, under certain circumstances. The entries are of the form:

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

where `id` is a unique two character identification code, `rstate` are 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. The run-levels are described in *inittab(5)*.

For example, you might want the system console to have a *getty* (be able to log in) for every state you define:

```
co:2345:respawn:/etc/getty console H
```

The action `respawn` tells *init* to continuously re-create a *getty* process at the console in run-levels two thru five. Leaving the rstate field null (as shown below) will cause execution in run-levels 0 through 6:

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

This is highly recommended for the system console.

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. If you make changes to these files: save a copy of the original file, document the changes and keep a copy of that documentation with this manual.

After *init* begins, but before it makes the first transition into states 0-6, all entries marked *boot* or *bootwait* are executed. As shipped, */etc/bcheckrc* is run now, and executes an *fsclean*, and if necessary *fsck*, for all file systems listed in */etc/checklist*.

Once the booting processes have been run, *init* comes up in the "initdefault" run-level as defined in */etc/inittab*. See the *init(1M)* and *inittab(5)* entries in the *HP-UX Reference* for more information. If you choose system security (from *reconfig*, described in Chapter 2, "After Installing HP-UX"), your system will come up in run-level 2, which is the default multi-user run-level. If you did not choose system security, your system will come up in run-level 1, which is the default single-user run-level. For information on changing the run-level, see the section on "Changing the System's Run-Level" later in this chapter.

Single-user vs. Multi-user

There are two default run-levels on your HP-UX system: run-level 1 and run-level 2. Whether you purchased a single-user or a multi-user system, you will, by default, come up in run-level 1 if you did not choose system security, and come up in run-level 2 if you chose system security (see Chapter 2, "After Installing HP-UX" for a description of system security).

In addition to run-levels 1 and 2, HP-UX has a true single-user 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. `bootwait` and `boot` entries are not executed. In this run-level no *getty*'s are issued, nor are any other actions taken by the script */etc/rc*. *syncer*, *bcheckrc*, and *fsck* also do not get executed. **Run-level s is not recommended by Hewlett-Packard** since in run-level s certain processes that monitor and check your system do not run.

The /etc/bcheckrc Program

The */etc/bcheckrc* (Boot CHECK Run Command) program is called from *init* before your system makes the first transition to any of states 0-6. *bcheckrc* checks to see if the system was properly shutdown.

bcheckrc calls *fsclean(1M)*. *fsclean* checks each file system listed in */etc/checklist* to see if there might be a consistency problem. To do this, *fsclean* looks in the primary superblock of each file system. In the superblock is a *clean byte*. When a file system is created, the clean byte is set to FS_CLEAN. When the file system is mounted (*mount*), the clean byte is set to FS_OK. If the file system is unmounted during a proper shutdown (see "Shutting Down Your System"), the clean byte is reset to FS_CLEAN. If, when *fsclean* checks the clean byte, the file system is unmounted, but the clean byte is not set to FS_CLEAN, the file system might be in an inconsistent state.

If *fsclean* finds the wrong value in the clean byte, *bcheckrc* will run *fsck(1M)* automatically (using the *preen* mode), correcting most errors found (see the discussion on the preen mode in "Appendix A: FSCK"). After running *fsck*, the system will reboot.

Anytime *fsck* makes changes to the root (/) file system the system must be rebooted to force the memory resident disc information to be consistent with the changes *fsck* made to the disc.

You may, on a system with multiple file systems, see the system come up, run *fsck*, reboot, then run *fsck* again. The first *fsck* fixes the root volume, the second fixes all the additional, as yet unmounted, discs. The second *fsck* does not cause the system to reboot because the discs it fixes are not mounted.

If the *fsck(1M)* command run by *bcheckrc* fails for any reason, *bcheckrc* starts a shell with the prompt `(in bccheckrc)#`, along with instructions to run *fsck* manually. **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 [CRTL] [D]. *bcheckrc* will then
proceed.

The /etc/rc File

Each time *init* changes run-level, either at boot time or when invoked manually, */etc/inittab* is read. After reading */etc/inittab* and signalling processes as required, a line in */etc/inittab* invokes */etc/rc*. This script uses the following two arguments returned by doing a `who -r`: the current run-level and the number of times this run-level has been entered previously. At power-up, the values of these arguments are 2 and 0 respectively, if you have not modified your system. The following describes the actions that take place whenever */etc/rc* is invoked. This is followed by a description of the contents of *rc*.

Upon starting, the */etc/rc* script sets the environment variables PATH (the default search path that the system uses to find commands) and TZ (for time zone). */etc/rc* next exports the TZ variable (using the *export* command). Exporting TZ causes *rc* (and any child process of *rc*) to override the default time zone (EST7ETD); for more information, see the *ctime(3C)* entry in the *HP-UX Reference* manual.

Next, */etc/rc* sets up and initializes the system console by using the *stty* command to set such attributes as the baud rate, communications protocol and tab settings.

/etc/rc in Run-level 2

You may customize */etc/rc* to perform functions which you wish to occur every time the system is booted, or whenever there is a change in run-level which *init* does not handle. As shipped, the */etc/rc* shell script

- sets the date (an interactive operation),
- mounts the file systems configured in */etc/checklist* (see *checklist(5)*),
- executes the */etc/cron* program,

  cron executes commands at specific dates and times, according to the instructions submitted by the *crontab* command (see *crontab(1)*).
- performs miscellaneous "housekeeping" chores, such as preserving editor files (if they exist), starting up the "lp daemon" (which files that are no longer needed by the system,
- saves various logging files (by renaming them) and prints *revision(1)* information about the HP-UX operating system software,

Returning Control to the /etc/init Process

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.

Execution of /etc/getty

As shipped to you, */etc/getty* is invoked only for the system console in run-level 2. If you did not choose system security from *reconfig* after you installed the system, no *getty*s will run.

The first command (*/etc/getty*) executed for each login terminal specifies the location of the terminal and its default communication protocol. Additionally, it causes the first `login:` prompt to be displayed. Eventually, the getty process slot is occupied by your shell (see the following section, "Login").

When that process is terminated (when you log out), the */etc/init* process is signalled and "wakes up". *init* then checks */etc/inittab* to see if the process that signalled it is flagged as continuous in the *inittab* entry. If the process is continually "respawned", *init* again invokes the command in the command field of the appropriate *inittab* entry as described above (i.e., 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. If you do, *getty* will send an error message to the console, wait 20 seconds, and then exit. On earlier versions of HP-UX, a *getty* to a non-existent terminal would consume large amounts of SPU time. This is no longer the case.

Login

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.

NOTE

You will receive a login prompt only if you chose system security with *reconfig* after you installed your system.

---

1. The login process begins when you supply a user name in response to the `login:` prompt generated by */etc/getty*. Once the user name has been entered, */etc/getty* executes *login* with the supplied user name; */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 you to enter the password. The supplied password is encrypted and compared to the encrypted password stored in */etc/passwd*. If a valid user name is supplied and that name has no password associated with it, you are logged in without further prompting.

   For security reasons, if the user name supplied 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.

   Successful logins are recorded in the */etc/wtmp* file. Unsuccessful logins are recorded in the */etc/btmp* file.

3. *login* next 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. Next, *login* changes the current working directory to that supplied in the home directory field in */etc/passwd*.

5. *Login* then 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*, though the most common are */bin/sh*, */bin/csh*, and */bin/pam*. If no command is present in the command field, */bin/sh* is executed by default.

6. Next, assuming a shell was generated in step 5, the shell executes the shell script */etc/profile* for the Bourne shell (*/bin/sh*) or */etc/csh.login* for the Berkley C shell (*/bin/csh*). 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 super-user (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.

7. */etc/profile* and */etc/csh.login* also execute the *stty* command to set a terminal's characteristics. In addition, they define the path for the MAIL environment variable and they perform the following tasks:

   - displays the message-of-the-day (contained in */etc/motd*).
   - uses *mail -e* to detect if any mail is present; if there is any mail, the message `You have mail.` is displayed.
   - executes `news -n` which displays the names of all new files added to */usr/news* since the last *login*.

8. Finally, the shell executes the script *.profile* (for the Bourne shell), *.cshrc* (for the C shell), or *.environ* (for PAM) if it exists in your home (login) directory. Typically, a *.profile*, *.cshrc*, or *.environ* file is created by the system administrator for each user ad is customized by the user for his environment on the HP-UX system. For example, you might create or modify your *.profile* file to alter the primary and secondary prompts, change one or more environment variables, set up your terminal for a particular application, or invoke application programs automatically.

   In addition to this, the C shell also executes the file *.login* (if it exists) when you first log in. *.login* is executed following the execution of *.cshrc*.

9. Now that you have successfully logged in, the shell waits for your first command.

Changes You May Want to Make

You should now have an understanding of how the operating system is loaded and initialized and the procedures used to log in on the system. This is a good time to consider how you can modify the boot and login procedures to customize your HP-UX system. This section is designed to help you in this process.

The discussions of system boot and login presented earlier in this chapter have been condensed in the following presentation to provide a "functional tour" through these same procedures. In this tour, the boot and login procedures are reduced to simple statements of function. Suggestions are made as to the type, scope, method and advisability of the modifications that can be made. **Note that changes to the files */etc/inittab*, */etc/rc*, and */etc/passwd* should not be major ones; the system may not boot if these files are radically modified. Before modifying these files you should create a recovery system and check it out.** The procedures for making the suggested modifications are either detailed here or in Chapter 5 ("The System Administrator's Toolbox").

NOTE

The sequence of events described below occur **only if you chose system security** from *reconfig*. See Chapter 2, "After Installing HP-UX", for a description on choosing system security.

The computer is powered up and the first operating system found is loaded.

The boot ROM code cannot be changed. However, by understanding the search algorithm the boot ROM uses to find an operating system, you can load any supported system that you want.

init is brought up and run.

init calls *bcheckrc*, and if your system was improperly shutdown, *fsck* is run.

/etc/rc will *mount* all the file systems listed in */etc/checklist*.

The system automatically goes into run-level 2 causing certain commands from */etc/rc* to be executed.

As shipped, the */etc/rc* file executes the *cron* command which runs programs at a scheduled time.

If *cron* is executed by */etc/rc*, you may want to add entries using the *crontab* command to automatically and periodically perform certain procedures. For example, you might add entries for *cron* to:

- back up the system.
- call other HP-UX systems for mail and other *uucp* transactions.
- execute system accounting commands (see chapter 6).

Commands are executed from the command field of run-level 2 entries in */etc/inittab*.

Insure "2" is in the `rstate` field of each terminal */etc/getty* line. The terminal */etc/getty* should have state numbers for each state you want the user of that terminal to be able to log in. You should have the console defined in all states.

A user attempts to log in to HP-UX.

To keep track of all bad login attempts, you must create a log file called */etc/btmp*. Once this file exists, the system uses it to log unsuccessful login attempts. You can use this file to help determine if unauthorized users are attempting to login. A summary of these attempts may be viewed using *lastb(1)*.

The system also keeps track of all logins and logouts in a log file called */etc/wtmp*, if created. A variety of login and logout information may be accessed with the *last(1)* command.

You can restrict which "tty" device (login communication port) the `root` user can login on by creating a file called */etc/securetty*. This file should contain the device names of the tty files where the `root` user can log in. The entries only contain the name of the special (device) file for that tty but not the pathname (typically, the */dev* directory). The file can contain more than one entry but only a single entry per line. If you do not explicitly create this file, the user `root` may log in on any tty device. Note that this security feature does not restrict a normal user from using *su(1)* to become the super-user on any device; see the *login(1)* entry in the *HP-UX Reference*. You can restrict use of the *su* command by executing `chmod 500 /bin/su`. Here is a typical */etc/securetty* file with two entries:

```
console
tty05
```

A user successfully logs in; after some initialization, login changes the user's current working directory to his login directory.

To set the user's login directory, edit the file */etc/passwd* (see *passwd(5)* in the *HP-UX Reference* manual).

Login executes a command from the user's entry in ***/etc/passwd.***

Typically, the command invokes a shell for the user. However, you may wish to execute an application program for the user. This is advisable when the user has no knowledge of HP-UX and only wants to use the system to run a specific application. If no entry exists in the */etc/passwd* command field, */bin/sh* is executed by default.

For example, suppose that an inexperienced user wanted to access HP-UX only to run the program *testx* (a program written by your company that tests widgets) contained in his login directory. You might add an entry to */etc/passwd* of the form:

```
john::135:12:run testx only:/users/john:/users/john/testx
```

The name `john` is the user's login name and the values `135` and `12` are his user ID and group ID respectively. The entry `run testx only` is a comment in */etc/passwd*. The entry `/users/john` is the user's login directory. The last entry is the command field; it specifies that when the user logs in, the program `/users/john/testx` is automatically run (instead of the shell). In this example, after the user `john` finishes executing the program `testx`, he will automatically be logged off the system.

The same thing happens in the more common case of a user executing a shell: when the shell is terminated (with a CTRL-D or `exit` command given by the user), the user is logged off.

The command field of */etc/passwd* is also useful for enabling a user to access information without logging onto the system. For example, the system is shipped with a *passwd* entry containing the user name `who` with the command */bin/who* in the entry's command field. Supplying `who` in response to the login prompt causes a list of all system users (all users currently logged in including the user `who`) to be displayed on the terminal. The user is logged in only for the duration of the *who* program and is logged off when the program terminates. This login lets anyone determine a valid login name; you may wish to delete this entry to provide more security.

As shipped to you, */etc/passwd* has entries for the users `who` (used as described above), `date` (which executes the *date* command providing handy access to the time and date), and `sync` (which executes the *sync* command). The *sync* command writes all the information contained in the system's I/O buffers (in RAM) to the disc.

The shell scripts */etc/profile* or */etc/csh.login* are automatically executed.

/etc/profile (for the bourne shell) or */etc/csh.login* (for the Berkeley C shell) allows you to force execution of one or more commands for each user when she logs in, every time that she logs in. This is ideal for forcing the execution of commands that each user should execute at login. For example, */etc/profile* and */etc/csh.login* (as shipped to you) assume that */etc/motd* (message-of-the-day) contains one or more messages and send the contents of that file to each user's terminal at login (via the *cat* command; the output appears on the user's standard output). To change the message sent to each user, simply edit */etc/motd*. Note from the previous discussion of */etc/profile* that many of these commands are only executed if the user is executing */bin/sh* or */bin/rsh*. */etc/csh.login* is executed if the user is executing */bin/csh*.

Edit */etc/profile* and */etc/csh.login* when you want to alter their function. New commands can be added to these scripts; old commands can be removed or modified. The */etc/profile* and */etc/csh.login* shipped to you are commented and, with the foregoing documentation, should be fairly easy to follow. Changes made to these scripts do not go into effect until the script is executed (either at login or explicitly).

The shell scripts *.profile* (for the bourne shell), *.cshrc* and *.login* (for the C shell), or *.environ* (for PAM) are automatically executed, if they exist in the user's login directory.

.profile, *.login*, or *.environ* is used to execute those commands that each user wants executed at login, as well as each time a new shell is spawned. For example, a user may want prompts from the shell that are different from the default prompts. To change these, the user simply adds `PS1="new prompt"` and `PS2="new secondary prompt"` to his these scripts, in the Bourne shell, and `set prompt = "[\!]:  "` in the C shell. Remember, in the Bourne shell variables must be exported to have an effect upon subsequent processes. In the C shell you must use `setenv`.

Following the execution of *.cshrc* in the C shell, the file *.login* is executed at log in.

Edit $HOME/*.profile* or $HOME/*.login* (where $HOME is the user's login directory path name set at login) when you want to alter its function. New commands can be added to the script; old commands can be removed or modified. Typically, a user uses *.profile* to:

- set and export (with the *export* command) environment variables such as shell prompts (PS1 and PS2) and the default search path (PATH).
- execute commands at login (for example, the *who* command, to see who else is on the system and the *ls* command to list the names of files in the login directory).
- set terminal options with the *tset(1)* command.

Do not change the *stty* settings until you understand how they work. Many commands (such as the *vi* text editor) require settings similar to those shipped with your system in order to operate properly. See the *stty(1)* command in the *HP-UX Reference* and "The vi Editor" article in the *HP-UX Concepts and Tutorials* manual.

Changes made to the *.profile* script do not go into effect until the script is executed. By typing

```
. .profile                # In the Bourne shell

source .chsrc             # In the C shell
```

in your current shell, *.profile* or *.cshrc* will execute in the current environment.

Changing the System's Run-level

Your system will "come up" in run-level 1 (if you did not choose system security) or run-level 2 (if you chose system security). This means your system either executed the *inittab* commands associated with run-level 1 or run-level 2.

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 when entering that run-level. For example, if you have not added a "respawn" *getty* (see discussion below) entry for run-level 3 for the console, entering run-level 3 from run-level 2 will cause the console to die!

Each terminal or RS232 port that is used as an incoming port needs to have a process called a *getty* running on the line. *inittab* is used to set up the gettys. **As shipped, *inittab*** will only run gettys if you have a secure system (run-level 2).

For each incoming port you must have one line in *inittab*, which looks like:

```
03:236:respawn:/etc/getty tty03 H
```

03 is the two-character "process ID" field (by convention the name of the associated tty); 236 are run-levels 2, 3 and 6 (there should be at least one run-level entry for each incoming port); `respawn` tells *init* to continuously restart the getty; and the colons (:) are mandatory separators and should be used as shown. You can enter comment lines in the following fashion:

```
xx::off:#This is a comment
```

See the *inittab(5)* entry in the *HP-UX Reference* for further details.

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 as "respawn" for the new run-level.

Once all the users that need to be off the system are off, change run-levels by making an entry of the form where:

`init new_rstate` [Return]

new_rstate is the number of the new run-level.

Depending on how the new run-level is defined, even the foregoing may not be necessary. For instance, 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. Watch for side effects however. Consider the case where a user logs off, you then change run-levels (from say, run-level 2 to run-level 4) and when the user attempts to log in, he cannot because an */etc/getty* entry does not exist for him in run-level 4 the way you have defined it.

When the system enters the new run-level, its actions are similar to the actions executed are those identified by the new run-level number. And of course some files, such as */etc/rc*, have no entries for run-levels other than run-levels 1 and 2. The */etc/rc* script may be customized to include more run-levels.

Summary of Environment Files

As you have seen, 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 those listed in the previous section to determine which files to modify. All of the files listed here have commented versions that were shipped with your system. Commented versions of files discussed here that do not have a specific path name (because they are generally user files in a user's home directory) are contained in the */etc* or */etc/newconfig* directories.

The */etc/newconfig* directory is shipped with updates of the HP-UX Operating System and generally contains new versions of the files shipped in the */etc* directory. If you have never updated your system, you will not have an */etc/newconfig* directory.

NOTE

The system may not boot if */etc/inittab*, **/etc/rc, and /etc/passwd** are radically modified. The parts of these files that are shipped should not be changed, though additions can be made as necessary for terminals, commands, and users. You should check the */etc/passwd* and */etc/group* files with the *pwck(1M)* and *grpck(1M)* commands.

/etc/inittab

This file contains entries for the different run-levels (supplied or created) to which the system administrator may want to change the system.

/etc/rc

This shell script defines miscellaneous actions for each system run-level change. It will take different actions for different run-level. This script typically contains commands to:

- run *expreserve* to preserve editor files (see the *ex(1)* entry).
- mount file systems via the *mount* command.
- start the *cron* program running.
- clean up any logging files.

/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 additional user who wants to log in on your system.**

/etc/group

This text file identifies the users that form a group. It contains a list of users and associates those users with a group name and a group ID.

/etc/motd

This text file contains messages that are printed to each user when he logs in. If you have a message that you want to communicate with every user (such as a message specifying a new system update), write the message in this file by using your favorite 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 and can be used by you to communicate with users on the system. You can also change the directory permissions to allow any users 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*, in $HOME/*.profile*, or $HOME/*.lo gin* (the user's personal version of profile), the file you placed in the */usr/news* directory will be printed when the user logs in. Depending on the options used with *news*, a user only receives the message once. See the *news(1)* entry in the *HP-UX Reference* for details.

/etc/profile or/etc/csh.login

These shell scripts are automatically executed for each user, each time she successfully log into the shell. This is an ideal location to place commands that you require every user 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 (see the *cat(1)* entry in the *HP-UX Reference*). These scripts are also an ideal location to defne and export default environment variables (such as PATH and TZ) in case the user does not set them in his *.profile* shell script.

/etc/wtmp

A text file used by the system to keep a history of logins, logouts, and date changes. Created automatically by the system, the contents of this file are accessed with the *last(1)* command.

/etc/btmp

A text file which—if it exists— is used by the system to keep track of bad login attempts. You must explicity create this file to use this feature. The contents of this file are accessed with the *lastb(1)* command.

/etc/securitty

A text file which—if it exists—specifies which tty files the `root` user can login on. You must explicity create this file and place the tty special file names in it to use this feature.

$HOME/.profile, $HOME/.cshrc, $HOME/.login, $HOME/.environ

These shell scripts, when located in the user's home (login) directory, are automatically executed each time the user successfully logs in. It is a good location place commands that customize the user's environment and that perform functions the user wants executed each time he logs in.

For example, they may include a definition of the default shell prompt (the `PS1` and `PS2` environment variables in the Bourne shell, `prompt` in the C shell), the default search path (the PATH environment variable), and some editor information (the EXINIT 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.

$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(1)* family of HP-UX editors (*vi*, *ex*, etc.). The file *.exrc* must exist in the user's home directory ($HOME) to use these features. The editor searchers for *$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 is generally defined and exported from either */etc/profile* or $HOME/*.profile*. The *.exrc* file serves a function similar to EXINIT. See the appendix to "The vi Editor" article in the *HP-UX Concepts and Tutorials* manual for further details.

/usr/lib/terminfo

This subsystem identifies terminal capabilities for programs such as the *vi* text editor. It defines terminal attributes for all Series 200/300 models and HP supported terminals. It also contains terminal attributes for terminal **not** supported by Series 200/300 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. 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*, and *fsclean* commands.

The file */etc/checklist* is shipped with a single special (device) file name: */dev/root*. This file corresponds to the hard disc on which you installed the system and which contains the root file system. You should add entries for each additional disc drive containing a file system which you want automatically mounted, and for each disc drive used as a swap device.

/etc/catman

Executing the *catman(1M)* command expands the *nroff* versions of manual pages (used by the *man(1)* 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 disc space — the expanded files will use slightly more storage space than the originals. This roughtly doubles the disc usage for manual pages because the original files remain intact. By default, running *catman* causes manual pages in all the */usr/man/manX* directories (where X is 0 through 9, `l`, `n`, or `p`) to be processed and stored in the corresponding */usr/man/catX* directories.

The *cat* directories are not shipped with your system but **must** be explicity created for *catman* to run successfully. Use the following script to create the appropriate *cat* directories:

```
cd /usr/man
for num in 1 2 3 4 5 6 7 8 l n p
do
  mkdir cat$num
done
```

Once the *cat* directories exist, you have three alternatives:

- 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 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. Because the */usr/man/cat* directories now exist, the first time *man(1)* is executed for any given manual entry, the entry is processed, added to the appropriate *cat* directory, and used in subsequent accesses.

  The third alternative is recommended if you can spare some disc space but do not want to use any more that is necessary. With the "build-as-you-go" alternative, the system only fills the *cat* directories as manual pages get accessed by *man(1)*.

When the processed *man* pages exist, you can remove the *nroff* source files.

Table of Contents

The System Administrator's Toolbox 5

Organized alphabetically by task or procedure, this chapter is designed to guide you through your designated tasks as the system administrator. Each major heading in the chapter identifies one or more administrative tasks. The procedure supplies at least one method for achieving the stated task. The following topics are covered:

- Adding/Moving Peripheral Devices[1]
- Adding/Removing Users[1]
- Adding to the checklist File
- Backing up and Restoring the File System
- Becoming the Root User
- Booting HP-UX
- Changing a Password
- Changing and Creating System States
- Climbing the HP-UX Tree
- Communicating with System Users
- Configuring HP-UX[1]
- Controlling Disc Use
- Creating A New File System
- Creating Groups/Changing Group Membership
- Creating a Recovery System
- Initializing Media
- Mounting and Unmounting Volumes
- Removing Optional Filesets and Partitions
- Setting the System Clock
- Setting Up the LP Spooler[1]
- Shutting Down the System
- Updating HP-UX and Installing Optional Products

[1] The *Application Execution Environment User's Manual* has an alternate (simplified) method for this task.

NOTE

If you execute a command and get the error:

command: `file not found`

you may need to load the command using the procedures in "Updating HP-UX".

It is a good idea to make a copy of any of the "configuration" files shipped with your system before you change them. The "Summary" section at the end of Chapter 4 contains a list and a brief description of these files. Other critical files should also be copied before altering them. Any files considered to be critical are mentioned in the procedure that discusses their alteration. You should also log the permission mask, owner, and group ids of any files you backup, so that a clean recovery can be insured. Following this precaution provides you with a backup copy of the original version of the file.

Your first task on a newly installed HP-UX system is to create a recovery system. Read "Creating a Recovery System" for a discussion on the importance of a recovery system, and the steps to create a recovery system.

Adding/Moving Peripheral Devices

The HP-UX operating system requires the existence of special (device) files in order to perform I/O to peripheral devices, such as disc drives, printers, tape drives, and terminals. Special files and an associated topic, block and character I/O, are discussed in the "Concepts" chapter of this manual. This section introduces you to the tools necessary to set up peripherals and their associated special files. The section also discusses terminal hardware configuration and the HP-UX modifications required for communicating with terminals.

You can use either this section to add peripheral devices, or you can use the "Setting Up." sections in the *Application Execution Environment Users Manual*.

NOTE

The pseudo device files **must not** be removed from the system under any circumstances, if HP-UX is to operate properly.

To see all of the currently installed special files on your system, log in on the system and type:

```
ls -R /dev [Return]
```

The system will respond by listing all the special files under the */dev* directory. The listing should include all of the special files shown in the following *Default Special (Device) Files* table. The table also includes brief explanations of the devices associated with these files and other useful information.

You may want to get a listing of a file that will be useful later in this section. If you have a printer already hooked up to your system, and have created the file */dev/lp*, type:

```
config -a
pr /etc/mkdev > /dev/lp & [Return]
```

which tells HP-UX to print the contents of the file *mkdev*, send the output to the printer, and to accomplish this as a background process. *mkdev* is created with the command *config -a*.

Table 5-1: 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
root	b	dynamic	dynamic	Root psuedo device file

Before delving into the software aspects of adding/moving peripheral devices, be sure your peripheral hardware is set up correctly. The guidelines given in the *Peripheral Installation Guide* supplied with your computer will help you configure your hardware. Make sure that you gracefully shut down your system and power off your peripherals before you change any of the address switches on your peripherals. To shut down the system, type:

/etc/shutdown -h [Return]

Wait for the "halted" message to appear on your screen, then *reboot* the system.

For details on *shutdown* and *reboot*, see the section "Shutting Down the System" in this chapter.

Overview of the Task

There are several basic steps required to add or move peripheral devices to your system. Here is an overview of the tasks you will need to accomplish; they are explained in detail later:

1. Using the guidelines covered in the *Peripheral Installation Guide* supplied with your computer and the installation manual supplied with the peripheral device, determine the best place (in terms of HP-IB Bus Addresses, shared sets of I/O resources, expected usage, etc.) to locate the peripheral.
2. Connect the peripheral device. If the peripheral device requires an interface card, set the appropriate switches on the card and install the card in the computer. **Never install 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.

3. Ascertain whether the peripheral device will be addressed as a block or character device, or both (disc drives will require both modes of access). Block and character I/O are discussed in the "Concepts" chapter of this manual, and examples are provided later in this section.

4. Next, determine if the special (device) file necessary to communicate with the peripheral device already exists on your HP-UX system. Some special files were shipped with your system and are shown in the table above. Default files reside in the */dev* directory and follow the naming conventions explained in the *intro(4)* entry in the *HP-UX Reference* manual.

5. If the appropriate special file does not exist for the device in question, you will have to create one. There are two ways to create special files — using the *mknod* command to create a particular device file, or editing, and executing the *mkdev* shell script to create one or more device files. Your choice of which method to use depends on how many device files you need to create and how experienced you are at the process of creating them. Use the *mkdev* script, created by executing `config -a`, or the tables given later in this section to get the parameters (major and minor numbers) needed by *mknod* to create the device files.

6. Make sure your kernel contains the correct drivers (kernel code) to use the peripherals. If you have just installed your HP-UX system, your kernel contains (by default) all possible drivers. If you have reconfigured your kernel, with *reconfig* or *config*, you may not have the full kernel, and you should check the */etc/conf/dfile* file for the list of installed drivers.

   To find out what drivers are necessary to use your peripherals, you have several options:

- 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.
- Look up your product number in the *Configuration Reference Manual*. The drivers necessary for your peripheral are given.

Determining the Peripheral's Location

If the peripheral you wish to add/move is an HP-IB device, determine the select code and bus address where the device will reside. The Installation Guide supplied with your computer lists several guidelines to help you identify an appropriate location for your HP-IB peripheral. The guidelines for interface selection are reviewed here; you should still consult your Installation Guide to determine all available HP-IB bus addresses for devices on these interfaces.

- The system root device (hard disc) is usually located at select code 14, bus address 0 on a disc (high speed) HP 98625A or HP 98625B interface card.
- 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 common.
- A 7971 9-track tape must be placed on a low speed HP-IB. A bus address of 3 is common.
- A 7974 or 7978 9-track tape should be placed on a disc HP-IB, if possible. You can also use that same bus for the root device. A bus address of 3 is common.
- Avoid putting flexible disc drives on the same interface as the root device.
- Plotters and the HP9111 graphics tablet should be placed on separate low speed HP-IB interfaces when possible. Bus addresses of 5 (for plotters) and 6 (for the graphics tablet) are common.

Connecting the Peripheral

Connect the peripheral to your computer at the location you have just determined. The computer's Installation Guide provides instructions for installing an interface card and identifying its select code. The manual supplied with the peripheral details the procedure for connecting it to the computer and setting its address if it has one.

Terminal hardware configuration is covered in the *Peripheral Installation Guide*. You must create the associated special 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 disc drives yourself: HP7911, HP7912, HP7914, HP7933, or HP7935. These SS/80 and CS/80 disc 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 special file** or a **character special file**. If you have a mountable file system, the device file used for communicating with the device holding the file system must be a block special file. File systems can be mounted from hard disc drives, cartridge tape drives, and flexible discs. Cartridge tape and flexible disc are generally used for mounting file systems ONLY if the system is a recovery system.

Character special files are used for communicating with any device that does not hold a mountable file system: terminals, printers, plotters, digitizers, magnetic tape drives, flexible disc drives, and, on occasion, disc mass storage devices. Communicating with a mass storage device (such as a disc) with a character special file causes the system to treat the disc like a magnetic tape drive (see the *disc(4)* entry in the *HP-UX Reference*). Use a character special file to communicate to flexible disc or cartridge tape if you are going to use the *tar(1)*, *tcio(1)*, *cpio(1)*, or *dd(1)* commands, or the LIF or BIF utilities.

In most cases, discs should have both block and character special file entries since discs usually hold mountable file systems. Any cartridge tape or flexible disc drives used for your recovery system should have entries for both block and character special files. All other devices should have character special file entries.

Creating Special Files

This subsection explains how to use the *mknod* command and the *mkdev* script to create special (device) files. The *mkdev* command contains example *mknod* command lines. You will need to modify these lines to reflect your special device files. You can create a copy of the *mkdev* script by executing `config -a`, and view it by executing `more /etc/mkdev`.

Each *mknod* line has the form:

`/etc/mknod` *path_name file_type major minor* [Return]

where *path_name* is the pathname of the special file to be created. You should select a file name for the special file which easily identifies the associated peripheral. The entry *intro(4)* in the *HP-UX Reference* manual describes a naming convention for special files. Use this naming convention to make your system easier to support and maintain. See *mknod(1M)* for additional details of the *mknod* command line. Special files should be put in the directory */dev*. Many commands expect to find special files in */dev* and will fail if the required special file is not there.

file_type is a single character `b`, `c`, `n` or `p`; `b` specifies that the file is a block special file, `c` specifies that the file is a character special file, `n` specifies that the file is a network special file and `p` specifies that the file is a named pipe (FIFO). See *mknod(1M)* for making network special files.

major is the number of the driver used to communicate with the peripheral. A table of major numbers is provided later in this section. For the devices needing both a character special, and a block special, device file, there is a character major number and a block major number.

minor is a value specifying the minor device with its actual address on the I/O bus. The minor number is made up of the select code, bus address, unit and volume numbers. If you set up your peripherals according to the guidelines in the *Peripheral Installation Guide* you already have all the correct minor numbers written on your worksheet.

Be aware that the *mkdev* command automatically sets up the correct access modes on special files. If you choose to use the *mknod* command (instead of modifying, then executing the *mkdev* script), determine the needed major parameters from the tables given later in this section. Follow the guidelines in the "Setting Appropriate Permission Masks" section to insure that correct access modes are set on the special files you create.

Using the mkdev Script

NOTE

Save your old *mkdev* before re-executing *config -a*. Each time you execute *config -a* to create a *mkdev* script, you will overwrite the previous *mkdev*.

Create a copy of *mkdev* by executing `config -a`. Customize the file by following the instructions provided below and by using the detailed comments contained in the script.

NOTE

You must edit and customize the *mkdev* file before using it.

After you have modified the file, you can execute the script by logging in as `root` and typing:

`/etc/mkdev` [Return]

You may wish to save the informative and diagnostic messages produced by the script for later examination. Do this by executing the following command, where *log_file* is the path name of the file where you wish to receive diagnostics:

```
/etc/mkdev  | tee log_file [Return]
```

After you have executed the *mkdev* script, you can examine the results by typing:

```
/bin/ls -lR  /dev  [Return]
```

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 special 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 respectively (see *ls(1)*).

If you make a mistake, delete the special files you wish to change and re-create them by editing and executing the *mkdev* script again.

NOTE

Do not delete the special files listed in Table 5-1. Deleting these required files will cause severe problems because the system needs these files to operate properly. For example, you will not be able to access your root device if you delete its special file. Table 5-1 has a list of the default special files.

Editing the mkdev Script

The *mkdev* script contains a series of *mknod* "templates" for creating every allowable device/address combination. They are called templates because they are models that must be modified to reflect the actual parameters you want associated with the peripheral's special file.

Because the script contains a lot of information, the first time you edit it be prepared to spend some time with the script and read through its comments. Do not expect to absorb all of the details; only try to familiarize yourself with its organization and general approach. Do not let its size and density overwhelm you; subsequent modifications become easier.

The script is organized into device classes. Separate sections and templates exist for: miscellaneous devices, terminals, CS/80 and SS/80 mass storage devices, non-CS/80 and SS/80 mass storage devices, magnetic (9-track) tape devices, printers, general HP-IB devices (plotters, digitizers, graphics printers, etc.), HIL, and CRT graphics devices.

In general, modifying the *mkdev* script consists of the following steps:

1. Find the template or device class that corresponds to the peripheral.
2. Read the script's instructions pertaining to that device class.
3. Decide what name to use for the special file associated with the device.
4. Copy the *mknod* template if indicated by the instructions.
5. Modify the *mknod* template to correspond to the name you chose. Use the *intro(4)* naming convention or one of your own.
6. Fill in any of the template's missing parameters (or placeholders for parameters) where the instructions indicate; this includes the major and minor numbers for some device classes.
7. Remove the comment sign (#) in front of the modified template so the line will be executed when you run the script.

Filling in the mknod Template

The following sections are structured by "device classes" and contain the information you need to create special (device) files using *mknod*. These device class sections are sequenced just as in the *mkdev* script; most of this material duplicates information from *mkdev*; it is provided here for your convenience.

If you are creating device files for devices other than the devices in the *mkdev* script, see the appropriate manual. For instance, if you are creating device files for HP-HIL devices, read the manual on using HP-HIL devices in *Concepts and Tutorials*.

The rest of the "Adding/Moving Peripheral Devices" section contains:

- A table of all major numbers for each supported device class. This table can be obtained by executing `lsdev` [Return].
- Specifics on Miscellaneous Devices.
- Specifics on Terminals and Modems.
- Specifics on Pseudo Terminals.
- Specifics on Hard Discs and Cartridge Tapes.
- Specifics on Magnetic Tape Devices.
- Specifics on Printers.
- Specifics on Plotters and Digitizers.
- Terminal Configuration Information.
- Special Considerations for Terminals.

NOTE

If you use the *mknod* command outside the *mkdev* script, you must set the permissions on your device files to be the same as the permissions specified in *mkdev*.

Here are the major numbers for each supported device class. The specific tables in each of the following sections also contain the applicable major numbers.

Major Numbers for Devices

Major #	Devices	
MISCELLANEOUS		
0	console	
2	tty	
3	mem	
3	kmem	
3	null	
8	swap	
255	root	
TERMINALS		
1	cua01	
1	cul01	
1	tty01	
1	tty0x	
CS/80 and SS/80 DEVICES		
4/0	hd	Use 4 for character devices, 0 for block devices.
4/0	ct	Use 4 for character devices, 0 for block devices.
4/0	md	Use 4 for character devices, 0 for block devices.
NON-CS/80 DEVICES		
Internal flexible discs		
6/1	fd.0	Use 6 for character devices, 1 for block devices.
6/1	fd.1	Use 6 for character devices, 1 for block devices.
Amigo discs		
11/2	9121.0	Use 11 for character devices, 2 for block devices.
11/2	9121.1	Use 11 for character devices, 2 for block devices.
MAGNETIC TAPE DEVICES		
5	HP 7970	
9	HP 7974	
9	HP 7978	
PRINTERS		
7	lp	
26	ciper	
SRM		
13	srm_node0	
13	srm_node1	

Major Numbers for Devices (continued)

Major #	Devices
PTYMAS and PTYSLV	
16	ptym0
17	pty0
16	ptym1
17	pty1
IEEE802 and ETHERNET	
18	lan
18	ieee
19	ether
PLOTTERS / DIGITIZERS / PRINTERS (RAW MODE) / HP-IB(4)	
21	hpib-device
21	hpib-channel
GPIO	
22	gpio
KEYBOARDS	
23	raw_8042
24	hil
25	hilkbd

NOTE

Each section of *mkdev* sets a permission mask before executing the *mknod* commands. If you use *mknod* commands outside the *mkdev* script, you must insure that permission masks are properly set up.

Miscellaneous Devices

The miscellaneous device class are the special files that the system needs to run properly. Each HP-UX installation must have the special files */dev/null*, */dev/console*, */dev/mem*, */dev/kmem*, */dev/root*, */dev/swap* and */dev/tty*. The special file */dev/null* is a null file (a "bit bucket") used by many HP-UX commands. The special file */dev/console* identifies the system console and the special file */dev/tty* is a synonym for the control terminal associated with a process group.

These miscellaneous special (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 *mkdev* script and removing the comment sign (the # character) from in front of the corresponding entry. Alternatively, recreate it with the `mknod(1M)` command using the character/blocked designation, major, and minor numbers given below.

Device	C/B	Major	Minor	Notes
console	c	0	0x000000	Mandatory (do not change)
tty	c	2	0x000000	Mandatory (do not change)
null	c	3	0x000002	Mandatory (do not change)
mem	c	3	0x000000	Mandatory (do not change)
kmem	c	3	0x000001	Mandatory (do not change)
root	b	dynamic	dynamic	Mandatory (do not change)
swap	c	8	0x000000	Mandatory (do not change)

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

Communication ports — user terminals as well as modems — need to be identified by one or more special (device) files, depending on the intended use of the port. `tty` files are required for terminals (hard-wired ports). For ports that receive incoming signals ("dial in" modems), the naming convention, `ttyd`, should be used for files . Ports that transmit signals ("dial out") require `cua` and `cul` special files.

The general template for ports is:

Device	C/B	Major	Minor	Notes
ttyxx	c	1	0xScAdUV	Sc, Ad, U, and V are explained below
ttydxx	c	1	0xScAdUV	Sc, Ad, U, and V are explained below
cuaxx	c	1	0xScAdUV	Sc, Ad, U, and V are explained below
culxx	c	1	0xScAdUV	Sc, Ad, U, and V are explained below

`xx` is a two digit line identifier (in the device file name). The minor number is a hex number representing the following:

Sc — the select code of the interface being used

Ad — the port address for each port.

If you are using a single port interface such as the HP 98626A card, the port address will be 00 (so the minor number would be `0xSc000V` in our first example above).

U — Always 0 for terminals and modems.

V — The hex representation of a 4-bit binary number. The 4 bits correspond to `0DCO`.

- 0 — Always 0
- D — 0=direct connect, 1=modem
- C — 0=CCITT protocol (Europe), 1=Simple protocol (U.S.)
- O — 0=dialout modem, 1=dial-in modem

An example configuration might be:

Port #	tty File	cua File	cul File
1	/dev/console	no entry	no entry
2	/dev/ttyd00	/dev/cua00	/dev/cul00
3	/dev/tty01	no entry	no entry
4	/dev/tty02	no entry	no entry
.	.	.	.
.	.	.	.
.	.	.	.
N	/dev/ttyxx	no entry	no entry

where *xx* is the two digit tty/cua/cul number described above. An example of creating the `tty`, `ttyd`, `cua`, and `cul` special files for a port is provided in the "Creating Special Files for Ports" section below. are not connecting a modem device.

Creating Special Files for Ports

Assume that you want to create special files for a modem at select code 20 (decimal 20 = hexidecimal 14), (using an HP98626 card), and associate it with line 10 (i.e., */dev/ttyd10*). Because the modem will be used as a dial-in and dial-out port, use the following forms of *mknod*.

```
mknod  /dev/cua10   c   1 0x140001
mknod  /dev/cul10   c   1 0x140001
mknod  /dev/ttyd10  c   1 0x140000
```

Notice that the minor number for the `cua` and `cul` special files, ends with a `1`. There are now three special (device) files associated with the dial-in and dial-out modem at select code 20.

Suppose you want to set up a direct-connect for a HP 98642 on line 11 at select code 13 (13 decimal = hexidecimal d), type:

```
mknod  /dev/tty11 c 1 0x0d0100
```

NOTE

A single-user HP-UX Series 200/300 system can have a maximum of 2 ports: a system console and 1 additional port which is used with *uucp*.

A multi-user HP-UX Series 200/300 system can have a maximum of 16 ports. (This is the maximum number of ports that may be used by *login*). For some systems (and for certain applications), using this maximum number of ports may be counter-productive in terms of performance, disc space and available memory.

Pseudo Terminals

Often applications need some form of software support which enables an application program to pretend it has a terminal. In HP-UX 5.0, this facility has been added with a new facility called a **pseudo terminal**. A pseudo terminal is a pair of character devices: a **master** device and a **slave** device. The slave device provides processes (in this case, user applications) an interface identical to that described in *termio(4)* of the *HP-UX Reference* manual.

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.

According to HP-UX naming conventions, all pseudo terminal devices are located in the directories `/dev/pty` (slaves), and `/dev/ptym` (masters) The master side special 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.**

Device	C/B	Major	Minor	Notes
ptyXX	c	16	0x00YYYY	Master side of pseudo terminal
ttyXX	c	17	0x00YYYY	Slave side of pseudo terminal

YYYY is a unique hexadecimal value, in the range of 0 to *npty-1*, used to identify the relationship between master and slave. *npty* is a configurable parameter, see "Configuring HP-UX" if you wish to change this value, or "Appendix D" if you wish to read about the parameter.

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.

As an example:

```
mknod /dev/ptym/ptyp0 c 16 0x000000
mknod /dev/pty/ttyp0  c 17 0x000000
```

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.

Your application's documentation will tell you how many pseudo terminals you need. For example, HP Windows/9000 needs 3 master/slave pairs per window.

For more information on pseudo terminals, see both the *termio(4)* and *pty(4)* sections of the *HP-UX Reference* manual.

Hard Discs, Flexible discs, and Cartridge Tapes

Each hard disc, flexible disc, and cartridge tape drive must have a special file associated with it. You must create both a block special file and a character or **raw** special file for each hard disc, flexible disc, and cartridge tape drive. The major number is different for block and character files. he address-dependent minor number is the same for both block and character entries.

The minor numbers in this device class have special meaning. Each minor number consists of a select code, HP-IB address, unit number, and volume number. For each possible HP-IB address where hard discs may be located, there can be several minor numbers. For example, if you have an HP7946A disc/tape drive, the disc drive has a unit number of 0 and the tape drive has a unit number of 1. select code = 7, HP-IB address = 0 disc drive minor number = 0x070000 cartridge tape minor number = 0x070010

There are *mknod* templates that correspond to each minor number at each possible address, including a template for the root device (*/dev/root*) that was installed with your system. You need to determine which template you need based on your disc/tape drive and on its controller option (shared or separate). The *Configuration Reference Manual* has information on all hard disc, cartridge tape, and flexible disc drives.

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.

To help clarify our discussion, *mt* refers to 9-track magnetic tape; *ct* refers to cartridge magnetic tape; *md* refers to microflexible disc. Distinguish the name of a character device from a block device by prefacing the raw (character) device name with "r". We suggest that when naming device files, you use the following representation:

- **hd79xx** and **rhd79xx** represent one of the following CS/80 discs: HP 7911, HP 7912, HP 7914, HP 7933, or HP 7935. Replace the **xx** in the entries with the disc's model number.
- **ct79xx** and **rct79xx** represent the cartridge tapes for the drives that have this device.
- **mt79xx** and **rmt79xx** represent 9-track magnetic tape.
- **mk91xx** and **rmd91xx** represent microflexible disc.

For example, for the HP 9134 disc drive, the special file names are:

```
/dev/hd9134        (block)
/dev/rhd9134       (character)
```

The *hd/ct* combination drives with a single controller, the *ct* will be unit 1. If you ordered a dual controller option for a *hd/ct* drive, such as the hp 7114TC, the unit number for both devices is 0, but the bus addresses for the two controllers must be set to different values.

You can replace the supplied device names with an appropriate name that contains the model number of your hard disc; see the naming conventions in the *intro(4)* entry of the *HP-UX Reference*.

The "root" Disc

There is a default special device file, named *root*, created during the installation process and necessary for certain operations of the HP-UX file system (such as checking file system integrity with the *fsck* command). This psuedo 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:

```
mknod /dev/root b 255 0xFFFFFF
```

Note that the major and minor numbers shown by performing a `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 *mknod* it on every system.

Magtape Volume and Unit Numbers

The unit number field of the minor number has special meaning when creating special files for the HP 7970E, HP 7971A, HP 7974A or HP 7978A/B magnetic tape drives. The single hexadecimal unit 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 the following table (bits 6 and 7 select the tape density, while bits 4 and 5 represent the unit number, and "x"s represent "don't care"):

Bit Settings

7	6	5	4	Selects
1	0	x	x	Density = 6250 bpi (HP 7978 only)
0	1	x	x	Density = 1600 bpi (All mag tapes)
0	0	x	x	Density = 800 bpi (HP 7974, opt 800 only)
x	x	0	0	Select Unit 0
x	x	0	1	Select Unit 1, etc.

The volume number field of the minor number also has special meaning when creating special 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 the following table:

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.

Each bit of the minor number is explained in the *Configuration Reference Manual*, under the HP part number of your magtape drive.

For more information on the use of magnetic tape, see Chapter 3 (Concepts) of this manual, *Configuration Reference Manual*, or the *mt(4)* section of the *HP-UX Reference* manual.

Printers

Although all printers on HP-UX are character devices, they can be used in different "modes": you can communicate with printers using drivers that either interpret or do not interpret the data. For a complete list of Series 300 supported printers and their major and minor numbers see the *HP 9000 Series 300 Configuration Reference Manual.*

Volume Numbers for Printer Special Files

The volume number field of the minor number has special meaning when creating special files for printers. The single digit hexadecimal value is made up of four bits. The table below shows and describes the bits involved:

Bit Order	When Clear (0)	When Set (1)
0	Printer	Plotter (raw) (overrides all other bits)
1	Normal	No overprint capability
2	Normal	Uppercase only
3	Normal	No page eject on open and close.

Using Printers as Spooled Devices

Printers can be accessed, through the line printer spooler (see *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, create the printer's special file as explained here, then follow the instructions in the "Setting Up the LP Spooler" section of this chapter to change the special file's ownership, group, and permission mode. 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 given earlier in this section and 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; see the *intro(4)* entry in the *HP-UX Reference* for a suggested naming convention. Also, execute `umask $defumask` (or its equivalent, `umask 111`) before using *mknod*.

Plotters and Digitizers

To create a special file for a device in this class, first determine the peripheral's location (i.e., interface and bus address). Use the guidelines given earlier in this section and in the Installation Guide supplied with your computer to choose an appropriate location. Then use the *HP 9000 Series 300 Configuration Reference Manual* to find the major number that corresponds to that specific device. Finally, insert a mknod line in your *mkdev* script to create the special file for the printer using the *mknod*.

Useful Naming Conventions

You must assign a **unique** special file name to each entry you create. The *intro(4)* entry in the *HP-UX Reference* explains special file naming conventions. Generally, use `rlp` followed by the product number for raw printers, `plt` followed by the product number for plotters, and `dig` followed by the product number for digitizers. If more than one device with file names. For example, to differentiate between two HP 9872 plotters, name the first one `plt9872` and the second `plt9872.1`.

Terminal Configuration Information

The Installation Guide supplied with your computer discusses the hardware aspects of hooking up a terminal to your system. This section offers the software configuration information you need.

After dealing with the hardware hookups, terminals must be configured so they can "talk" to HP-UX. Series 200/300 HP-UX requires that terminals be configured to have the characteristics listed below. If a particular configuration option is not available on your HP terminal, then the option is already properly chosen (as a default value) by the terminal.

Use the *HP 9000 Series 300 Configuration Reference Manual*, "Appendix A", for typical terminal and datacomm parameters. The manual supplied with the terminal describes how to use the function keys to configure the terminal. Generally, you will press a key that chooses the "terminal configuration" option and alter the appropriate fields by answering prompts from the terminal's configuration program.

Except when using the terminal as a system console (discussed below), you may use any **baud rate** that the terminal will handle. The baud rate setting on the terminal **must** match the baud rate parameter in the *getty* command located in the terminal's entry in the `/etc/inittab` file as discussed below.

If you are using the terminal as the system console, set the terminal's baud rate at 9600 to match the HP-UX expectation for the system console. After the system is installed and running, you may change some of the configuration parameters to suit your own needs. For example, changing the HP-UX expectation for a particular baud rate is done by modifying one of the parameters to the *getty* command located in the `/etc/inittab` file (and associated with the terminal in question). Further information is provided in the section that follows ("Special Considerations for Terminals"), in the *getty(1M)*, *gettydef(5)*, and *inittab(5)* entries in the *HP-UX Reference*.

Special Considerations for Terminals

When a terminal is added to the system, you must perform the steps described in the preceding section and 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` entries have the form:

model_number location

where *model_number* is the the product number of the terminal or computer (as defined in `/etc/terminfo`) and *location* is the special (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`:

```
9836    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
```

/etc/inittab entries

The file, */etc/inittab* is described in "System Boot and Login". This section discusses entries specific to terminals.

Most `/etc/inittab` entries have the form:

id:rstate:action:`/etc/getty` *-txxx special_file_name N # comment field*

The first three fields (shown as *id*, *rstate*, and *action*) are discussed in *init(1M)* and *inittab(5)* in the *HP-UX Reference* manual. The typical values for these fields are: `id` = unique two character string, `rstate` = 2, and `action` = `respawn` (for continuous).

The two character string `id` 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 the getty will be run in (see *init(1M)*. 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).

The fourth field must contain the `/etc/getty` command; it is immediately followed by three parameters. The first parameter, *-t xxx*, is the optional time-out option for use with modems. The second parameter, *special_file_name*, is the file name (*tty04*) — not the complete path name (`/dev/tty04`) — of the terminal's or modem's character special file. The named file must reside in the `/dev` directory. The third parameter to *getty*, represented by *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, see the *getty(1M)* and *gettydef(5)* 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` 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 special file (`tty04`) for the terminal on which *getty* is invoked. 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. `respawn` causes the "login:" prompt to be redisplayed after a user logs out. Refer to the "System Boot and Login" chapter in this manual, and to the *getty(1M)*, *gettydef(5)*, and *inittab(5)* entries in the *HP-UX Reference* for further details.

Dealing with an Unresponsive Terminal

Some common user errors, such as trying to display the contents of a binary file (for example, an executable program) on a terminal, will result in the terminal being left in some unusable configuration.

If, for whatever reason, a terminal will not respond (or does not appear to respond) to your commands, several solutions are available. The first is to simply log off the system (using *exit*, [CTRL]-[D], or whatever the recommended method for the shell you are running) and then login again. Generally this will clear up any problems.

A second solution is to do a hard reset of your terminal. If you do a hard reset, the tabs may become unset. To determine if you need to reset the tabs, execute *ls*. If the files are listed in a column instead of across the page, you need to reset tabs. To reset the tabs, execute the */usr/bin/tabs* command.

If you have properly set up the /etc/ttytype file, the *tset(1)* command will generally clear the problem. To use this, type the following **exactly** as shown — blanks are significant. You may not see anything echoed on the screen.

[CTRL]-[J] `stty sane erase "^H" kill "^U" echo`[CTRL]-[J]

This sets the "erase" character to [CTRL]-[H] and the "kill" character to [CTRL]-[U]. When the screen and keyboard response returns, type:

`tset` [Return]

Your terminal should now exhibit proper behavior. This may not work if the terminal is in 8-bit mode.

As another option to this procedure execute the *tabs(1)* command.

Adding/Removing Users

The material in this section covers only the software or configuration aspects of adding/removing a user to/from the HP-UX system. If the user will have his own terminal, you need to install the terminal and do some associated configuration before the user can log in to the system; see the "Adding/Moving Peripheral Devices" section of this appendix.

Each user is defined by an entry in */etc/passwd*. Without this entry, the user cannot log in. To add a user to the system, you must add a line to this file and do a few other things. A complete description of the */etc/passwd* file can be found in the *passwd(5)* entry in the *HP-UX Reference* manual.

You can also use *reconfig* to add and remove users. See "Adding and Removing Users" in Chapter 10 of the *Application Execution Environment Users Manual* for details on using *reconfig*.

Two approaches for adding users to the system are offered here. Both approaches require the aforementioned entry in the */etc/passwd* file and the procedure for creating that entry is explained next. Following that, a listing of a shell script (that partially automates the process of adding users to the system) is supplied. Finally, a step-by-step method for adding users is presented. If you expect to add a few users to the system, it will probably be worth your time to type in the shell script listed below to ease the task of adding users. If you have a single-user system or expect to add only one or two users to your system, the step-by-step process will probably be your best choice. Both approaches accomplish the same task; the "automation" of adding users is the only functional difference between the two.

Creating the /etc/passwd Entry

To create an entry in the */etc/passwd* file for the new user, first log in to the system as the super-user `root`.

If this is the first time you are following this procedure, make a copy of the original */etc/passwd* 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, using the text editor of your choice (such as *ed*, *vi*, or *ex*), edit the file */etc/passwd*. 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

The colon character (:) is used to delimit the various fields in the entry.

user_name is the user's login name, consisting of 1 to 8 lowercase letters, and at least one number and/or special character, and any other characters you desire.

: : represents an empty password field. Passwords and the *passwd* command are discussed later in this section.

user_id is the real user ID — a **unique** integer value that the system uses to identify the user. If the real user ID is 0, then that user has super-user 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 6001, for this field. Userids greater than 6001 will be mapped onto 6001.

NOTE

There should be only one entry per real user ID; the user whose real user ID is 0 should be named root.

---

group_id is the real group ID — an integer value shared by all members of the same group. This entry corresponds with the group entry in */etc/group*; see the "Creating Groups/Changing Group Membership" section in this chapter for details.

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 is the absolute path name of the user's login directory. This 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. A user's login directory is usually a subdirectory of the */users* directory and has the same name as the user's login name. For example, a user whose last name is Young might have login name **young** and home directory `/users/young`. The directory should be owned by the user.

command is 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) 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.

The *pwck(1M)* command should be used to verify the format of */etc/passwd/* Once you are satisfied with the contents of */etc/passwd*, write the modified file to the disc and terminate the editing session.

The "Makeuser" Script

This section contains the shell script for adding users to the system. This script assumes that certain files are located where they were when the system was shipped. If you have moved these files, edit the script to match their new locations.

To use this script, you need to:

- Log in to the system as the super-user `root`.
- Use one of the text editors to create the */etc/makeuser* file by typing in the listing below. The name */etc/makeuser* is only a suggestion.
- Change the mode of the file by typing:

  chmod 744 /etc/makeuser [Return]

  This gives you read, write and execute permission on the file but restricts the access of all other users to read permission.
- After creating the new user's */etc/passwd* entry, execute the script by typing:

  /etc/makeuser *user_name* [Return]

  where *user_name* is the new user's username from the */etc/passwd* entry.

Here is the "makeuser" shell script:

```
: #/etc/makeuser: create a new user

USERS=/users
if [ $# != 1 ]
then
        echo "Usage: makeuser name"
        exit 1
fi
if [ -d $USERS/$1 ]
then
        echo "Home directory already exists."
        exit 1
fi
if grep \^$1\: /etc/passwd >/dev/null
then
        :
else
        echo "No password entry."
        exit 1
fi
mkdir $USERS/$1
chown $1 $USERS/$1

ls -ld $USERS/$1
ls -la $USERS/$1
echo "Remember to add the new user to a group."
```

The Step-by-Step Method

If you decided not to use the "makeuser" script, here is a step-by-step procedure for accomplishing the same task. The following procedure assumes that certain files are located where they were when the system was shipped. It also assumes that an appropriate entry for the new user already exists in the */etc/passwd* file.

1. Create a login directory for the user with the *mkdir* command by typing:

   `mkdir /users/`*user_name* [Return]

   where *user_name* is the new user's name and the entire path name (`/users/`*user_name*) matches the `login_directory` field of the user's */etc/passwd* entry.

2. Create the *.exrc* and login files for the user (the name of the login file depends on the default shell, e.g. *.profile* and *.exrc*) If the login file exists in a user's login directory, the shell attempts to execute that file at the end of the login process. This file typically contains shell commands and environment variable definitions which customize the user's environment and/or automatically run one or more programs. If the file *.exrc* exists in the user's login directory, it is used to map terminal characteristics and key definitions for some of the HP-UX text editors.

3. Because you (the user `root`) created the new user's directory and copied the *.profile* and *.exrc* files into his directory, you own both his directory and his files. To change the ownership to the user, type:

   chown *user_name* `/users/`*user_name* [Return]

   (where *user_name* is the user and `/users/`*user_name* is his login directory) to change the directory ownership and type:

   chown *user_name* `/users/`*user_name*`/.[a-z]*` [Return]

   (where *user_name* is the user) to change the ownership of the files. The specification `/users/`*user_name*`/.[a-z]*` matches all the files in the user's directory that begin with a period and are followed by a lower-case letter and then anything else.

4. Check the ownership and permissions of the user's directory by typing:

   `ls -ld /users/`*user_name* [Return]

   and check the status of the user's files by typing:

   `ls -la /users/`*user_name* [Return]

   See the *ls(1)* entry in the *HP-UX Reference* for an explanation of the display.

5. If you are using the group access features available on HP-UX, see the "Creating Groups/Changing Group Membership" section in this chapter.

Whether you used the "makeuser" method or the step-by-step method, the new user is now installed on the system. A few optional considerations still bear examination.

Some Optional Items

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 the *chgrp(1)* and *group(5)* entries in the *HP-UX Reference*.

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.

If you are adding a terminal for this user, see the "Adding/Moving Peripheral Devices" section to set up the terminal and add entries to the */etc/ttytype* and */etc/inittab* files.

Using reconfig

You can also use the "Adding a user" option in *reconfig(1M)*. This procedure is explained in the *Application Execution Environment User's Manual*.

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 super-user. Then type:

passwd *user_name* [Return]

and respond to the system's prompt for a new password; see the *passwd(1)* entry in the *HP-UX Reference* for details. This will set the new user's password to the password you typed.

Forcing the User to Create a Password

If you neither want to create a password for the user nor leave it up to him to create one, you can set a parameter that forces him to create a password the first time he logs in to the system. To accomplish this requires that, in the user's */etc/passwd* entry, the password field's optional **aging** field contains two periods. The optional aging field is separated from the password field with a comma (see the *passwd(5)* entry for details). 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 from the System

To remove a user from the system,

1. remove the user's $HOME directory and his other directories and files to release the disc space for other users. The easiest way to remove the user's files is to type:

   find / -user *user_name* **-exec rm {} \;**

   Note that this **removes all** the user's files and directories.

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 "System Boot and Login" chapter in this manual, and *inittab(5)* in the *HP-UX Reference* for details.

Suspending a User from the System

If, for whatever administrative reason, you wish to keep a user off the system for a time, you can temporarily revoke his login privileges by modifying his line in */etc/passwd*. If his password is replaced with an asterisk (*), he cannot use the system until you delete that asterisk (and then give him a new password). The following example illustrates this:

```
atilla:*:101:5:Atilla the Hun:/users/atilla:/bin/sh
```

Adding to the Checklist File

The */etc/checklist* file contains a list of file systems and swap devices. The following HP-UX programs use the checklist file for information:

- During bootup, or if you execute the *mount -a* command all **file systems** (type "rw" and "ro") in */etc/checklist* will automatically be mounted,
- during bootup (if you add `swapon -a` to */etc/checklist*), or if you execute *swapon -a*, all **swap devices** (type "sw") in */etc/checklist* will be enabled,
- if you execute *fsck* without a list of filesystems, all filesystems in */etc/checklist* will be checked,

The format of an entry in */etc/checklist* is:

```
dev_name   mnt_dev   mount_fld   type   order   0   comment
```

Where:

dev_name — The device name used by *fsck* and *swapon*. If the file system is the root file system, */dev_name* must be the **block special device file name** associated with the root file system (usually */dev/root*). If the file system is a non-root file system, *dev_name* must be the **character special device file name** associated with the file system (for example, */dev/rhd* rather than */dev/hd*). If the file system is marked as a swap device, this must be the **block special device file name**.

mnt_dev — The **block special device file name** used by *mount* and *swapon*.

mount_fld — The directory where *mnt_dev* is to be mounted.

type — The possible values for type are:

- rw read/write permission for the file system
- ro read-only permission for the file system
- sw swap device
- xx ignore this entry

order — *fsck* will check the file systems in the order you specify. The root file system (*/dev/root*) should always be "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". You will probably only use "1" and "2".

0 — This field always has a value of "0".

comment — This field is preceeded by a "#". You can put any comment in this field.

For example, the entries in Table 5-2 will allow your system to automatically (at bootup):

- *fsck* both the */dev/root* and the */dev/rhd* file systems (*fsck* ignores all "sw" and "xx" entries),
- mount */dev/hd* on the directory */hd2* with read/write permission (using the *mount -a* command automatically executed at bootup),
- turn on swapping (if you added the command */etc/swapon -a* to the */etc/rc* file).

Also, the *shutdown* command executes *umount -a*, and *fsck*, when run manually, reads the list of file systems from */etc/checklist* if you do not provide a list of filesystems in the command line.

If you are temporarily removing a disc from the system, you should invalidate the entries for that disc by changing the disc's type to "xx".

Table 5-2: Example /etc/checklist Entries

/dev/root	/dev/root	/	rw	1	0	# root
/dev/rhd	/dev/hd	/hd2	rw	2	0	# 7945
/dev/hd	/dev/hd	/hd2	sw	2	0	# 7945 swap

Note that root is assumed to be a swap device, so you don't need a separate swap entry for the root file system. For all other file systems, if they are also swap devices, you must have both a file system entry (if you wish it to automatically be mounted) and a swap entry (if you wish it to be automatically enabled for swapping).

Backing up and Restoring the File System

The backup process lets you 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 disc, or 9-track tape. Recovery procedures are also detailed in this section.

To minimize the chance of loss, backups should be stored 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.

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. Complete backups (as compared with partial ones) take significant time and the media costs involved can be high.

Daily Archive Backups

One backup strategy is to make complete backups of the file system on a daily basis. A complete backup is often called an **archive backup**. Restoring the file system from a full archive backup consists of restoring the most recent backup tape or flexible disc. 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.

Incremental Backups

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.

Mixing Archive and Incremental Backups

The ability to use both archive and incremental backups leads to a useful backup strategy that is cost-effective and time-efficient without sacrificing the integrity of the file system. This method involves making archive (complete) backups of the file system once per week and supplementing this with daily incremental backups.

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.

There are two backup scripts provided with your system, *backup* and *backupf*. Use *backupf* if your backup will be on flexible disc. Use *backup* if your backup will be on cartridge tape or 9-track tape. If you use 9-track tape, you must modify the *backup* script. The backup scripts (*backupf* or *backup*) can be used to do either a full or an incremental backup. You can customize these scripts, as explained later in this chapter.

Hewlett-Packard recommends the following backup schedule:

- full archival backup weekly or biweekly
- store the backup at least 2 weeks (see Figure 5-1)
- daily incremental backup
- take an archive backup and store it in a permanent archive monthly.

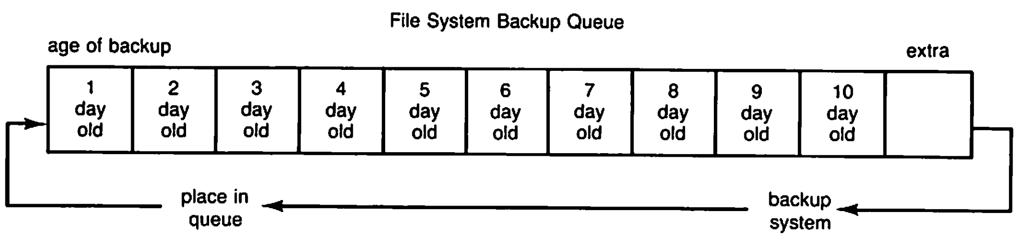


Figure 5-1: File System Backup Queue

You should continue making/using 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).
- you feel it is necessary to create a new archive backup for any reason.

Backing Up the System

The file system can be backed up onto the following media:

- cartridge tape,
- flexible disc,
- magnetic tape.

You can either:

- perform a regular system backup using either the *backup* command or the *backupf* command, or
- backup selected files using the *cpio* or *tar* commands to copy selected files to a backup volume.

The following sections describe how to use the correct backup script for your backup media.

Backing up Onto Flexible Disc

Use the backup script, *backupf*, to backup your file system onto flexible discs.

To prepare to backup, answer the following question:

- Is character special file associated with your flexible disc drive called */dev/rfd*?

  If not, you must edit */etc/backupf* to assign the correct device file to the `outdev` variable or use the procedures in "Adding/Moving Peripheral Devices" in this chapter to create the character special device file called */dev/rfd*.

1. Login as the super-user `root`. See the section "Becoming the Root User" in this chapter. If you are not the super-user you will have problems copying files that you do not own or have permission to access.
2. Execute the shutdown command by typing:

   shutdown 0 [Return]

   For more information on the shutdown procedure, see the section "Shutting Down the System" in this chapter.
3. Insert the flexible disc.
4. Perform the backup.

The *backupf* script will write (log) 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.

If the *backupf* command runs out of room on the flexible disc, it prompts you to insert a new medium.

- **If you are doing an archival backup**, do the following steps:
  a. Change to the root directory by typing:

     cd / [Return]

  b. Type:

     /etc/backupf -archive [Return]

  c. Check the file system by typing in:

     fsck -p *[filesystem_name]* [Return]

  For more information on *fsck* see "Checking Your File System" in this chapter.

NOTE

Do not run *fsck -p* without reading the "Using the FSCK command" (Appendix A).

- **If you are performing an incremental backup**, do the following:
  a. Change to the root directory by typing:

     cd / [Return]

  b. Type:

     /etc/backupf [Return]

- **If you are backing up selected files**, do the following:
  a. Go to the directory containing the files:

     cd *directory_name* [Return]

  b. Assuming your flexible disc drive is associated with the device file */dev/rfd* (if not, replace "rfd" with the correct character special device file name), use the *cpio* command as follows:
     - to back up all files and subdirectories from the current directory, type in:

       `find . -print | cpio -ocBx > /dev/rfd` [Return]

     - to back up all files in your current directory, type in:

       `ls | cpio -ocBx > /dev/rfd` [Return]

     - to back up selected files in your current directory and subdirectories, type in:

       `ls` *file-name dir-name/file* `| cpio -ocBx > /dev/rfd` [Return]

  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.

5. When the backup has finished, remove the flexible disc.
6. Label the flexible disc with the date and the type of backup (archive or incremental) and store it in a secure place.
7. Examine the information and messages sent to the file, */etc/backuplog* to determine if any errors occurred during the backup process.

Backing up Onto Cartridge Tape

Use the backup script, *backup*, to backup your file system onto cartridge tape.

To prepare to backup, answer the following questions:

- Is character special file associated with your cartridge tape drive called */dev/rct*?

  If not, you must edit */etc/backupf* to assign the correct device file to the `outdev` variable or use the procedures in "Adding/Moving Peripheral Devices" in this chapter to create the character special device file called */dev/rct*.

1. Login as the super-user `root`. See the section "Becoming the Root User" in this chapter. If you are not the super-user you will have problems copying files that you do not own or have permission to access.

2. Execute the shutdown command by typing:

   `shutdown 0` [Return]

   For more information on the shutdown procedure, see the section "Shutting Down the System" in this chapter.

3. Insert the cartridge tape.

4. Perform the backup.

   The *backup* script will write (log) 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.

   If the *backup* command runs out of room on the cartridge tape, it prompts you to insert a new tape.

   - **If you are doing an archival backup**, do the following steps:

     a. Change to the root directory by typing:

        `cd /` [Return]

     b. Type:

        `/etc/backup -archive` [Return]

     c. Check the file system by typing in:

        `fsck -p` *[filesystem_name]* [Return]

     For more information on *fsck* see "Checking Your File System" in this chapter.

     NOTE

     Do not run *fsck -p* without reading the "Using the fsck Command" (Appendix A).

   - **If you are performing an incremental backup**, do the following:

     a. Change to the root directory by typing:

        `cd /` [Return]

     b. Type:

        `/etc/backup` [Return]

- **If you are backing up selected files**, do the following:
  a. Go to the directory containing the files:

     `cd` *directory_name* [Return]

  b. Assuming your cartridge tape drive is associated with the device file */dev/rct* (if not, replace "rct" with the correct character special device file name), use the *cpio* command as follows:

     - to back up all files and subdirectories from the current directory, type in:

       ```
       find .-print | cpio -ocBx | tcio -o  /dev/rct [Return]
       ```

     - to back up all files in your current directory, type in:

       ```
       ls | cpio -ocBx | tcio -o /dev/rct [Return]
       ```

     - to back up selected files in your current directory and subdirectories, type in:

     `ls` *file_name dir_name/file_name* `| cpio -ocBx | tcio -o /dev/rct` [Return]

  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.

5. When the backup has finished, remove the cartridge tape. Be certain that the cartridge tape is logically "unloaded" **before physically removing it from the tape drive**; see the "Special Considerations" material later in this section.

6. Label the cartridge tape with the date and the type of backup (archive or incremental) and store it in a secure place.

7. Examine the information and messages sent to the file, */etc/backuplog* to determine if any errors occurred during the backup process.

Backing up Onto Magnetic Tape
Use the backup script, *backup*, to backup your file system onto magnetic tape.

To prepare to backup, you must edit the */etc/backup* script and replace these lines:

```
cpio -ocx                    |
tcio -o $outdev
```

with this line:

```
cpio -ocBx > /dev/rmtxx
```

where */dev/rmtxx* is the special character device file associated with your magnetic tape drive.

1. Login as the super-user `root`. See the section "Becoming the Root User" in this chapter. If you are not the super-user you will have problems copying files that you do not own or have permission to access.

2. Execute the shutdown command by typing:

   shutdown 0 [Return]

   For more information on the shutdown procedure, see the section "Shutting Down the System" in this chapter.

3. Mount the magnetic tape.

4. Perform the backup.

   The *backup* script will write (log) 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.

   If the *backup* command runs out of room on the magnetic tape, it prompts you to change tapes.

   - **If you are doing an archival backup**, do the following steps:

     a. Change to the root directory by typing:

        cd / [Return]

     b. Type:

        /etc/backup -archive [Return]

     c. Check the file system by typing in:

        fsck -p *[filesystem_name]* [Return]

     For more information on *fsck* see "Checking Your File System" in this chapter.

     NOTE

     Do not run *fsck -p* without reading the "Using the fsck Command" (Appendix A).

- **If you are performing an incremental backup**, do the following:
  a. Change to the root directory by typing:

     `cd /` [Return]

  b. Type:

     `/etc/backup` [Return]

- **If you are backing up selected files**, do the following:
  a. Go to the directory containing the files:

     `cd` *directory_name* [Return]

  b. Assuming your magnetic tape drive is associated with the device file */dev/rmt* (if not, replace "rmt" with the correct character special device file name), use the *cpio* command as follows:

     - to back up all files and subdirectories from the current directory, type in:

       `find .-print | cpio -ocBx > /dev/rmt` [Return]

     - to back up all files in your current directory, type in:

       `ls cpio | -ocBx > /dev/rmt` [Return]

     - to back up selected files in your current directory and subdirectories, type in:

       `ls` *file_name dir_name/file_name* `| cpio -ocBx > /dev/rmt` [Return]

  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.

5. When the backup has finished, remove the magnetic tape.

6. Label the magnetic tape with the date and the type of backup (archive or incremental) and store it in a secure place.

7. Examine the information and messages sent to the file, */etc/backuplog* to determine if any errors occurred during the backup process.

Restoring the System

If you need to restore the system due to major problems, and the system is still functioning, log in as the super-user `root` and run the *shutdown* and *fsck* commands as referenced in steps 2 and 3 under the heading "A Standard Backup Procedure" above. 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*.

If you did not need to use *fsck* to repair the system, or after *fsck* has repaired the system, follow the procedures below. You can use these procedures to recover one or more specific files from a backup, also.

Before restoring a backup, you must know how the backup was made. If you used cartridge tape, you used both *cpio* and *tcio*. If you used flexible disc or 9-track, you used only *cpio*.

To restore the file system:

1. Log in as the super-user, `root`, if you need to restore the entire file system. Follow the procedure in "Becoming the Root User" in this chapter. If you will restore only specific files, you must be either the file's owner or root.
2. Write protect the tape or flexible disc your backup is stored on.
3. 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** before continuing with this process. If you wish to restore only a single file from a multiple-cartridge tape backup, insert the correct tape and answer "y" to the resynch prompt.
4. Once the medium is ready, enter one of the following command forms. 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. Select the appropriate form of the command based on the information supplied above.

   For flexible disc or magnetic tape, type in:

   `cpio -iBcdmuvx [patterns]` < *special_file* [Return]

   For cartridge tape, type in:

   `tcio -i` *special_file* | `cpio -icdmuvx [patterns]` [Return]

Note that *special_file* is the name of the character special file associated with the backup device (usually */dev/rfd*, */dev/rmt*, or */dev/rct*). *[patterns]* is an optional parameter used to specify which files to recover. If you wish to recover all the files, do not specify a pattern. If you wish to recover specific files, list them (separated by a blank space) where you see *[patterns]*.

If Disaster Strikes

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 have set up a recovery system after your initial installation (using the procedures described in "Creating a Recovery System") you can boot your system and rebuild your file system from your backups. The recovery system is a functional HP-UX system on cartridge tape or flexible disc.

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 Chapter 2, "Installing HP-UX", and reinstall the system.

NOTE

If you have updated your system since you first installed HP-UX, contact your local Hewlett-Packard Sales Office to obtain a current installation tape or set of flexible discs. Otherwise you will have to do one (or more) updates before you even begin restoring files.

Once you have re-installed the original system and it is operating properly, use the forms of the *cpio* and *tcio* commands given above to copy (restore) the most recent archive and incremental backup(s) from the backup device to the system's root device.

Special Considerations

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.

Dealing with Backup Media

If the end of the medium is reached during the backup process, a message is sent to the standard error file (usually the system console) by the *tcio* and *cpio* commands. For example:

```
tcio: to continue, type new device name when ready, return implies
same device
```

When this occurs, change the backup medium and type:

special_file_name [Return]

where *special_file_name* is the special (device) file name associated with the backup device (for example, if you are using flexible disc, the special (device) file is probably */dev/rfd*). The backup process will then continue.

Performing Backups Automatically

Incremental backups can be performed automatically by using the *crontab* command to schedule *backup* (where it gets executed by the *cron* "clock daemon"); see the *cron(1)* and *crontab(1)* entries in the *HP-UX Reference*.

Perform the following:

1. Login as the super-user, root.
2. Create a file (e.g. x) with this line:

   ```
   55 23 * * 1-5 /etc/backup
   ```

3 Use the *crontab* command with the files name as an argument (e.g. `crontab x`)

   This creates a file: /usr/spool/cron/crontabs/root.

   cron will automatically do a backup at 11:55 p.m.

When running backups with *cron*, there is no way of knowing if the system is inactive. Choose a time when you think no one will be working. Send a message to all users requesting them to log off.

If you run backups from *cron*, 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:
   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:
   a. Be certain that the printer associated with the *backup* script's error output device is on-line.
   b. Install a blank backup medium.

System Restoration and Shared Files

The system cannot write to a file that is marked as shared and being executed with *exec*. (Shared files are discussed in the "Concepts" chapter.) Thus, files being 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.

Becoming the Root User

You need to be logged in with the USER NAME `root` to perform most HP-UX system administrative tasks.

If you have a non-secure system, the only way to become root is to type in:

su root [Return]

If you have a secure system, you can use the above method, or you can log in as root. When you receive the login prompt, enter the word `root`:

login: root [Return]

then supply the password.

Booting HP-UX

Booting HP-UX (bringing up the system) is discussed in detail in Chapter 4 of this manual. This section provides some reminders and a few additional suggestions to supplement that material.

- 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 installation guide supplied with your HP-UX system.
- 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 200/300 computer. If the disc drive has not completed its power-up sequence (which may require several minutes on some disc drives), the boot ROM will not be able to access the disc and load the system.
- The boot ROM follows a specific search for the system console. Descriptions of the system console are in Chapter 4 of this manual.
- The */etc/bcheckrc* script will execute the *fsck* command at bootup whenever the system was incorrectly shutdown (see "Shutting Down the System" in this chapter). 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 certain disaster. Refer to Appendix A ("Using the FSCK Command") in this manual for details on checking the file system.

On single-user systems or small multi-user systems, it may be useful 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 states 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.

Changing a Password

The "Adding/Removing Users" section in this chapter discusses creating passwords for users. This section discusses how either 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` [Return]

The *passwd(1)* command prompts for the existing (old) password before allowing the user to continue. Once the correct old password is entered, the command prompts for a new password. Enter at least 7 characters and/or digits of your choosing followed by at least one numeric and/or special character, and [Return]. Actually, fewer characters may be used but they must be entered three times before the system will accept them (this is only true if you are super-user). 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 a user has forgotten his password, he 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 the user a new password.

To change a user's password, become super-user and type:

`passwd` *user_name* [Return]

where *user_name* is the user's login name. You will be prompted for the user's new password. Only the super user may use this method for changing a password.

To protect the security of the system, */etc/passwd* should be owned by `root` (the super-user) and no one should have write permission to the file. Not even the super-user. If you, as the super-user, want to modify */etc/passwd*, temporarily change the permission using *chmod(1)*, modify */etc/passwd*, then change the permission of */etc/passwd* back. You could also, as the super-user, use the enforced write command of your editor (":w!" in *vi*) to override the file protection. Actually, non-super users should not be allowed to write to **any** of the files contained in the */etc* directory, or to the directory itself.

If you change the password for the user `root`, you may want to write down the password and keep it in a secure place. If this password is lost or forgotten, no one can log in as the super-user. If you ever forget your super-user password, the only way to gain access is to use the recovery system (see "Creating a Recovery System" in this chapter). 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 disc.**

Changing and Creating System Run-levels

One of the tools you need as the system administrator is the ability to move properly from one system run-level to another. Also, you may find it useful to create new system run-level 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 changing the run-level of the system and creating new system run-levels.

As discussed in the "System Boot and Login" 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(1M)* command. Also, anyone having write permission to the file */etc/inittab* can create new run-levels or re-define existing run-levels. Even if this user lacks the capability to invoke *init* to enter a modified run-level, his ability to re-define existing run-levels in */etc/inittab* could wreak havoc upon your system. Make sure that the aforementioned permissions are correct.

If you purchased a single-user system, the system was shipped to you such that run-levels 2 and s use only the system console. Run-level 2 is the mode you will normally want to run your single-user system because */etc/rc* is set up to execute processes in run-level 2 (see "System Boot and Login")

You can create other run-levels, and have other logins, on your single-user system but you cannot have more than one user logged in at a time. Adding *getty* entries to run-level 2 on a single-user system has no effect other than to waste a large amount of processor time. If yours is a single-user system, note that much of the following material applies primarily to multi-user systems.

Changing the System's Run-level

Entering Run-level s

Many of the system maintenance tasks you perform as system administrator require the system to be in single-user mode. In single-user mode (run-level s), the only access to the system is 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*) can be run in run-level s.

When taking the system from any numbered run-level (run-levels 0 thru 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, type in:

> `shutdown` *grace*

This will automatically warn all users that they have *grace* seconds to log off, *syncs* the system, kills all processes, and safely brings 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 super-user to change the system's run-level.

1. If any users are on the system, warn them 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(1)* or *wall(1)* 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 and, in the process, interrupts whatever they are doing (but does not stop execution). Avoid using *wall* unless you feel it is necessary.

   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 as them to log off; their proceses will not be killed.

2. After users are off the system as necessary, force the system to write the contents of its I/O buffers to the file system by typing:

   `sync` [Return]

3. Next, change to the desired run-level by typing:

   `/etc/init` *new_run-level* [Return]

   where *new_run-level* is the number of the run-level you wish to enter. If any outstanding user processes should not be present in the new run-level, you must have the appropriate */etc/inittab* entries to kill them.

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 thru 6).

Creating New System Run-levels

You can create new run-levels if you find it useful. It is acceptable to make certain, suggested modifications to run-level 2 (such as the addition of more *getty* entries to */etc/inittab* previously mentioned). Before creating a new run-level, take the following precautions:

- make a copy of the original */etc/inittab* file (using the *cp(1)* command) and save the original version of the file under a different name (such as `/etc/orig_inittab`). If anything goes wrong, you will still have a relatively untouched version of the file.
- change the `initdefault` entry in your test version to "s". This means when you boot, you will come up in run-level s, and can **change** to run-level 2 (`init 2`). If your new run-level 2 does not work, you will still successfully come up.

NOTE

Before making any changes to */etc/inittab*, save a copy of the */etc/inittab* file. In case you corrupt your file, you will still have a copy of a working version of */etc/inittab*. If you do not have the *initdefault* line as shipped, you may not be able to boot your system. After thoroughly testing your changes, re-insert the initdefault line.

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 two-character id used to identify a process or process group;
- 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(5)* 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. As before, warn all users to log off the system and follow the other procedures in the "Changing the System's Run-level" section above **before you change run-levels**.

In a few cases, such as when a newly-created run-level closely matches an existing run-level (i.e., 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 their *getty*, the process will be killed. Watch for side effects however. Consider the case where a user logs off, you then change run-levels (from say, run-level 2 to run-level 4) and when the user attempts to log back in, he cannot because an */etc/getty* entry does not exist for him in your run-level 4 definition.

Whenever the system enters the newly-defined or created run-level, its actions are similar to those described in the "System Boot" section (of the "System Boot and Login" chapter in this manual) except that the commands executed are those identified by the new run-level number. And of course some files, such as */etc/rc*, have no entries for run-levels other than run-level 2.

Example /etc/inittab

The following is an example */etc/inittab* for a system that contains a system console and six terminals. Run-level s is single-user run-level. 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 preferred to work from his own terminal rather than from the system console.

```
is:2 :initdefault:
bl:  :bootwait:/etc/bcheckrc </dev/syscon >/dev/syscon 2>2&1
bc:  :bootwait:/etc/brc 1>/dev/syscon 2>&1
sl:  :wait:(rm -f /dev/syscon; ln /dev/systty /dev/syscon;) 1>/dev/console
rc:  :wait:/etc/rc <dev/syscon >/dev/syscon 2>&1
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
```

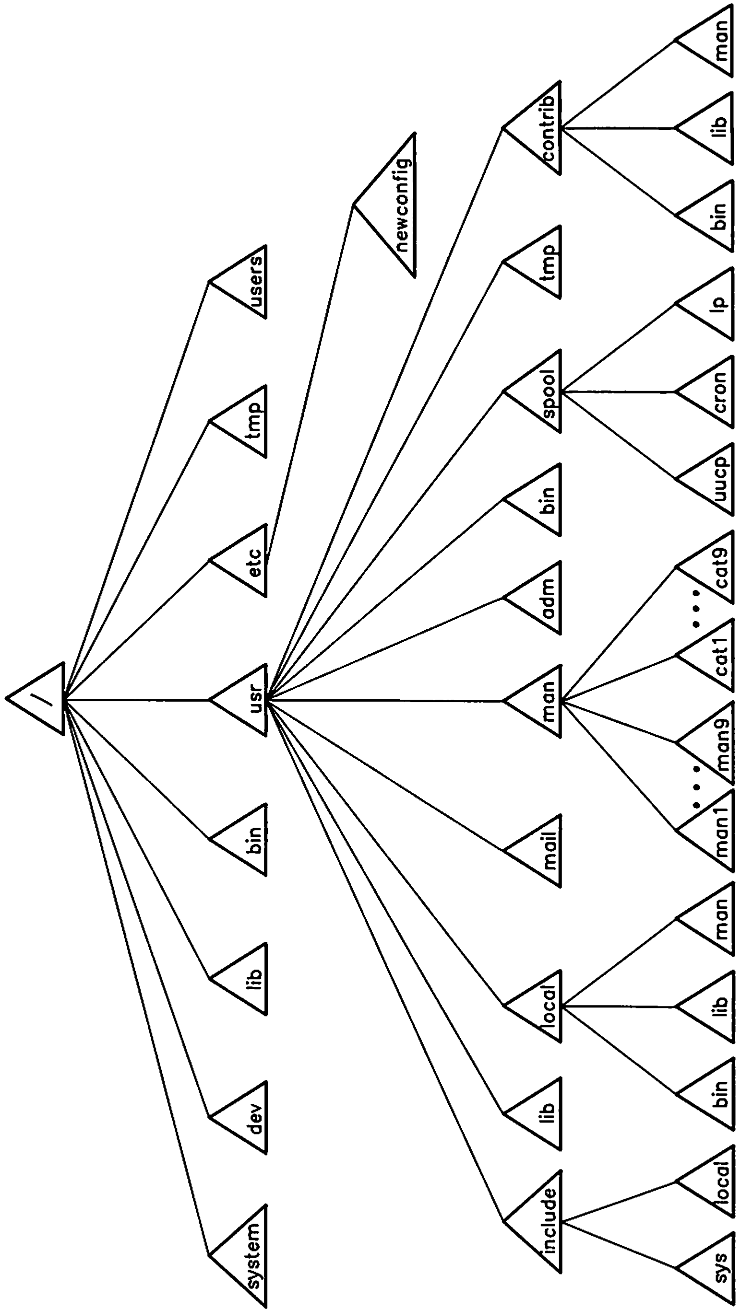
Climbing the HP-UX Tree

As you must know by now, the file system of HP-UX is organized in a tree structure The base of the tree is the root (although one wonders why we don't call it the trunk) of the file system, and the file name / is associated with the root. Under the root (and not above), are eight standard directories created when you installed your system: `bin`, `dev`, `etc`, `lib`, `system`, `tmp`, `users`, and `usr`.

This section describes the basic purpose of the major directories in your HP-UX tree. You will find this useful as you add files and modify your system in the future. As you read the descriptions, reference them to the figure which follows.

- **/bin**—contains frequently used commands, and those required to boot, restore, recover and/or repair the system.
- **/dev**—contains special device files used to communicate to peripherals. For more information, see *mknod(1M)*.
- **/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.
- **/lib**—frequently used object code libraries and related utilities are placed in this directory.
- **/system**—contains object code for drivers; also contains the boot area.
- **/tmp**—a place to put temporary files (those normally with short lifetimes and which may be removed without notice).
- **/users**—user home directories go below this directory.
- **/usr**—less frequently used commands and other miscellaneous files are stored under this directory.
- **/usr/adm**—system administrative data files lay here.
- **/usr/bin**—less frequently used commands and those not required to boot, restore, recover, and/or repair the system go here.
- **/usr/include**—high-level C language header files (shared definitions).
- **/usr/include/sys**—low-level (kernel-related) C language header files.
- **/usr/include/local**—localized C language header files.

- **/usr/lib**—less frequently used object code libraries, related utilities, and miscellaneous data files go here.
- **/usr/local**—localized files should be placed here.
- **/usr/local/bin**—localized commands should go here.
- **/usr/local/lib**—localized object code libraries are placed here.
- **/usr/local/man**—put any on-line manual pages for localized systems in this directory.
- **/usr/mail**—where your mail box resides.
- **/usr/man**—all on-line documentation shipped with your system can be found here.
- **/usr/man/man1 ... man9**—the unformatted version of *man(1)* pages.
- **/usr/man/cat1 ... cat9**—*man(1)* pages already processed to speed access go here.
- **/usr/spool/public**—used for free access of files to other systems via *uucp* or LAN.
- **/usr/spool**—spooled (queued) files for various programs.
- **/usr/spool/uucp**—queued work files, lock files, log files, status files, and other files for *uucp(1)*.
- **/usr/spool/cron**—spooled jobs for *cron(1)* and *at(1)*.
- **/usr/spool/lp**—control and working files for the lp spooler go here.
- **/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.
- **/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.


/
users
tmp
etc
usr
bin
lib
dev
system
newconfig
contrib
tmp
spool
bin
adm
man
mail
local
lib
include
man
lib
bin
lp
cron
uucp
cat9
cat1
man9
man1
man
lib
bin
local
sys

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(1)* command provide a way to get brief announcements to the system users.
- 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.
- Longer messages or even major documents intended for specific users are best sent with the *mail(1)* command.
- To write to users who are already logged in, use the *write(1)* (for specific users) or *wall(1)* (for all users) commands. *write* is intended for user-user dialog, while *wall* is intended for system-wide announcements. Note that if a user has executed the *mesg(1)* command with the **-n** (no) option, write permission to that user's terminal is denied and the *write* command will not work.
- When the *wall* (write all) command is run by the super-user, 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 super-user, 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.

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, or mail.

Configuring HP-UX

Series 200/300 HP-UX allows you to configure, or customize, various attributes of your system. You may require more swap space than the default amount of space. Perhaps you don't want the full kernel taking space in main memory. Or maybe, for a special application, you need to change selected system parameters to make the application more efficient. The *config* command allows you to customize your system to meet these requirements.

NOTE

Do not confuse the command, *config* (discussed here), and the command *reconfig*, discussed in Chapter 2, the section "After Installing HP-UX"

There are 2 ways to configure your system:

- Use the *reconfig* command

  The *reconfig* command only installs drivers - you must use *config* to change swap space or system parameters.

  For most Series 200/300 installations, only kernel drivers will need to be configured. See the "Creating A New Operating System" in the *Application Execution Environment User's Manual* or *reconfig* in the *HP-UX Reference* for details. If you need to configure more swap space or change system parameters, you must use *config*.

- Use the *config* command

  If you are only changing the kernel device drivers, use the *reconfig* command. If you need to change swap space or system parameters, you must use the *config* command. The rest of this section will discuss what you can change using the *config* command. Use this section with the man page for *config(1M)*.

What Can Be Configured

When you configure your system, you create a new kernel. You may also change the way your processes work by changing operating system parameters or process's swap space.

The *config* command lets you configure 4 areas of the kernel:

- device drivers and I/O cards: you must configure all the device drivers and I/O cards you need each time you use *config*—the system has no defaults,
- root and swap devices: optional— the system provides defaults,
- selected system parameters: optional— the system provides defaults,
- System V IPC kernel code: optional— the system provides defaults.

The section, "Parts to Configure", discusses each of the above areas.

config will add kernel parts (drivers) you specify or change parameters that govern how to manage your system. Any drivers not specifically mentioned will be removed.

Using Config

Programs and files used for configuring your system are:

- */etc/config*,
- */etc/master*,
- */etc/conf/*libraries— object code,
- */etc/conf/h*— .h files,
- */etc/conf/machine* — .h files,
- */etc/conf/dfile.[min,full,full.lan,support]*— sample configuration files.
- */etc/conf/dfile*—the configuration file created with *reconfig*. This file reflects the drivers currently installed (unless you have created a new operating system using *config*, and used a different *dfile*.

NOTE

You must not use *config* in the root directory (/) or you will overwrite your kernel.

To use *config*, follow these steps:

1. Log in as the super-user, root.
2. Shutdown the system. This will bring your system into single-user mode, and will make your system inactive. Follow the procedure in "Shutting Down the System" in this chapter.
3. Execute *fsck* (see Appendix A for details).
4. *cd* to some directory other than / (for example, */etc/config*). 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.**
5. Create a dfile (description file). The dfile is a user-provided description of the system configuration. It contains all device drivers you need in your kernel, plus swap space configuration (if not the default values) and system parameter configuration (if not the default values).

   You can either create your own dfile, or use one of the sample dfiles sent with your system. The sample dfiles contain only device drivers and are described in the "Device Drivers and I/O Cards" section. See the samples in */etc/conf*: *dfile.full, dfile.full.lan, dfile.min, dfile.support*.
6. Execute *config* on your dfile to create both a C program and a makefile. See *config(1M)* for available options.

   ```
   /etc/config dfile
   ```

   Executing *config* creates *conf.c* and *config.mk*. *conf.c* is a "c" program that contains external references to the kernel driver code. Satisfying these external references will bring code from the system libraries into the kernel. *config.mk* is a makefile.

   Check to see that you have these files by typing `ls`. You should see your dfile, *conf.c*, and *config.mk*.

7. Create the new hp-ux kernel in the current directory by executing:

   `make -f config.mk` [Return]

   config.mk will list all code brought into the kernel:

   ```
     /etc/conf/libmin.a
        cs80.o
        mux.o
        muxs.o
        :
   /etc/conf/libdevelop.a
        amigo.o
        ciper.o
        :
   ```

   Executing *conf.mk* creates a file, *hp-ux*, in your current directory.

8. Backup the existing kernel by typing in:

   `cp /hp-ux /SYSBCKUP` [Return]

   If the newly configured kernel won't boot, use the bootrom's *attended mode* to access the backup kernel. To use the attended mode, press the space bar during bootup. This halts the automatic boot mechanism and allows you to choose the system to load. You should select "B" (backup) option instead of the "H" (HP-UX) option. For example, select "1B" instead of "1H".

9. Move the new kernel to the / directory.

   `cp ./hp-ux /hp-ux` [Return]

10. Reboot

    `exec reboot` [Return]

11. If you have added a second swap disc, enable the second disc by executing:

    `swapon` *second_swapdevice* [Return]

    Add this entry to */etc/rc* for automatically enabling the second disc when you boot.

A useful option for *config* is the *-a* option:

-a If no file is specified, this creates a mkdev script (see section "Adding/Moving Peripheral Devices" in this chapter for a description of *mkdev*)

-a dfile If a file is specified, it must be a dfile. Providing a file name indicates to *config* that you are supplying addresses for devices. *config* will produce both a mkdev script and a list of mknod commands for each device. See *mknod* in the *HP-UX Reference* for details

Parts to Configure

Device Drivers and I/O Cards

You MUST configure all the kernel drivers each time you execute *config*. If you do not include all the hardware drivers you need, you will be unable to communicate with your peripherals associated with those drivers. Peripherals are the hardware associated with the files in */dev*.

This area of customization enables you to configure hardware drivers, cards in the backplane, and software drivers (pseudo drivers). Pseudo drivers don't talk to any hardware device. LAN and Windows/9000 drivers are examples of pseudo drivers.

Four sample script files came with your system to show how different device drivers and I/O cards can be combined to form different kernels. The sample files can be found in */etc/conf*. The files are:

- *dfile.full*—this file will configure an HP-UX system with all the drivers supported in this release, minus LAN,
- *dfile.full.lan*—*dfile.full* plus the LAN driver.
- *dfile.support*—this file reflects the set of drivers and I/O cards available in releases prior to 5.0.
- *dfile.min*—this file contains one possible minimal (but still useful) system configuration.

In addition to the sample files, the dfile you created, or the dfile created by *reconfig* if you used *reconfig*, will be in this directory.

In determining the device drivers for the hardware you have, consult the "alias" portion of the */etc/master* file. If you wish to determine the device drivers from the */etc/master* file using alias, do the following:

1. make a list of the model numbers of all your peripheral devices, I/O cards, and software (LAN, Windows/9000, etc.).
2. look up the part number in the */etc/master* file and add the corresponding driver name to your dfile. For example, if you have a 7945 disc, look up 7945 in */etc/master*. You will see the following entry:

   ```
   7945        cs80
   ```

   You would add "cs80" to your dfile.

After you have the drivers you need (you have created a new kernel with *config*), you must create device files. 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. See the section "Adding/Moving Peripheral Devices" for information on creating device files.

Swap Space and Root Device

Swap space is an area on your disc (or discs) reserved for the virtual memory system (see "Memory Management" in the "Concepts" chapter). The root device is the disc where you have your root directory.

The kernel initialization procedures, executed immediately after your operating system is loaded, will use the following rules to determine specifications for rootdev and swapdev:

1. If either the *rootdev* or *swapdev* is already specified by conf.c, then use values from conf.c.
2. If *rootdev* has not been specified in conf.c, the root device will default to the boot device.
3. If *swapdev* has not been specified in conf.c, the swapping device will be assigned to the same device as *rootdev*.

Determining the Amount of Swap Space Needed

Swap space holds a process image of all the processes that are running. If you do not have enough swap space, HP-UX will not execute your program. HP-UX will either return an error (for example ENOMEM) or kill your 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 7 possible messages, describing when and why the process was killed.

There are 3 ways to choose the amount of swap space. They are listed in order of increasing difficulty:

1. Use the system default.

   When you created your file system, *newfs* created a default amount of swap space. The default swap space is probably large enough for your installation.

2. Estimate the amount of swap space you need by adding the space required by your largest application to the default amount of swap space.

   Determine the swap space required by your application from either the applications documentation or by using some of the methods from option 3. Determine the default amount of swap space either from the messages on your display at boot up (512 byte units), from the */etc/disktab* file (in bytes), or from the "Disc Parameters Menu" during installation (in Kbytes).

3. Calculate your swap space requirements using the following formula and the worksheet in Figure 5-2.

   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 the formula only if options one and two won't work. Make sure you count each process (using the process ID) rather than each program name.

   Swap space computation:

   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. The *file(1)* command tells if a file contains shared text ("pure" means shared).

   B = SUM (all data sizes) for all processes. If the process does not used shared code, then add its code size (see "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(2)* or *malloc(3)*.

   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.

   D = SUM (sticky code sizes) for all sticky code files 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 an "t" in their permissions.

E = SUM (all existing shared segments' sizes) for all shared segments created by users via *shmget*. HPWindows/9000 uses about 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. 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. The first unit allocated is *dmmin* 1K blocks. Each successive unit is twice as large as the previous one, until *dmmax* is reached. From this point, all units are *dmmax* 1K 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.

Swap space = A + B + C + D + E + F + G.

Executing *size(1)* on the executable file will give you a starting point for the amount of space it requires. 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. Again, there is no simple way to determine he dynamic data size unless you know your program's run-time logic.

A

For shared code, fill out the code space needed by the process. Count shared code only once.

Process ID	Code size
____________	____________
____________	____________
____________	____________
____________	____________
____________	____________

B

For each shared process listed above, 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 (minimim 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
____________	____________
____________	____________
____________	____________
____________	____________
____________	____________

Figure 5-2

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 segment give the shared segment size. Note that if you are running HP Windows/9000 you must add at least 4½ 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 is:

A + B + D + C + D + F + G = Total swap space

___ + ___ + ___ + ___ + ___ + ___ + ___ = ____________

Figure 5-2 (cont'd)

Creating Swap Space

You can specify swap devices (the secondary storage devices holding swap space) by putting a line in your *dfile* that looks like:

```
swap devname address swplo [nswap]
```

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 preceeding 0x.

See "Adding/Moving Peripheral Devices" in this chapter for a discussion on minor device numbers.

swplo Swap area's location in decimal.

If (swplo = -1), then the swap space is put on the disc, after the file system.
If (swplo = 0), then allocate the entire disc to the swap space— don't reserve any area before the swap area.
If (swplo >= 1), then reserve *swplo* space at the beginning of the disc, leaving (disc-size - swplo) for the swap area.

nswap Size of swap area in number of 1 Kbyte units.

Swap space can be on the same disc as your file system, on a separate disc, or both (see figure 5.3). If you need more than the default swap space, use the swap configuration in either 5.3.c or 5.3.d. An explanation of each configuration follows.

NOTE

If you rebuild your root file system, you must have another bootable system. This can be either another root disc, or a recovery system. This procedure is described later in this section.

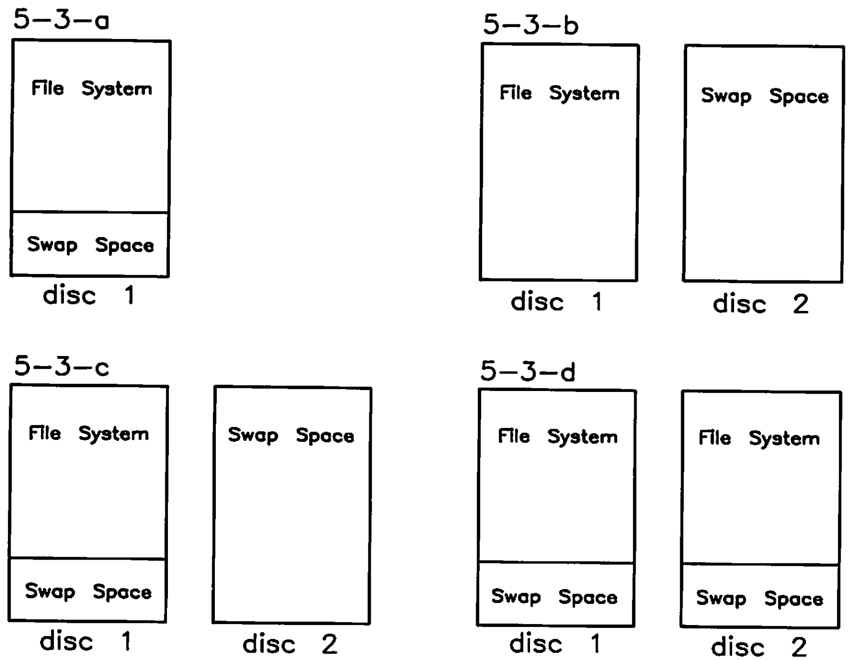


Figure 5-3: Swap Space

Possible configurations for swap space are:

- Figure 5-3-a shows the default configuration.

  Use *newfs* to create your file system (see "Creating File Systems" in this chapter)

- Figure 5-3-b shows one disc (probably the root disc) with no default swap area after the file system, and a second disc dedicated to swapping. You must remake your file system, using the entire disc for the file system, then put the following line in your dfile[1]:

```
swap cs80 E0100 0 /* disc 2 */
```

- Figure 5-3-c shows one disc (probably the root disc) with a the default swap space and file system size, and a second disc dedicated to swapping.

  The entry in your dfile for Figure 5-3-c would look like[1]:

```
swap cs80 E0000 -1 /* disc 1 */
swap cs80 E0100  0 /* disc 2 */
```

- Figure 5-3-d shows two file system discs, each with the default swap space.

  On each disc, a file system was created using *newfs*. The entries in your dfile would look like[1]:

  ```
  swap cs80 E0000 -1 /* disc 1 */
  swap cs80 E0100 -1 /* disc 2 */
  ```

- Remake your file system, decreasing the file system size.

  Decreasing your file system's size increases your swap size. This option would look similar to Figure 5-3-a, 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 = disc size - pa×DEV_BSIZE. *pa* is the size of partition a in blocks (only one partition per disc on Version 5.1 of HP-UX). DEV_BSIZE is defined in */usr/include/sys/param.h*.

  To change the swap space, change *pa* and *nc* in your */etc/disktab* file so that: pa = 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
  ```

  Again, change only pa and nc.

NOTE

If you increase the size of the swap space on your file system disc without re-making the file system, you may loose parts of your file system and the system will probably crash.

How to Remake Your File System, decreasing the file system size.

Decreasing your file system's size increases your swap size. This option would look similar to Figure 5-3-a, but the swap space would be larger—less file system space.

[1] All the examples assume disc 1 is at select code 14, bus address 0, and disc 2 is at select code 14, bus address 1.

To rebuild your filesystem, you can either use another root disc or re-install your system. To rebuild your filesystem using another root disc, 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.

```
                           disc size - swap size in bytes
   file_system_size = ________________________________
                                    DEV_BSIZE
```

   DEV_BSIZE is the size of blocks and is defined in */usr/include/sys/param.h*.

   For example, if DEV_BSIZE = 1024 bytes, and you need 20 971 520 bytes of swap space on your HP 7945 disc, you would need 36 352 blocks for your file system:

```
                             58 195 968 - 20 971 520
   file_system_size = ____________________________ = 36 352 blocks
                                      1024
```

3. Create another root file system on hard disc using the *file_system_size* you determined in step 2.

   In the example in step 2, you would make your file system using the command line:

```
   newfs -s 36352 /dev/rhd2 hp7945
```

   See "Creating A New File System" in this chapter for specific details on the file system creation procedure.

4. Copy your existing file system onto the new root disc.

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. Backup the entire file system using the procedures in this chapter, the section "Backing Up and Restoring HP-UX".

3. Re-install HP-UX, this time specifying the larger swap size in the "Disc Parameters Menu". See Chapter 2 for instructions on how to install HP-UX.

4. Restore your files from the backup created in step 1.

Enabling Swap Space

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(1)* command after you boot the new kernel. For examples in Figures 5-3-c and 5-3-d, if the second disc corresponded to the device file */dev/hd2*, and had a minor number of 0x0e0100, the second device could be enabled by inserting "`swapon /dev/hd2`" in */etc/rc*. Once you have enabled a device for swapping, it can not be disabled without halting the system.

Creating Raw Disc Space

Sometimes you need raw disc space for development. The default disc arrangement does not leave any raw (unused) disc space. To create raw disc space, use the following steps:

1. Create your file system, leaving enough space for both your swap space and raw disc space (using the procedures shown in "creating Swap Space").
2. Put an entry into your dfile to specify the amount of swap space.
3. Execute *config*.

When you first boot the system, it displays both the swap start and swap size in 512 byte units. *config* requires the swap size in 1024 byte units.

For example, if your system originally displayed the following:

```
swap start = 88032 swap size=20384 (in 512 byte units)
```

This translates to:

```
swap start = 44016 swap size=10192 (in 1K byte units)
```

You can partition your original swap space into three separate spaces as follows:

```
raw read/write area:        start=44016       size=512
cushion area:               start=44528       size=5
smaller swap area:     swap start=44533  swap size=9675
```

To accomplish this, put the following line in your dfile:

```
swap cs80 0e0000 -1 -9675
```

The above procedure leaves unused disc space between the file system and the swap space starting at 44016.

System Parameters
HP-UX allows you to configure certain operating system related parameters (attributes). These attributes determine how your system manages memory, limits table sizes, and determines other system limits.

NOTE

HP-UX system parameters should not be modified unless you *fully* understands the ramifications of doing so.

There are two types of parameters: system parameters and System V IPC code capability. System V IPC code capability is described in the next section. Each parameter is described in detail in "Appendix D: System Parameters". Read the entry for the system parameter *thoroughly* before you attempt to change the parameter.

The format for specifying system parameters in the *dfile* is:
parameter (decimal-number or *formula)*
where *parameter* is one of the parameters described in "Appendix D".

The attributes that you can configure are:

- **accounting code parameters**—used by system accounting.
  Parameters: *timeslice, acctsuspend, acctresume.*
- **time information**—used by the system to determine the time from Greenwich Mean Time and differences due to daylight savings time.
  Parameters: *dst, timezone.*
- **parity errors**—selects action for a parity error.
  Parameter: *parity_option.*
- **limiter for system resource allocation**—used to calculate values of other global system parameters.
  Parameter: *maxusers.*
- **file system parameters**—number of open files, number of open inodes in the system, number of file system buffer-cache buffer headers, number of file locks, and maximum number of shared text descriptors.
  Parameters: *nfile, ninode, nbuf, nflocks, ntext.*

- **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. See the description below.
  Parameters: *maxdsiz, maxssiz, maxtsiz.*
- **memory parameters**—memory guaranteed to be available for virtual memory and/or system overhead at any time and size of area used by *argdev.*
  Parameter: *unlockable_mem, argdevnblk.*
- **pseudo-teletypes**—maximum number of pseudo-teletypes.
  Parameter: *npty.*

User Process Size Limits

Each of the system parameters is described in "Appendix D: System Parameters". The 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 3-9 (Chapter 3) shows how the parameters, *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 Series 200 or Series 300, Model 310, to 16 Mbytes, and limited on a Model 320 to 4 Gbytes, regardless of the size of these parameters.

If you change these parameters, make sure they can still work together. See Chapter 3, the subsection "Logical Address Space Management", the "Memory Management" section 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 system parameters, described above.

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*.

Controlling Disc Use

As system administrator, you should keep track of the amount of disc space available to users and the distribution of free disc space across file systems. The following procedure will help you evaluate your disc use and identify future disc needs.

1. The *du(1)* command should be executed regularly (weekly or bi-weekly) and the output kept in an accessible file for later comparison. This method lets you spot users who are rapidly increasing their disc usage.

2. Use the *find(1)* 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
```

3. Use the *df(1)* command to list the amount of free disc space on a volume.

   The *df* command returns the number of free blocks left on the file system available for the ordinary user (not super-user); it does not count the number of blocks reserved by *minfree*. If you wish to see a more detailed report of disc usage, type:

```
df -t
```

   An example of output from `df -t` is:

```
/       (/dev/hd        ):     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 93 918 blocks (512 byte blocks) available to the ordinary user, and 165 510 512-byte blocks (715 904 - 550 394 = 165 510) available to the super-user.

   If you have reached the *minfree* threshold (discussed in "Creating File Systems" in this chapter), only the super-user can continue to allocate blocks.

Some files, if present, are written to automatically by certain HP-UX commands (to monitor system use and for general house-keeping). Some files are created automatically by the commands that require them. Both of these sorts of files are called logging files. If not periodically checked and cleared, these files simply continue to grow.

Here are some typical logging files:

- */etc/utmp* (binary) - current login status.
- */etc/wtmp* (binary) - history of logins, logouts and date changes.
- */etc/btmp* (binary) - history of failed login attempts. This file will not be created by your system; it will be used only if it exists.
- */etc/mnttab* (binary) - mounted devices (not the official list kept by the system in the kernel).
- */usr/adm/sulog* (ASCII) - history of use of the *su* command.
- */usr/lib/cron/log* (ASCII) - history of actions by *cron*.

You should clean these files on a daily or weekly basis. The only way to avoid logging is to link log files to */dev/null* (not recommended).

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 create another file system. The following are the steps you should follow to create a file system.

1. Connect the mass storage device on which the file system will exist to your HP-UX system. See the "Adding/Moving Peripheral Devices" section earlier in this chapter and the installation manuals supplied with your Series 200 or Series 300 computer and/or the mass storage device for hardware installation details.

2. Create both a character special and a block special device file for the mass storage medium. Use the instructions in the "Adding/Moving Peripheral Devices" section of this chapter. You need both a character special and a block special device file.

3. Once the mass storage device is properly installed and turned on, run the media initialization utility *mediainit(1)*. This utility initializes the medium on which the file system will reside. If the medium has been initialized before, you may skip this step. Read "Initializing the Media" in this chapter, or read the *mediainit(1)* entry in the the *HP-UX Reference*.

4. Determine if you have any special system requirements.

   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. See the section "Configuring HP-UX" in this chapter for a description of swap space, and the "Customizing your File System" section for guidelines on special requirements.

5. Use *mkfs* or *newfs* to create the file system on the initialized disc. You must be the super-user `root` to create a file system.

   An HFS file system can be created using either *newfs* or *mkfs*. *newfs* is a friendly front end to *mkfs*, and is the recommended command to use.

   If you wish to create a file system using only default values, type in:

   `newfs` *device_file disc-type* [Return]

   where *device_file* is the special device file associated with the disc drive where the file system will reside, and *disc-type* is the name for the disc in the text file */etc/disktab*. See *newfs(1M)* for a full list of parameters.

   newfs uses defaults from both *mkfs* and from the file, */etc/disktab*. The *mkfs* defaults are listed in Table 5-3. The *disktab* file contains disc-specific information for *newfs*. See *disktab(5)* in the *HP-UX Reference* or look in the file, */etc/disktab*.

If you determined in step 5 that you need to create a non-default file system, follow the syntax for *newfs* or *mkfs* in the *HP-UX Reference*. An example of creating a file system specifying the swap space is shown in the section "Configuring HP-UX" in this chapter.

mkfs will not put the boot program on the disc for you. You must explicitly put the boot program on disc by using *dd*.

NOTE

The *mkfs* and *newfs* programs will list the alternate superblock locations. You should write down the alternate superblock locations. If your primary superblock becomes corrupted, you will need to use an alternate superblock with *fsck* to repair your system.

---

6. Next, mount the new file system. This is described in the section "Mounting and Unmounting Volumes". The basic steps are:

```
# create a directory on which to mount the new file system
  mkdir /mount_directory

# mount the new file system onto '/mount_directory'
  mount /dev/fname /mount_directory
```

   where */dev/fname* is the name of the block special file associated with the mass storage medium.

7. Add the new file system to */etc/checklist* by inserting the name of the character special device file name associated with the file system.

   If the newly created file system is intended as a permanent addition, you may wish to modify */etc/checklist* so the new file system will be checked and mounted when the system is booted.

Table 5-3: mkfs Defaults

parameter	range	default	comments
size	not applicable	none	In DEV_BSIZE[1] blocks. If using *newfs* size can be taken from */etc/disktab*. Size is the total disc size minus the swap size in DEV_BSIZE[1] blocks.
block-size	4K or 8K	MAXBSIZE (8K)	Specified in bytes
frag-size	DEV_BSIZE[1] to block-size	1024	Specified in bytes. Must be a multiple of DEV_BSIZE[1]
#tracks/cylinder	greater than 0	16	Taken from */etc/disktab* if using *newfs*
#cylinders/group	1 to 32	16	
% free space	0 to 100	10 %	Once this threshold has been crossed only the super-user can continue to write.
revolutions/seconds	not applicable	60	If using *newfs* the parameter is revolutions/minute and the value can be taken from */etc/disktab*.
number of bytes per inode	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.

[1] DEV_BSIZE can be found in */usr/include/sys/param.h*

Customizing Your File System

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 disc.

- Decreasing the blocksize to 4 Kbyte blocks may be a more efficient use of file system space if you have small files.

- Decreasing the file system size will give you more swap size. More swap space enables you to run larger programs, but decreases the area where you can store files.
- 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.

Creating Groups/Changing Group Membership

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

The *group_name* field contains the name of the group.

The *password* field is used by *newgrp*. Placing an asterisk (*) in the *password* field prevents non-group members from switching to this group.

The *group_id* field is the unique integer ID shared by all group members.

The *member1, member2, ..., memberN* 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 disc and terminate the editing session.

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*.

To change to a different group if you only have */etc/group*, use the *newgrp(1)* command, to change your effective group ID to that of another group. At login, each user is placed in the group specified by his group ID entry in */etc/passwd*.

Creating a Recovery System

Once your system has been installed, the first thing you should do is make a recovery system. If you can't boot from your root disc, you can use your recovery system to boot and repair your file system. You may be unable to boot because your root disc is too corrupt, or because you forgot your root password.

A recovery system is built using a shell script, */etc/mkrs*. *mkrs* will build a small HP-UX system with a minimal kernel so you can boot.

The recovery system is easy to build, and is valuable if you ever need it. You can build a recovery system in multi-user mode, you don't need to have your users log off. You can build your recovery system on either cartridge tape or on two 3½ inch double-sided, double-density micro discs. The cartridge tape can hold more files, but is *much* slower for both creating and using the recovery system.

NOTE

If you change the swap space of your system, you MUST create a new recovery system. If you fail to create a new recovery system, the old one may overwrite and destroy your root file 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.

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	Directory containing a small subset of HP-UX commands. The actual commands varies 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 special files for the root disc and the recovery drive). The device files are created automatically.
/disc	A directory that can be used to mount a file system.
/etc	*/etc* contains the tools necessary to fix your root file system: *sbtab, fsck, mknod, mount*, and *umount*. It also contains small *inittab, init, profile*, and *rc* files, which are necessary for booting.
/tmp	

To create the recovery system:

1. Log in as the super-user, root.

   You will be accessing privileged commands, so you must have super-user privileges.

2. Create the device files in */dev* for the device you wish to create the recovery system on.

   See "Adding/Moving Peripherals" for procedures on how to create device files. You must have both character and block mode device files for the root disc and for the tape or flexible disc drive used for a recovery system.

3. Create the recovery system by using the *mkrs* command. If *mkrs* doesn't exist on your system (you receive a message `execute permission denied`), follow the guidelines in "Updating HP-UX" to install the "ACONFIG" file set. *update* will automatically load all file sets dependent on ACONFIG.

   mkrs has the following form:

   ```
   mkrs [-f rcdev] [-r rootdev] [-t type] [-v]
   ```

   rcdev is the name of the **block mode device file** for the device on which you are creating your recovery system. *rcdev* must be the same block special device file name you created in step 2. If your device files are named *ct* and *rct*, you don't need to specify this option; *ct* is the default. Do not use the full path name (e.g. use -f mt not -f /dev/mt).

rootdev is assumed to be *hd* by default. If your root device files are not named *hd* and *rhd*, you must use the *-r* option. Do not use the full path name (e.g. use -r hdl not -r /dev/hdl).

type can be either *ct* (cartridge tape) or *md* (micro disc). The default is ct.

For example, if your root file system is associated with the device file, */dev/hd*, and you will be creating your recovery system on the flexible disc drive associated with the device file */dev/md*, you would type in:

```
mkrs -f md -t md [Return]
```

If you are creating your recovery system on micro discs, you will need two discs. You will be prompted to insert the second disc when *mkrs* is finished with the first disc.

The *mkrs* process takes about 30 minutes on a micro disc and about 1 hour on a cartridge tape.

5. Boot the recovery system to check it out. For this step, you will need to shut down the system. You probably want to test-boot the recovery system during off hours. Follow the steps 1 - 3 under "Booting the Recovery System".

6. Put the recovery system in a safe place and LOCK IT!

   When you boot using the recovery system, you come up as root. 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).

Booting the Recovery System

To use your recovery system, you will need to boot, then use the recovery tools to gain access to your root disc. Follow these steps to boot your system.

1. **Power up all system swap devices** (generally just your root file system's hard disc) and the recovery system device. Your recovery system uses the swap space on your system, so the these devices must be powered up.

2. Turn on your computer and (immediately) hold down the space bar to 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 9122 flexible disc drive, and your regular system on a 9133, you might see (on the right side of the display):

```
9133; 1400, 0, 0
   1H SYSHPUX
   1B SYSBCKUP
9122; 0700, 0, 0
   2H SYSHPUX
```

   You would select [2][H].

   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 only 2 or 3 minutes from the micro disc, and about 10 minutes from the cartridge tape.

   When it's done, you will see the words:

```
Welcome to the HPUX Recovery System
```

You are now in the recovery system.

Using the Recovery System

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:

- assumes you cannot boot your regular system; you suspected a problem and used the recovery system to boot,
- uses the term "root device" for the device that is root under normal circumstances (disc drive associated with your root file system),
- assumes your root device is called */dev/hd* (block special file) and */dev/rhd* (character special file),
- assumes your recovery device is called */dev/fd* (block special file) and */dev/rfd* (character special file).

NOTE

Use "ls -l /dev" to determine what device files are actually present on the recovery system.

---

1. If you think your root device is corrupted, then run *fsck* on the **character** special file associated with your root device (e.g. */dev/rhd*). The file */etc/sbtab* on the recovery system should identify the location of duplicate superblock (it will be an empty file if */etc/sbtab* did not exist on your root file system when the recovery system was made). For example, if one of the locations is block 16, then use `fsck -b 16 /dev/rhd`.

   If you think the boot area on your root device has been corrupted, then copy the boot area from the recovery system to the root device as follows:

   ```
   dd if=/dev/rfd of=/dev/rhd count=1 bs=8k
   ```

   If the fsck appears to solve the problem then do a *sync*, wait for the busy light on the tape or floppy drive to remain off, halt the system, and reboot from your hard disc. If the reboot fails then come up on the recovery system again and go on with step 2. If the boot succeeds but other problems still exist on the disc (such as missing or corrupted files) then go to step 3.

2. Once the integrity of the file system is ensured, mount the root device using:

```
mount /dev/hd /disc
```

The strategy at this point is to at least get the root volume back to the point where the system can be booted from it. The critical files are listed below. If any of these files are suspect on the root volume then take the indicated actions to fix the file.

/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
rm /disc/bin/rsh
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 go edit it later), then create a single line *inittab*. Do this as follows:

```
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).

etc/ioctl.syscon — If you have changed the device used as the console then 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):

```
rm /disc/etc/ioctl.syscon
```

/dev/console — This file (which is also linked to */dev/syscon* and */dev/systty*) could be corrupted or the file might not match the console, resulting in the system not being bootable. Use the following to compare these files on the recovery system and the root device.

```
ls -l /dev /disc/dev
```

If the files on the root volume do not match those on the recovery system, then correct the problem by doing the following:

```
# remove the 3 files, console, syscon, and systty
rm /disc/dev/console /disc/dev/syscon /disc/dev/systty

# use the appropriate parameters to make /dev/console match
# the /dev/console file on the recovery media (using mknod).
# Note that you cannot copy (using cp) device files

# 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 possibilities:

a. If there is a backup kernel file on the root volume that is not corrupted then use it to reboot from the hard disc. Once you have rebooted you can either create a new kernel (using *config*) or copy the backup kernel to */hp-ux*. A common name for a backup kernel is */SYSBACKUP*.

b. If there is no usable backup kernel then you need to copy */hp-ux.min* from the recovery system to the root device. If your recovery system is on cartridge tape, */hp-ux.min* will be on your one recovery tape. If your recovery system is on flexible disc, */hp-ux.min* will be on your second recovery disc. To use the second recovery disc to boot from, do the following:

   - *sync* and wait for the busy light on the floppy drive to remain off.
   - Turn off your computer, remove flexible disc #1, insert flexible disc #2, power on your computer, hold the space bar down, and once given the choice, choose to boot from the flexible disc.
   - The kernel on flexible disc #2 was built so that it uses the root volume as root (and not the flexible disc).
   - Go to step 4.

3. You should now be able to boot from the root volume. Do a *sync* (while still running on the recovery media), wait for the busy light on the tape or flexible disc drive to remain off, halt the system and reboot.

4. If you have rebooted from the root volume, you can now fix other possible problems as described below.

   - 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. Until you have tested your newly edited *inittab*, you should make the default state s, i.e., like the initdefault line shown above, which causes the system to come up in single user state when booted. You can switch to other states using *init x*, where x=1,2,....

   - if you lost other system files you should *update* your system. *update* requires that at least the following commands be available. The flexible disc containing the command is also listed. All of these commands can be gotten from the #1 tape.

```
/etc/mkfs               SYS_CORE (gray) #1
/usr/bin/mediainit      SYS_CORE (gray) #1 (found as /mediainit)
/bin/sh                 SYS_CORE (gray) #2
/bin/mkdir              SYS_CORE (gray) #2
/bin/pwd                SYS_CORE (gray) #2
/bin/cpio               SYS_CORE (gray) #2
/usr/bin/lifcp          SYS_CORE (gray) #2
/usr/bin/tcio           SYS_CORE (gray) #2
/etc/update             SYS_CORE (gray) #2
/etc/sysrm              SYS_CORE (gray) #2
```

   If any of these are missing, they can be gotten from your install tape/flexible disc by mounting the tape/flexible disc and copying the missing command(s) to the root device.

Initializing Media

A command, *mediainit*, must be used on your flexible discs, cartridge tape, and hard discs before you can put files on them. *mediainit* prepares the disc or tape for error-free operation.

NOTE

Do not initialize a mounted file system.

---

1. Install (set up and connect) the disc or tape drive if you have not already done so. Follow the guidelines in the *Peripheral Installation Guide*.
2. Create a character device file for the tape or disc drive if you have not already done so. Follow the guidelines in the section "Adding/Moving Peripherals" in this chapter.
3. Initialize the media by typing in the *mediainit(1)* command using the character special device file's *pathname* (e.g. */dev/rfd* rather than */dev/fd* for flexible disc).

   `mediainit` *[options] pathname* [Return]

   The choice of the interleave factor can have a substantial impact on disc performance (both hard disc and flexible disc). The interleave factor should be 2 for flexible disc. For hard disc, use the interleave factor recommended in the *Configuration Reference Manual*, listed under the HP part number of your disc drive.

NOTE

If you add new hard disc to your system, you may need to reinitialize the new hard disc before you can use it. If you change your system configuration (such as moving the system disc from the built-in HP-IB to the HP 98625A interface, or adding DMA) it **may** be necessary to reinitialize the disc. Check with your local HP sales office.

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*), type in:

  `mediainit /dev/rct` Return

- If you are initializing a flexible disc (associated with the character special device file */dev/rfd*), you must specify an interleave factor of 2. Type in:

  `mediainit -i 2 /dev/rfd` Return

Remember, it is usually necessary to initialize media only once.

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 disc. It is possible to have other file systems on **different** discs; any other mass storage device supported by Series 200/300 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.

Once a block special (device) file exists for the new disc device, and a file system has been created on the disc, the file system the device contains can be mounted. See the "Adding/Removing Peripheral Devices" section in this chapter for information on creating special (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 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 filesystem to */etc/checklist*, */etc/bcheckrc* will mount and check the file system at bootup. */etc/bcheckrc* uses the clean byte to determine if the system was properly shut down. If the clean byte is FS_OK, then it was not unmounted with the *umount* command, and could be corrupted. *bcheckrc* will run *fsck* if the clean byte is FS_OK to correct corruption before continuing. */etc/rc* mounts each file system in */etc/checklist* after *bcheckrc* has checked them.

To Mount a File System

You must create a special (device) file (if one does not yet exist) for the mass storage device containing the file system which is to be mounted. The special file must be a block special file; see the "Adding/Removing Peripheral Devices" section in this chapter for details.

Before attempting to mount a file system, be certain the mass storage device associated with the file system is powered up and is on-line. If the file system is on removable medium (such as a flexible disc), insert the media in the mass storage device at this time. Do not remove the flexible disc until it is unmounted.

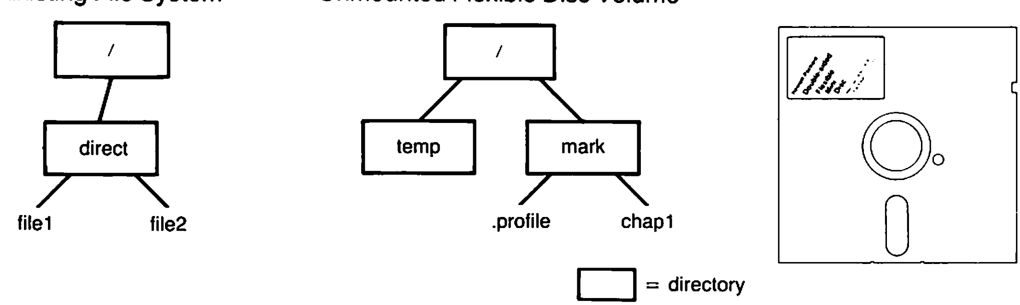
Decide what directory in the HP-UX file system will be used to mount the file system. It is best if you choose a directory that is at the root of the HP-UX file system. That is to say, its complete pathname should consist of only the / character followed by the name of the directory.

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.

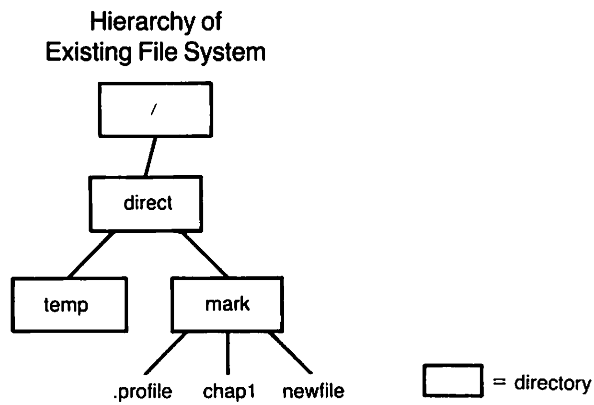
The diagrams that follow, for the sake of illustration, show a file system mounted on a directory that does contain files but this is **not** standard practice. Consider a directory called */direct* that contains two files, `file1` and `file2`. Assume that you want to mount a flexible disc (that contains a hierarchical file system of its own as shown in the following illustration) 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.

Hierarchy of Existing File System

File System Hierarchy of Unmounted Flexible Disc Volume



The preceding diagram shows the */direct* directory on the existing mounted file system. It also shows the file system hierarchy on the as-yet-unmounted flexible disc.

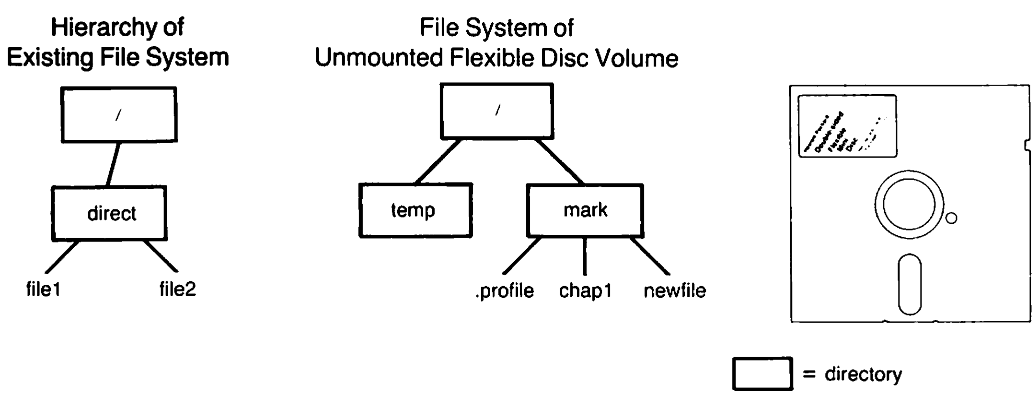


The preceding diagram shows the file system hierarchy once the file system is mounted on the */direct* directory. The file `newfile` has been added to the new file system after it was mounted. Notice that the files `file1` and `file2` previously available in the */direct* directory are no longer accessible; the files are still there, they just cannot be accessed until the file system is unmounted.

The file system was mounted with a command of the form:

/etc/mount */dev/fd_name* **/direct** [Return]

where */dev/fd_name* is the block special (device) file associated with the mass storage device containing the flexible disc and `/direct` is the directory on which the file system is mounted.



The preceding diagram 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.

The file system was unmounted with a command of the form:

`/etc/umount` */dev/fd_name* [Return]

where */dev/fd_name* is the block special (device) file associated with the mass storage device containing the file system.

NOTE

Always unmount a file system before removing it from its mass storage device (as in removing the flexible disc). Removing a mounted file system from its mass storage device before unmounting it is likely to corrupt the file system.

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. 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` *special_fname* [Return]

   where *special_fname* is the pathname of the block special (device) file of the device associated with the mounted file system. This command fails if there are open files on the file system you are attempting to unmount.
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 disc), the medium can now be removed safely.

Mounting/Unmounting File Systems using /etc/checklist

If you wish to mount all file systems in */etc/checklist*, enter the following:

```
mount -a
```

The full path name where each file will be mounted must also exist in */etc/checklist*.

If you wish to unmount all file systems in */etc/checklist*, enter the following:

```
umount -a
```

Special Considerations

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:

- having 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.
- if a file has been accessed and the sticky bit is set in the file's protection mode, the use count of the file is always greater than zero; the file is always open.

Removing Optional Products and Filesets

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 file sets.

1. Determine which file sets you wish to remove.

   A list of HP-UX file sets is given in Appendix C. Appendix C does not list the file sets for optional products, such as LAN. A file, */etc/filesets*, contains a complete list of file sets you have installed on your system.

2. Become the root user.

3. Shut down the system, by typing in:

   shutdown 0 [Return]

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* [Return]

   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 (see "Updating Your HP-UX System").

5. Return your system to normal operating mode by typing in:

   reboot [Return]

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, and optionally your Series 200, Model 236, if you have powered down—they have a battery which keeps the clock current.

Only the super-user can change the system clock. To set the current time and date:

1. Login as the super-user `root`.
2. Insure that the time zone environment variable TZ is set properly (for more information refer to *tzset* under the *ctime(3)* entry in the *HP-UX Reference* manual).

   Typically, TZ's value is set with a variable declaration (as shown below) in the file */etc/profile*. It can also be set from an application program with the *tzset* library routine.

   As shipped to you, the system is set up to run in the Mountain Time Zone. To change the time zone to your time zone, modify the */etc/profile* file to contain two entries of the form:

   ```
   TZ=XXXHYYY
   export TZ
   ```

   where `XXX` and `YYY` are three letter representations of the standard and daylight time zones for your area and `H` represents the difference between current local time and Greenwich Mean Time, in hours. The `export TZ` line will remain the same regardless of the time zone. For example, in Denver, Colorado you would enter the following:

   ```
   TZ=MST7MDT
   export TZ
   ```

   where MST stands for Mountain Standard Time and MDT stands for Mountain Daylight Time. Here are some other examples:

   - For St. Clair Shores, Michigan: `TZ=EST5EDT`
   - For Norman, Oklahoma: `TZ=CST6CDT`
   - For Corvallis, Oregon: `TZ=PST8PDT`
   - For Hawaii: `TZ=HST10` since Hawaii has no Daylight Savings Time.

3. Terminate the *cron* process if it is running.

   To terminate *cron*, first locate the cron process information by typing in:

   ps -ef [Return]

   Then terminate *cron* by typing:

   kill -9 *pid* [Return]

   where *pid* is the process id associated with *cron*.

4. Now that the time zone is set, you can set the correct time and date (using the *date* command) by typing an entry of the form:

   date MMddhhmm{yy} [Return]

   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, 85 specifies 1985 as the current year.

5. Restart *cron* if you terminated it in step 3.

   To restart *cron*, type in:

   cron [Return]

When *date* is executed it echoes the time and date on your screen.

Special Considerations

The *make* program (see 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*; see 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.

If you are only changing the clock by a small amount (20 - 30 minutes), *cron*'s behavior should not present a problem and no corrective action is necessary.

Setting Up the LP Spooler

HP-UX provides a series of commands, collectively referred to as the LP Spooler, to configure and control line printer spooling. Line printer spooling is a mechanism by which printing requests and their associated files get stored temporarily in a spool directory until they can be printed. The LP Spooler can be customized to spool to different printers and allows printers to be grouped into various classes to increase the overall efficiency of the system. Some of the LP commands are available to all users; others, only to the system administrator.

This section presents some terminology and an overview of the spooler's operation followed by a brief description of all LP commands: those available to all users and those available only to the super-user (the user `root`) or LP Spooler administrator (the user `lp`). It then describes how to install and configure the LP Spooler, and concludes with some tips on monitoring and maintaining the spooler.

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(1)* options. When *lp(1)* is invoked, it associates a unique ID with the request and passes the request to the **LP scheduler** (invoked by the *lpsched(1M)* 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(1M)* 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 models for creating new interface programs.

The *lp(1)* command directs output to the default destination unless a destination is specified when *lp(1)* is executed. The **default destination** may be set or changed by the system administrator. A **destination** is either a printer or a class and 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(1M)* command whereas physical destinations are defined using *mknod(1M)* — which associates physical devices with special (device) files. A single physical destination may be associated with one or more logical destinations. *Lp(1)* requests are directed to a logical destination as long as it has been set to accept requests (see *accept(1M)*). When a corresponding physical destination (a printer) is available and has been enabled (see *enable(1)*), the request is transferred to it.

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(1)* command. The user may address a specific printer or a specific request ID number. See the *lp(1)* entry in the *HP-UX Reference*.
- **disable(1)** Disables one or more physical printers such that they will not print *lp(1)* requests. See the *enable(1)* entry in the *HP-UX Reference*.
- **enable(1)** Activates one or more physical printers to print *lp(1)* 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(1)* 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.

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(1)* 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(1M)* command options are used.
- **lpmove(1M)** Moves requests queued by the LP scheduler from one destination to another. The LP scheduler must **not** be running when *lpmove(1M)* is used. See the *lpsched(1M)* entry in the *HP-UX Reference*.

- **lpsched(1M)** Schedules requests taken by *lp(1)* for spooling to line printers.
- **lpshut(1M)** Shuts down the LP scheduler. See the *lpsched(1M)* entry in the *HP-UX Reference*.
- **reject(1M)** Rejects *lp(1)* requests on one or more logical destinations where a "destination" is a printer or class of printers. See the *accept(1M)* entry in the *HP-UX Reference*.

Installing the LP Spooler

To install the LP Spooler, first log in as the super-user (`root`).

Three configuration files **must** be checked and/or modified for the LP Spooler to work properly: */etc/passwd*, */etc/group*, and */etc/rc*. The contents of these configuration files depend on the version of your HP-UX system. If you have just **installed** a new system, the files should be in the */etc* directory and contain the appropriate information; if you have just **updated** an existing system, you need to update your existing files (*passwd*, *group*, and *rc*) with the information contained in the files in the */etc/newconfig* directory. See the section "Updating the HP-UX System" in this chapter and the file */etc/newconfig/Update_info* for details on different revisions of HP-UX. Note that the */etc/newconfig* directory is only shipped with updates. Check the files for the following information and, if necessary, update your existing files:

- The */etc/passwd* file should contain:

  ```
  lp::9:2::/usr/spool/lp:/bin/sh
  ```

  providing ownership of the LP Spooler to the user `lp`. There may be other users also associated with the group *bin* in your particular configuration.

- The */etc/group* file should contain:

  ```
  bin::2:root,bin,lp
  ```

  providing group ownership of the LP Spooler to the user `bin`.

- The */etc/rc* file should contain:

  ```
  # Start lp printer scheduler
          rm -f /usr/spool/lp/SCHEDLOCK
          /usr/lib/lpsched
          echo line printer spooler started
  ```

  to start up the LP scheduler every time the system is booted.

You may want to add password protection to the `lp` user. To do this, log in as the user `lp` and execute the *passwd* command; see the *passwd(1)* entry in the *HP-UX Reference* for details.

To set up a particular printer to be used with the LP Spooler, you can edit and use the */etc/mklp* script, or type in the commands directly from the keyboard. The first step is to make a special (device) file for the printer using the *mknod* command. The major number depends on the interface and protocol used; the minor number depends on the interface and select code to which the printer is connected; see the */etc/mkdev* script or the "Adding/Moving Peripheral Devices" section for details.

Assume, for example, that you want to set up an HP 2934A printer as a spooled device. The printer is on the HP-IB at select code 5, bus address 1. To install the printer, type the following commands (followed by [Return]):

```
/etc/mknod  /dev/lp2934  c  7 0x050100
chown  lp  /dev/lp2934
chgrp  bin  /dev/lp2934
chmod  600  /dev/lp2934
```

This sequence creates the special (device) file for the HP 2934 printer and changes the file's owner to `lp`, its group to `bin`, and its protection mode to read/write access for the owner only.

Once this is done and the LP Spooler is configured (described in the following section), spooling requests can take the form:

```
pr myfile | lp -dlp2934
```

which pipes the output of *pr(1)* (the file `myfile`) to *lp* and specifies the destination as the HP 2934 printer.

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). Alternately, you can take an existing special file for the printer and create a link to it from the pathname */dev/lp*.

Configuring the LP Spooler System

The */etc/mklp* script is available which you can edit and execute. This section describes each step which also appears in the script. Configuring the LP Spooler system requires the following steps:

1. Log in as super-user and shut down the LP scheduler with:

   ```
   /usr/lib/lpshut
   ```

2. Determine for each line printer, how the LP Spooler system will communicate with the printer. This is done by specifying a **model** script when you invoke *lpadmin(1M)*.

   Models are shell scripts 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 model scripts for a generic "dumb" printer, the HP 2225A, HP 2631G, HP 2686A, HP 2688A, HP 2934A, and the HP 9000 Model 520 internal thermal printer. 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.

3. When you have selected a model, execute the *lpadmin* command with its **-p** option to name a printer. 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 special (device) file — the physical destination.

   `-mhp2934a` specifies a model in the */usr/spool/lp/model* directory.

   `-h` specifies that the printer is "hard-wired".

4. For each of the printers defined with *lpadmin*, execute *accept* and *enable* to allow requests to reach the printer:

```
/usr/lib/accept lp
/usr/bin/enable lp
```

   Note that `lp` is the name of the logical destination. In step 3, the *lpadmin* command associated it with a particular physical destination by specifying the device file name.

5. Select a printer to be the system default. For example, if the printer `lp` is to be the default, execute:

```
/usr/lib/lpadmin -dlp
```

6. Now restart the LP scheduler with:

```
/usr/lib/lpsched
```

   and see that the LP spooler's "scheduler" is properly running by executing:

```
lpstat -t
```

7. If the scheduler is not running, you must remove the file SCHEDLOCK before it will work properly. Do this by typing:

```
rm -f /usr/spool/lp/SCHEDLOCK
```

   and then repeat step 6 above. The `SCHEDLOCK` file acts as a "semaphore" to keep more than one scheduler from running at any given point in time. The *lpshut* command automatically removes the `SCHEDLOCK` file when it terminates the LP scheduler. The *shutdown* command executes *lpshut*.

Other LP Spooler Adminstrator Duties

There are several other 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;
- starting and stopping the LP scheduler;
- grouping printers into classes;
- removing destinations (printers and classes of printers);
- moving requests to other destinations.

Determining LP Spooler Status

The command *lpstat(1)* 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 Jul 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
```

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 special (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.

You can specify any combination of the above options on an *lpstat* command line.

In addition to using zero or more of *lpstat*'s options, you can also follow the command with particular request IDs, in which case *lpstat* provides status information about those requests.

Controlling the LP Scheduler

The LP scheduler services all *lp* requests by routing them to an interface program associated with the specified printer or class of printers. Interface programs control the actual printing on the devices.

To start the LP scheduler running, use:

```
/usr/lib/lpsched
```

The scheduler must be running for the LP Spooler to be available for use. However, you **must** shut down the scheduler before using either *lpadmin* or *lpmove*. To shut down the scheduler, use:

```
/usr/lib/lpshut
```

Remember to re-start the scheduler once you are through using *lpadmin* or *lpmove*.

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. For example, you could associate the printer described above to a class with:

```
/usr/lib/lpadmin -plp -cclass1
```

This creates the class `class1` (unless it already exists) and inserts the printer `lp` into it.

Removing Destinations

LP Spooler destinations (printers, classes, or both) are removed with the *lpadmin* command. 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

Occasionally it is useful to move requests from one destination to another, such as when one printer is down for repairs. The *lpmove* command is provided for this purpose. Before using the command make sure that *lpsched* is not running. To shut down the LP scheduler, execute:

```
/usr/lib/lpshut
```

You can use *lpmove* in one of the following ways.

1. Move all requests for printer `lp1` to printer `lp2`:

   ```
   /usr/lib/lpmove lp1 lp2
   ```

2. Move the request with the ID `lp1-103` to printer `lp2`:

   ```
   /usr/lib/lpmove lp1-103 lp2
   ```

Lpmove never checks the acceptance status of the new printer (whether or not *accept* has been executed on it) when it moves requests; therefore, you should execute:

```
lpstat -alp2
```

to see if `lp2` can accept requests before actually redirecting requests to it.

Shutting Down the System

Improperly powering down the computer (or an "on-line" mass storage device) can cause the file system to become corrupt. *shutdown(1M)* terminates, 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.

The *shutdown(1M)* command — among other things — kills all unnecessary processes, forces the contents of the file system's I/O buffers to be written to the disc (with the *sync(1)* command), and takes the system into the single-user state. It will also optionally halt or reboot the system.

To shutdown the system from a normal operating mode, perform the following steps.

1. Login as the super-user `root`. See "Becoming the Root User" in this chapter for more information on how to become the super-user.
2. Move to the root directory of the file system by entering the command:

   `cd /` [Return]
3. Execute the shutdown command.

   The *shutdown* command allows you to specify a *grace_period*, which is the number of seconds you want *shutdown* to wait before terminating all processes. You can also use the *-r* option to automatically reboot the system after reaching run-level "s", or the *-h* option to halt the system.

   The *shutdown* command looks like (see the examples below):

   /etc/shutdown *[-r|-h] grace_period* [Return]

   If *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 hold the [CTRL] key depressed as you press [D] to signify the end of the message.

   If *grace_period* is omitted, then after waiting the specified amount of time, *shutdown* asks if you want to continue. When *shutdown* completes its task, it displays a message telling you to halt the system when you are ready.
4. If you have halted the system (*shutdown -h*) you may now power down the system if you wish. If you wish to bring the system back up, go to step 5.

If you have not halted the system, and you wish to power down the system, you have two alternatives:

a. If you ran *shutdown* without the *-h* option, you are now in run-level s, and can halt the system by typing in:

```
reboot -h [Return]
```

b. If you have not yet run the *shutdown* program, you are still in the normal operating run-level, and can halt the system by typing in:

```
shutdown -h [Return]
```

5. If you do not wish to power down the system, and now need to resume normal operating run-level, you should type in:

```
reboot [Return]
```

For example:

- To activate a newly configured kernel, you should shutdown the system with no grace period, and automatically reboot:

```
shutdown -r 0 [Return]
```

- If you wish to install an interface card, halt the system by typing:

```
shutdown -h 0 [Return]
```

Wait for the "halted" message, then turn the power off to the computer.

- If you wish to backup your system, you should change to run-level "s" by typing:

```
shutdown 0 [Return]
```

After running backup, bring the system back up with all the daemons running by typing:

```
reboot [Return]
```

NOTE

Do not execute *shutdown -r* from run-level "s". You must reboot using the *reboot* command.

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

```
reboot -h [Return]
```

√Updating Your HP-UX System

This section describes the steps necessary to update your HP-UX system, as well as how to install optional products such as the SRM access utilities (Shared Resource Manager), LAN (Local Area Network) and optional partitions of your operating system. Since the process of updating or installing optional products could involve changes to the HP-UX kernel, you should carefully follow the preparatory steps below before proceeding. Note that the procedure is exactly the same for updating and for optional product installation. You should read the definitions and discussion in the first part of chapter 2 ("Installing HP-UX").

An overview of the entire update procedure is:

1. prepare the system for an update
2. locate the product (optional product, operating system update, or optional partition),
3. load the update tools if you are updating your operating system,
4. perform the update,
5. exit the update program
6. Check for additional information in the *Installation Notes* or in the file called `/etc/newconfig/Update_info`.

Each of these 6 steps is discribed in the following subsections.

Preparing to Update

The HP-UX kernel could be modified when you update the system. Because of this, it is always possible to crash your disc while updating. By following these simple steps, you will be able to minimize any risks to the integrity of your computer system and data.

Shutdown your system by following the procedure in "Shutting Down the System".
You do not need to specify any option, other than the grace period, when shutting down the system for an update.

Back up your file system
If you make a mistake, and possibly corrupt your file system, while updating your computer, 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 ("The System Administrator's Toolbox") for more information on how to archive your system.

Do an fsck

Now that your system is in single-user mode, you should do a file system check using the *fsck(1M)* command. For more on how to use the *fsck* command see Appendix A, "Using the FSCK Command".

If the *fsck* showed no problems with your file system, and your system is in single-user mode, you are ready to begin the update and optional product installation process.

You should now be running in a single-user mode, and logged in as the super-user, `root`.

Find /dev Major and Minor Numbers

You need to know the major and minor numbers of the drive used to read the update media, and for the hard disc being updated as well. **Be sure you know the correct values of these numbers before you continue**. You can obtain this information by typing:

```
ll /dev/source /dev/dest Return
```

where *source* is the tape drive or flexible disc drive your update will be loaded from, and *dest* is the hard disc you will be updating (probably your root disc). Make sure you write these values down, because the */etc/update* program uses screen menus—you won't be able to "scroll" back to find these numbers.

Locate and Write-protect the Product

You will be performing the update either with cartridge tape or with flexible discs.

If your update is on 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 CS/80 data cartridge 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.**

If your update is on flexible discs, locate the write-protect mechanism on the label end, and the back side of the disc. Slide the write protect shutter toward the label end of the disc, so you uncover the write protect hole. Holding the flexible disc with the label up and toward you, insert the disc into your disc drive.

Load the Update Tools

The update tools need to be loaded ONLY if you are updating your operating system with a new set of update media. If you are adding an optional partition or loading an optional product, you do not need to load the update tools. If you are not updating your operating system, go to the section called "Perform the Update".

If you are updating your operating system, type in:

```
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 special device file name assigned to the same cartridge tape or flexible disc drive you just inserted the update media in. 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.

Perform the Update

NOTE

Before you perform the update, you must have completed the preperatory steps to ensure your system is ready. You must have copied the major and minor numbers of the cartridge tape drive or flexible disc drive the update will be on, and the major and minor numbers of the disc you will be performing the update to (usually your root disc).

If you have a non-HP terminal, you **MUST** execute the update program with a *-m* option. This will turn off all the menus in the update program. You will be prompted for the appropriate choices, rather than see the menu, on your screen.

1. Type in:

   /etc/update [Return]

   Your system will reboot to remove any remaining processes. You will see your normal boot messages, and then the screen should clear and the menu shown in Figure 5-4 will appear. This is the main update utility menu. All update procedures are treated as sub-tasks from this menu.

```
*******************************************************************
                 HP-UX UPDATE UTILITY -- MAIN MENU
*******************************************************************
                           Select Choice


Source device is:      /dev/update.src   Major: -1  Minor: fffffff
Destination device is: /dev/update.dest  Major: -1  Minor: fffffff

DISPLAY options for a new partition
EXIT update
CHANGE source device
CHANGE destination device
```

Figure 5-4: Main Utility Menu

Note the four softkeys at the bottom of your screen:

- **NEXT** will move the highlight to the next item in each menu.
- **PREVIOUS** will move the highlight to the previous item in each menu (the item listed above the current item).
- **SELECT** will execute the currently highlighted option.
- **QUIT** will exit the *update* program at any time.

Both the source and destination device have the major number listed as -1 and the minor number as fffffff. This prevents you from accidentally accessing the wrong device. Since these are impossible values, you must enter the correct major and minor numbers, which you should have written down earlier.

2. CHANGE the source device.

By using the **NEXT** and **PREVIOUS** softkeys, choose the "CHANGE source device" option on the main menu and press **SELECT**. You will see the top half of the menu shown in Figure 5-5.

```
*************************************************************
        HP-UX UPDATE UTILITY -- CHANGE DEVICE MENU
*************************************************************



Current source device address is:  Major: -1  Minor: ffffffff

                    NEW Major Number?  4
                    NEW Select Code?   7
                    NEW Bus Address?   4
                    NEW Unit Number?   0
                    NEW Volume Number? 0
```

Figure 5-5: Change Device Menu

As you are prompted for each item, enter the source device's major number, select code, bus address, unit and volume numbers in **decimal format**. You have these numbers written down in hexidecimal format. In our example the major number was 4, and the minor number was 0x070400 in hexadecimal format (do not type in any leading zeros for these values). The minor number has the format:

`0xScBaUV`

where 0x indicates the number is in base 16, Sc is the select code, Ba is the bus address, U is the unit number, and V is the volume.

Once you have entered the volume number, the main menu will appear. Notice that the new source device values are now shown. Check that the values shown on the menu match those you have written down. It is possible that you could make a mistake while converting from hexadecimal to decimal format.

3. CHANGE the destination device.

   Use the **NEXT** and **PREVIOUS** softkeys to choose the "CHANGE destination device" option on the menu and press **SELECT**. You should now see a screen very similar to the menu shown in Figure 5-5, only you will be changing the destination device rather than the source device.

   In exactly the same manner as you just changed the source device, enter the new destination device major and minor numbers, which you have written down. We will use major number 0 and minor number 0x0e0500 in our example.

   Once you have entered the new volume number, the update program will attempt to *mount(1)* the device you have listed at that address. If it is the root device (which is the normal case), the following prompt will appear near the bottom of the console:

   ```
   Cannot mount the destination device.
   Is this the ROOT device?  ('y'or'n') >>
   ```

   Because the method of updating the root volume versus another hard disc is different, you must tell the program that you are indeed updating the root volume by typing [y]. Otherwise, you should enter an [n].

   Once you have entered the correct addresses for the source and destination devices, you see these values reflected in the main menu, similar to our example menu in Figure 5-6. If either of these values are wrong, you can go back to change either the source or the destination device. **Do not continue if you are unsure of these values!** Use the **QUIT** softkey if you are not sure of these device addresses, and go back to step 1 to begin again.

```
*******************************************************************
                  HP-UX UPDATE UTILITY -- MAIN MENU
*******************************************************************
                            Select Choice


Source device is:      /dev/update.src   Major:  4  Minor: 70400
Destination device is: /dev/update.dest  Major:  0  Minor: e0500

DISPLAY options for a new partition
EXIT update
CHANGE source device
CHANGE destination device
```

Figure 5-6: Main Menu After Changing Devices

4. Choose the "DISPLAY" option.

   Using the **NEXT** and **PREVIOUS** keys, choose the "DISPLAY options for a new partition" menu item and press **SELECT**. The *update* procedure will now read the update tape or flexible disc to get a list of available options, which takes a couple of minutes. You should see the screen shown in Figure 5-7.

```
          ****************************************************
                HP-UX UPDATE UTILITY -- READING FILE MENU
          ****************************************************
                               reading


         Please insert media for a new partition.
         When the busy light goes off, press [Return]
```

Figure 5-7: Reading File Menu

   Press [Return]. The *update* procedure will read the tape or flexible disc, and a new main menu will appear.

   If you are performing the update from cartridge tape, perform steps 5 and 6, then go to step 8. If you are performing the update from flexible discs, skip steps 5 and 6, go to step 7.

5. If you are updating from flexible disc, go to step 7.

   The new menu is similar to the menu shown in Figure 5-8.

```
*******************************************************************
                HP-UX UPDATE UTILITY -- MAIN MENU
*******************************************************************
                          Select Choice



 Load "98515A" partition
 Load "SYS_CORE" partition

 Process ALL partitions
 DISPLAY options for a new partition
 EXIT update
 CHANGE source device
 CHANGE destination device
```

Figure 5-8: Partition Menu for Update on Cartridge Tape

- If you wish to process all the partitions, use the **NEXT** and **PREVIOUS** softkeys to move to the "Process ALL partitions" menu item, and press the **SELECT** softkey. The update program will automatically load all partitions and file sets. You will be prompted when to change media, and will be returned to the menu in Figure 5-6 when everything is loaded. You can then EXIT the update program. It takes approximately one hour to load everything from 1 tape.
- If you wish to "load" a partition, use the **NEXT** and **PREVIOUS** softkeys to move to the "Load xxxxxx partition" menu item, and press the **SELECT** softkey.

  Go to Step 6.

- If you wish to display the options for a new partition, use the **NEXT** and **PREVIOUS** softkeys to move to the "DISPLAY options for a new partition" menu item, and press the **SELECT** softkey. The *update* procedure will now read the update tape to get a list of available options, which takes a couple of minutes. You should see the screen shown in Figure 5-7.
- If you are finished loading partitions, keep using the "DISPLAY" option to work with more media.
- If you wish to exit the update procedure, use the **NEXT** and **PREVIOUS** softkeys to move to the "EXIT update" menu item, and press the **SELECT** softkey.

6. This step is performed if you chose to "Load" a partition from cartridge tape (see step 5). If you are updating from flexible disc, go to step 7.

   You will see a menu similar to that in Figure 5-9. On the new menu, notice a partition name on the upper left segment of the screen, and the list of product "file sets".

```
Partition:  "SYSCORE"              Media Number:  "1"
***********************************************************
           HP-UX UPDATE UTILITY -- FILE SET MENU
***********************************************************
                       select choice


Load "ACORE" fileset
Load "ACORE2" fileset


Process ALL file sets
Use the [Quit] softkey to exit the menu
```

Figure 5-9: File Set Menu for Updates on Cartridge Tape

At this point you have several options:

- If you wish to load all the file sets at one time, use the softkeys to select the `Process ALL file sets` option. This procedure could take 20 to 60 minutes to load depending upon the processor and peripherals being used. When finished, you will be returned to the fileset menu (Figure 5-9).

  Press the **QUIT** softkey to exit. You will then return to the partition menu (Figure 5-8).

- If you wish to load just the core system (basically the update for HP-UX itself), use the softkeys to select the option similar to `Load "ACORE" file set`. Note that in each "Load xxxxxx" option, the xxxxxx is the product number or name for that file set.

- If you wish to load the core system and certain optional products, select the options you want. You will select one loading option at a time. Each time, that fileset will be loaded, along with all dependencies, and you will return to the fileset menu. When you have loaded all the filesets you wish, press the **QUIT** softkey to exit. You will be returned to the partition menu (Figure 5-8).

- If you wish to exit the fileset menu, press the **QUIT** softkey to return to the partition menu, Figure 5-8.

The *update* program will immediately load this fileset if it has never done so previously, but if it has, the *update* program will inform you that the product exists on your system and asks you if you want it removed. [y] will remove the old fileset, and re-load the new one. Answering [n] will prevent this fileset from being loaded.

NOTE

When loading the HP-UX operating system or SYS_CORE partition you should answer [y] for each piece queried, unless you are sure that it is okay to not re-load a particular piece.

When the selected option is complete, the main menu (shown in Figure 5-8) will be re-displayed, but with one or more of the "load" options removed. Those options corresponding to the partitions you have completely loaded will not be displayed.

7. Perform this step if you are updating from flexible discs.

   You will see a menu similar to that in Figure 5-10. On the new menu, notice a partition name on the upper left segment of the screen, and the list of product "file sets".

```
Partition:  "SYSCORE"              Media Number:  "1"
***********************************************************
           HP-UX UPDATE UTILITY -- MAIN MENU
***********************************************************
                        select choice


Load "98515A" file set
Load "ACORE" file set

Process ALL file sets
DISPLAY options for a new partition
EXIT update
CHANGE source device
CHANGE destination device
```

Figure 5-10: File Set Menu for Updates on Flexible Disc

At this point you have several options:

- If you wish to load all the file sets at one time, use the softkeys to select the `Process ALL file sets` option. This procedure could take 20 to 60 minutes to load depending upon the processor and peripherals being used.
- If you wish to load just the core system (basically the update for HP-UX itself), use the softkeys to select the option similar to `Load "98515A" file set`. Note that in each "Load xxxxxx" option, the xxxxxx is the product number or name for that file set.

- If you wish to load the core system and certain optional products, select the options you want.

  You will select one loading option at a time. Each time, that fileset will be loaded, and you will return to the fileset menu. When you have loaded all the filesets you wish, press the **QUIT** softkey to exit. You will be returned to the main menu, shown in Figure 5-10.

- If you wish to load more operating system partition, select the "DISPLAY options for a new partition" menu item. You will go back to the menu shown in Figure 5-6. Continue with step 4.

- If you wish to abort the entire update procedure, select the "EXIT update" menu item.

The *update* program will immediately load this file set if it has never done so previously, but if it has, the *update* program will inform you that the product exists on your system and asks you if you want it removed. [y] will remove the old file set, and re-load the new one. Answering [n] will prevent this file set from being loaded.

NOTE

When loading the HP-UX operating system or the ACORE fileset, you should answer [y] for each piece queried.

When the selected option is complete, the main menu (similar to the menu shown in Figure 5-10) will be re-displayed, but with one or more of the "load" options removed. Those options corresponding to the file sets you have already loaded will not be displayed. If you decided to load all the file sets, there will be no "load" options displayed.

8. Leave the update program.

   If you have loaded all the partitions you want in this session, 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. The system will reboot.

   NOTE

   Do not cycle power during the reboot.

   The reboot of your system may appear to take longer than normal. This is because the first time you reboot after doing an update, scripts execute which customize your system for the updated products you installed.

9. Check for Additional Information.

   Following this reboot process, you should log in and check to see if you have a file called `/etc/newconfig/Update_info`. Also look for any other update files in the */etc/newconfig* directory or in the */tmp/update.log* file. Follow instructions in these files, and instructions in the *Installation Notes*, if supplied.

System Accounting 6

Multi-user HP-UX allows concurrent sharing of computer resources among multiple users: several users can be logged in, all sharing disc space, memory, and the CPU. On multi-user systems, HP-UX System Accounting provides the means to:

- monitor disc space usage for individual users
- record connect session data (logins/logouts)
- collect resource utilization data (such as memory usage, and execution times) for individual processes
- charge fees to specific users
- generate summary files and reports that can be used to analyze system performance and bill users for resource consumption

What Is in This Chapter?

HP-UX System Accounting allows you to accomplish accounting tasks through a number of versatile commands. This chapter illustrates the use of these commands and contains the following sections:

- "Overview of System Accounting" provides the background information necessary to understand how to use System Accounting.
- "Daily Usage and Installation" shows the routine daily usage of System Accounting and shows you how to install it.
- "Disc Space Usage Accounting" illustrates the use of the accounting commands that monitor disc space utilization on a per-user basis.
- "Connect Session Accounting" describes the commands that record and report connect session accounting information.
- "Process Accounting" shows how to generate per-process accounting data and reports.
- "Charging Fees to Users" is the section where you learn how to charge fees to users.
- "Summarizing and Reporting Accounting Information" shows how to generate the main daily and monthly accounting reports that are used to monitor system performance and bill users.
- "Updating the Holidays File" describes how to set up the file */usr/lib/acct/holidays* on your system.
- "Fixing Corrupted Files" Occasionally, during day-to-day usage of System Accounting, certain files may become inconsistent or messed up; this section shows how to fix these files.
- "Sample Accounting Shell Scripts" provides listings of shell scripts that you might find useful on your system.

In addition to these sections, the appendix "System Accounting Files" contains brief definitions of all the files used by System Accounting.

NOTE

Much of the material in this chapter assumes greater knowledge of HP-UX than is required of the "average" user. In particular, System Accounting borrows many concepts from the previous chapters "Concepts", and "System Boot and Login." If you are unfamiliar with the concepts and terminology in those chapters, then you should review them.

Overview of System Accounting

In this section, the instrinsics of System Accounting are examined. Key terms are defined, commands are introduced, system data flow is described, and finally, you are shown the login and directory structure of System Accounting.

Definitions

The following terms are specific to System Accounting. Other terms are listed in the Glossary and the end of this manual.

prime/non-prime connect time

Prime time is the time during the day when the computer system is most heavily used—for example, from 9:00am to 5:00pm. Non-prime time is the remaining time during the day when the system is less heavily used—from 5:00pm to 9:00am in this example.

When reporting computer time usage, System Accounting distinguishes between prime and non-prime time usage. You can specify prime and non-prime time on your system by editing the file */usr/lib/acct/holidays*. (For details on the *holidays* file, see the section "Updating the Holidays File" in this chapter.

NOTE

Prime time is in effect only on weekdays (Monday through Friday); non-prime time is in effect during the weekends (Saturdays and Sundays) and on any holidays specified in the *holidays* file.

process accounting records

Once System Accounting is installed and turned on, the following occurs: whenever a process terminates, the kernel writes a process accounting record for the terminating process into the current process accounting file, */usr/adm/pacct* by default (you can specify that a file other than *pacct* be used as the process accounting file, if you want).

A process accounting record contains resource-usage data for a single process; it summarizes *how much* of the various resources the process used during its lifetime. Examples of information contained in process accounting records are:

- the user ID of the process's owner
- the name of the command that spawned the process
- the amount of time it took the process to execute

For greater detail on the contents and format of process accounting records, see *acct(5)* in the *HP-UX Reference Manual.*

total accounting records

These records, created by various accounting commands, contain summary accounting information for individual users. These records provide the basic information for many reports generated by System Accounting. Some examples of information contained in these records are:

- the ID and user name of the user for whom the total accounting record was created
- the total number of processes that the user has spawned during the accounting period for which the total accounting record was created
- fees for special services rendered to this user

The exact contents and format of total accounting records can be found in *acct(5)*. In addition, commands covered in later sections of this chapter show how these records are created and used by System Accounting.

Introduction to Commands

System Accounting provides many versatile commands to accomplish numerous varied tasks. There are commands that create data, commands that display data, commands that remove data, commands that merge data, and commands that summarize and report data. In addition, the output of one command may be the input to others, and so on.

Accounting commands can be logically categorized into six basic command groups:

- installation
- disc usage accounting
- connect session accounting
- process accounting
- fee charging
- summarizing and reporting accounting information

Descriptions of these command groups, along with a brief synopsis of each command, follow:

Installation

These commands insure that System Accounting is properly installed. They are used to turn accounting on when HP-UX is powered up and turn accounting off when the system is shut down. They may also do some file cleanups. Two such commands exist:

- **startup**—starts accounting when HP-UX is switched to multi-user mode. Invoked from */etc/rc*.
- **shutacct**—turns off accounting when HP-UX is turned off via the */etc/shutdown* shell.

Disc Usage Accounting

In general, these commands produce disc usage accounting information: they show disc space usage (in blocks) for individual users. They also produce total accounting records. There are four commands:

- **acctdusg** and **diskusg**—both commands show how many blocks of disc space users are consuming. They differ in command options, and the manner in which they produce the information—*acctdusg* takes its input from a list of path names created by *find*, and *diskusg* looks at the inodes of the file system to create its output.
- **acctdisk**—this command produces total accounting records. Its input is supplied (either directly or indirectly) from *acctdusg* or *diskusg*.
- **dodisk**—Produces total accounting records by using the *diskusg* and *acctdisk* commands. *dodisk* is normally invoked by *cron*.

Connect Session Accounting

Independently of System Accounting, the programs **login** and **init** record connect sessions by writing records into */etc/wtmp*. Accounting commands can display or fix this file, and can produce total accounting records for this file. There are five commands:

- **fwtmp**—displays the information contained in *wtmp*.
- **wtmpfix**—normalizes connect session records that span date changes (see *date(1)*). Also validates login names in connect session records.
- **acctcon1**—summarizes *wtmp* in ASCII readable format, producing one line per connect session.
- **acctcon2**—takes input of the format produced by *acctcon1* and produces total accounting records as output.
- **prctmp**—used to display the session record file. (The session record file is normally */usr/adm/acct/nite/ctmp*.)

Process Accounting

When process accounting is turned on, the kernel writes a process accounting record to *pacct* whenever a process terminates. A number of accounting commands exist that summarize and report this accounting information. In addition, certain commands turn process accounting on or off and insure that *pacct* doesn't become too large. There are eight process accounting commands in all:

- **accton**—turns process accounting on or off, depending on whether or not a filename argument is supplied with the command. If no filename is given, then process accounting is turned off; the kernel stops writing process accounting records to *pacct*. If a filename is specified, then the kernel starts writing process accounting records to the specified filename.

  accton uses the system call **acct(2)** to turn process accounting on or off. In addition, only the super-user can execute *accton*.

- **ckpacct**—checks the size of the process accounting file *pacct*. If *pacct* becomes too large, then a new *pacct* file is created via *turnacct switch*. **If disc space becomes critically short, then process accounting is turned off until sufficient space is available.** This command is normally invoked by *cron*.
- **turnacct on | off | switch**—performs one of three functions, depending on which argument (on, off, or switch) is specified. *turnacct on* turns process accounting on by calling *accton* with the default filename argument */usr/adm/pacct*; *turnacct off* turns process accounting off by calling *accton* with no filename argument; *turnacct switch* renames the current *pacct* file (so that it is no longer the current process accounting file) and creates a new, empty *pacct* file.

- **acctcom**—displays process accounting records contained in *pacct* (or any specified file).
- **acctcms**—takes *pacct* as input, and produces summary accounting information by command, as opposed to by process.
- **acctprc1**—produces readable process accounting information, mainly for input into *acctprc2*.
- **acctprc2**—takes input of the form produced by *acctprc1* and produces total accounting records.

Charging Fees

Occasionally, you may want to charge a user for something. For example, you might charge fees to users for fixing any damaged files that they have. The **chargefee** command allows you to charge fees to specific users.

Summarizing and Reporting Accounting Information

This group of commands summarizes and reports the data created through the command groups described above. These are the commands that are probably used most frequently; they represent the highest level of accounting commands. Five such commands exist:

- **prtacct**—takes as input total accounting records and displays the records in ASCII readable format.
- **acctmerg**—combines the contents of separate total accounting files into a single total accounting file. Allows the merging of disc, process, and connect session total acounting records.
- **runacct**—the main accounting shell script. Normally invoked daily by *cron*, this command processes disc, connect session, process, and fee accounting information and produces summary files and reports. It accomplishes its task by proceeding through various states. In each successive state it invokes accounting commands to perform a specific task. For example, in one state, total accounting records for connect sessions are created; in another, disc, connect session, process, and fee total accounting records are merged to create one total accounting file.
- **prdaily**—invoked by *runacct* to format a report of the previous day's accounting data; the report is stored in the file **/usr/adm/acct/sum/rpt***mmdd* where *mmdd* is the month and day of the report. *runacct* may also be used to display a report of the current day's accounting information.
- **monacct**—invoked once a month (or accounting period), this command summarizes daily accounting files and produces a summary files for the accounting period.

System Data Flow

At this point, you have the rudimentary knowledge necessary to understand how System Accounting works; you know some important definitions and should basically know what the various commands do. The purpose of this section is to help you visualize how the different commands work together to create accounting data.

Figure 6-1 illustrates how accounting data is created. The diagram is broken into five separate sub-diagrams, each one representing the data flow for a given command group. The following notational conventions are used:

Symbol	Description
source ⇒ dest	Wide arrows represent the *transfer of data* from a source to a destination. The source is at the start of the arrow; the destination, at the point. For example, the inodes of the file system are the source of information used by *diskusg*, which in turn is the source of disc usage reports that are inputs to *acctdisk*.
cause → object	Thin arrows represent cause-effect relationships. The cause lies at the start of the arrow; the object affected lies at the point. For example, *turnacct on* invokes *accton* which then signals the kernel to begin writing process accounting records to *pacct*.
files	Boxes with rounded corners represent files or groups of files. In a more general sense, they represent the inputs to and outputs from the various commands.

NOTE

The installation commands do not appear in the diagram, because they aren't directly involved in the data creation process; they merely insure that it happens.

NOTE

The commands *runacct* and *prdaily* are shown as having no inputs. This isn't exactly true: they do have inputs, but they get their inputs by executing other accounting commands. In essence, their inputs are the same basic inputs of the other command groups.

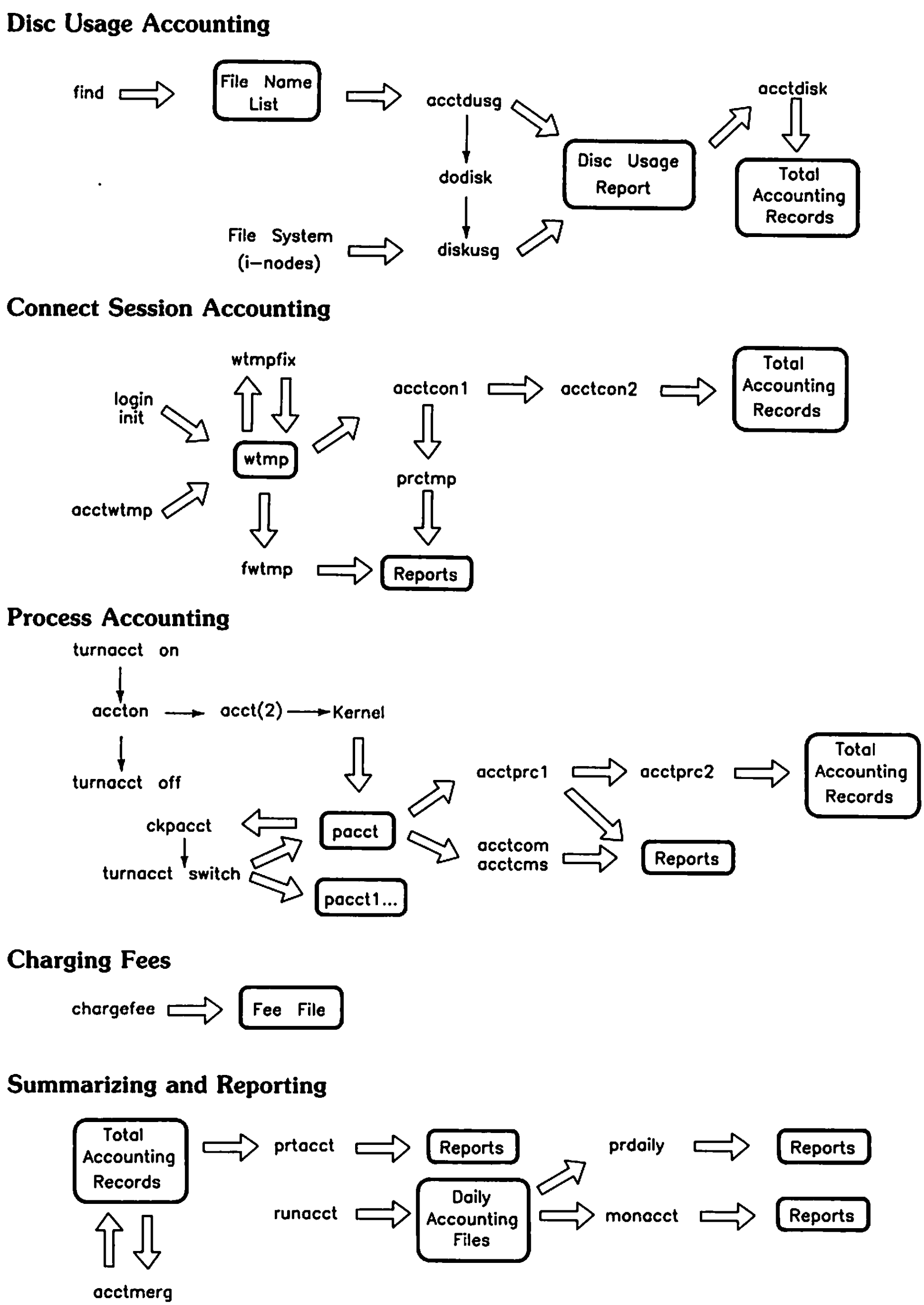


Figure 6-1. System Accounting Data Flow Diagram

Login and Directory Structure

You now know the basics, but you still can't begin learning the day-to-day usage of accounting commands until you know where to log in. In addition, you should know the accounting directory structure—where the various commands, directories, and files are located. These two topics are discussed here.

Logging In

The login name for System Accounting is **adm**; the user ID for *adm* is 4. The *adm* login is a member of the **group** *adm*, and the group *adm* has a group ID of 4, also.

The home directory for the *adm* login is **/usr/adm**. You log in to System Accounting the same way you do for any account—simply supply the login name to the HP-UX login prompt:

```
login: adm
```

NOTE

The integrity of accounting data files must be maintained if System Accounting is to generate accurate reports. For this reason, it is highly recommended that a password be used with the *adm* login.

Directory Structure

System Accounting uses a multi-level directory structure to organize its many accounting files. Each directory in this structure stores related groups of files, commands, or other directories. (See the appendix "System Accounting Files" for definitions of the accounting data files.)

Figure 6-2 illustrates this structure, and descriptions of each directory follow:

- ***/usr/adm***—contains all active data-collection files, such as *pacct* and *fee*.
- ***/usr/adm/acct***—contains the *nite*, *sum*, and *fiscal* directories described below.
- ***/usr/adm/acct/nite***—stores data files that are processed daily by *runacct*.
- ***/usr/adm/acct/sum***—cumulative summary files updated by *runacct* are kept here.
- ***/usr/adm/acct/fiscal***—periodic (monthly) summary files created by *monacct* are found here.
- ***/usr/lib/acct***—System Accounting commands reside here.
- ***/etc***—contains *wtmp*, and shell scripts *rc* and *shutdown*.

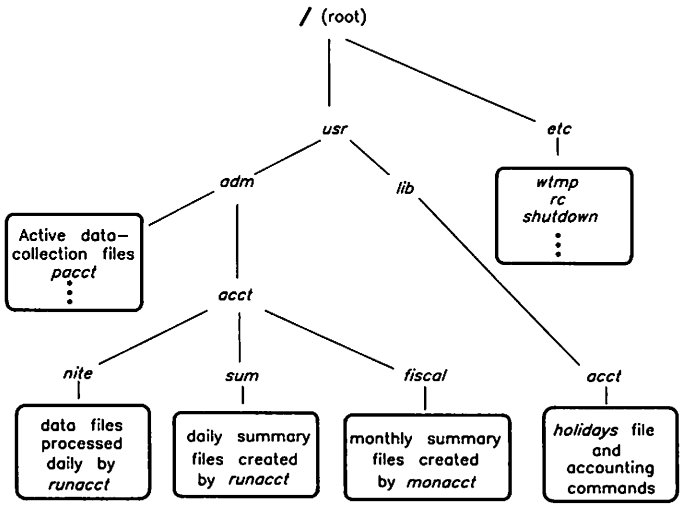


Figure 6-2. System Accounting Directory Structure

Daily Usage and Installation

Now that the basics have been covered, you can start learning how to use System Accounting on a daily basis. The purpose of this section is to show you:

1. How System Accounting automatically creates daily and monthly accounting data and reports.
2. What you must do to get Accounting running on your system.

After reading this section, you should be able to install Accounting on your system. Once properly installed, Accounting will automatically generate daily and monthly accounting data and reports.

Summary of Daily Operation

The daily operation of System Accounting is summarized by the following steps:

1. When HP-UX is switched into multi-user mode, the system initialization shell script *rc* executes the accounting command *startup*. The purpose of *startup* is to start Accounting, and it performs the following functions:
   a. Calls *acctwtmp* to add a boot record to *wtmp*. This record is marked by storing "acctg on" in the device name field of the *wtmp* record.
   b. Turns process accounting on via *turnacct on*. *turnacct on* executes *accton* with the filename argument */usr/adm/pacct*.
   c. It removes work files left in the *sum* directory by *runacct*.
2. A report of the previous day's accounting information can be created by running *prdaily*. Obviously, this step is omitted the first day that Accounting is installed, because the previous day's accounting information doesn't yet exist. However, after *runacct* has been executed, *prdaily* will generate valid reports.
3. The *ckpacct* command is executed every hour via *cron* to insure that the process accounting file *pacct* doesn't become too large. If *pacct* grows past a set maximum number of blocks, *turnacct switch* is invoked, which creates a new *pacct* file. (Other conditions may also limit the size of the process accounting file or turn process accounting off; for more details, see the discussion of *ckpacct* in the "Process Accounting" section.) The advantage of having several smaller *pacct* files is that *runacct* can be restarted faster if a failure occurs while processing these records.

4. The *chargefee* program can be used to charge fees to users. It adds records to the file *fee*. These records are processed during the next execution of *runacct* and merged in with total accounting records.

5. *runacct* is executed via *cron* each night. It processes the active fee file and the process, connect session, and disc total accounting files. It produces command and resource-usage summaries by login name.

6. When the system is turned off using *shutdown*, the *shutacct* command is executed. The purpose of *shutacct* is to stop Accounting, and it performs the following functions:

   a. Writes a termination record to *wtmp* via the command *acctwtmp*. This record is marked by having "acctg off" in the device name field.

   b. Turns process accounting off by calling *turnacct off*.

How to Install System Accounting

Not all users require accounting services on their systems. For this reason, HP-UX System Accounting is provided as an option: if you want to use Accounting, you must install it yourself. Installation procedure is covered here.

There are four steps in the installation process:

1. Updating */etc/rc*
2. Updating */etc/shutdown*
3. Creating *crontab* entries
4. Setting PATH for accounting commands

Each of these steps must be carried out to insure that System Accounting automatically creates daily and monthly accounting information. Detailed descriptions of each step follow.

Updating /etc/rc

The system initialization shell script *rc* must be updated to automatically start System Accounting when the system is switched into multi-user mode. This requires adding the following entry in the **state 2** section:

```
/bin/su - adm -c /usr/lib/acct/startup
```

Updating /etc/shutdown

To insure that accounting is turned off when the system is brought down via *shutdown*, you must add the *shutacct* command to the *shutdown* shell script. The call to *shutacct* should be placed in the section of *shutdown* where all processes are killed by the */etc/killall* command. By calling *shutacct* after *killall*, process accounting information can be captured for the processes that were terminated by *killall*. The entry for the *shutacct* command should be made as follows:

```
/usr/lib/acct/shutacct
```

NOTE

If you do not use */etc/shutdown* to bring your system down, then you should use some other means—such as *turnacct off* or *shutacct* by itself—to turn system accounting off before shutting down your system.

Creating crontab Entries

To automate the daily and monthly creation of accounting data, you should create a *crontab* file that *cron* can use to automatically run certain accounting commands. This process entails the following steps:

1. Log in to System Accounting as the user *adm*.
2. Use an editor to create the *crontab* file containing the accounting commands that are to be run automatically by *cron*. (The actual entries to make in this file are shown after these steps.)
3. Execute the *crontab(1)* command, specifying the file created in step 2 as input. This step insures that the *crontab* file created in step 2 will be scanned by *cron* every minute. After invoking this command, the step 2 file will be stored in the file:

   ```
   /usr/spool/cron/crontabs/adm
   ```

4. At this point, you are finished creating *crontab* entries. If you ever want to change the entries, simply re-edit the file created in step 2 and use the *crontab(1)* command again.

The following entries, accompanied by a description of each, should be made in the *crontab* file created in step 2:

- `0 4 * * 1-6 /usr/lib/acct/runacct 2> /usr/adm/acct/nite/fd2log`

  runacct, the main accounting shell script, should be executed daily (during non-prime hours) to generate daily accounting reports. The above entry executes *runacct* at 4:00am every Monday through Saturday. Error messages will be redirected to the file */usr/adm/acct/nite/fd2log*, if any errors occur while *runacct* executes.

- `0 2 * * 4 /usr/lib/acct/dodisk`

  dodisk creates total accounting records that summarize disc space usage for individual users. This entry runs *dodisk* at 2:00am every Thursday morning.

- `5 * * * * /usr/lib/acct/ckpacct`

  To insure that the process accounting file, *pacct*, doesn't get too large, the command *ckpacct* should be executed hourly. This entry invokes *ckpacct* at five minutes into every hour.

- `15 5 1 * * /usr/lib/acct/monacct`

  The monthly merging of accounting data is facilitated through the *monacct* command. This entry allows *monacct* to generate a monthly total report and total accounting file. *monacct* will be executed at 5:15am on the first day of every month.

NOTE

The dates and times shown in the *crontab* entries above are only suggestions; you can tailor *crontab* entries to suit your needs. However, if you use different entries than those shown here, be sure that *monacct* is run at such a time as to allow *runacct* sufficient time to finish.

Setting PATH for Accounting Commands

Finally, you should set the PATH shell variable in */usr/adm/.profile* so that System Accounting knows where to look for commands. Path should be set as follows:

```
PATH=/usr/lib/acct:/bin:/usr/bin:/etc:/usr/adm
```

Disc Space Usage Accounting

System Accounting provides the means to monitor disc space utilization for individual users. In this section, disc space usage accounting commands are explained. Before reading this discussion, you may want to review the "File System Implementation" section of the "Concepts" chapter.

Disc usage commands provide two main functions: they report disc usage (in blocks) for individual users and create disc total accounting records (supplied as inputs to commands such as *prtacct* or *runacct*).

Reporting Disc Space Usage

Two commands—*acctdusg* and *diskusg*—report disc usage for individual users; both commands show the number of disc blocks allocated to specific users. However, each command has slightly different options. In addition, each differs in the manner in which it produces accounting information.

acctdusg

acctdusg takes from standard input a list of path names, usually created by the *find* command. For each file in the list, *acctdusg* identifies the owner of the file, computes the number of blocks allocated to the file, and adds this amount to a running total for the file's owner. When finished looking through the list, *acctdusg* displays the information accumulated for each user: user ID, user name, and number of blocks used.

This command is useful for reporting disc usage information for specific users or files. For example, suppose you want to know how many blocks of disc space you are using: your user ID is 351, user name is *bill*, and your home directory is */users/pseudo/bill*. The following illustrates how you would use the *find* and *acctdusg* commands to show this information.

```
$ find /users/pseudo/bill -print > bills.files
$ acctdusg < bills.files
00351   bill                30
$ rm bills.files
```

In the above example, `bill` is using 30 blocks of disc space. The series of commands shown could easily have been combined into one line, such as:

```
$ find $HOME -print | acctdusg
00351   bill                30
```

The next example shows how to use *acctdusg* to generate disc usage information for all files in the system:

```
$ find / -print | acctdusg
00350   fred              11
00351   bill              30
00352   mike              17
00353   sarah             13
00354   molly             18
00000   root               3
00004   adm               36
00001   bin             2434
```

Two options are included with *acctdusg*:

-u *no_owners* — If -u is given, then path names of the files for which no owner is found are written into the file *no_owners*. This option could potentially find users who are trying to avoid disc charges.

-p *p_file* — The password file */etc/passwd* is the default file used by *acctdusg* to determine ownership of files. If the -p option is used, then *acctdusg* will use *p_file* instead. This option is not needed if your password file is */etc/passwd*.

The shell script **grpdusg** provided in the section called "Sample Accounting Shell Scripts" displays disc accounting information for users in a given group. It illustrates the use of the -u option with *acctdusg*.

diskusg

This command reports disc usage information in the same format as *acctdusg*—user ID, user name, and total disc blocks used. However, *diskusg* generates disc accounting information by looking through the **inodes** of a specified special file (see *inode(5)* and the "File System" section of the "Concepts" chapter for more information on inodes and special files.) Therefore, *diskusg* is faster and more accurate than *acctdusg*.

The syntax of the *diskusg* command is:

diskusg [*options*] [*files*]

It generates a disc usage report from data in *files*, if specified; otherwise standard input is used. *diskusg* is normally invoked with the *files* argument. When specified, *files* are the special filenames of the devices containing the inode information used by *diskusg* to generate its report. *files* is normally a special file from the */dev* directory.

The following options may be used with *diskusg*:

- `-s` This tells *diskusg* that: (1) input is in *diskusg* output format, and (2) that all lines for a single user should be combined into a single line. This option is used to merge data from separate files, each containing the output from using *diskusg* on different devices.

- `-v` This option is useful for finding users who are trying to avoid disc space accounting charges. When this option is specified, *diskusg* writes records to *stderr* (standard error output) showing the special file name, inode number, and user ID of files that apparently have no owner.

- `-i` *fnmlist* Causes *diskusg* to ignore the data on those file systems whose file system name is in *fnmlist*. *fnmlist* is a list of file systems seperated by commas or enclosed within quotes.

- `-p` *p_file* This is the same as the `-p` option of *acctdusg*.

- `-u` *u_file* This option produces **exactly** the same output as the `-v` option. The difference between the two options is that `-v` writes its output to *stderr*; this option writes its output to the file *u_file*.

The output of *diskusg* is normally used by *acctdisk* to create disc total accounting records. In addition, *diskusg* is normally called by *dodisk*.

The following example creates disc usage information for all users whose files reside on the disc whose device file is */dev/rhd*. **Note that the file system used in this example is the same as was used in the previous *acctdusg* example.**

```
$ diskusg /dev/rhd
0      root     10616
1      bin      778
4      adm      96
350    fred     14
351    bill     32
352    mike     20
353    sarah    16
354    molly    22
355    horatio  2
501    guest    2
```

The differences between *diskusg* and *acctdusg* are best illustrated by comparing their outputs. Note that:

1. *acctdusg* places leading zeros on user IDs; *diskusg* doesn't.
2. *acctdusg* counts files **only under each users $HOME directory**. Files that users own in directories other than their home directory (for example, files in the */tmp* directory) are counted as files with no owner.
3. Two extra users—`horatio` and `guest`—show up in the output of *diskusg* when compared with the output from

   ```
   find / -print | acctdusg
   ```

   This occurred because the directories of these two users were empty; therefore, no disc usage totals were generated. However, *diskusg* looked at inodes and saw that `horatio` and `guest` were actually using two blocks for the directories themselves.
4. If two or more users have links to a particular file, then *acctdusg* will prorate disc space usage for the file between each user. For example, if three users had a link to a 300-block file called *skurbnich.dat*, each user would be charged for 100 blocks of this file.

Creating Total Accounting Records

Two commands are used to create total accounting records: *acctdisk*, and *dodisk*.

acctdisk

acctdisk takes from standard input records of the format produced by *acctdusg* and *diskusg*. From these records, *acctdisk* produces disc total accounting records that may be inputs to *prtacct* or *runacct*.

The following would write disc total accounting records to the file *disktacct* for all users in the group `pseudo`:

```
find / -group pseudo -print | acctdusg | acctdisk > disktacct
```

The next example would generate disc total accounting records for all users who have files on the disc `rhd`. The total accounting records are written to the file *disktacct*.

```
diskusg /dev/rhd | acctdisk > disktacct
```

acctdisk has no options and is normally invoked by *dodisk*.

dodisk

dodisk is normally invoked by *cron* to create disc total accounting records for daily usage by System Accounting. The syntax for *dodisk* is:

`dodisk` [-*o*] [*files* ...]

In the default case, *dodisk* creates disc total accounting records on the special files whose names are stored in **/etc/checklist**: the special file names are supplied as input to *diskusg*, which pipes its output to *acctdisk*, which in turn creates total accounting records.

If the `-o` option is used, *dodisk* creates total accounting records more slowly by using *acctdusg* instead of *diskusg*.

If *files* are used, disc accounting will be done on these file systems only. If the `-o` option is used, then *files* should be mount points of mounted file systems; if omitted, they should be the special file names of mountable filesystems.

NOTE

See the "Daily Usage and Installation" section of this chapter for more information on how *dodisk* should be invoked by *cron*.

It is possible for malicious users to defeat disc space accounting by giving their files away to other users with *chown(2)* or *chown(1)* (by default, all users can execute them). To avoid this, take away the ability to use these commands from some or all users with the *setprivgrp(1M)* command. To let only the super-user use the change-ownership abilities, add the following line to */etc/rc*:

```
setprivgrp -n CHOWN
```

To let one or more groups of users use the change-ownership abilities, add a line for each group to */etc/rc*, similar to the following:

`setprivgrp` *group_name* `CHOWN`

NOTE

Taking away the change-ownership ability may cause problems when running some commands or applications.

Connect Session Accounting

Whenever a user logs in or out of HP-UX, the program *login* records the connect session in the file */etc/wtmp*. Records in *wtmp* contain the following information:

- the terminal name on which the connect session occurred,
- the login name of the user,
- the current time/date at login or logout, and
- other status information (see *utmp(5)* for details).

System Accounting provides commands that allow you to write records to *wtmp*, to display and manipulate *wtmp*, and to create total accounting records from *wtmp*. These commands are covered in this section.

Writing Records to wtmp – acctwtmp

The command *acctwtmp* allows you to write records to *wtmp* for whatever reason you might have. *acctwtmp* is normally invoked by *startup* and *shutacct* to record when System Accounting was turned on and off respectively. The format of the command is:

`acctwtmp` "*reason*"

where *reason* is string describing the reason for writing the record to *wtmp*. **Note that *acctwtmp*** does not directly write records to *wtmp*: it writes a record containing the terminal name, current time, and reason string to standard output. To actually write the record to *wtmp* you must append the output from *acctwtmp* to the *wtmp* file as follows:

`acctwtmp` "*reason*" >> */etc/wtmp*

The *reason* string may be any combination of letters, numbers, spaces, and the dollar sign ($), but may not exceed 11 characters in length. (*reason* must be enclosed in double quotes as shown.)

Displaying Connect Session Records - fwtmp

To display the contents of *wtmp*, you can use the command *fwtmp*. When no options are used, *fwtmp* takes from standard input records of the format contained in *wtmp*; it writes to standard output the ASCII readable equivalent of the input records. The output of this command can either:

1. be edited, via a HP-UX editor such as *vi(1)*, and then rewritten to *wtmp* using special *fwtmp* options described below; or
2. supplied as input to commands which convert the information to total accounting records.

The syntax of *fwtmp* is:

`fwtmp` [*-ic*]

The options can be used in any combination. The following table describes what the different combinations do.

Option	Description
`-ic`	Denotes that input is in ASCII readable form and is to be converted to binary form. This is essentially the opposite of using *fwtmp* without any options.
`-i`	Both input and output are in ASCII readable format. This is the same as performing an ASCII to ASCII copy.
`-c`	Both input and output are in binary format—a binary to binary copy.

The following example shows the output produced by *fwtmp* and is followed by descriptions of each column in the report:

```
$ fwtmp < /etc/wtmp
                  system boot    0  2 0000 0000 479472540 Mar 12 03:49:00 1985
root         co console          0  7 0000 0000 479475173 Mar 12 04:32:53 1985
                  acctg on       0  9 0000 0000 479493135 Mar 12 09:32:15 1985
mike         a1 ttya1          352  7 0000 0000 479493590 Mar 12 09:40:00 1985
mike         a1 ttya1          352  8 0011 0000 479496000 Mar 12 10:20:00 1985
sarah        07 tty07          353  7 0000 0000 479518335 Mar 12 16:32:15 1985
bill         10 tty10          351  7 0000 0000 479521475 Mar 12 17:24:35 1985
sarah        07 tty07          353  8 0011 0000 479522478 Mar 12 17:41:18 1985
bill         10 tty10          351  8 0011 0000 479526487 Mar 12 18:48:07 1985
             co console          0  8 0011 0000 479526488 Mar 12 18:48:08 1985
                  acctg off      0  9 0000 0000 479526493 Mar 12 18:48:13 1985
                  system boot    0  2 0000 0000 479389800 Mar 12 05:00:00 1985
```

Column	Description
1	The login name of the user who logged in or out.
2	*/etc/inittab* id (this is usually the number of the line on which the connect session took place).
3	The name of the device on which the connect session occured.
4	Process id of the user who logged in or out.
5	Entry type. This field contains information on the type of record—for example, it shows whether the record is a login record (entry type=7), logout record (entry type=8), or if the record was written by *acctwtmp* (entry type=9). See *utmp(5)* for more details on this field.
6—7	Exit status for connect session. See *login(1)* and *utmp(5)* for details.
8	Time that entry was made (in elapsed seconds since January 1, 1970).
9—12	The equivalent of column 8 in date/time format showing month, day, time of day (in 24-hour format), and year.

Fixing wtmp Errors – wtmpfix

When a user logs into HP-UX, the *login* program stores the value seven (7) in the entry type field of the connect session record. When the same user logs out, an entry type of eight (8) is recorded. You can see this by examining the sample output created by *fwtmp* in the previous section. Note that in the example, login records preceed their corresponding logout records in chronological order.

Occasionally, this time-stamped ordering becomes inconsistent: logout records might preceed login records. (This occurs when the date and time are reset while users are still logged in.) When this happens, the commands that create connect session total accounting records will not work properly.

Fortunately, there is a command that fixes corrupted *wtmp* files: *wtmpfix*. *wtmpfix* takes as input *wtmp* binary records and corrects the time/date stamps to be consistent; its standard output is also binary *wtmp* records. Its syntax is:

`wtmpfix` [*files*]

If *files* is given, then input is taken from *files*. A dash (-) can be used in place of *files* to indicate standard input. **Note that if you specify `wtmp` as both input to and output from this command, `wtmp` will be destroyed.** Therefore, take care not to destroy *wtmp*. The following shows how to properly fix *wtmp* using *wtmpfix*:

```
$ wtmpfix /etc/wtmp > wtmp.temp
$ fwtmp -c < wtmp.temp > /etc/wtmp
$ rm wtmp.temp
```

Creating Total Accounting Records

This final set of connect session accounting commands is used to create connect session total accounting records. Before reading any further, you may want to review Figure 6-1 (in the "System Data Flow" section).

acctcon1

acctcon1 converts a sequence of login/logoff records (of the format contained in *wtmp*) read from its standard input to a sequence of records, one per login session. Its input is normally redirected from *wtmp*; its output is columnar ASCII and can be supplied as input to *prctmp* or *acctcon2*.

The use of *acctcon1* is illustrated below by first displaying the contents of *wtmp* with *fwtmp*, and then using *acctcon1* to create connect session summary file. The columnar data produced by *acctcon1* is described after the report.

```
$ fwtmp < /etc/wtmp
               system boot    0  2 0000 0000 479472540 Mar 12 03:49:00 1985
root        co console        0  7 0000 0000 479475173 Mar 12 04:32:53 1985
               acctg on       0  9 0000 0000 479493135 Mar 12 09:32:15 1985
mike        a1 ttya1        352  7 0000 0000 479493590 Mar 12 09:40:00 1985
mike        a1 ttya1        352  8 0011 0000 479496000 Mar 12 10:20:00 1985
sarah       07 tty07        353  7 0000 0000 479518335 Mar 12 16:32:15 1985
bill        10 tty10        351  7 0000 0000 479521475 Mar 12 17:24:35 1985
sarah       07 tty07        353  8 0011 0000 479522478 Mar 12 17:41:18 1985
bill        10 tty10        351  8 0011 0000 479526487 Mar 12 18:48:07 1985
            co console        0  8 0011 0000 479526488 Mar 12 18:48:08 1985
               acctg off      0  9 0000 0000 479526493 Mar 12 18:48:13 1985
$ acctcon1 < /etc/wtmp
520095488   353    sarah   1665    2478    479518335   Tue Mar 12 16:32:15 1985
521012224   352    mike                    479493590   Tue Mar 12 09:40:00 1985
520095488   351    bill    0       5012    479521475   Tue Mar 12 17:24:35 1985
521011712   0      root    41047   6488    479475173   Tue Mar 12 04:32:53 1985
```

Descriptions of the columnar data produced by *acctcon1* follow:

Column	Description
1	Shows the device address (in decimal equivalent of major/minor device address) at which the connect session occurred.
2	Gives the user ID for the connect session record.
3	Displays the login name for the user.
4	Shows the number of prime connect time seconds that were used during the connect session.
5	Shows non-prime connect seconds.
6	The connect session starting time (in seconds elapsed since January 1, 1970) is displayed here.
7-11	The remaining columns convert column six to date/time format.

In addition to its normal usage, *acctcon1* has four options:

Option	Description
`-p`	This option tells *acctcon1* not to produce one record per connect session. Instead, *acctcon1* simply echos its input—one line per *wtmp* record—showing line name, login name, and time (in both seconds and day/time format). Using this option is similar to using *fwtmp*, except that this option doesn't show status information, whereas *fwtmp* does.
`-t`	*acctcon1* maintains a list of lines on which users are logged in. When it reaches the end of its input, it emits a session record for each line that still appears to be active. It normally assumes that its input is a current file, so that it uses the current time as the ending time for each session in progress. The `-t` flag causes it to use, instead, the last time found in its input, thus assuring reasonable and repeatable numbers for non-current files.
`-l` *file*	This option causes a line usage summary report to be placed in *file*. This report shows each line's name, number of minutes used, percentage of total elapsed time used, number of sessions charged, number of logins, and number of logins and logoffs. This report can be used to keep track of line usage, identify bad lines, and find software/hardware oddities. **Note that hang-up, termination of *login(1)*, and termination of the login shell each generate logoff records; therefore, the number of logoffs is often three to four times the number of connect sessions.** Shown below is an example of the line use file (`line_use`) created from the same *wtmp* file used in the previous *acctcon1* example; the standard output of *acctcon1* has been redirected into the file *ctmp*.
`-o` *file*	Using the `-o` option (e.g., `acctcon1 -o f_overall`) causes *file* to be filled with an overall record for the accounting period, giving starting time, ending time, number of reboots, and number of date changes.

```
$ acctcon1 -t -l line_use < /etc/wtmp > ctmp
$ cat line_use
TOTAL DURATION IS 899 MINUTES
LINE    MINUTES PERCENT # SESS  # ON    # OFF
console 856     95      1       1       1
tty07   69      8       1       1       1
ttya1   40      4       1       1       1
tty10   84      9       1       1       1
TOTALS  1049    --      4       4       4
```

prctmp

The *prctmp* command is simple. Its only function is to put headings on the output created by *acctcon1*: *prctmp* makes a readable report from the output of *acctcon1*.

prctmp takes its input from standard input; therefore, to create a *prctmp* report from *acctcon1* information, you can simply pipe the output from *acctcon1* into *prctmp* as follows:

```
$ acctcon1 < /etc/wtmp | prctmp
```

prctmp will respond by generating a report with appropriate headings over the columns of output from *acctcon1*.

acctcon2

acctcon2 creates connect session total accounting records from standard input of the format created by *acctcon1*. In other words, to create connect session total accounting records, simply send the output from *acctcon1* into the input of *acctcon2*.

The total accounting records created by *acctcon2* are sent to standard output. So if you want to store these records, you must redirect standard output. The following command line shows how to write total accounting records from the connect session record file (*wtmp*) into the file `ctacct`:

```
$ acctcon1 < /etc/wtmp | acctcon2 > ctacct
```

Process Accounting

Process accounting commands provide the means to accumulate execution statistics—such as memory usage, CPU time, number of input/output transfers—for individual processes. This section describes how to:

1. Turn process accounting on,
2. Turn process accounting off,
3. Make sure that the process accounting file (*pacct*) doesn't become too large,
4. Display process accounting records,
5. Generate a command summary report, and
6. Create total accounting records from the process accounting file.

You might find it helpful to look at the System Data Flow Diagram (Figure 6-1) when reading this section.

Turning Process Accounting On

Before System Accounting can generate process accounting data, process accounting must be turned on; two commands can be used to accomplish this task: *turnacct on* and *accton*. After process accounting has been turned on, the kernel will write a process accounting record, for every terminating process, into the current process accounting file (*pacct* by default).

NOTE

The *startup* command, placed in the system initlization shell script */etc/rc*, automatically turns process accounting on. Therefore, if you have updated */etc/rc* for System Accounting (as described in the section "How to Install System Accounting"), process accounting will automatically be activated, and you should seldom need to use the commands described here.

These commands are described only for your benefit should you ever need to manually turn process accounting on or off.

---

turnacct on

The command used most often to activate accounting is *turnacct on*; only the super-user and the *adm* login can execute this command. *turnacct on* assumes that the process accounting file is the default file *pacct*. The action of *turnacct on* can be summarized as follows:

1. Check to see if the process accounting file *pacct* exists.
2. If *pacct* doesn't exist, then create a new *pacct* file.
3. Turn process accounting on by invoking *accton* with the filename argument *pacct*.

To execute this command, simply enter `turnacct on` to the HP-UX prompt. Note that only the adm login and the super-user can execute this command.

accton

Again, only the super-user and the *adm* login can execute *accton*. When invoked with a filename argument, *accton* turns on process accounting and makes the specified filename the current process accounting file. For example,

```
$ accton /usr/adm/pacct
```

tells the kernel to start writing process accounting records to the file `/usr/adm/pacct`. The next example would activate process accounting and make the current process accounting file `/usr/adm/XX107`:

```
$ accton /usr/adm/XX107
```

NOTE

You must make sure that the filename you specify is an existing file; otherwise, *accton* will fail.

Note that in the System Data Flow Diagram (Figure 6-1), *accton* is shown calling another routine, *acct(2)*. *acct(2)* is the system call that actually tells the kernel to start writing process accounting records. See the *HP-UX Reference* for more details on *acct(2)*.

Turning Process Accounting Off

Two commands are used to turn process accounting off: *turnacct off* and *accton* (with no filename argument). These commands tell the kernel to stop writing records to the current process accounting file.

NOTE

If you have updated the */etc/shutdown* shell script as described in the section "How to Install System Accounting," you may seldom ever use these commands. The reason is that the *shutacct* command, added to */etc/shutdown*, automatically turns process accounting off.

---

turnacct off

turnacct off can be executed by only the super-user and the *adm* login. *turnacct off* turns process accounting off by invoking the *accton* command without the optional filename argument. You execute this command by typing:

```
$ turnacct off
```

accton

When *accton* is invoked without the optional filename argument, process accounting is turned off. You would enter this command as:

```
$ accton
```

As shown in the System Data Flow Diagram (Figure 6-1), *accton* tells the kernel to stop writing process accounting records by using the system call *acct(2)*.

Checking the Size of pacct

On a multi-user system, many processes can execute during a single hour. Therefore, process accounting files have the potential to become quite large. System Accounting has built-in mechanisms that insure that the default process accounting file *pacct* doesn't become too large. The two commands used for this purpose are: *turnacct switch* and *ckpacct*.

NOTE

The commands described here work only on the default process accounting file, *pacct*.

---

ckpacct

The command *ckpacct* is normally invoked by *cron* every hour to insure that the current process accounting file *pacct* hasn't become to large. The format of *ckpacct* is:

`ckpacct` [*blocks*]

If the size of *pacct* exceeds the *blocks* argument, 1 000 by default if *blocks* is not specified, then *turnacct switch* is executed, which renames the current *pacct* file and creates a new *pacct* file.

NOTE

If the amount of free space falls below a certain threshold, *ckpacct* will automatically turn off process accounting via *turnacct off*. On 5.0 HP-UX, accounting is turned off when free space falls below 2%, and back on when it goes above 4%. On 5.1 HP-UX these percentages are can be configured using the *acctresume* and *acctsuspend* system parameters (See "Configuring HP-UX" in the "System Administrator's Toolbox"). These percentages are in addition to the *minfree* attribute. When free space goes over the specified percentage, process accounting will be reactivated.

The kernel may also enforce a size limit on the size of *pacct*. This will take precedence over the limit set by *ckpacct*. See *acctsh(1M)* and *acct(2)* in the *HP-UX Reference Manual* for more details.

---

turnacct switch

turnacct switch is used to create a new *pacct* file when the current *pacct* file is too large. The action of *turnacct switch* can be summarized as follows:

1. Process accounting is temporarily turned off.
2. The current *pacct* file is renamed to *pacct***incr**, where **incr** is a number starting at **1** and incrementing by one for each additional *pacct* file that is created via *turnacct switch*.
3. Since the old *pacct* file was renamed to *pacct***incr**, a new, current *pacct* file is created.
4. Process accounting is restarted; the kernel starts writing records to the newly created *pacct* file.

The example below illustrates the effect of using the *turnacct switch* command. In the example, *turnacct switch* is executed from the **adm** home directory */usr/adm*. Comment lines begin with a cross-hatch(#) and are included in the example only as explanatory material.

```
$ #
$ # First, list all the process accounting files
$ # (at this point, there is only one).
$ #
$ ll pacct*
-rw-rw-r--    1 adm     adm      2196 Mar 21 12:44 pacct
$ #
$ # Now execute turnacct switch, which will rename the current
$ # pacct file to pacct1 and will create a new pacct file.
$ #
$ turnacct switch
$ #
$ # Now verify this by listing all process accounting
$ # files again.
$ #
$ ll pacct*
-rw-rw-r--    1 adm     adm        72 Mar 21 12:46 pacct
-rw-rw-r--    1 adm     adm      2196 Mar 21 12:44 pacct1
$ #
$ # The current process accounting file is pacct.  The previous
$ # process accounting file is now named pacct1.
$ #
```

Displaying Process Accounting Records - acctcom

The *acctcom* command allows you to display records from any file containing process accounting records. Normally you would use this command to display records from the *pacct* files (*pacct*, *pacct1*, *pacct2* ...).

acctcom is a very versatile command; its syntax follows:

`acctcom` [[*options*][*file*]] ...

If no *file* is specified, *acctcom* uses the current *pacct* file as input. Input can also be taken from standard input. Some of *acctcom*'s options allow you to select only the records that you want to see; other options control the format of the report.

The information contained in this section is organized as follows:

- First, definitions are given for the columnar data produced by *acctcom*.
- Command options that control the format of the report are discussed.
- Options that allow you to select particular records are described.
- Finally, to help you understand how to use *acctcom*'s options, sample *acctcom* reports are shown.

Definitions of Information Produced by acctcom

acctcom generates a columnar report with descriptive headings on each column. Each line of the report represents the execution statistics that a particular process accumulated during its lifetime. The following table defines the standard columns in the report—i.e., the columns that are displayed when none of *acctcom*'s options are specified.

Column Header	Definition
`COMMAND NAME`	The name of the command or program that spawned the process is shown here. Whenever you enter a command, you are spawning a process. For example, if you enter the command `$ ll /usr/lib/acct` you are creating a process with the command name `ll`. If a command requiring super-user privileges is executed, a # appears before the command name.
`USER`	The login name of the user who created the process is displayed here.
`TTYNAME`	This is the name of the terminal from which the process was executed. If the process was not executed from a known terminal (for example, if it was executed via *cron*), then a question mark(?) appears in this column.
`START TIME`	The time that the process began executing (in *hh:mm:ss* format) is displayed here.
`END TIME`	This is the time (*hh:mm:ss*) that the process finished executing.
`REAL (SECS)`	The number of seconds that elapsed from `START TIME` to `END TIME` is shown in this column.
`CPU (SECS)`	This column shows how much of the CPU's time a process used during its execution.
`MEAN SIZE(K)`	This is a rough estimate (in kilobytes) of the amount of memory that a process used during execution. This estimate is determined from the current process's memory usage at each system clock interrupt. It is, therefore, subject to statistical sampling errors. Only the memory resident pages of a process are counted; no pages in the swap space are counted. Shared code and data is divided among the processes using it. The size is divided by the number of processes sharing the code or data.

The table below defines the columns that are not displayed on the standard report, but which can be displayed by using *acctcom* options.

Column Header	Definition
F	For a process created by *fork* which does not do an *exec*, this column takes the value **1**; commands which require super-user privileges show a **2**; super-user commands which do a *fork* without an *exec* show a **3**; otherwise, this column shows a **0**.
STAT	This column displays the system exit status. (This is **not** the status returned by *exit(2)* to a parent process during *wait*). When a process terminates normally, this field shows a **0**. If a command terminates abnormally, then a value other than zero is shown. For example, if you interrupt a command with the DEL key, this column will contain a **2**.
HOG FACTOR	The hog factor is computed as the CPU time divided by REAL time; it provides a relative measure of the available CPU time used by the process during its execution. For example, a hog factor of less than 0.50 indicates that the process spent less than half of its time using the CPU. A hog factor of 0.75 indicates that a process spent 75% of its time using the CPU.
KCORE MIN	Provides a combined measurement of the amount of memory used (in kilobytes) and the length of time it was used (in minutes). It is computed as follows: `KCORE MIN = CPU (SECS) * MEAN SIZE(K) / 60`
CPU SYS	This is the portion of total CPU time that was spent executing operating system code, such as system calls (for example, writing to disc).
USER (SECS)	This is the remaining portion of CPU time. User CPU time is the amount of time actually spent executing a process's code (rather than system code).
CPU FACTOR	Whenever you execute a command, the CPU spends part of its time actually executing the command's code (user CPU time) and spends the rest of its time performing system functions, such as writing to the disc or terminal (system CPU time). That is, total CPU time is comprised of both *system* and *user* CPU time: `CPU (SECS) = CPU SYS + USER (SECS)` The CPU factor shows the ratio of user CPU time to total CPU time: `CPU FACTOR = USER (SECS) / (CPU SYS + USER (SECS))` For example, if a command has a CPU factor of 0.35, that means that the CPU spent 35% of its time executing user code and 65% performing system functions.

CHARS TRNSFD — The number of characters (bytes) read and/or written by the command is displayed in this column.

BLOCKS R/W — This column shows the number of file system blocks read and/or written as a result of executing this command. **This number is not directly related to CHARS TRNSFD and may vary each time the command is executed,** because BLOCKS R/W is affected by directory searches made before opening files, other processes accessing the same files, and general file system activity.

Report Format Options

When no report format options are specified, *acctcom* will produce a report containing only the default information. Optional information can be displayed only by using the report format options. Definitions of the report format options follow:

Option	Description
-a	Causes average statistics to be displayed at the end of the report. The following information is shown: • total number of commands processed (cmds=xxx) • average real time per process (Real=x.xx) • average CPU time per process (CPU=x.xx) • average USER CPU time per process (USER=x.xx) • average SYS CPU time per process (SYS=x.xx) • average characters transfered (CHARS=x.xx) • average blocks transferred (BLK=x.xx) • average CPU factor (USR/TOT=x.xx) • average HOG factor (HOG=x.xx)
-b	Using this option will display the process records in reverse order: most recently executed commands will be shown first.
-f	Prints the *fork/exec* flag (F column) and process exit status (STAT column) on the report.

-h Causes the optional HOG FACTOR column to be displayed, instead of the standard mean memory size column MEAN SIZE(K).

-i The optional I/O counts—CHARS TRNSFD and BLOCKS R/W—replace the standard MEAN SIZE(K) column in the report.

-k Replace the standard MEAN SIZE(K) column with KCORE MIN.

-m Show the default column MEAN SIZE(K) on the report. This option is used to include MEAN SIZE(K) when it has been bumped off by another option. For example,

```
$ acctcom -km
```

produces a report showing both KCORE MIN and MEAN SIZE(K).

-r Include the optional CPU FACTOR column in the report.

-t Show separate system and user CPU times (CPU SYS and USER (SECS) respectively).

-v Using this option will suppress the printing of column headings at the top of the report.

-q This option is the same as the -a option, except that individual process accounting records are not displayed—only the averages are displayed.

-o *ofile* Copy the input process accounting records to *ofile*.

Record Selection Options

The options described here allow you to select the records that are included in the report produced by *acctcom*. The table shown below defines and provides examples for each option:

Option **Description**

-l *line* Display only the processes that were executed from the user terminal **/dev/***line*. For example,

```
$ acctcom console
```

would display records only for the processes that were created from the terminal console.

-u *user* Show only the processes belonging to *user*. *user* can be any of the following:

- a user ID (for example, `acctcom -u 355`)
- user name (`acctcom -u horatio`)
- a cross-hatch # (`acctcom -u#`)
- a question mark ? (`acctcom -u?`)

If # is specified as the user name, then only the commands that require super-user priveleges will be displayed by acctcom. If **?** is given as the user, then only the processes with unknown process IDs will be displayed.

As an example, the following two commands are equivalent:

```
$ acctcom -u 0
$ acctcom -u root
```

-g *group* Show only the processes belonging to *group*. *group* may be specified as either a group name or group ID. For example, suppose the group *pseudo* with group ID *300* is defined in */etc/group*; then the following two commands are equivalent:

```
$ acctcom -g 300
$ acctcom -g pseudo
```

-s *time* Select processes existing **at or after** *time*. Time is given in 24-hour format—*hr*[:*min*[:*sec*]]. The following example would display all the processes that existed at or after 3:30pm:

```
$ acctcom -s 15:30
```

-e *time* Select processes that existed **at or before** *time*. Time is supplied in 24-hour format *hr*[:*min*[:*sec*]]. The next example would display all the processes that existed between midnight and 12:15am:

```
$ acctcom -e 0:15
```

-S *time* Select processes **starting** at or after time where *time* is in 24-hour format. The following example would display all the processes that **started** at 1:30:42pm or after:

```
$ acctcom -S 13:30:42
```

`-E` *time* — Display only the processes that **terminated** at or before *time*, where time is in 24-hour format *hr*:[*min*[:*sec*]]. Note both the `-S` and `-E` options with the same *time* argument will cause *acctcom* to display only the processes that existed at the specified *time*. For example, to see all the processes that existed at exactly 30 minutes past noon, you would enter:

```
$ acctcom -S 12:30 -E 12:30
```

`-n` *pattern* — Show only the commands matching *pattern*. *pattern* can be a regular expression as described in *ed(1)*, except that + means one or more occurrences. For example, to display all processes that were created by executing the *ls* command, you would enter:

```
$ acctcom -n ls
```

To display all the commands that start with *acct*, enter:

```
$ acctcom -n acct
```

To see all the commands that contain the letter **m** in their spelling, you would type:

```
$ acctcom -n .*m.*
```

`-H` *factor* — Display only those processes whose hog factor exceeds *factor*. For example,

```
$ acctcom -H 0.85
```

would display all the processes that spent over 85% of their execution time in the CPU. You can use this option to find greedy processes—processes that are hogging the CPU.

`-O` *time* — Show only those processes whose system CPU time exceeds *time*, specified in seconds. The following example would be used to determine which processes took more than 8.25 seconds of operating system CPU time to execute:

```
$ acctcom -O 8.25
```

This option could be used to determine which processes are making heavy use of the operating system calls.

`-C` *sec* — Show only the processes whose total CPU time (`SYS + USER`) exceeds *sec* seconds. The next example would display all the processes that used over 5.28 seconds of CPU time to execute:

```
$ acctcom -C 5.28
```

-I *chars* Display only the processes tranferring more characters than the limit given by *chars*. For example,

```
$ acctcom -I 10240
```

will display all the processes that transferred over ten kilobytes of characters (10 240 = 10 × 1 024 bytes).

Sample Reports

The following sample report illustrates the use of *acctcom* without any options. The report generated is the standard report produced when no options are specified.

```
$ acctcom
ACCOUNTING RECORDS FROM:  Thu Mar 21 12:52:26 1985
COMMAND                      START    END          REAL     CPU    MEAN
NAME       USER     TTYNAME  TIME     TIME        (SECS)  (SECS) SIZE(K)
#accton    root     console  12:52:26 12:52:26      0.12    0.10   19.00
ls         sarah    tty07    14:04:08 14:04:08      0.28    0.23   16.50
ckpacct    adm      ?        14:30:00 14:30:05      5.13    1.45   24.00
pwd        bill     tty10    15:09:07 15:09:07      0.48    0.22   22.50
find       sarah    tty07    18:51:37 18:51:39      2.73    0.15   26.50
tabs       root     console  19:10:18 19:10:18      0.92    0.13   23.50
stty       root     console  19:10:19 19:10:19      0.88    0.08   26.00
mail       bill     tty10    19:10:21 19:10:22      1.78    0.23   28.50
news       root     console  19:10:23 19:10:23      0.73    0.12   23.00
acctcom    adm      ttya0    19:53:16 19:53:38     22.58    2.55   28.50
```

Now display all the processes created between 7:00pm and 7:30pm by the user *root*. In addition, include the optional CPU factor and average statistics in the output.

```
$ acctcom -S 19:00 -E 19:30 -u root -ah
START AFT: Thu Mar 21 19:00:00 1985
END BEFOR: Thu Mar 21 19:30:00 1985
COMMAND                      START    END          REAL     CPU     HOG
NAME       USER     TTYNAME  TIME     TIME        (SECS)  (SECS)  FACTOR
tabs       root     console  19:10:18 19:10:18      0.92    0.13    0.14
stty       root     console  19:10:19 19:10:19      0.88    0.08    0.09
news       root     console  19:10:23 19:10:23      0.73    0.12    0.16
cmds=3 Real=0.84   CPU=0.11   USER=0.02    SYS=0.09   CHAR=26.12   BLK=11.50
USR/TOT=0.19 HOG=0.13
```

Sample reports are helpful, but the best way to learn the various *acctcom* options is to use them. Take a few minutes to experiment with this command; it is very powerful and can provide you with much useful information if used properly.

Command Summary Report - acctcms

The *acctcms* command takes process accounting records as input; but instead of reporting on the individual processes, *acctcms* generates a report on the commands that generated the process accounting records. The action of *acctcms* can be summarized as follows:

1. *acctcms* looks through the input process accounting records and accumulates execution statistics for each unique command name. This information is stored in internal summary format—one record per command name.
2. Depending on the *acctcms* options used, the command summary records created in step 1 are sorted.
3. The command summary records are written to standard output in the internal summary format mentioned in step 1. (To get an ASCII, readable report of this information, you would use the -a option described later.)

The syntax of the *acctcms* command is:

`acctcms` [*options*] *files*

where *files* is a list of the input process accounting files for which the command summary report is to be generated.

Producing a Readable Report - the -a option

By default, the output of *acctcms* is in internal summary record format; if you display it to your terminal, all you see is gibberish. To get a human-readable report, use the -a option.

The -a option causes *acctcms* to produce a report with descriptive column headings. Total and average (mean) execution statistics for each command are displayed—one line per command—along with total and average statistics over all commands in the report. Descriptions of the columnar data produced by *acctcms* follow:

Column Header	Description
COMMAND NAME	The name of the command for which execution statistics are summarized. Unfortunately, all shell procedures are lumped together under the name **sh**, because only object modules are reported by the process accounting system.
NUMBER CMDS	The total number of times that the command was invoked.
TOTAL KCOREMIN	The total amount of kcore minutes accumulated for the command. (See the "Definitions of Information Produced by acctcom" for a more accurate description of kcore minutes.)
TOTAL CPU-MIN	The total CPU time that the named command has accumulated.
TOTAL REAL-MIN	Total accumulated real time seconds are displayed in this column.
MEAN SIZE-K	The average amount of memory (in kilobytes) consumed by the command.
MEAN CPU-MIN	The average CPU time consumed per command invocation is shown here; the following equation shows how it is computed: `MEAN CPU-MIN = TOTAL CPU-MIN / NUMBER CMDS`
HOG FACTOR	This column shows the average hog factor over all invocations of the command. It is computed as: `HOG FACTOR = TOTAL CPU-MIN / TOTAL REAL-MIN`
CHARS TRNSFD	This column shows the total number of characters transferred by the command. Note that this number may sometimes be negative.
BLOCKS READ	A total count of the physical blocks read and written by the given command. (See the section "Displaying Process Accounting Records — acctcom" for details on the significance of this total.)

NOTE

When only the -a option is specified, the report is sorted in descending order on the TOTAL KCOREMIN column: commands using more TOTAL KCOREMIN are shown before those using fewer TOTAL KCOREMIN. This report gives a relative measure of the amount of memory used over time by the various commands: commands toward the start of the report are making more use of memory resources than are commands toward the end of the report.

Other Options

In addition to the -a option, several other options can be used to control the format of the report generated by *acctcms*. Some options specify which field to sort the report on; other options control the printing of prime/non-prime time usage. The following table defines these options and illustrates their use.

Option	Description
-c	Sort the commands in descending order on TOTAL CPU-MIN, rather than the default TOTAL KCOREMIN. This report can be used to determine which commands are using most of the computer's CPU time.
-n	Causes the report to be sorted in descending order on the column named NUMBER CMDS. Commands toward the start of this report are the ones used most frequently; commands toward the end are used least often.
-j	All commands invoked only once are combined on one line of the report; this line is denoted by having "***other" in the COMMAND NAME column. This option is useful for shortening a report that has many one-invocation commands.
-o	**Used only with the -a option**, -o causes the report to be generated only for commands that were executed during non-prime time (as specified in the *holidays* file). You can use this option to get a non-prime time command summary report.
-p	**Also used only with the -a option**, -o elicits a report only only for commands that were executed during prime time (as specified in *holidays*). This option is used to get a prime time command summary report.
-apo	When the options -o and -p are used together with -a, a combination prime and non-prime time report is produced. The output of this report is same as that produced by -a alone, except that the NUMBER CMDS, TOTAL CPU-MIN, and TOTAL REAL-MIN columns are divided into two columns—one for prime time totals, the other for non-prime time. (Prime time columns have a (P) header, while non-prime time columns are headed by (NP).)
-s	Specifies that any named input files following the -s on the command line are already in internal summary format. This option is useful for merging previous *acctcms* reports with current reports. The following example uses -s to create a command summary report from previous process accounting files (pacct?) and the current process accounting file (pacct). The final ASCII report is stored in the file ascii_summary.

```
$ acctcms pacct? > old_summary
$ acctcms pacct > new_summary
$ acctcms -as old_summary new_summary > ascii_summary
```

Sample Report

The ASCII reports produced by *acctcms* contain more than 80 characters per line. When these reports are displayed at an 80-column terminal, the lines wrap around on the screen. In addition, if the report is printed on an 80-column printer, some of the rightmost columns will be lost. Therefore, be sure to use either:

1. a printer with compressed print capabilities, so that all of the report will fit on standard computer paper; or

2. a printer with enough columns to display all the information—for example, a 132-column printer.

The following example generates a command summary report for the current process accounting file (no file is specified, so the current *pacct* file is assumed). By giving the `-j` option, all the commands that were executed only once are grouped under the command name `***other`. Note also that total execution statistics for all commands are grouped under the command name `TOTALS`.

```
$ acctcms -aj
                              TOTAL COMMAND SUMMARY
COMMAND NUMBER    TOTAL    TOTAL     TOTAL  MEAN     MEAN     HOG    CHARS BLOCKS
NAME      CMDS KCOREMIN  CPU-MIN REAL-MIN SIZE-K  CPU-MIN  FACTOR TRNSFD  READ

TOTALS      61    17.63   0.38     164.49  46.25    0.01     0.00 104553   1027

acctcms     17    12.13   0.16       0.35  76.72    0.01     0.45  49192    306
sh           8     2.43   0.09     152.86  26.79    0.01     0.00   9043    163
more         3     0.73   0.02      10.50  31.00    0.01     0.00  21618     83
ll           6     0.61   0.04       0.11  16.50    0.01     0.33   5715     95
acctcom      4     0.58   0.02       0.07  28.50    0.01     0.30  15319     42
***other     9     0.54   0.02       0.14  25.26    0.00     0.16    459    161
cat          4     0.19   0.01       0.35  22.97    0.00     0.02   3112     52
rm           2     0.11   0.00       0.02  22.22    0.00     0.29      0     29
chmod        2     0.10   0.00       0.01  22.00    0.00     0.35      0     15
accton       2     0.08   0.00       0.02  19.00    0.00     0.29      0     22
sed          2     0.08   0.01       0.04  14.50    0.00     0.13     73     38
echo         2     0.05   0.00       0.02  20.00    0.00     0.16     22     21
```

Creating Total Accounting Records

Two commands—*acctprc1* and *acctprc2*—are used to create total accounting records from the process accounting files. The output from *acctprc1* is supplied as input to *acctprc2* which produces the total accounting records. These commands are normally invoked by *runacct* to produce daily accounting information.

acctprc1

This command reads process accounting records from standard input, adds login names corresponding to the user ID of each record, and then writes for each process an ASCII line showing:

- the ID of the user that created the process
- the user's login name
- prime CPU time in ticks (a "tick" is one fiftieth of a second)
- non-prime CPU time, also in ticks
- mean memory size (in pages—4 Kbytes on Series 300, 2 Kbytes on Series 200)

The format of *acctprc1* is:

`acctprc1` [*ctmp*]

Input must be redirected from a process accounting file.

The following example creates a file, `ascii_ptacct`, containing ASCII process accounting information that can be used to create process total accounting records. This file is created from the current process accounting file *pacct*.

```
$ acctprc1 <pacct >ascii_ptacct
```

Normally, *acctprc1* gets login names from the password file *passwd*, which is sufficient on systems where each user has a unique user ID. However, on systems where different users share the same user ID, the *ctmp* file should be specified; it helps *acctprc1* distinguish different login names that share the same user ID.

When specified, *ctmp* is expected to contain a list of login sessions of the form created by *acctcon1*, sorted by user ID and login name.

acctprc2

This command reads from standard input records of the form created by *acctprc1*; it then summarizes the records by user ID and name, and then writes the sorted summaries to standard output as total accounting records. The following example creates total accounting records for all processes in the current process accounting file *pacct*; the total accounting records are stored in the file `ptacct`.

```
$ acctprc1 <pacct |acctprc2 >ptacct
```

Charging Fees to Users – chargefee

System Accounting provides the capability to charge fees to specific users; the *chargefee* command is used to accomplish this task. *chargefee* allows you to charge generic *units* to a specific login name. The syntax of this command is:

`chargefee` *login_name number*

where *number* is the number of units to be charged to a particular user, and *login_name* is the login name of the user to whom *number* units are to be charged.

NOTE

number can be any whole number in the range -32 768 to 32 767; when charging fees, keep in mind that the sum of each user's fees must also be within this range.

chargefee accumulates fee charge records in the file */usr/adm/fee*. These records are then merged with other accounting records via *runacct*.

Examples

The following example charges 25 units to the user whose login name is `horatio`:

```
$ chargefee horatio 25
```

Suppose you inadvertently charged 247 units to the user named `zimblits`, and you want to return his charges to their original value. You would enter the following:

```
$ chargefee zimblits -247
```

Summarizing and Reporting Accounting Information

This final group of commands summarizes and reports accounting information. Certain commands display and merge total accounting files, while others generate the daily and monthly reports used to analyze system performance and bill users for resource usage. The following commands are discussed in this section:

- *prtacct*—displays total accounting records
- *acctmerg*—merges total accounting files
- *runacct*—generates daily summary files and reports
- *prdaily*—displays the daily summary files and reports created by *runacct*
- *monacct*—creates monthly summary files and reports

Displaying Total Accounting Records – prtacct

The *prtacct* command allows you to display the contents of a process accounting file. Its format is

`prtacct` *file* "*heading*"

where:

- *file* is the name of the total accounting file to be displayed
- "*heading*" is a comment to be included in the standard report header produced by *prtacct*

The format of the *prtacct* report is described next and is followed by an example.

Report Format

prtacct produces a columnar report with one line per total accounting record. Descriptive column headings are included in the report. Definitions of each column follow:

Column Header	Description
UID	The user ID of the owner of the total accounting record—i.e., the ID of the user for whom the total accounting record was created.
LOGIN NAME	The login name of the owner of the total accounting record is displayed here.
CPU (MINS)	The total amount of CPU time (in minutes) that the user has consumed. This column is divided into prime and non-prime columns (PRIME and NPRIME respectively). Information in these columns is created through process accounting commands.
KCORE-MINS	This represents a cumulative measure of memory and CPU time that a user consumed (see "Definitions of Information Produced by acctcom" for a more precise definition). Information in this column is also divided into PRIME and NPRIME columns. This information is created through process accounting commands.
CONNECT (MINS)	Identifies the real time used (in minutes). In essence, what this column identifies is the amount of time that the user was logged in to the system. This column is also subdivided into PRIME and NPRIME columns. The connect session accounting commands are the source of this information.
DISK BLOCKS	The total number of disc blocks allocated to the user is shown here. This information is created via disc space accounting commands.
# OF PROCS	The total number process spawned by the user is displayed here. This information is created via the process accounting commands.
# OF SESS	This column shows how many times the user logged in. Connect session accounting commands create this data.
# DISK SAMPLES	This column indicates how many times the disc accounting was run to obtain the average number of disc blocks listed in the DISK BLOCKS column.
FEE	The number of fee units charged via *chargefee* is displayed here.

Example

The following example displays disc total accounting records. First the total accounting records are created via disc space accounting commands; then they are displayed using *prtacct*. When examining this report, take note of the following:

1. The similarities between this and the sample report produced by *diskusg* (see "Disc Space Usage Accounting").

2. Only the columns relating to disc space usage have non-zero values, because the total accounting records were created only from disc space usage accounting commands.

```
$ for file_system in 'cat /etc/checklist'
> do
> diskusg $file_system >dtmp.'basename $file_system'
> done
$ diskusg -s dtmp.* |sort +0n +1 |acctdisk >disktacct
$ prtacct disktacct "DISC TOTAL ACCOUNTING RECORDS"
```

Mar 26 17:01 1985 DISC TOTAL ACCOUNTING RECORDS Page 1

	LOGIN	CPU (MINS)		KCORE-MINS		CONNECT (MINS)		DISK	# OF	# OF	# DISK	FEE
UID	NAME	PRIME	NPRIME	PRIME	NPRIME	PRIME	NPRIME	BLOCKS	PROCS	SESS	SAMPLES	
0	TOTAL	0	0	0	0	0	0	11598	0	0	10	0
0	root	0	0	0	0	0	0	10616	0	0	1	0
1	bin	0	0	0	0	0	0	778	0	0	1	0
4	adm	0	0	0	0	0	0	96	0	0	1	0
350	fred	0	0	0	0	0	0	14	0	0	1	0
351	bill	0	0	0	0	0	0	32	0	0	1	0
352	mike	0	0	0	0	0	0	20	0	0	1	0
353	sarah	0	0	0	0	0	0	16	0	0	1	0
354	molly	0	0	0	0	0	0	22	0	0	1	0
355	horatio	0	0	0	0	0	0	2	0	0	1	0
501	guest	0	0	0	0	0	0	2	0	0	1	0

Merging Total Accounting Files - acctmerg

Normally executed by *runacct*, the *acctmerg* command merges separate total accounting files into a single total accounting file. All the total accounting records for a particular user name and ID are merged together to form one total accounting record for the given user name and ID. This command is useful for merging disc, connect session, and process total accounting files together to form a single, comprehensive total accounting file.

acctmerg reads standard input and up to nine additional files, all in total accounting record format (or an ASCII version thereof). Its syntax is

`acctmerg` [*options*] [*file*] ...

where:

- *options* control the report format and the manner in which records are merged.
- *file* is one of up to nine files (in addition to standard input) that are to be merged into a single total accounting file, written to standard output.

Command Options

The following options may be used with *acctmerg* to control the report format and the manner in which the total accounting records are merged:

Option	Description
`-a`	*acctmerg* normally produces output as total accounting records. Using the `-a` option causes *acctmerg* to produce output in ASCII. Note that the output generated by using this option is the same as the report produced by *prtacct*, except that no report headings or totals are displayed—only the columnar data is shown.
`-i`	In the default case, *acctmerg* assumes that its input files contain total accounting records. If `-i` is specified, then *acctmerg* will expect input files to be in the ASCII format created by the `-a` option.
`-p`	This option simply echos input records—no merging or processing is done. The output is displayed in the format produced by the `-a` option.
`-t`	Produces a single total accounting record that summarizes all input records. To see the ASCII version of this record, you must use the `-t` and `-a` options together: `$ acctmerg -t -a <tot_acct_recs` **Note that `-t` and `-a` can be specified in any order, but they must be specified seperately as shown.**

-u Normally, *acctmerg* merges records that have the same user ID and user name. Using **-u** causes *acctmerg* to merge records on the basis of same user ID only—i.e., disregard the user name as a key on which to merge records.

-v This option causes *acctmerg* to produce output in verbose ASCII format. The same report is produced as the **-a** option, except that floating point numbers are displayed in more precise notation:

<*mantissa*>e<*exponent*>

The **-a**, **-v**, and **-i** options are useful if you wish to edit total accounting records. For example, suppose that you have created a total accounting file (*ptacct*) containing process total accounting records, and you want to make some adjustments to these records. The following sequence could be used to make "repairs" to this file.

```
$ acctmerg -v -a <ptacct >ptacct.ascii
      edit ptacct.ascii as desired ...
      then copy the changes back to ptacct
$ acctmerg -i <ptacct.ascii >ptacct
```

Example

The following example creates disc, process, and connect session total accounting records, merges them together, and stores the merged file in the file `merged_file`.

```
$ # First, create disc space usage total accounting records (dtacct)...
$ #
$ for fs in `cat /etc/checklist`
> do
> diskusg $fs >dtmp.`basename $fs`
> done
$ diskusg -s dtmp.* |sort +0n +1 |acctdisk >dtacct
$ #
$ # Now create connect session total accounting records (ctacct)...
$ #
$ acctcon1 </etc/wtmp |acctcon2 >ctacct
$ #
$ # Create process total accounting records (ptacct)...
$ #
$ >ptacct
$ for p_file in pacct*
> do
> acctprc1 <$p_file |acctprc2 >>ptacct
> done
$ #
$ # Now merge all the total accounting files (?tacct) into
$ # a single total accounting file (tacct)...
$ #
$ acctmerg dtacct ctacct <ptacct >tacct
```

Creating Daily Accounting Information – runacct

runacct is the main daily accounting shell procedure. It is normally initiated via *cron* during non-prime hours. *runacct* processes disc, connect session, process, and fee accounting files. It prepares cumulative summary files for use by *prdaily* and/or for billing purposes. This section discusses the following aspects of *runacct*:

- files processed by *runacct*
- the states that *runacct* progresses through while executing
- recovery from *runacct* failure
- restarting *runacct*
- reports produced by *runacct*

Files Processed by runacct

The following files, processed by *runacct*, are of particular interest to the reader. (Filenames are given relative to the directory */usr/adm/acct*.)

- *nite/lineuse* contains usage statistics for each terminal line on the system. This report is especially useful for detecting bad lines. If the ratio of logoffs to logins on a particular line exceeds 3 to 1, then there is a good possibility that the line is failing.
- *nite/daytacct* contains total accounting records from the previous day.
- *sum/tacct* contains accumulated total accounting records for each day's total accounting records (*nite/daytacct*) and can be used for billing purposes. It is restarted each month or fiscal period by the *monacct* shell script.
- *sum/daycms* is produced by *acctcms*. It contains the daily command summary. The ASCII version of this file is in *nite/daycms*.
- *sum/cms* holds the accumulation of each day's command summaries (*sum/daycms*). A new *sum/cms* file is created each month by *monacct*. The ASCII version of this file is in *nite/cms*.
- *sum/loginlog* maintains a record of the last time each user logged in.
- *sum/rprtMMDD* is the main daily accounting report created by *runacct*. This report can be printed via *prdaily*.

runacct takes care not to damage files in the event of errors. A series of protection mechanisms are used that attempt to recognize errors, provide intelligent diagnostics, and terminate processing in such a way that *runacct* can be restarted with minimal intervention. To accomplish these goals, the following actions are performed by *runacct*:

- *runacct*'s progress is recorded by writing descriptive messages to the file *nite/active*.
- All diagnostics output during the execution of *runacct* are redirected to the file *nite/fd2log*.
- If the files *lock* and *lock1* exist when *runacct* is invoked, an error message will be displayed, and execution will terminate.
- The *lastdate* file contains the month and day that *runacct* was last run and is used to prevent more than one execution per day.
- If *runacct* detects an error, a message is written to */dev/console*, mail is sent to *root* and *adm*, locks are removed, diagnostics files are saved, and execution is terminated.

The States of runacct

In order to allow *runacct* to be restartable, processing is broken down into separate re-entrant *states*. As *runacct* executes, it records its progress by writing the name of the most recently completed state into the file called **/usr/adm/statefile**. After processing for a state is complete, *runacct* examines *statefile* to determine which state to enter next. When *runacct* reaches the final state (CLEANUP), the *lock* and *lock1* files are removed, and execution terminates. The following table describes *runacct*'s states:

State	Action
SETUP	The command *turnacct switch* is executed. The process accounting files, *pacct?*, are moved to *Spacct?.MMDD*. The */etc/wtmp* file is moved to *nite/wtmp.MMDD* with the current time added on the end.
WTMPFIX	*nite/wtmp.MMDD* is checked for correctness by *wtmpfix*. Some date changes will cause *acctcon1* to fail, so *wtmpfix* attempts to adjust the time stamps in the *nite/wtmp.MMDD* file if a date change record appears.
CONNECT1	Connect session records are written to *ctmp*. The *lineuse* file is created, and the *reboots* file, showing all of the boot records found in *nite/wtmp.MMDD*, is created.
CONNECT2	*ctmp* is converted to connect session total accounting records in the file *ctacct.MMDD*.
PROCESS	The *acctprc1* and *acctprc2* programs are used to convert the process accounting files, *Spacct?.MMDD*, to the total accounting records in *ptacct?.MMDD*. The *Spacct* and *ptacct* files are correlated by number so that if *runacct* fails, the unnecessary reprocessing of *Spacct* files will not occur. One precaution should be noted: **When restarting *runacct* in this state, remove the last *ptacct* file; if you don't, *runacct* will not finish.**
MERGE	Merge the process and connect session total accounting records to form *nite/daytacct*.
FEES	Merge in any ASCII *tacct* records from the file *fee* into *nite/daytacct*.
DISK	On the day after the *dodisk* shell script runs, merge *nite/disktacct* with *nite/daytacct*.
MERGETACCT	Merge *nite/daytacct* with *sum/tacct*, the cumulative total accounting file. Each day, *nite/daytacct* is saved in *sum/tacctMMDD*, so that *sum/tacct* can be recreated in the event it becomes corrupted or lost.
CMS	Merge in today's command summary with the cumulative summary file *sum/cms*. Produce ASCII and internal format command summary files.

`USEREXIT` Any installation-dependent (local) accounting programs can be run in this state. For example, you might want to execute commands that generate daily billing data for individual users (the shell script `acct_bill` in the section "Sample System Accounting Shell Scripts" could be used for this purpose). To have local accounting programs executed by *runacct*, simply enter the commands in *runacct* in the code for the `USEREXIT` state of *runacct*.

`CLEANUP` Clean up the temporary files, run *prdaily* and save its output in the file *sum/rprtMMDD*, remove the locks, then exit.

Recovering from Failure

It is possible that *runacct* might fail and terminate abnormally. The primary reasons for *runacct* failure are:

- a system "crash"
- not enough disc space remaining in */usr*
- a corrupted *wtmp* file

If the *nite/activeMMDD* file exists, check it first for error messages. If the *nite/active* file and lock files exist, check *fd2log* for any mysterious messages. The following are error messages produced by *runacct* and the recommended recovery actions:

- `ERROR: locks found, run aborted`

  The files *lock* and *lock1* were found. These files must be removed before *runacct* can be restarted.

- `ERROR: acctg already run for` *date*`: check /usr/adm/acct/nite/lastdate`

  The date in *lastdate* and today's date are the same. Remove *lastdate* before restarting *runacct*.

- `ERROR: turnacct switch returned rc=`*?*

  Check the integrity of *turnacct* and *accton*. The *accton* program must be owned by *root* and have the *setuid* bit set.

- `ERROR: Spacct?.MMDD already exists`

  File setups probably already run. Check the status of files, then run setups manually.

- `ERROR: /usr/adm/acct/nite/wtmp.MMDD already exists, run setup manually`

  You must perform the `SETUP` step manually, because the daily *wtmp* file already exists.

- `ERROR: wtmpfix errors see /usr/adm/acct/nite/wtmperror`

  wtmpfix detected a corrupted *wtmp* file. See the section "Fixing Corrupted Files" for details on fixing *wtmp* errors.

- `ERROR: connect acctg failed: check /usr/adm/acct/nite/log`

  acctcon1 encountered a bad *wtmp* file. Again, see the section "Fixing Corrupted Files" on how to fix the file.

- `ERROR: Invalid state, check /usr/adm/acct/nite/active`

  the file *statefile* is probably corrupted. Check *statefile* and read *active* before restarting.

Restarting runacct

runacct is normally run via *cron* only once per day. However, if an error occurs while executing *runacct* (as described above), it may be necessary to restart *runacct*. *runacct* has the following syntax:

`runacct` [*mmdd* [*state*]]

When called without arguments, *runacct* assumes that it is being invoked for the first time on the current day; this is how *runacct* is invoked by *cron*. The argument *mmdd* is necessary if *runacct* is being restarted and specifies the month and day for which *runacct* will rerun the accounting. The entry point for processing is based on the contents of *statefile*. To override *statefile*, include the desired entry *state* on the command line.

For example, to start *runacct*, you would enter:

```
$ nohup runacct 2> /usr/adm/acct/nite/fd2log &
```

To restart *runacct* on the 26th day of March:

```
$ nohup runacct 0326 2> /usr/adm/acct/nite/fd2log &
```

To restart *runacct* at state `WTMPFIX` on June 1st:

```
$ nohup runacct 0601 WTMPFIX 2>/usr/adm/acct/nite/fd2log &
```

Daily Reports

runacct generates five basic reports upon each invocation. Brief descriptions of each report follow. **Detailed descriptions of the reports are found in the following section, "Displaying runacct Reports — prdaily."**

- **Daily Line Usage Report**—summarizes connect session accounting since the last invocation of *runacct*. It provides a log of system reboots, power failure recoveries, and any other records dumped into */etc/wtmp* via *acctwtmp*. In addition, it provides a breakdown of line utilization.

- **Daily Resource Usage Report**—gives a summary of resource usage per individual user: it basically merges all the total accounting records for individual users and displays the records, one per user.

- **Daily Command Summary**—summarizes resource usage data for individual commands since the last invocation of *runacct*. The data included in this report is useful in determining the most heavily used commands; you can use these commands' characteristics of resource utilization when "tuning" your system.

  This report is sorted by `TOTAL KCOREMIN`, an arbitrary but often-good yardstick for calculating "drain" on a system.

- **Monthly Total Command Summary**—This report is exactly the same as the Daily Command Summary, except that the Daily Command Summary contains command summary information accumulated only since the last invocation of *runacct*, while the Monthly Total Command Summary summarizes commands from the start of the fiscal period to the current date. In other words, the monthly report reflects the data ' accumulated since the last invocation of *monacct*.

- **Last Login**—simply gives the date each user last logged in to the system. This could be a good source for finding likely candidates for the archives, or getting rid of unused login directories.

Displaying runacct Reports - prdaily

As *runacct* finishes executing, it deposits a report of the current day's accounting in the file */usr/adm/acct/sum/rpt***mmdd**, where *mmdd* is the month and day that the report was generated. The *prdaily* command is used to display the contents of any daily report file created by *runacct*. Its syntax is

`prdaily` [-*l*] [-*c*] [*mmdd*]

where:

- *mmdd* is an optional report date. If no date is specified, *prdaily* produces a report of the current day's accounting information. Previous days' accounting reports can be displayed by using the *mmdd* option and specifying the exact report date desired.
- The `-l` option prints a report of exceptional usage by login name for the specified date. This option is used to determine which users are consuming excessive amounts of system resources. The limits for exceptional usage are kept in the file */usr/lib/acct/ptelus.awk* and can be edited to reflect your installation's requirements.
- Valid only for the current day's accounting, the `-c` option is used to get a report of exceptional resource usage by command. This option is used to determine which commands are using excessive amounts of system resources. The limits for exceptional usage are maintained in the file */usr/lib/acct/ptecms.awk* and can be edited to reflect your system's needs.

The reports produced by *runacct* were described briefly in the previous sub-section. Now the reports are discussed in more detail.

Daily Line Usage Report

In the first part of this report, the FROM/TO banner should alert you to the period reported on. The times are the date-time that the last report generated by *runacct*, and the date-time that the current report was generated. It is followed by a log of system reboots, shutdowns, power failure recoveries, and any other records dumped into *wtmp* by the *acctwtmp* command.

The second part of the report is a breakdown of line utilization. The TOTAL DURATION shows how long the system was in a multi-user state. The columns of the report are defined in the following table:

Column	Description
LINE	The terminal line or access port being reported on.
MINUTES	The total numbber of minutes that the line was in use during the accounting period.
PERCENT	The percentage of TOTAL DURATION that the line was in use: PERCENT = (MINUTES / TOTAL DURATION) * 100
# SESS	Shows the number of times that this port was accessed for a *login* session.
# ON	Historically, this column displayed the number of times that the port was used to log a user on; but since *login* can no longer be executed explicitly to log in a new user, this column should be identical to # SESS.
# OFF	This column reflects not only the number of times a user logged off, but also any interrupts that occurred on the line. Generally, interrupts occur on a port when *getty(1M)* is first invoked when the system is brought down to a multi-user state. This column comes into play when # OFF exceeds # ON by a large factor. This usually indicates that the multiplexer, modem, or cable is going bad, or that there is a bad connection somewhere. The most common cause of this is an unconnected cable dangling from the multiplexer.

During real time, *wtmp* should be monitored as this is the file that connect session accounting is taken from. If it grows rapidly, execute *acctcon1* to determine which line is the noisiest. If the interrupting is occuring at a furious rate, general system performance will be affected.

Daily Resource Usage Report

This report gives a by-user breakdown of system resource usage. The format of this report is the same as that produced by the *prtacct* command. See the Report Format table for the *prtacct* command for definitions of the columnar data found in this report.

Daily and Monthly Command Summary

These two reports are the same, except that the Daily Command Summary reports information only for commands executed since the last invocation of *runacct*; the Monthly Command Summary contains information on commands executed since the last invocation of *monacct*.

The output of this report is identical to that produced by *acctcms*. For definitions of the data found in this report, see the discussion of *acctcms* in the "Process Accounting" section.

Last Login

This report simply shows the last date and time that each user logged in. The longer it has been since a particular user logged in, the more likely it is that the user's files could be archived, or maybe even that the user could be removed from the system.

Creating Monthly Accounting Reports - monacct

monacct creates monthly summary files and reports; the resulting output is stored in the directory */usr/adm/acct/fiscal*. After creating its monthly reports, it removes the old daily accounting files from the directory */usr/adm/acct/sum* and replaces them with new summary accounting files.

monacct should be invoked once each month or accounting period. Its syntax is

`monacct` *number*

where *number* indicates the which month or period it is (01=January, 12=December). If *number* is not specified, *monacct* assumes that it is being invoked for the current month; this default is useful if *monacct* is executed via *cron* on the first day of each month (as described in the "Daily Usage and Installation" section).

Descriptions of the files created in the *acct/fiscal* directory follow:

- *cms?*—contains the total command summary file for the accounting period denoted by ?. The file is stored in internal summary format. Therefore, to display this file, you must use the *acctcms* command. The following example shows how to display this file for the month of June:

  ```
  $ acctcms -a -s /usr/adm/acct/nite/fiscal/cms06
  ```

- *fiscrpt?*—contains a report similar to that produced by *prdaily*. The report shows line and resource usage for the month represented by ?. The following would display the fiscal accounting file for the month of November:

  ```
  $ cat /usr/adm/acct/nite/fiscal/fiscrpt11
  ```

- *tacct?*—is the total accounting file for the month represented by ?. To display this file, you must use the *prtacct* command. The following would display the total accounting summary file for the month of January:

  ```
  $ prtacct /usr/adm/acct/fiscal/tacct01 "JANUARY TOTAL ACCOUNTING"
  ```

Updating the Holidays File

The file ***/usr/lib/acct/holidays*** contains the information that System Accounting needs to distinguish between prime and non-prime time. It contains the following information:

1. **Comment Lines**. Comment lines are entered by placing an astrisk (*) as the first character in the line; they may appear anywhere in the file.
2. **Year Designation Line**. This line should be the first non-comment line in the file and must appear only once. The line consists of three four-digit numbers (leading blanks and tabs are ignored). The first number designates the year; the second denotes the time (in 24-hour format) that prime time starts; the third gives the time that prime time ends and non-prime time starts.

   For example, to specify the year as 1985, prime time at 9:00 a.m., and non-prime time at 4:30 p.m., the following entry would be appropriate:

   ```
   1982   0900   1630
   ```

   A special condition allowed for in the time field is that 2400 is automatically converted to 0000.
3. **Company Holiday Lines**. These entries follow the year designation line. Company holidays are days when few people should be using the computer. Therefore, System Accounting assumes that non-prime time is in effect during the entire 24 hours of a specified holiday.

   Company holiday lines have the following format:

   day_of_year   Month   Day   Description of Holiday

   The *day_of_year* field is a number in the range 1 through 366, corresponding to the day of the year for the particular holiday (leading blanks and tabs are ignored). The remaining fields are simply commentary and are not used by other programs.

NOTE

As delivered, the *holidays* file contains valid entries for Hewlett-Packard's prime/non-prime time, and holidays. You should check this file and edit it as necessary to reflect your organization's requirements.

Fixing Corrupted Files

System Accounting files may become corrupted or lost. Some of these files can simply be ignored or restored from the file save backup. However, certain files must be fixed in order to maintain the integrity of System Accounting.

Fixing wtmp Errors

The *wtmp* files seem to cause the most problems in the daily operation of System Accounting. When the date is changed and HP-UX is switched into multi-user mode, a set of date change records is written into */etc/wtmp*. The *wtmpfix* command is designed to adjust the time stamps in the *wtmp* records when a date change is encountered. However, some combinations of date changes and reboots won't be caught by *wtmpfix* and cause *acctcon1* to fail. The following steps show how to "patch" a damaged *wtmp* file.

```
$ cd /usr/adm/acct/nite
$ fwtmp <wtmp.MMDD >wtmp.temp
        Using an editor, delete corrupted records or
        delete all records from beginning up to the date change
$ fwtmp -ic <wtmp.temp >wtmp.MMDD
$ rm wtmp.temp
```

If the *wtmp* file is beyond repair, create a null *wtmp* file. This will prevent any charging of connect time. *acctprc1* will not be able to determine which login owned a particular process, but it will be charged to the login that is first in the password file for that user ID.

Fixing tacct Errors

If your installation is using System Accounting to charge users for system resource usage, the integrity of *sum/tacct* is quite important. If *sum/tacct* ever becomes corrupted, then check the contents of *sum/tacctprev* with the command *prtacct*. If it looks correct, then the latest *sum/tacct.MMDD* should be patched up, and *sum/tacct* should then be recreated. A simple patch procedure would be:

```
$ cd /usr/adm/acct/sum
$ acctmerg -a -v <tacct.MMDD >tacct.temp
        Using an editor, remove the bad records and
        write duplicate UID records to another file
$ acctmerg -i <tacct.temp >tacct.MMDD
$ acctmerg tacctprev <tacct.MMDD >tacct
$ rm tacct.temp
```

Remember that *monacct* removes all the *tacct.MMDD* files; therefore, *sum/tacct* can be recreated by merging these files together.

Sample Accounting Shell Scripts

grpdusg

This shell script displays disc space usage totals for the users who are members of a specified group. The syntax of this command is:

```
grpdusg group_name
```

where *group_name* is the name of the group for which disc space accounting information is to be generated.

For example,

```
$ grpdusg pseudo
```

generates disc space usage information for all the users in the group pseudo.

The Shell Script

```
# Check for the group-name parameter.
#
if      [ $# -ne 1 ]
then    echo "\nUsage:  grpdusg group-name\n"
        exit 1
fi
echo    "\nOne moment please...\n"
#
# Use the find command to find all the files whose owners are members of
# group-name.  Pipe the output from find into acctdusg which will accumulate
# disc space usage information for the users in group-name.
# NOTE:
#       - accounting data is temporarily stored in _${1}_tmp
#       - error messages are stored temporarily in _${1}_err
#       - if files exist that have no owners, then the names of
#         these files are stored in _no_owners
#
fn=_${1}_
find / -group $1 -print 2>${fn}err |acctdusg -u _no_owners >${fn}tmp
#
# Remove the _no_owners file if its size is not greater than zero.
#
if      [ -s _no_owners ]
then    echo "\nFiles having no owners exist--check _no_owners\n"
else    rm _no_owners
        echo "\nAll files have owners-- _no_owners not created\n"
fi
#
# Use echo and awk to display disc usage totals for this group.
#
echo "\nDisc space usage information (group is ${1}):\n"
awk 'BEGIN {print "\n_UID____USER NAME________BLOCKS"}
     { sum += $3 ;              # add up total disc blocks used
       print $0                 # display information for user
     END { print "\nTOTAL DISC SPACE USAGE= ", sum, "blocks" }' ${fn}tmp
#
# Remove temporary files, then exit.
#
rm ${fn}*
```

acct_bill

acct_bill takes as input a total accounting file and produces as output billing totals for all users found in the input file. The syntax of *acct_bill* is:

```
acct_bill [ mmdd ]
```

If the optional *mmdd* is not specified, then *acct_bill* takes as input the current day's total accounting file (*acct/nite/daytacct*); if *mmdd* is given, then input is taken from the total accounting file for the date specified by *mmdd* (*acct/sum/tacct***mmdd**). Output is written to the file *bills***mmdd**, where **mmdd** is the date given with the command, or the current date if *mmdd* was not specified with the command.

Examples

To generate billing information for the current day, simply enter:

```
$ acct_bill
```

and the billing information will be stored in the file *acct/sum/bills***mmdd**, where *mmdd* is the current date.

To create billing information for January 23rd, you would enter:

```
$ acct_bill 0123
```

after which the billing information would be stored in the file called *acct/sum/bills0123*.

To automatically generate daily billing totals for all users, you should call *acct_bill* without the date argument from the USEREXIT state of *runacct*.

Output Produced by acct_bill

The output of *acct_bill* contains one line per user and has the following format:

user_ID *user_name* *billing_amount*

where *user_ID* and *user_name* identify the user who is being billed, and *billing_amount* shows the total amount that the useris to be charged.

billing_amount is computed by multiplying *accounting coefficients* (found in the shell script) by columns of the report generated by *prtacct*. Assuming that billing amounts are in dollars, the coefficients (as they are currently shown) produce the following billing amounts:

- 10 cents for every minute of prime CPU time consumed
- five cents for every minute of non-prime CPU time consumed
- a half cent for every prime kcore minute used
- two-tenths of a cent for every non-prime kcore minute
- a half cent for every prime connect time minute
- two-tenths of a cent for every non-prime connect minute
- two-and-a-half cents for every block of disc space used
- two-and-a-half cents for every process spawned by the user
- ten cents for every connect session
- each fee unit charged via *chargefee* counts as one cent

You should experiment with this command by altering the coefficients to see how *billing_amount* is affected. After gaining confidence with this shell script, you can alter the coefficients to suit your installation's needs.

The Shell Script

```
_date=`date +%m%d`
_outfile=/usr/adm/acct/sum/bills
_infile=/usr/adm/acct
#
# Set _infile and _outfile, based on whether or not MMDD was given
#
if      [ $# -eq 0 ]
then                    # Generate billing data for current day.
        _infile=${_infile}/nite/daytacct
        _outfile=${_outfile}${_date}
else                    # Create billing data for date given (MMDD).
        _infile=${_infile}/sum/tacct${1}
        _outfile=${_outfile}${1}
fi
#
# Create a file containing the ASCII equivalent of the input total
# accounting file (tacct_ASC.tmp_).  The file can then be supplied as input
# to awk, which will generate billing data for each user.
#
acctmerg -a -t <$_infile >$_outfile  # output TOTAL amount first
acctmerg -a <$_infile >>$_outfile    # append users' total accounting records
#
# Using awk, compute billing totals for each user in the total accounting file.
#
awk 'BEGIN {
             # ***************************************************
             #    A C C O U N T I N G   C O E F F I C I E N T S
             # ***************************************************
             cpu_P =0.10  # 0.10 monetary units per minute of prime CPU time
             cpu_NP=0.05  # 0.05 monetary units per non-prime CPU minute used
             kcm_P =0.005 # for prime kcore minutes consumed
             kcm_NP=0.002 # for non-prime kcore minutes used
             con_P =0.005 # prime connect (real) time
             con_NP=0.002 # non-prime connect time used
             blk =  0.025 # number of blocks used
             prc =  0.025 # number of processes spawned
             ses =  0.10  # number of connect sessions
             fee =  0.01  # 100 charge units per monetary unit
             # ***************************************************
           }
     # Start computing billing amounts for each user.
     { _sum = cpu_P*$3 + kcm_P*$5 + con_P*$7        # compute prime usage
       _sum+= cpu_NP*$4+ kcm_NP*$6+ con_NP*$8       # add non-prime usage
       _sum+= blk*$9 + prc*$10 + ses*$11 + fee*$13  # add remaining amounts
       printf "%-8s %-10s %10.3f\n", $1, $2, _sum   # display user total
     }' tacct_ASC.tmp_ >$_outfile  # write output from awk to appropriate file
rm tacct_ASC.tmp_                  # remove the temporary ASCII file
```

Using the FSCK Command A

Introduction

The file system consistency check (*/etc/fsck*) corrects inconsistent information in your file system.

You must have a thorough understanding of the file system before making any *fsck* decisions. Read "File System Implementation" in the "Concepts" chapter before going on.

fsck should be performed:

- during bootup, if you had an unclean shutdown (did not use *shutdown* or *reboot*).

  It should be done before the system is taken into run-level 2. As shipped, your system should do this automatically if it detects an improper shutdown, via the `bcheckrc` entry in */etc/inittab*. An improper shutdown means you didn't shut down your system using the *shutdown* command described in "Shutting Down Your System" in the "Toolbox" chapter. *fsclean* is used to detect an improper shutdown.

- **any** time you suspect problems with the HP-UX file system.

  There is no specific problem you can relate to file system corruption. If your HP-UX system doesn't behave like you think it should, run *fsck*.

- You should run *fsck* occasionally even though *fsclean* (run from */etc/bcheckrc*) does not indicate the need to do so. The backup script has an option to run *fsck -n* after the backup is performed; you should use this option. *fsck -n* reports discrepencies, but does not fix them, so it can be safely run by *cron*.

The *fsck* program, when run on the root file system, **MUST use the block device** (i.e., /dev/root). If the character device is used, *fsck* doesn't know that it was the root file system, and won't issue a reboot instruction when needed.

The *fsck* program can be run in several different modes.

-p Preening mode.

The preening mode is the option used by *bcheckrc*. This option fixes many potential problems, but never removes data. When you preen the system, you are not running interactively. *fsck* will decide what to do, and if it can't deal with a situation, it will terminate. For the inconsistencies the preening mode can fix, it will print a message identifying the file system, and the corrective action taken. The preening option will fix the following inconsistencies:

- Unreferenced inodes
- unreferenced files and fifos
- link counts in inodes too large
- missing blocks in the free list
- blocks in the free list also in files
- wrong counts in the superblock
- clean byte marked wrong

Other problems will cause *fsck -p* to terminate and prompt for manual execution of *fsck*. Since it was created to be run non-interactively, when it finishes it does not tell you if you need to reboot the system. You must determine that from the return code (see *fsck(1M)*).

-y Yes mode.

Using the *-y* option can be very dangerous. This option causes *fsck* to answer YES to all questions, which might remove data. **Do not use the -y option if you have important data on your file system** unless you have first used the *-n* option and understand the potential damage.

-n No mode

Using the *-n* option causes *fsck* to answer NO to all questions. This will never remove data, so is very safe. You can use the *-n* option any time; multiuser (though not recommended), singleuser, or background.

If you use *fsck* with the *-n* option in multi-user mode, you will probably come up with some inconsistancies due to file system action. However, you will never damage your system.

default Interactive mode.

The interactive mode allows you to choose whether to perform an action or not.

-q Quiet mode.

fsck prints only the messages that require a response.

The system should **always** be in a single-user state and quiescent (inactive and not being written on) before executing the *fsck* command. Use *fsck* in run-level s, after the *shutdown* command is executed. Running *fsck* when there is file system activity may cause loss of data.

fsck should be executed on a file system that is unmounted (except for the root file system). If you shutdown the system using *shutdown*, all file systems except root will be unmounted and you will be in run-level s.

fsck should be executed using an **unmounted character special device file**, not a block special device file, except when checking the root file system. See "Adding/Moving Peripherals" in the "Toolbox" chapter for a discussion on block and character devices, and naming conventions for device files.

Only the system administrator should run *fsck*. If this check discovers an inconsistency, corrective action must be taken. This article will tell you how to use *fsck* to insure file system integrity.

A directory with the name */lost+found* must exist on the file system being examined **before** *fsck* is run. */lost+found* should have been created on your root volume when you installed HP-UX on your system. The *fsck* command uses this directory for any problem files that it finds. After you run *fsck*, examine the files placed in /lost+found and move them to where they belong or remove them. You should clear the /lost+found directory before you execute *fsck* again.

To place these files, follow this procedure:

1. Change to the lost+found directory (`cd /lost+found`).
2. Find out what type of file it is (executable, text, etc), and who owns the file, by typing:

   ```
   file * [Return]
   ll * [Return]
   ```

   If the file is text, see what's in it by typing:

   `more` *filename* [Return]
3. If the file is executable, you can try one of two things:
   a. If the file has an SCCS id string, the `what` command will list it.
   b. If the file does not have an SCCS id string, use the `strings` command to print the literal strings from the file. The strings (e.g. error message strings) may help identify the owner.
4. From this information, determine where the file belongs, or who it belongs to, and move the file to the correct directory.

How FSCK Handles Inconsistencies

Conventions

fsck is a multi-pass file system check program. Each phase of the *fsck* program invokes a different file system pass. After the initial setup, *fsck* performs successive phases over each file system, checking blocks and sizes, pathnames, connectivity, reference counts, and the free-block map (possibly rebuilding it), and doing some cleanup.

Read the "Introduction" section for a discussion on the different modes (options) you can run *fsck* in.

When an inconsistency is detected while running interactively, *fsck* reports the error condition. If a response is required, *fsck* prints a prompt message and waits for a response. When preening, *fsck* will choose a response and note it on the screen. In this section, each error message and possible responses are presented.

The error conditions are organized by the phase of the *fsck* program in which they can occur. The error conditions that may occur in more than one phase are discussed in "Initialization Phase Errors" below.

Initialization Phase Errors

During the initialization phase, before the file system check can be performed, tables have to be set up and certain files opened. This section lists error conditions resulting from command line options, memory requests, opening of files, status of files, file system size checks, and creation of the scratch file. All of the initialization errors are fatal if you are preening. See the *fsck(1M)* entry in the HP-UX Reference manual for further information.

"C" option?

The character represented by *C* is not a legal option to *fsck*. Legal options are *-b*, *-y*, *-n*, *-q*, and -p. *fsck* terminates on this error condition. See the *fsck(1M)* entry in the HP-UX Reference manual for further information.

cannot alloc NNN bytes for "XXX"

XXX is either *blockmap*, *freemap*, *statemap*, or *lncntp*. *fsck*'s request for memory failed. This should never happen. *fsck* terminates on this error condition. Contact your local HP Sales and Service Office for assistance.

Can't open checklist file:

The default file system checklist file (*/etc/checklist*) can not be opened for reading. *fsck* terminates on this error condition. Check for the existence of the file, and the access modes of the file.

Can't stat root

fsck's request for statistics about the root directory (/) failed. This should never happen. *fsck* terminates on this error condition. Contact you local HP Sales and Service Office for assistance.

Can't stat ...
Can't make sense out of ...

fsck's request for statistics about the file system failed. When running manually, it ignores this file system and continues checking the next file system given. If this happens, check for the existence, and the access modes, of the file system.

"file_system_name" is not a block or character device; OK

You have given *fsck* a regular file name by mistake. You should check the file type of the file system.
Possible responses to the OK prompt are:

YES Ignore this error condition.

NO Ignore this file system and continue checking the next file system given.

Can't open ...

The file system listed cannot be opened for reading. When running manually, it ignores this file system and continues checking the next file system given. Check the access modes of the file system.

"file_system_name": (NO WRITE)

Either the *-n* flag was specified or *fsck*'s attempt to open the file system, "file_system_name" for writing failed. When running manually, all the diagnostics are printed, but no modifications are attempted to fix them.

Other messages:

```
MAGIC NUMBER WRONG
NCG OUT OF RANGE
CPG OUT OF RANGE
NCYL DOES NOT GIVE WITH NCG*CPG
SIZE PREPOSTEROUSLY LARGE
TRASHED VALUES IN SUPER BLOCK
```

and will be followed by the message:

file_system: `BAD SUPER BLOCK: superblock_address`
`USE -b OPTION TO FSCK TO SPECIFY LOCATION OF AN ALTERNATE`
`SUPER-BLOCK TO SUPPLY NEEDED INFORMATION; SEE fsck(1M).`

The superblock has been corrupted. An alternative superblock must be used. See the discussion on alternative superblocks under the "Superblock Consistency" section in this appendix.

INTERNAL INCONSISTENCY: "message"
An internal problem occured in *fsck*. "*message*" will indicate the problem. This should never happen. Contact your local HP Sales and Service Office for assistance.

CAN NOT SEEK: BLK "bn" (CONTINUE)
fsck's request for moving to the specified block number in the file system failed. This should never happen. Contact your local HP Sales and Service Office for further assistance.
Possible responses to the CONTINUE prompt are:

YES Attempt to continue to run the file system check. Often, however, the problem will persist. This error condition will not allow a complete check of the file system. A second run of *fsck* should be made to re-check this file system.

NO Terminate the program.

CAN NOT READ: BLK ... (CONTINUE)
fsck's attempt to read a specified block number in the file system failed. This can happen when you interrupt *fsck* before it finishes. Contact your local HP Sales and Service Office for further assistance.

Possible responses to the CONTINUE prompt are:

YES Attempt to continue to run the file system check. Often, however, the problem will persist. This error condition will not allow a complete check of the file system. A second run of *fsck* should be made to re-check this file system.

NO Terminate the program.

CAN NOT WRITE: BLK ... (CONTINUE)
fsck's attempt to write a specified block number in the file system failed. The disc is probably physically write-protected. Remove write protections from the disc, and re-run *fsck*.

Possible responses to the CONTINUE prompt are:

YES Attempt to continue to run the file system check. Often, however, the problem will persist. This error condition will not allow a complete check of the file system. A second run of *fsck* should be made to re-check this file system.

NO Terminate the program.

Phase 1 Errors: Check Blocks and Sizes

This phase concerns itself with the inode list. This section lists error conditions resulting from checking inode types, setting up the zero-link-count table, examining inode block numbers for bad or duplicate blocks, checking inode size, checking block count, and checking inode format. All errors in phase 1 are fatal if you are preening the file system, except for INCORRECT BLOCK COUNT and BAD INDIRECT ADDRESS.

CG ... : BAD MAGIC NUMBER

The magic number of cylinder group is wrong. This usually indicates that the cylinder group maps have been destroyed. When running manually, the cylinder group is marked as needing to be reconstructed.

UNKNOWN FILE TYPE I=... (CLEAR)

The mode word of the inode indicates that the inode is not a character special, block special, regular, network special, fifo, or directory inode.

Possible responses to the CLEAR prompt are:

YES De-allocate the inode by zeroing its contents. This will always invoke the UNALLOCATED error condition in Phase 2 for each directory entry pointing to this inode.

NO Ignore this error condition.

LINK COUNT TABLE OVERFLOW (CONTINUE)

An internal table for *fsck* containing allocated inodes with a link count of zero has no more room.

Possible responses to the CONTINUE prompt are:

YES Continue with the program. This error condition will not allow a complete check of the file system. A second run of *fsck* should be made to re-check this file system. If another allocated inode with a zero link count is found, this error condition is repeated.

NO Terminate the program.

"block_number" BAD I=...

The inode represented by "I=" contains the block number,"block_number". This block number is out of the range of the file system. This error condition may invoke the `EXCESSIVE BAD BLKS` error condition in phase 1 if this inode has too many block numbers outside the file system range. This error condition will always invoke the `BAD/DUP` error condition in Phase 2 and Phase 4.

EXCESSIVE BAD BLKS I=... (CONTINUE)

There are more than 10 blocks with a block number out of the range of the file system associated with the inode.

Possible responses to the CONTINUE prompt are:

YES Ignore the rest of the blocks in this inode and continue checking with the next inode in the file system. This error condition will not allow a complete check of the file system. A second run of *fsck* should be made to re-check this file system.

NO Terminate the program.

"block_number" DUP I=...

The inode contains block number, "block_number", which is already claimed by another inode. This error condition may invoke the `EXCESSIVE DUP BLKS` error condition in phase 1 if this inode has too many block numbers claimed by other inodes. This error condition will always invoke Phase 1b and the `BAD/DUP` error condition in Phase 2 and Phase 4.

EXCESSIVE DUP BLKS I=... (CONTINUE)

There are more than 10 blocks claimed by other inodes.

Possible responses to the CONTINUE prompt are:

YES Ignore the rest of the blocks in this inode and continue checking with the next inode in the file system. This error condition will not allow a complete check of the file system. A second run of *fsck* should be made to re-check this file system.

NO Terminate the program.

DUP TABLE OVERFLOW (CONTINUE)
An internal table in *fsck* containing duplicate block numbers is full.

Possible responses to the CONTINUE prompt are:

YES Continue with the program. This error condition will not allow a complete check of the file system. A second run of *fsck* should be made to re-check this file system. If another duplicate block is found, this error condition will repeat.

NO Terminate the program.

PARTIALLY ALLOCATED INODE I=... (CLEAR)
The inode is neither allocated nor unallocated.

Possible responses to the CLEAR prompt are:

YES De-allocate the inode by zeroing its contents.

NO Ignore this error condition.

INCORRECT BLOCK COUNT I=... (CORRECT)
The block count for the inode, *inode_number*, is *X* blocks, but should be *Y* blocks. When you are preening the count is corrected.

Possible responses to the CORRECT prompt are:

YES replace the block count of the inode with *Y*.

NO ignore this error condition.

BAD INDIRECT ADDRESS: IND BLOCK n[m] = val I=... (CORRECT)
An indirect address block, allocated in the inode indicated by "I=", contains block address for regions beyond the allocated size of the file. When you are preening, these entries are zeroed.

Possible responses to the CORRECT prompt are:

YES Zero the entry.

NO ignore this error condition. Later attempts by the operating system to extend the file into this region may cause a system crash.

Phase 1b: Rescan for more Dups

When a duplicate block is found in the file system, the file system is rescanned to find the inode which previously claimed that block. This section lists the error condition when the duplicate block is found.

"block_number" DUP I=...

The inode contains the block number, "block_number", which is already claimed by another inode. This error condition will always invoke the BAD/DUP error condition in Phase 2. You can determine which inodes have overlapping blocks by examining this error condition and the DUP error condition in Phase 1.

Phase 2: Check Path-Names

This phase concerns itself with removing directory entries pointing to error conditioned inodes from Phase 1 and Phase inode mode and status, directory inode pointers in range, and directory entries pointing to bad inodes. All errors in this phase are fatal if you are preening your file system.

ROOT INODE UNALLOCATED. TERMINATING

The root inode (inode number 2) has no allocated mode bits. This should never happen. The program will terminate. Contact your local HP Sales and Service Office for further assistance.

NAME TOO LONG

The path name shown is too long. This is usually indicative of loops in the file system name space. This can occur if the super user has made circular links to directories. The offending links must be removed.

ROOT INODE NOT DIRECTORY (FIX)

The root inode (inode number 2) is not directory inode type.

Possible responses to the FIX prompt are:

YES Change the root inode's type to be a directory. If the root inode's data blocks are not directory blocks, a very large number of error conditions will be produced.

NO Terminate the program.

DUPS/BAD IN ROOT INODE (CONTINUE)
Phase 1 or Phase 1b found duplicate blocks or bad blocks in the root inode (inode number 2) for the file system.

Possible responses to the CONTINUE prompt are:

YES Ignore the DUPS/BAD error condition in the root inode and attempt to continue to run the file system check. If the root inode is not correct, then this may result in a large number of other error conditions.

NO Terminate the program.

I OUT OF RANGE I=... (REMOVE)
NAME has an inode number (I), which is greater than the end of the inode list.

Possible responses to the REMOVE prompt are:

YES The directory entry (NAME) is removed.

NO Ignore this error condition.

UNALLOCATED I=... (REMOVE)
There are two possible error messages that start like this. One is for directory entries and one is for files. "DIR" or "FILE" has a directory inode (I) without allocated mode bits. The owner, mode, size, modify time, and directory/file name are printed.

Possible responses to the REMOVE prompt are:

YES The directory entry is removed.

NO Ignore this error condition.

DUP/BAD I=... (REMOVE)
There are two possible error messages that start like this. One is for directory entries and one is for files. Phase 1 or Phase 1b found duplicate blocks or bad blocks associated with the "DIR" or "FILE" having directory inode, "I". The owner, mode, size, modify time, and directory are printed.

Generally, the inode with the earliest modify time is incorrect, and should be cleared. To be safe, you should exit *fsck* (Break) and inspect the two files to determine which is corrupt, then re-run *fsck*.

Possible responses to the REMOVE prompt are:

YES The directory entry is removed.

NO Ignore this error condition.

ZERO LENGTH DIRECTORY I=... (REMOVE)
The directory entry's size is zero. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the REMOVE prompt are:

YES the directory entry is removed. This will always invoke the BAD/DUP error condition in phase 4.

NO ignore this error condition.

DIRECTORY TOO SHORT I=... (FIX)
The directory entry's size is less than the minimum size for a directory. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the FIX prompt are:

YES increase the size of the directory to the minimum directory size.

NO ignore this error condition.

DIRECTORY CORRUPTED I=... (FIX)
A directory entry has been found with an inconsistent internal state. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the FIX prompt are:

YES Throw away all entries up to the next directory boundary. This drastic action can throw away directory entries, and should be taken only after other recovery efforts have failed.

NO Skip to the next directory boundary and resume reading, but do not modify the directory.

BAD INODE NUMBER FOR '.' I=... (FIX)
The directory entry doesn't have an inode number for '.' which is equal to the inode number. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the FIX prompt are:

- YES — change the inode number for '.' to be equal to the inode number given after `I=`.
- NO — leave the inode number for '.' unchanged

MISSING '.' I=... (FIX)
The directory doesn't have its first directory entry allocated. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the FIX prompt are:

- YES — make an entry for '.' with inode number equal to the inode number given after `I=`.
- NO — leave the directory unchanged

MISSING '.' I=...
CANNOT FIX, FIRST ENTRY IN DIRECTORY CONTAINS ...
The directory has, as its first entry, the file name given. *fsck* cannot resolve this problem. The file system should be mounted and the offending entry moved elsewhere. To do this, exit the *fsck* program, mount the file system (you can force a mount by using *mount -f*, find the file name, and move it to a different directory. The file system should then be unmounted and *fsck* should be run again. The owner, mode, size, modify time, and directory name are printed.

MISSING '.' I=...
CANNOT FIX, INSUFFICIENT SPACE TO ADD '.'
The directory does not have '.' as its first entry. *fsck* cannot resolve this problem. If this happens, contact your local HP Sales and Service office. The owner, mode, size, modify time, and directory name are printed.

EXTRA '.' ENTRY I=... (FIX)
The directory has more than one entry for '.'. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the FIX prompt are:

YES remove the extra entry for '.'.

NO leave the directory unchanged

BAD INODE NUMBER FOR '..' I=... (FIX)
The directory's inode number for '..' does not equal the parent of the inode number (I). The owner, mode, size, modify time, and directory name are printed.

Possible responses to the FIX prompt are:

YES change the inode number for '..' to be equal to the parent of the inode given (I).

NO leave the inode number for '..' unchanged

MISSING '..' I=... (FIX)
The directory doesn't have its second directory entry allocated.

Possible responses to the FIX prompt are:

YES make an entry for '..' with inode number equal to the parent of the inode number given (I).

NO leave the directory unchanged

MISSING '..' I=...
CANNOT FIX, SECOND ENTRY IN DIRECTORY CONTAINS ...
The directory has, as its second entry, the file name given. *fsck* cannot resolve this problem. The file system should be mounted and the offending entry moved elsewhere. To do this, exit the *fsck* program, mount the file system (you can force a mount by using *mount -f*, find the file name, and The file system should then be unmounted and *fsck* should be run again.

MISSING '..' I=...
CANNOT FIX, INSUFFICIENT SPACE TO ADD '..'
The directory does not have '..' as its second entry. *fsck* cannot resolve this problem. If this happens, contact your local HP Sales and Service office.

EXTRA '..' ENTRY I=... (FIX)
The directory has more than one entry for '..'.

Possible responses to the FIX prompt are:

YES remove the extra entry for '..'.

NO leave the directory unchanged

Phase 3: Check Connectivity

This phase concerns itself with the directory connectivity seen in Phase 2. This section lists error conditions resulting from unreferenced directories, and missing or full */lost+found* directories.

UNREF DIR I=... (RECONNECT)
The directory inode (I) was not connected to a directory entry when the file system was traversed. The owner, mode, size, and modify time of the directory inode are printed. If you are preening, the directory is reconnected if its size is non-zero, otherwise it is cleared.

Possible responses to the RECONNECT prompt are:

YES Reconnect the directory inode to the file system in the directory for lost files (*/lost+found*). This may invoke the `lost+found` error condition if there are problems connecting the directory inode to */lost+found*. This may also invoke the CONNECTED error condition in phase 3 if the link was successful.

NO Ignore this error condition. This error will always invoke the UNREF error condition in Phase 4.

SORRY, NO lost+found DIRECTORY
There is no */lost+found* directory in the root directory of the file system. *fsck* ignores the request to link a directory in */lost+found*. This will always invoke the UNREF error condition in Phase 4. Check access modes of */lost+found*. This error is fatal if you are preening the system.

SORRY. NO SPACE IN lost+found DIRECTORY
There is no space to add another entry to the */lost+found* directory in the root directory of the file system. *fsck* ignores the request to link a directory in */lost+found*. This will always invoke the UNREF error condition in Phase 4. Clean out unnecessary entries in */lost+found* or make */lost+found* larger and try again. This error is fatal if you are preening the system.

DIR I=... PARENT WAS I=...
This is an advisory message indicating a directory inode was successfully connected to the */lost+found* directory. The parent inode of the directory inode is replaced by the inode number of the */lost+found* directory.

Phase 4: Check Reference Counts

This phase concerns itself with the link count information seen in Phase 2 and Phase 3. This section lists error conditions resulting from unreferenced files, missing or full */lost+found* directory, incorrect link counts for files, directories, or special files, unreferenced files and directories, bad and duplicate blocks in files and directories, and incorrect total free-inode counts. All errors in this phase are correctable if your are preening your file system, except if you run out of space in */lost+found*.

UNREF FILE I=... (RECONNECT)

The inode (I) was not connected to a directory entry when the file system was traversed. The owner, mode, size, and modify time of the inode are printed. If you are preening, the file is cleared if either its size, or its link count, is zero. Otherwise it is reconnected.

Possible responses to the RECONNECT prompt are:

YES Reconnect the inode to the file system in the directory for lost files (usually */lost+found*). This may invoke the `lost+found` error condition if there are problems connecting the inode to */lost+found*.

NO Ignore this error condition. This will always invoke the CLEAR error condition in phase 4.

(CLEAR)

The inode mentioned in the immediately previous error condition cannot be reconnected. If you are preening, this error cannot occur, since lack of space to reconnect files is a fatal error.

Possible responses to the CLEAR prompt are:

YES De-allocate the inode mentioned in the immediately previous error condition by zeroing its contents.

NO Ignore this error condition.

SORRY. NO lost+found DIRECTORY

There is no */lost+found* directory in the root directory of the file system. *fsck* ignores the request to link a file in */lost+found*. This will always invoke the CLEAR error condition in phase 4. Check access modes of */lost+found*. If you are preening your file system, this error is fatal.

SORRY. NO SPACE IN lost+found DIRECTORY
There is no space to add another entry to the */lost+found* directory in the root directory of the file system. *fsck* ignores the request to link a file in */lost+found*. This will always invoke the CLEAR error condition in phase 4. Check size and contents of */lost+found*. This error is fatal if you are preening your file system.

LINK COUNT... (ADJUST)
The link count for the file, directory, or inode is one link count (COUNT=) but should be a different link count (SHOULD BE). The owner, mode, size, and modify time are printed. If you are preening, the link count is adjusted.

Possible responses to the ADJUST prompt are:

YES	Replace the link count of the file in the inode with the "SHOULD BE: number.
NO	Ignore this error condition.

UNREF ... (CLEAR)
The file or directory with the inode number, "I", was not connected to a directory entry when the file system was traversed. The owner, mode, size, and modify time of the inode are printed. If you are preening, the inode is cleared, since this is a file that was not connected because its size or link count was zero.

Possible responses to the CLEAR prompt are:

YES	De-allocate inode by zeroing its contents.
NO	Ignore this error condition.

BAD/DUP ... (CLEAR)
Phase 1 or Phase 1b found duplicate blocks or bad blocks associated with the file or directory inode given in "I=". The owner, mode, size, and modify time of the inode are printed. This error will not occur if you are preening, since it would have caused a fatal error earlier.

Possible responses to the CLEAR prompt are:

YES	De-allocate the inode by zeroing its contents.
NO	Ignore this error condition.

Often deleting only one of the files containing DUPS will cure the problem. *fsck* should be re-run to confirm that the problem was fixed. A `NO` means that *fsck* must be re-run to finish cleaning up the file system.

FREE INODE COUNT WRONG IN SUPERBLK (FIX)
The actual count of the free inodes does not match the count in the super-block of the file system. If you are preening, the count is fixed.

Possible responses to the FIX prompt are:

YES	Replace the count in the super-block by the actual count.
NO	Ignore this error condition.

Phase 5: Check Cyl groups

This phase concerns itself with the free-block maps. This section lists error conditions resulting from allocated blocks in the free-block maps, free-blocks missing from free-block maps, and the total free-block count not matching the count contained in the super-block.

CG ...: BAD MAGIC NUMBER
The magic number of the cylinder group is wrong. This usually indicates that the cylinder group maps have been destroyed. When running manually, the cylinder group is marked as needing to be reconstructed. If you are preening your system, this error is fatal.

EXCESSIVE BAD BLKS IN BIT MAPS (CONTINUE)
You should never get this message. If you do, contact your local HP Sales and Service office.

SUMMARY INFORMATION "t" BAD
where *t* is one or more of:

```
(INODE FREE)
(BLOCK OFFSETS)
(FRAG SUMMARIES)
(SUPER BLOCK SUMMARIES)
```

The indicated summary information was found to be incorrect. This error condition will always invoke the BAD CYLINDER GROUPS condition in phase 6. If you are preening, the summary information is recomputed.

"x" BLK(S) MISSING
A number of blocks (x) that are unused by the file system were not found in the free-block maps. This error condition will always invoke the BAD CYLINDER GROUPS condition in phase 6. If you are preening, the block maps are rebuilt.

FREE BLK COUNT WRONG IN SUPERBLOCK (FIX)
The actual count of free blocks does not match the count in the super-block of the file system. If you are preening, the counts are fixed.

Possible responses to the FIX prompt are:

YES	Replace the count in the super-block by the actual count.
NO	Ignore this error condition.

BAD CYLINDER GROUPS (FIX)
Phase 5 has found bad blocks in the free-block maps, duplicate blocks in the free-block maps, or blocks missing from the file system. If you are preening, the cylinder groups are reconstructed.

Possible responses to the FIX prompt are:

YES	Replace the actual free-block maps with new free-block maps.
NO	Ignore this error condition.

Phase 6: Salvage Cylinder Groups

This phase concerns itself with reconstructing the free-block maps. No error messages are produced.

Cleanup

Once a file system has been checked, a few cleanup functions are performed. This section lists advisory messages about the file system and modify status of the file system.

"f" files, "b" used, "r" free ("y" frags, "z" blocks)
This message indicates that the file system just checked has a total of "f" files, using "b" fragment-sized blocks, with "r" fragment-sized blocks available (free) for use. The numbers in parenthesis divides the free count into "y" free fragments and "z" free full sized blocks.

No action is required on your part.

******* REBOOT HP-UX *******

This message indicates that the root file system has been modified by *fsck*. If HP-UX is not re-booted immediately, the work done by *fsck* may be undone by the in-core (memory) copies of tables HP-UX keeps. If you are preening, *fsck* will exit with a code of 4. The *bcheckrc* script interprets an exit code of 4 by executing the *reboot* command.

NOTE

If you are preening, you will not get this message. *bcheckrc* will correctly interpret the exit code and automatically reboot, but if you execute *fsck -p* outside the *bcheckrc* program, you must check the return code to see if the system needs to be rebooted.

******* FILE SYSTEM WAS MODIFIED *******

This message indicates that the current file system was modified by *fsck*. If this file system is mounted or is the current root file system, *fsck* should be halted and HP-UX re-booted. If HP-UX is not re-booted immediately, the work done by *fsck* may be undone by the in-core (memory) copies of tables HP-UX keeps.

Notes

System Accounting Files B

Descriptions of the different files processed by HP-UX System Accounting are found in this appendix. The files are grouped according to the directory in which they are found.

Files in the /usr/adm directory

Filename	Contents
diskdiag	Diagnostic output from the execution of disc space accounting commands.
dtmp	Output from the *acctdusg* program.
fee	Output from the *fchargefee* command (ASCII total accounting records).
pacct	The current active process accounting file.
pacct?	Process accounting files switched via *turnacct switch*.

Files in the /usr/adm/acct/nite directory

Filename	Contents
active	Used by *runacct* to record progress. It contains warning and error messages. *activeMMDD* is the same as *active* after *runacct* detects an error.
ctacct.MMDD	Total accounting records created from connect session accounting.
ctmp	Output of *acctcon1*—connect session records.
daycms	ASCII daily command summary used by *prdaily*.
daytacct	Total accounting records for current day.
disktacct	Total accounting records created by the *dodisk* command.
fd2log	Diagnostic output from the execution of *runacct* (see *crontab* entry).
lastdate	The last day that *runacct* was executed, in *date +%m%d* format. (See *date(1)* for a description of *+%m%d* date format.)
lock & lock1	Used to control serial use of *runacct*.
lineuse	Terminal (tty) line usage report used by *prdaily*.
log	Diagnostics output from *acctcon1*.

logMMDD	Same as *log* after *runacct* detects an error.
reboots	Contains beginning and ending dates from *wtmp*, and a listing of reboots.
statefile	Used to record the current state being executed by *runacct*.
tmpwtmp	*wtmp* file, corrected by *wtmpfix*.
wtmperror	Error messages, if any, from *wtmpfix*.
wtmperrorMMDD	Same as *wtmperror* after *runacct* detects an error.
wtmp.MMDD	The previous day's *wtmp* file.

Files in the /usr/adm/acct/sum directory

Filename	Contents
cms	Total command summary file for current month in internal summary format.
cmsprev	Command summary file without latest update.
daycms	Command summary file for previous day in internal summary format.
loginlog	Shows the last login date for each user.
pacct.MMDD	Concatenated version of all process accounting files for the date *MMDD*. This file is removed after reboot.
rptMMDD	Daily accounting report for date *MMDD*.
tacct	Cumulative total accounting file for current month.
tacctprev	Same as *tacct* without latest update.
tacctMMDD	Total accounting file for date *MMDD*.
wtmp.MMDD	Saved copy of wtmp file for *MMDD*. Removed after reboot.

Files in the /usr/adm/acct/fiscal directory

Filename	Contents
cms?	Total command summary for month *?* in internal summary format.
fiscrpt?	Report similar to *prdaily* for the month *?*.
tacct?	Total accounting file for the month *?*.

Partitions and File Sets **C**

Your set of installation flexible discs (or tapes) demonstrates the concept of *partitions*. HP-UX is broken into **partitions** or parts and stored on micro discs (or on one tape). Each disc represents a subset or subsets of the whole HP-UX system. Each partition is composed of file sets. File sets are the smallest unit in installation. You can choose the file sets you want loaded onto the system. However, the file sets which make up the SYS_CORE partition are automatically loaded on your system by the installation (see "Mandatory vs. Non-mandatory File Sets" below). After these file sets are loaded, you can load other file sets depending on your needs.

By separating HP-UX into file sets, you can install only those file sets you will use (and save time and disc space). For example, if you do not plan to use a part of HP-UX (such as a text editor) you have the option of not loading the file set containing it on the system.

Except for guarding against the possibility of trying to install partitions or file sets which will not fit on your hard disc, not much planning is required. In the installation procedure you have a great deal of flexibility in what you want on (or off) the system. This flexibility also means you have to make several key decisions when installing.

How Partitions and File Sets Work

You can think of partitions, file sets, and files as "building blocks" of the AXE system. The smallest unit of information in this scheme is a **file**. Groups of related files are grouped into **file sets**. File sets, in turn, are grouped into **partitions**. Finally, the partitions — taken as a whole — comprise the HP-UX system. As noted above, when you receive HP-UX, the system will be partitioned into color coded discs (if you received HP-UX in a set of micro discs). Each of the color-coded partitions represents a grouping of file sets; each file set represents a grouping of files. The following diagram illustrates the relationship between the HP-UX system, partitions, file sets, and files.

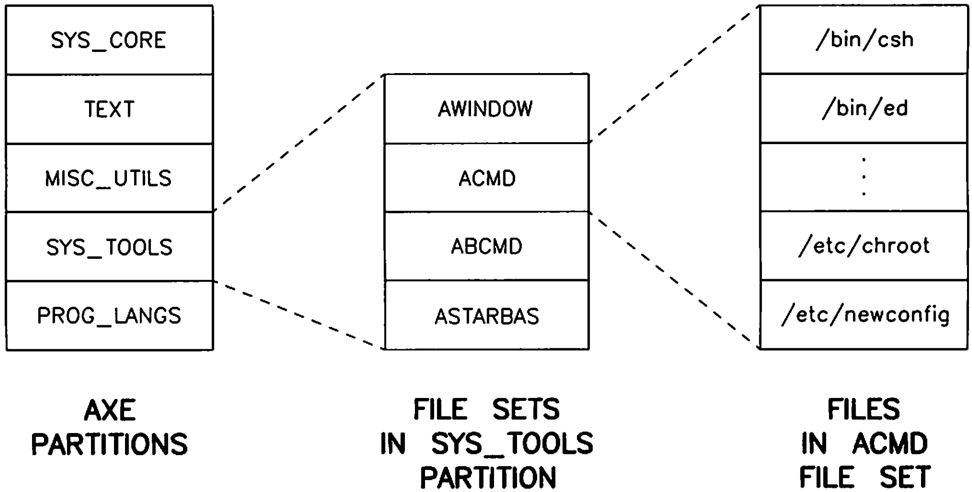


Figure C-1: HP-UX partitions, file sets and files.

Dependent File Sets

In some cases, one file set will depend on other file sets being present in the system. If such a **dependency** exists, the file set must be loaded along with (either before or after) the file set in question. When trying to load a file set that has a dependency, the computer informs you of the dependency. If necessary, the computer prompts you to insert a different disc to load from; it specifies which disc or tape to use. If more than one dependency exists, the computer may prompt for yet another disc to load from.

File sets are never split among discs although a given file set may have dependencies on other discs.

Mandatory vs. Non-mandatory Partitions

Only one partition **must** be installed for your HP-UX system to function: SYS_CORE. This partition contains the ACORE file sets which in turn contain essential facilities and commands used by other file sets. All the other partitions contain optional file sets. The SYS_CORE partition is **mandatory** for the operation of HP-UX and will be automatically loaded by the installation procedure. It is on the micro discs which have no colored labels. You can run HP-UX with only this single partition loaded.

All the other partitions are considered "non-mandatory"; they need not be installed for the proper operation of HP-UX. However, for most needs we recommend you install the SYS_TOOLS partition as well if you have enough disc space. This partition contains fundamental tools for file management, file manipulation, process control, peripheral support, and HP Windows/9000. The SYS_TOOLS partition is on the micro discs with the red labels.

Besides these two partitions, loading all other partitions and file sets depends on your intended use of the system. This factor must be weighed against your other needs for available storage space on your hard disc (or RAM, Random Access Memory, in the case of HP Windows/9000).

Which File Sets Do I Want?

After reading about what a file set is, you need to determine which file sets you feel you want on the system. All of the file sets are listed in this appendix with a short description. As you read through the list, you may wish to mark the file sets you feel you want on the system. Then, during installation, you can refer to this list for the file sets you want installed.

Recovering Lost Files

Most of the commands in the HP-UX system are contained in the file sets documented in this appendix. If any of these commands are removed or somehow destroyed, you can use *update* to re-load the file set containing the commands. In order to run update, you need to recover certain essential files. You can mount the installation media (tape or discs 1 and 2) and recover these files:

```
Micro disc 1:
etc/mkfs
mediainit

Micro disc 2:
bin/cpio
bin/mkdir
bin/sh
bin/pwd
usr/bin/lifcp
usr/bin/tcio
etc/update
etc/sysrm
```

NOTE

These files are not included in any one file set. Therefore, you cannot use update to recover them. The files are essential for update to be able to run. So, if these files get destroyed, you **must** recover them before you can run update.

Recovering Files

If you installed from a cartridge tape:

1. Make sure the tape is writable.
2. Insert tape in drive; wait for the drive light to remain off.
3. Mount the tape to an empty folder using:

   ```
   mount /dev/ct /disc
   ```

   where `/dev/ct` is a block device.

4. Perform the following copies depending on the file you are recovering:

NOTE

You can copy any or all of these files if needed.

```
cp /disc/etc/mkfs /etc
cp /disc/mediainit /usr/bin
cp /disc/bin/cpio /bin
cp /disc/bin/mkdir /bin
cp /disc/bin/sh /bin
cp /disc/bin/pwd /bin
cp /disc/usr/bin/lifcp /usr/bin
cp /disc/urs/bin/tcio /usr/bin
cp /disc/etc/update /etc
cp /disc/etc/sysrm /etc
```

5. Type: `sync`
6. Type: `umount /dev/ct`

If you installed from micro discs:

1. Make sure the first two micro discs are writable.
2. Insert micro disc 1 (with the grey label); wait for the drive light to remain off.
3. Mount the micro disc to an empty folder:

   ```
   mount /dev/fd /disc
   ```

   where `/dev/fd` must be a block device.
4. Enter the commands: `cp /disc/etc/mkfs /etc`
   `cp /disc/mediainit /usr/bin`
5. Unmount the first micro disc: `umount /dev/fd`
6. Insert micro disc 2 (grey label) into drive; wait for the drive light to remain off.
7. Mount the micro disc to an empty folder: `mount /dev/fd /disc`
   where `/dev/fd` must be a block device.

8. Enter any or all of the following commands:

```
cp /disc/bin/cpio /bin
cp /disc/bin/mkdir /bin
cp /disc/bin/pwd /bin
cp /disc/etc/update /etc
cp /disc/etc/sysrm /etc
cp /disc/usr/bin/lifcp /usr/bin
cp /disc/usr/bin/tcio /usr/bin
```

9. Unmount the second micro disc: `umount /dev/fd`

Partitions and Their File Sets

Table C-1 lists all file sets, the partitions they are contained in, and their dependencies.

Table C-1: File Sets and Partitions

Fileset	Size[1]	Partition	System	Dependencies
A98515A	688	SYS_CORE	AXE	AXE Core filesets
A98670B	788	SYS_CORE	PE	AXE Core filesets
A98680B	788	SYS_CORE	PE	AXE Core filesets
ACORE	720	SYS_CORE	AXE	AXE Core filesets
ACORE2	1331	SYS_CORE	AXE	AXE Core filesets
ACORE3	1335	SYS_CORE	AXE	AXE Core filesets
ACORE4	1295	SYS_CORE	AXE	AXE Core filesets
ACORE5	168	SYS_CORE	AXE	AXE Core filesets
A98597A	800	SYS_CORE	PE	
APCORE	1210	SYS_CORE	PE	Core filesets
APCORE2	1143	SYS_CORE	PE	Core filesets
AFTN_CORE	302	SYS_CORE	PE	Core filesets
APAS_CORE	210	SYS_CORE	PE	Core filesets
AACMD	1250	SYS_TOOLS	AXE	
ABCMD	1130	SYS_TOOLS	AXE	
AASTARBAS	635	SYS_TOOLS	AXE	AAWINDOW
AAWINDOW	607	SYS_TOOLS	AXE	AASTARBAS
ADSTARBAS	1317	SYS_TOOLS	PE	AASTARBAS,APSTARBAS, AAC,AAPROG,APC
APSTARBAS	1227	SYS_TOOLS	PE	ASTARBAS,AC,APROG,APC
AMSTARBAS	414	SYS_TOOLS	PE	
ADWINDOW	607	SYS_TOOLS	PE	AAWINDOW,APWINDOW
APWINDOW	206	SYS_TOOLS	PE	AAWINDOW,AASTARBAS
AMWINDOW	526	SYS_TOOLS	PE	
APSYSCOM	953	SYS_TOOLS	PE	APSYSCOM2
APSYSCOM2	843	SYS_TOOLS	PE	APSYSCOM
APEDITOR	316	SYS_TOOLS	PE	
APFILEMGR	119	SYS_TOOLS	PE	

[1] Size is in 512 byte blocks.

Table C-1: File Sets and Partitions (cont'd)

Fileset	Size[1]	Partition	System	Dependencies
APFILTER	698	SYS_TOOLS	PE	
APNLS	266	SYS_TOOLS	PE	
APCEUTILS	433	SYS_TOOLS	PE	
AAC	847	PROG_LANGS	AXE	
AAPROG	411	PROG_LANGS	AXE	
APC	863	PROG_LANGS	PE	APC2
APC2	735	PROG_LANGS	PE	APC
APFORTRAN	56	PROG_LANGS	PE	
APPROG	1252	PROG_LANGS	PE	APPROG2
APPROG2	752	PROG_LANGS	PE	APPROG
A98519A	936	PROG_LANGS	PE	AAC,AAPROG,APPROG, APPROG2
A98599A	935	PROG_LANGS	PE	AAC,AAPROG,APPROG, APPROG2
A98518A	775	PROG_LANGS	PE	AAC,AAPROG,APPROG, APPROG2
A98598A	774	PROG_LANGS	PE	AAC,AAPROG,APPROG, APPROG2
AACONFIG	1229	MISC_UTILS	AXE	AACONFIG2,AAC,AAPROG
AACONFIG2	1229	MISC_UTILS	AXE	AACONFIG,AAC,AAPROG
APAPPLIC	840	MISC_UTILS	PE	
APPERIPH	158	MISC_UTILS	PE	
APSCCS	539	MISC_UTILS	PE	
APACCT	749	MISC_UTILS	PE	
A50951A	1397	MISC_UTILS	PE	AACONFIG,AACONFIG2, AAC,AAPROG
A50952A	1397	MISC_UTILS	PE	AACONFIG,AACONFIG2, AAC,AAPROG
A50956A	1397	MISC_UTILS	PE	AACONFIG,AACONFIG2, AAC,AAPROG
A50957A	1397	MISC_UTILS	PE	AACONFIG,AACONFIG2, AAC,AAPROG

Table C-1: File Sets and Partitions (cont'd)

Fileset	Size[1]	Partition	System	Dependencies
A98693A	508	MISC_UTILS	PE	
AAMANUAL	1079	TEXT	AXE	
AHPUX_MAN	993	TEXT	PE	AHPUX_MAN2,AHPUX_MAN3, AHPUX_MAN4,APTEXT, APTEXT2
AHPUX_MAN2	888	TEXT	PE	AHPUX_MAN,AHPUX_MAN3, AHPUX_MAN4,APTEXT, APTEXT2
AHPUX_MAN3	1066	TEXT	PE	AHPUX_MAN,AHPUX_MAN2, AHPUX_MAN4,APTEXT, APTEXT2
AHPUX_MAN4	1110	TEXT	PE	AHPUX_MAN,AHPUX_MAN2, AHPUX_MAN3,APTEXT, APTEXT2
APTEXT	1011	TEXT	PE	
APTEXT2	588	TEXT	PE	
A98520A	1291	GRAPHICS	PE	A98520A - A98520A4
A98520A2	1412	GRAPHICS	PE	"
A98520A3	1438	GRAPHICS	PE	"
A98520A4	334	GRAPHICS	PE	"
A98600A	1290	GRAPHICS	PE	A98600A - A98600A4
A98600A2	1412	GRAPHICS	PE	"
A98600A3	1438	GRAPHICS	PE	"
A98600A4	333	GRAPHICS	PE	"
A98683X	1183	GRAPHICS	PE	

The file set list that follows is broken into sections, each representing a partition of the system. Within each section will be sub-sections which describe each file set, and list the files associated with the file set.

INSTALL NOTE

This first set of files is NOT a real file set list; these commands are put on the disc by the installation process.

```
bin
bin/mkdir
bin/pwd
bin/sh
bin/cpio
usr
usr/bin
usr/bin/mediainit
usr/bin/tcio
usr/bin/lifcp
etc
etc/mkfs
```

PARTITION: SYS_CORE

This partition is contained on the grey labeled micro discs. There are 6 micro discs to the SYS_CORE partition. These micro discs are manditory for AXE installation and minimal operation (the file sets in this partition are loaded automatically). The first two micro discs are used for the installation procedure.

FILE SET: 98515A

Description: Manditory file set.

```
hp-ux
etc
etc/conf
etc/conf/dfile
system
system/98515A
system/98515A/customize
```

FILE SET: A98670B

Description: Single-user kernel for S200

```
hp-ux
etc
etc/conf
etc/conf/dfile
system
system/98670B
system/98670B/customize
```

FILE SET: A98680B

Description: Multi-user kernel for S200

```
hp-ux
etc
etc/conf
etc/conf/dfile
system
system/98680B
system/98680B/customize
```

FILE SET: ACORE

Description: Manditory file set.

```
bin
bin/basename
bin/date
bin/expr
bin/line
bin/ls
bin/mv
bin/passwd
bin/rm
bin/rmdir
bin/sed
bin/stty
bin/sync
bin/who
etc
etc/conf
etc/conf/.amigo.help
etc/conf/.ciper.help
```

etc/conf/.del_user.help
etc/conf/.exit.help
etc/conf/.exit2.help
etc/conf/.int.help
etc/conf/.intro.help
etc/conf/.lp.help
etc/conf/.ncip.help
etc/conf/.os_opt.help
etc/conf/.plotter.help
etc/conf/.printer.help
etc/conf/.prod.help
etc/conf/.security.help
etc/conf/.stape.help
etc/conf/.tape.help
etc/conf/.term.help
etc/conf/.user.help
etc/init
etc/initconfig
usr/lib/mv_dir

FILE SET: ACORE2

Description: Manditory file set.

bin
bin/cat
bin/dd
bin/dirname
bin/echo
bin/find
bin/grep
bin/ipcrm
bin/ipcs
bin/kill
bin/login
bin/ps
bin/sleep
bin/su
etc
etc/backupf
etc/boot
etc/devnm
etc/fsck
etc/fsclean
etc/getanswer
etc/getty
etc/killall
etc/link
etc/mknod
etc/mount
etc/mvdir
etc/newconfig
etc/newconfig/.profile
etc/newconfig/README
etc/newconfig/Update_info
etc/newconfig/backup
etc/newconfig/bcheckrc
etc/newconfig/brc
etc/newconfig/checklist
etc/newconfig/disktab
etc/newconfig/gettydefs
etc/newconfig/group
etc/newconfig/inittab
etc/newconfig/motd
etc/newconfig/multi_user
etc/newconfig/passwd
etc/newconfig/profile
etc/newconfig/pstatus
etc/newconfig/qstatus
etc/newconfig/rc
etc/newconfig/single_user
etc/newconfig/ttytype
etc/newfs
etc/reboot
etc/reconfig
etc/setmnt
etc/shutdown
etc/syncer
etc/umount
etc/unlink
etc/wall

FILE SET: ACORE3

Description: Manditory file set.

bin/chgrp
bin/chmod
bin/chown
bin/touch
bin/uname
bin/wc
etc/swapon
usr
usr/bin
usr/bin/banner
usr/bin/cut
usr/bin/hostname
usr/bin/tset
usr/lib
usr/lib/hp2686a
usr/lib/hp2934a
usr/lib/terminfo
usr/lib/terminfo/1
usr/lib/terminfo/1/100
usr/lib/terminfo/1/110
usr/lib/terminfo/1/150
usr/lib/terminfo/1/1520
usr/lib/terminfo/1/1521
usr/lib/terminfo/1/1620

usr/lib/terminfo/1/1620-m8
usr/lib/terminfo/1/1640
usr/lib/terminfo/1/1640-m8
usr/lib/terminfo/1/1700
usr/lib/terminfo/1/1line
usr/lib/terminfo/1/1linepty
usr/lib/terminfo/2
usr/lib/terminfo/2/2382
usr/lib/terminfo/2/2392
usr/lib/terminfo/2/2393
usr/lib/terminfo/2/2394
usr/lib/terminfo/2/2397
usr/lib/terminfo/2/2500
usr/lib/terminfo/2/2621
usr/lib/terminfo/2/2621-48
usr/lib/terminfo/2/2621-ba
usr/lib/terminfo/2/2621-fl
usr/lib/terminfo/2/2621-nl
usr/lib/terminfo/2/2621-nt
usr/lib/terminfo/2/2621-pb
usr/lib/terminfo/2/2621-wl
usr/lib/terminfo/2/2621A
usr/lib/terminfo/2/2621P
usr/lib/terminfo/2/2621a
usr/lib/terminfo/2/2621k45
usr/lib/terminfo/2/2621nl
usr/lib/terminfo/2/2621nt
usr/lib/terminfo/2/2621p
usr/lib/terminfo/2/2621wl
usr/lib/terminfo/2/2622
usr/lib/terminfo/2/2622a
usr/lib/terminfo/2/2622p
usr/lib/terminfo/2/2623
usr/lib/terminfo/2/2623a
usr/lib/terminfo/2/2623p
usr/lib/terminfo/2/2624
usr/lib/terminfo/2/2624a
usr/lib/terminfo/2/2624p
usr/lib/terminfo/2/2625
usr/lib/terminfo/2/2626
usr/lib/terminfo/2/2626-12
usr/lib/terminfo/2/2626-12-s
usr/lib/terminfo/2/2626-12x40
usr/lib/terminfo/2/2626-ns
usr/lib/terminfo/2/2626-s
usr/lib/terminfo/2/2626-x40
usr/lib/terminfo/2/2626A
usr/lib/terminfo/2/2626P
usr/lib/terminfo/2/2626a
usr/lib/terminfo/2/2626p
usr/lib/terminfo/2/2627
usr/lib/terminfo/2/2627a
usr/lib/terminfo/2/2627p
usr/lib/terminfo/2/2628
usr/lib/terminfo/2/262x
usr/lib/terminfo/2/2640
usr/lib/terminfo/2/2640a
usr/lib/terminfo/2/2640b
usr/lib/terminfo/2/2644
usr/lib/terminfo/2/2645
usr/lib/terminfo/2/2647
usr/lib/terminfo/2/2647F
usr/lib/terminfo/2/2648
usr/lib/terminfo/2/2648A
usr/lib/terminfo/2/2648a
usr/lib/terminfo/2/2703
usr/lib/terminfo/2/2709
usr/lib/terminfo/3
usr/lib/terminfo/3/300
usr/lib/terminfo/3/300s
usr/lib/terminfo/3/3045
usr/lib/terminfo/3/31
usr/lib/terminfo/3/3101
usr/lib/terminfo/3/33
usr/lib/terminfo/3/333
usr/lib/terminfo/3/35
usr/lib/terminfo/3/36
usr/lib/terminfo/3/37
usr/lib/terminfo/3/380
usr/lib/terminfo/3/382
usr/lib/terminfo/3/3a
usr/lib/terminfo/3/3a+
usr/lib/terminfo/4
usr/lib/terminfo/4/40
usr/lib/terminfo/4/400
usr/lib/terminfo/4/4012
usr/lib/terminfo/4/4013
usr/lib/terminfo/4/4014
usr/lib/terminfo/4/4014-sm
usr/lib/terminfo/4/4015
usr/lib/terminfo/4/4015-sm
usr/lib/terminfo/4/4023
usr/lib/terminfo/4/4024
usr/lib/terminfo/4/4025
usr/lib/terminfo/4/4025-17
usr/lib/terminfo/4/4025-17ws
usr/lib/terminfo/4/4025cu
usr/lib/terminfo/4/4025ex
usr/lib/terminfo/4/4027
usr/lib/terminfo/4/4027-17
usr/lib/terminfo/4/4027-17ws
usr/lib/terminfo/4/4027cu
usr/lib/terminfo/4/4027ex
usr/lib/terminfo/4/4080
usr/lib/terminfo/4/4112
usr/lib/terminfo/4/4112-5
usr/lib/terminfo/4/4112-d
usr/lib/terminfo/4/4112-nd
usr/lib/terminfo/4/4113
usr/lib/terminfo/4/4114

usr/lib/terminfo/4/42
usr/lib/terminfo/4/42-nl
usr/lib/terminfo/4/43
usr/lib/terminfo/4/4424
usr/lib/terminfo/4/4424-2
usr/lib/terminfo/4/4424-el-2
usr/lib/terminfo/4/45
usr/lib/terminfo/4/450
usr/lib/terminfo/5
usr/lib/terminfo/5/5520
usr/lib/terminfo/7
usr/lib/terminfo/7/735
usr/lib/terminfo/7/743
usr/lib/terminfo/7/745
usr/lib/terminfo/8
usr/lib/terminfo/8/8001
usr/lib/terminfo/8/8510
usr/lib/terminfo/9
usr/lib/terminfo/9/9020
usr/lib/terminfo/9/912
usr/lib/terminfo/9/912-2p
usr/lib/terminfo/9/9122p
usr/lib/terminfo/9/912b
usr/lib/terminfo/9/912c
usr/lib/terminfo/9/912cc
usr/lib/terminfo/9/920
usr/lib/terminfo/9/920-2p
usr/lib/terminfo/9/9202p
usr/lib/terminfo/9/920b
usr/lib/terminfo/9/920c
usr/lib/terminfo/9/925
usr/lib/terminfo/9/950
usr/lib/terminfo/9/950-2p
usr/lib/terminfo/9/950-4p
usr/lib/terminfo/9/950-rv
usr/lib/terminfo/9/950-rv-2p
usr/lib/terminfo/9/950-rv-4p
usr/lib/terminfo/9/9502p
usr/lib/terminfo/9/9504p
usr/lib/terminfo/9/950rv
usr/lib/terminfo/9/950rv2p
usr/lib/terminfo/9/950rv4p
usr/lib/terminfo/9/9816te
usr/lib/terminfo/9/9816teb
usr/lib/terminfo/9/9826
usr/lib/terminfo/9/9826ite
usr/lib/terminfo/9/9835
usr/lib/terminfo/9/9836
usr/lib/terminfo/9/9836ite
usr/lib/terminfo/9/9836te
usr/lib/terminfo/9/9836teb
usr/lib/terminfo/9/9837
usr/lib/terminfo/9/9837ite
usr/lib/terminfo/9/9845
usr/lib/terminfo/9/98700
usr/lib/terminfo/a
usr/lib/terminfo/a/a980
usr/lib/terminfo/a/aa
usr/lib/terminfo/a/aaa
usr/lib/terminfo/a/aaa-18
usr/lib/terminfo/a/aaa-18-rv
usr/lib/terminfo/a/aaa-20
usr/lib/terminfo/a/aaa-22
usr/lib/terminfo/a/aaa-24
usr/lib/terminfo/a/aaa-24-rv
usr/lib/terminfo/a/aaa-26
usr/lib/terminfo/a/aaa-28
usr/lib/terminfo/a/aaa-29
usr/lib/terminfo/a/aaa-29-ctxt
usr/lib/terminfo/a/aaa-29-np
usr/lib/terminfo/a/aaa-29-rv
usr/lib/terminfo/a/aaa-29-rv-ctxt
usr/lib/terminfo/a/aaa-30
usr/lib/terminfo/a/aaa-30-ctxt
usr/lib/terminfo/a/aaa-30-rv
usr/lib/terminfo/a/aaa-30-rv-ctxt
usr/lib/terminfo/a/aaa-30-s
usr/lib/terminfo/a/aaa-30-s-ctxt
usr/lib/terminfo/a/aaa-30-s-rv
usr/lib/terminfo/a/aaa-30-s-rv-ct
usr/lib/terminfo/a/aaa-36
usr/lib/terminfo/a/aaa-36-rv
usr/lib/terminfo/a/aaa-40
usr/lib/terminfo/a/aaa-40-rv
usr/lib/terminfo/a/aaa-48
usr/lib/terminfo/a/aaa-48-rv
usr/lib/terminfo/a/aaa-59
usr/lib/terminfo/a/aaa-60
usr/lib/terminfo/a/aaa-60-rv
usr/lib/terminfo/a/aaa-60-s
usr/lib/terminfo/a/aaa-60-s-rv
usr/lib/terminfo/a/aaa-ctxt
usr/lib/terminfo/a/aaa-db
usr/lib/terminfo/a/aaa-rv
usr/lib/terminfo/a/aaa-rv-ctxt
usr/lib/terminfo/a/aaa-rv-unk
usr/lib/terminfo/a/aaa-s
usr/lib/terminfo/a/aaa-s-rv
usr/lib/terminfo/a/aaa-unk
usr/lib/terminfo/a/aaa18
usr/lib/terminfo/a/aaa20
usr/lib/terminfo/a/aaa22
usr/lib/terminfo/a/aaa24
usr/lib/terminfo/a/aaa26
usr/lib/terminfo/a/aaa28
usr/lib/terminfo/a/aaa29
usr/lib/terminfo/a/aaa30
usr/lib/terminfo/a/aaa36
usr/lib/terminfo/a/aaa40
usr/lib/terminfo/a/aaa48

usr/lib/terminfo/a/aaa59
usr/lib/terminfo/a/aaa60
usr/lib/terminfo/a/aaadb
usr/lib/terminfo/a/act4
usr/lib/terminfo/a/act5
usr/lib/terminfo/a/act5s
usr/lib/terminfo/a/addrinfo
usr/lib/terminfo/a/adm1
usr/lib/terminfo/a/adm1a
usr/lib/terminfo/a/adm2
usr/lib/terminfo/a/adm21
usr/lib/terminfo/a/adm3
usr/lib/terminfo/a/adm31
usr/lib/terminfo/a/adm36
usr/lib/terminfo/a/adm3a
usr/lib/terminfo/a/adm3a+
usr/lib/terminfo/a/adm3aplus
usr/lib/terminfo/a/adm42
usr/lib/terminfo/a/adm42-nl
usr/lib/terminfo/a/adm5
usr/lib/terminfo/a/aed
usr/lib/terminfo/a/aed512
usr/lib/terminfo/a/agile
usr/lib/terminfo/a/agiles
usr/lib/terminfo/a/aj
usr/lib/terminfo/a/aj830
usr/lib/terminfo/a/aj832
usr/lib/terminfo/a/alto
usr/lib/terminfo/a/altoh19
usr/lib/terminfo/a/altoheath
usr/lib/terminfo/a/ambas
usr/lib/terminfo/a/ambassador
usr/lib/terminfo/a/ampex
usr/lib/terminfo/a/annarbor
usr/lib/terminfo/a/ansi
usr/lib/terminfo/a/apple
usr/lib/terminfo/a/arpanet
usr/lib/terminfo/b
usr/lib/terminfo/b/bantam
usr/lib/terminfo/b/bc
usr/lib/terminfo/b/beacon
usr/lib/terminfo/b/beehiveIIm
usr/lib/terminfo/b/bg
usr/lib/terminfo/b/bg-ni
usr/lib/terminfo/b/bg-nv
usr/lib/terminfo/b/bg-rv
usr/lib/terminfo/b/bg1.25
usr/lib/terminfo/b/bg1.25-nv
usr/lib/terminfo/b/bg1.25-rv
usr/lib/terminfo/b/bg2.0
usr/lib/terminfo/b/bg2.0-nv
usr/lib/terminfo/b/bg2.0-rv
usr/lib/terminfo/b/bh3m
usr/lib/terminfo/b/bitgraph
usr/lib/terminfo/b/bitgraph-ni
usr/lib/terminfo/b/bitgraph-nv
usr/lib/terminfo/b/bitgraph-rv
usr/lib/terminfo/b/blit
usr/lib/terminfo/b/blit-pb
usr/lib/terminfo/b/blitlayer
usr/lib/terminfo/b/bussiplexer
usr/lib/terminfo/c
usr/lib/terminfo/c/c100
usr/lib/terminfo/c/c100-1p
usr/lib/terminfo/c/c100-4p
usr/lib/terminfo/c/c100-rv
usr/lib/terminfo/c/c100-rv-4p
usr/lib/terminfo/c/c100-rv-4p-na
usr/lib/terminfo/c/c100-rv-4p-pp
usr/lib/terminfo/c/c100-rv-na
usr/lib/terminfo/c/c100-rv-pp
usr/lib/terminfo/c/c1004p
usr/lib/terminfo/c/c100rv
usr/lib/terminfo/c/c100rv4p
usr/lib/terminfo/c/c100rv4pna
usr/lib/terminfo/c/c100rv4ppp
usr/lib/terminfo/c/c100rvna
usr/lib/terminfo/c/c100rvpp
usr/lib/terminfo/c/c100rvs
usr/lib/terminfo/c/c100s
usr/lib/terminfo/c/c104
usr/lib/terminfo/c/c108
usr/lib/terminfo/c/c108-4
usr/lib/terminfo/c/c108-4p
usr/lib/terminfo/c/c108-8
usr/lib/terminfo/c/c108-8p
usr/lib/terminfo/c/c108-8p-na
usr/lib/terminfo/c/c108-8p-rv-na
usr/lib/terminfo/c/c108-na
usr/lib/terminfo/c/c108-na-8p
usr/lib/terminfo/c/c108-rv
usr/lib/terminfo/c/c108-rv-4p
usr/lib/terminfo/c/c108-rv-8p
usr/lib/terminfo/c/c108-rv-na
usr/lib/terminfo/c/c108-rv-na-8p
usr/lib/terminfo/c/c108-w
usr/lib/terminfo/c/c108-w-8p
usr/lib/terminfo/c/ca
usr/lib/terminfo/c/ca22851
usr/lib/terminfo/c/carlock
usr/lib/terminfo/c/cb-unix
usr/lib/terminfo/c/cbblit
usr/lib/terminfo/c/cbunix
usr/lib/terminfo/c/cdc
usr/lib/terminfo/c/cdc456
usr/lib/terminfo/c/cdc456tst
usr/lib/terminfo/c/cdi
usr/lib/terminfo/c/cdi1203
usr/lib/terminfo/c/cg7900
usr/lib/terminfo/c/chromatics

usr/lib/terminfo/c/ci8510
usr/lib/terminfo/c/citoh
usr/lib/terminfo/c/citoh-6lpi
usr/lib/terminfo/c/citoh-8lpi
usr/lib/terminfo/c/citoh-comp
usr/lib/terminfo/c/citoh-elite
usr/lib/terminfo/c/citoh-pica
usr/lib/terminfo/c/citoh-prop
usr/lib/terminfo/c/citoh-ps
usr/lib/terminfo/c/compucolor2
usr/lib/terminfo/c/compucolorii
usr/lib/terminfo/c/concept
usr/lib/terminfo/c/concept100
usr/lib/terminfo/c/concept100-rv
usr/lib/terminfo/c/concept100-rv-
usr/lib/terminfo/c/concept108-4p
usr/lib/terminfo/c/concept108-8p
usr/lib/terminfo/c/concept108-na-
usr/lib/terminfo/c/concept108-rv-
usr/lib/terminfo/c/concept108-w-8
usr/lib/terminfo/c/ct82
usr/lib/terminfo/c/ct8500
usr/lib/terminfo/d
usr/lib/terminfo/d/d100
usr/lib/terminfo/d/d132
usr/lib/terminfo/d/d200
usr/lib/terminfo/d/d80
usr/lib/terminfo/d/d800
usr/lib/terminfo/d/datagraphix
usr/lib/terminfo/d/datamedia2500
usr/lib/terminfo/d/datapoint
usr/lib/terminfo/d/dataspeed40
usr/lib/terminfo/d/dd5000
usr/lib/terminfo/d/debug
usr/lib/terminfo/d/decwriter
usr/lib/terminfo/d/delta
usr/lib/terminfo/d/dg
usr/lib/terminfo/d/dg6053
usr/lib/terminfo/d/diablo
usr/lib/terminfo/d/dialogue
usr/lib/terminfo/d/dialogue80
usr/lib/terminfo/d/dialup
usr/lib/terminfo/d/digilog
usr/lib/terminfo/d/direct
usr/lib/terminfo/d/direct800
usr/lib/terminfo/d/dm1520
usr/lib/terminfo/d/dm1521
usr/lib/terminfo/d/dm2500
usr/lib/terminfo/d/dm3025
usr/lib/terminfo/d/dm3045
usr/lib/terminfo/d/dm80
usr/lib/terminfo/d/dm80-w
usr/lib/terminfo/d/dmdt80
usr/lib/terminfo/d/dmdt80-w
usr/lib/terminfo/d/dp3
usr/lib/terminfo/d/dp3360
usr/lib/terminfo/d/ds40
usr/lib/terminfo/d/ds40-2
usr/lib/terminfo/d/dt80
usr/lib/terminfo/d/dt80-w
usr/lib/terminfo/d/dtc
usr/lib/terminfo/d/dtc300s
usr/lib/terminfo/d/dtc382
usr/lib/terminfo/d/dumb
usr/lib/terminfo/d/dw
usr/lib/terminfo/d/dw1
usr/lib/terminfo/d/dw2
usr/lib/terminfo/d/dw3
usr/lib/terminfo/d/dw4
usr/lib/terminfo/e
usr/lib/terminfo/e/env230
usr/lib/terminfo/e/envision230
usr/lib/terminfo/e/ep40
usr/lib/terminfo/e/ep4000
usr/lib/terminfo/e/ep4080
usr/lib/terminfo/e/ep48
usr/lib/terminfo/e/esprit
usr/lib/terminfo/e/ethernet
usr/lib/terminfo/e/ex3000
usr/lib/terminfo/e/exidy
usr/lib/terminfo/e/exidy2500
usr/lib/terminfo/f
usr/lib/terminfo/f/f100
usr/lib/terminfo/f/f100-rv
usr/lib/terminfo/f/f1720
usr/lib/terminfo/f/f1720a
usr/lib/terminfo/f/falco
usr/lib/terminfo/f/falco-p
usr/lib/terminfo/f/fixterm
usr/lib/terminfo/f/fox
usr/lib/terminfo/f/freedom
usr/lib/terminfo/f/freedom-rv
usr/lib/terminfo/f/freedom100
usr/lib/terminfo/g
usr/lib/terminfo/g/gigi
usr/lib/terminfo/g/gsi
usr/lib/terminfo/g/gt100
usr/lib/terminfo/g/gt100a
usr/lib/terminfo/g/gt40
usr/lib/terminfo/g/gt42
usr/lib/terminfo/h
usr/lib/terminfo/h/h1000
usr/lib/terminfo/h/h1420
usr/lib/terminfo/h/h1500
usr/lib/terminfo/h/h1510
usr/lib/terminfo/h/h1520
usr/lib/terminfo/h/h1552
usr/lib/terminfo/h/h1552-rv
usr/lib/terminfo/h/h19
usr/lib/terminfo/h/h19-a

usr/lib/terminfo/h/h19-b
usr/lib/terminfo/h/h19-bs
usr/lib/terminfo/h/h19-pb
usr/lib/terminfo/h/h19-smul
usr/lib/terminfo/h/h19-u
usr/lib/terminfo/h/h19A
usr/lib/terminfo/h/h19a
usr/lib/terminfo/h/h19b
usr/lib/terminfo/h/h19bs
usr/lib/terminfo/h/h19u
usr/lib/terminfo/h/h2000
usr/lib/terminfo/h/h89
usr/lib/terminfo/h/heath
usr/lib/terminfo/h/heath-19
usr/lib/terminfo/h/heath-ansi
usr/lib/terminfo/h/heathkit
usr/lib/terminfo/h/heathkit-a
usr/lib/terminfo/h/hp
usr/lib/terminfo/h/hp110
usr/lib/terminfo/h/hp150
usr/lib/terminfo/h/hp2382
usr/lib/terminfo/h/hp2382a
usr/lib/terminfo/h/hp2392
usr/lib/terminfo/h/hp2392a
usr/lib/terminfo/h/hp2393
usr/lib/terminfo/h/hp2393a
usr/lib/terminfo/h/hp2394
usr/lib/terminfo/h/hp2394a
usr/lib/terminfo/h/hp2397
usr/lib/terminfo/h/hp2397a
usr/lib/terminfo/h/hp2621
usr/lib/terminfo/h/hp2621-ba
usr/lib/terminfo/h/hp2621-fl
usr/lib/terminfo/h/hp2621-nl
usr/lib/terminfo/h/hp2621-nt
usr/lib/terminfo/h/hp2621-pb
usr/lib/terminfo/h/hp2621-wl
usr/lib/terminfo/h/hp2621a
usr/lib/terminfo/h/hp2621k45
usr/lib/terminfo/h/hp2621nl
usr/lib/terminfo/h/hp2621nt
usr/lib/terminfo/h/hp2621p
usr/lib/terminfo/h/hp2621wl
usr/lib/terminfo/h/hp2622
usr/lib/terminfo/h/hp2622a
usr/lib/terminfo/h/hp2622p
usr/lib/terminfo/h/hp2623
usr/lib/terminfo/h/hp2623a
usr/lib/terminfo/h/hp2623p
usr/lib/terminfo/h/hp2624
usr/lib/terminfo/h/hp2624a
usr/lib/terminfo/h/hp2624p
usr/lib/terminfo/h/hp2625
usr/lib/terminfo/h/hp2626
usr/lib/terminfo/h/hp2626a
usr/lib/terminfo/h/hp2626p
usr/lib/terminfo/h/hp2627
usr/lib/terminfo/h/hp2627a
usr/lib/terminfo/h/hp2627p
usr/lib/terminfo/h/hp2628
usr/lib/terminfo/h/hp2640a
usr/lib/terminfo/h/hp2640b
usr/lib/terminfo/h/hp2644
usr/lib/terminfo/h/hp2645
usr/lib/terminfo/h/hp2648
usr/lib/terminfo/h/hp2648a
usr/lib/terminfo/h/hp2703
usr/lib/terminfo/h/hp2709
usr/lib/terminfo/h/hp44
usr/lib/terminfo/h/hp45
usr/lib/terminfo/h/hp9020
usr/lib/terminfo/h/hp9816te
usr/lib/terminfo/h/hp9816teb
usr/lib/terminfo/h/hp9826
usr/lib/terminfo/h/hp9835
usr/lib/terminfo/h/hp9836
usr/lib/terminfo/h/hp9836te
usr/lib/terminfo/h/hp9836teb
usr/lib/terminfo/h/hp9837
usr/lib/terminfo/h/hp9845
usr/lib/terminfo/h/hp98700
usr/lib/terminfo/h/hpex
usr/lib/terminfo/h/hpsub
usr/lib/terminfo/i
usr/lib/terminfo/i/i100
usr/lib/terminfo/i/i3101
usr/lib/terminfo/i/i400
usr/lib/terminfo/i/ibm
usr/lib/terminfo/i/ibm3101
usr/lib/terminfo/i/infoton
usr/lib/terminfo/i/intertec
usr/lib/terminfo/i/intertube
usr/lib/terminfo/i/intertube2
usr/lib/terminfo/i/intext
usr/lib/terminfo/i/intext2
usr/lib/terminfo/i/intextii
usr/lib/terminfo/i/ips
usr/lib/terminfo/i/iq120
usr/lib/terminfo/i/iq140
usr/lib/terminfo/i/isc
usr/lib/terminfo/i/isc8001
usr/lib/terminfo/i/it
usr/lib/terminfo/i/it2
usr/lib/terminfo/i/ite
usr/lib/terminfo/j
usr/lib/terminfo/j/jerq
usr/lib/terminfo/k
usr/lib/terminfo/k/k45
usr/lib/terminfo/k/klc
usr/lib/terminfo/k/ktm

```
usr/lib/terminfo/l
usr/lib/terminfo/l/la120
usr/lib/terminfo/l/layer
usr/lib/terminfo/l/lp
usr/lib/terminfo/l/lpr
usr/lib/terminfo/m
usr/lib/terminfo/m/mdl110
usr/lib/terminfo/m/megatek
usr/lib/terminfo/m/microb
usr/lib/terminfo/m/microbee
usr/lib/terminfo/m/microkit
usr/lib/terminfo/m/microterm
usr/lib/terminfo/m/microterm5
usr/lib/terminfo/m/mime
usr/lib/terminfo/m/mime-3a
usr/lib/terminfo/m/mime-3ax
usr/lib/terminfo/m/mime-adm3a
usr/lib/terminfo/m/mime-adm3ax
usr/lib/terminfo/m/mime-fb
usr/lib/terminfo/m/mime-hb
usr/lib/terminfo/m/mime1
usr/lib/terminfo/m/mime2
usr/lib/terminfo/m/mime2a
usr/lib/terminfo/m/mime2a-s
usr/lib/terminfo/m/mime2a-v
usr/lib/terminfo/m/mime2as
usr/lib/terminfo/m/mime2av
usr/lib/terminfo/m/mimefb
usr/lib/terminfo/m/mimehb
usr/lib/terminfo/m/mimei
usr/lib/terminfo/m/mimeii
usr/lib/terminfo/m/minansi
usr/lib/terminfo/m/mkt
usr/lib/terminfo/m/mransi
usr/lib/terminfo/m/mw2
usr/lib/terminfo/n
usr/lib/terminfo/n/nec
usr/lib/terminfo/n/netronics
usr/lib/terminfo/n/network
usr/lib/terminfo/n/netx
usr/lib/terminfo/n/nomad
usr/lib/terminfo/n/nuc
usr/lib/terminfo/n/nucterm
usr/lib/terminfo/o
usr/lib/terminfo/o/o31
usr/lib/terminfo/o/oadm31
usr/lib/terminfo/o/obitgraph
usr/lib/terminfo/o/obitgraph-nv
usr/lib/terminfo/o/obitgraph-rv
usr/lib/terminfo/o/oblit
usr/lib/terminfo/o/oc100
usr/lib/terminfo/o/oconcept
usr/lib/terminfo/o/ojerq
usr/lib/terminfo/o/omron
usr/lib/terminfo/o/ovi300
usr/lib/terminfo/o/owl
usr/lib/terminfo/p
usr/lib/terminfo/p/patch
usr/lib/terminfo/p/patchboard
usr/lib/terminfo/p/pbox
usr/lib/terminfo/p/pc
usr/lib/terminfo/p/pe550
usr/lib/terminfo/p/plasma
usr/lib/terminfo/p/plugboard
usr/lib/terminfo/p/print
usr/lib/terminfo/p/printer
usr/lib/terminfo/p/printerbox
usr/lib/terminfo/p/printing
usr/lib/terminfo/p/ps
usr/lib/terminfo/p/pty
usr/lib/terminfo/q
usr/lib/terminfo/q/qume
usr/lib/terminfo/q/qume5
usr/lib/terminfo/r
usr/lib/terminfo/r/ramtek
usr/lib/terminfo/r/rayterm
usr/lib/terminfo/r/reach
usr/lib/terminfo/r/regent
usr/lib/terminfo/r/regent100
usr/lib/terminfo/r/regent20
usr/lib/terminfo/r/regent200
usr/lib/terminfo/r/regent25
usr/lib/terminfo/r/regent40
usr/lib/terminfo/r/regent40+
usr/lib/terminfo/r/regent40-s
usr/lib/terminfo/r/regent60
usr/lib/terminfo/r/regent60-na
usr/lib/terminfo/s
usr/lib/terminfo/s/s1500
usr/lib/terminfo/s/sb1
usr/lib/terminfo/s/sb2
usr/lib/terminfo/s/sb3
usr/lib/terminfo/s/sbg
usr/lib/terminfo/s/sbi
usr/lib/terminfo/s/screwpoint
usr/lib/terminfo/s/sexidy
usr/lib/terminfo/s/smarterm
usr/lib/terminfo/s/smarterm-s
usr/lib/terminfo/s/smartvid
usr/lib/terminfo/s/sol
usr/lib/terminfo/s/sol1
usr/lib/terminfo/s/sol2
usr/lib/terminfo/s/soroc
usr/lib/terminfo/s/spinwriter
usr/lib/terminfo/s/sun
usr/lib/terminfo/s/sun1
usr/lib/terminfo/s/superbee
usr/lib/terminfo/s/superbrain
usr/lib/terminfo/s/switch
usr/lib/terminfo/s/swtp
```

usr/lib/terminfo/s/synertek
usr/lib/terminfo/s/system1
usr/lib/terminfo/t
usr/lib/terminfo/t/t10
usr/lib/terminfo/t/t1061
usr/lib/terminfo/t/t1061f
usr/lib/terminfo/t/t16
usr/lib/terminfo/t/t3700
usr/lib/terminfo/t/t3800
usr/lib/terminfo/t/t500
usr/lib/terminfo/t/tab
usr/lib/terminfo/t/tab132
usr/lib/terminfo/t/tab132-rv
usr/lib/terminfo/t/tab132-w
usr/lib/terminfo/t/tab132-w-rv
usr/lib/terminfo/t/tec
usr/lib/terminfo/t/tec400
usr/lib/terminfo/t/tec500
usr/lib/terminfo/t/tek
usr/lib/terminfo/t/tek4012
usr/lib/terminfo/t/tek4013
usr/lib/terminfo/t/tek4014
usr/lib/terminfo/t/tek4014-sm
usr/lib/terminfo/t/tek4015
usr/lib/terminfo/t/tek4015-sm
usr/lib/terminfo/t/tek4023
usr/lib/terminfo/t/tek4024
usr/lib/terminfo/t/tek4025
usr/lib/terminfo/t/tek4027
usr/lib/terminfo/t/tek4112
usr/lib/terminfo/t/teleray
usr/lib/terminfo/t/teletec
usr/lib/terminfo/t/televideo950
usr/lib/terminfo/t/terak
usr/lib/terminfo/t/terminet
usr/lib/terminfo/t/terminet1200
usr/lib/terminfo/t/terminet300
usr/lib/terminfo/t/tex
usr/lib/terminfo/t/ti
usr/lib/terminfo/t/ti700
usr/lib/terminfo/t/ti733
usr/lib/terminfo/t/ti735
usr/lib/terminfo/t/ti745
usr/lib/terminfo/t/ti800
usr/lib/terminfo/t/tn1200
usr/lib/terminfo/t/tn300
usr/lib/terminfo/t/trs-80
usr/lib/terminfo/t/trs80
usr/lib/terminfo/t/ts-1
usr/lib/terminfo/t/ts1
usr/lib/terminfo/t/tty
usr/lib/terminfo/t/tty33
usr/lib/terminfo/t/tty37
usr/lib/terminfo/t/tty40
usr/lib/terminfo/t/tty43
usr/lib/terminfo/t/tty4424
usr/lib/terminfo/t/tty4424-2
usr/lib/terminfo/t/tty4424-el-2
usr/lib/terminfo/t/ttywilliams
usr/lib/terminfo/t/tvi
usr/lib/terminfo/t/tvi-2p
usr/lib/terminfo/t/tvi2p
usr/lib/terminfo/t/tvi912
usr/lib/terminfo/t/tvi912-2p
usr/lib/terminfo/t/tvi9122p
usr/lib/terminfo/t/tvi912b
usr/lib/terminfo/t/tvi912c
usr/lib/terminfo/t/tvi912cc
usr/lib/terminfo/t/tvi912cup@
usr/lib/terminfo/t/tvi920
usr/lib/terminfo/t/tvi920-2p
usr/lib/terminfo/t/tvi9202p
usr/lib/terminfo/t/tvi920b
usr/lib/terminfo/t/tvi920c
usr/lib/terminfo/t/tvi925
usr/lib/terminfo/t/tvi950
usr/lib/terminfo/t/tvi950-2p
usr/lib/terminfo/t/tvi950-4p
usr/lib/terminfo/t/tvi950-ap
usr/lib/terminfo/t/tvi950-b
usr/lib/terminfo/t/tvi950-ns
usr/lib/terminfo/t/tvi950-rv
usr/lib/terminfo/t/tvi950-rv-2p
usr/lib/terminfo/t/tvi950-rv-4p
usr/lib/terminfo/t/tvi9502p
usr/lib/terminfo/t/tvi9504p
usr/lib/terminfo/t/tvi950b
usr/lib/terminfo/t/tvi950ns
usr/lib/terminfo/t/tvi950rv
usr/lib/terminfo/t/tvi950rv2p
usr/lib/terminfo/t/tvi950rv4p
usr/lib/terminfo/u
usr/lib/terminfo/u/ubell
usr/lib/terminfo/u/ubellchar
usr/lib/terminfo/u/unitrm18
usr/lib/terminfo/u/unknown
usr/lib/terminfo/v
usr/lib/terminfo/v/vc103
usr/lib/terminfo/v/vc203
usr/lib/terminfo/v/vc303
usr/lib/terminfo/v/vc303-a
usr/lib/terminfo/v/vc403a
usr/lib/terminfo/v/vc404
usr/lib/terminfo/v/vc404-na
usr/lib/terminfo/v/vc404-s
usr/lib/terminfo/v/vc404-s-na
usr/lib/terminfo/v/vc415
usr/lib/terminfo/v/vi200
usr/lib/terminfo/v/vi200-f
usr/lib/terminfo/v/vi200-ic

usr/lib/terminfo/v/vi200-rv
usr/lib/terminfo/v/vi200-rv-ic
usr/lib/terminfo/v/vi300
usr/lib/terminfo/v/vi300-aw
usr/lib/terminfo/v/vi300-rv
usr/lib/terminfo/v/vi300-ss
usr/lib/terminfo/v/vi550
usr/lib/terminfo/v/viewpoint
usr/lib/terminfo/v/virtual
usr/lib/terminfo/v/visual
usr/lib/terminfo/v/vitty
usr/lib/terminfo/v/vk100
usr/lib/terminfo/v/vt100
usr/lib/terminfo/v/vt100-am
usr/lib/terminfo/v/vt100-bot-s
usr/lib/terminfo/v/vt100-nam
usr/lib/terminfo/v/vt100-nam-w
usr/lib/terminfo/v/vt100-nav
usr/lib/terminfo/v/vt100-nav-w
usr/lib/terminfo/v/vt100-np
usr/lib/terminfo/v/vt100-s
usr/lib/terminfo/v/vt100-s-bot
usr/lib/terminfo/v/vt100-s-top
usr/lib/terminfo/v/vt100-top-s
usr/lib/terminfo/v/vt100-w
usr/lib/terminfo/v/vt100-w-am
usr/lib/terminfo/v/vt100-w-nam
usr/lib/terminfo/v/vt100-w-nav
usr/lib/terminfo/v/vt100am
usr/lib/terminfo/v/vt100nam
usr/lib/terminfo/v/vt100s
usr/lib/terminfo/v/vt100w
usr/lib/terminfo/v/vt125
usr/lib/terminfo/v/vt132
usr/lib/terminfo/v/vt50
usr/lib/terminfo/v/vt50h
usr/lib/terminfo/v/vt52
usr/lib/terminfo/w
usr/lib/terminfo/w/wy100
usr/lib/terminfo/x
usr/lib/terminfo/x/x1700
usr/lib/terminfo/x/x1720
usr/lib/terminfo/x/x1750
usr/lib/terminfo/x/xitex
usr/lib/terminfo/x/xl83
usr/lib/terminfo/y
usr/lib/terminfo/y/ya
usr/lib/terminfo/z
usr/lib/terminfo/z/z19
usr/lib/terminfo/z/z30
usr/lib/terminfo/z/zen30
usr/lib/terminfo/z/zenith

FILE SET: ACORE4

Description: Manditory file set.

etc/newconfig/d.profile
usr
usr/bin
usr/bin/asa
usr/bin/awk
usr/bin/cancel
usr/bin/disable
usr/bin/enable
usr/bin/lp
usr/bin/lpr
usr/bin/lpstat
usr/bin/more
usr/bin/tabs
usr/bin/tar
usr/bin/what
usr/lib
usr/lib/accept
usr/lib/expreserve
usr/lib/lpadmin
usr/lib/lpfx
usr/lib/lpmove
usr/lib/lpsched
usr/lib/lpshut
usr/lib/more.help
usr/lib/nls
usr/lib/nls/n-computer
usr/lib/reject
usr/lib/tabset
usr/lib/tabset/xerox1720
usr/lib/tabset/vt100
usr/lib/tabset/teleray
usr/lib/tabset/stdcrt
usr/lib/tabset/std
usr/lib/tabset/diablo
usr/lib/tabset/beehive
usr/lib/tabset/9837
usr/lib/tabset/9836
usr/lib/tabset/9826
usr/lib/tabset/3101
usr/spool
usr/spool/lp
usr/spool/lp/class
usr/spool/lp/interface
usr/spool/lp/member
usr/spool/lp/request
usr/spool/lp/model
usr/spool/lp/model/dumb
usr/spool/lp/model/hp2225a
usr/spool/lp/model/hp2631g
usr/spool/lp/model/hp2686a
usr/spool/lp/model/hp2934a
usr/preserve
usr/tmp
lib

lost+found
tmp
users
users/guest
system
system/ACORE4
system/ACORE4/customize

FILE SET: ACORE5

Description: Manditory file set.

bin
bin/pam
system
system/ACORE5
system/ACORE5/customize
system/ACORE5/revlist
users
users/nowindow
users/nowindow/Autost
users/nowindow/.environ
usr
usr/lib
usr/lib/nls
usr/lib/nls/n-computer
usr/lib/nls/n-computer/pam.cat

FILE SET: A98597A

Description: Multi-user Kernel for S300

hp-ux
etc
etc/conf
etc/conf/dfile
system
system/98597A
system/98597A/customize

FILE SET: APCORE

Description: First PE core fileset

bin/crypt
bin/df
bin/diff
bin/env
bin/hp9000s200
bin/hp9000s500
bin/mail
bin/mesg
bin/ncheck
bin/newgrp
bin/nohup
bin/pdp11
bin/ranlib
bin/sed
bin/sum
bin/tty
bin/u370
bin/u3b
bin/u3b10
bin/u3b2
bin/u3b5
bin/vax
bin/write
etc/cron
etc/fsdb
etc/grpck
etc/last
etc/lastb
etc/magic
etc/newconfig/.proto
etc/newconfig/at.allow
etc/newconfig/cron.allow
etc/newconfig/crontab
etc/newconfig/master
etc/newconfig/mklp
etc/newconfig/queuedefs
etc/newconfig/termcap
etc/ocpio
etc/pwck
etc/revck
etc/setprivgrp
etc/tunefs
etc/whodo

FILE SET: APCORE2

Description: Second PE core fileset

usr/bin/at
usr/bin/batch
usr/bin/bdiff
usr/bin/bifdf
usr/bin/col
usr/bin/crontab
usr/bin/ctags
usr/bin/diff3
usr/bin/egrep
usr/bin/expand
usr/bin/fgrep
usr/bin/file
usr/bin/getopt
usr/bin/getprivgrp
usr/bin/groups
usr/bin/id
usr/bin/logname
usr/bin/news
usr/bin/prealloc

usr/bin/rtprio
usr/bin/tput
usr/bin/unexpand
usr/bin/xargs
usr/lib/cron
usr/lib/help/ad
usr/lib/help/bd
usr/lib/help/cm
usr/lib/help/cmds
usr/lib/help/co
usr/lib/help/de
usr/lib/help/ge
usr/lib/help/prs
usr/lib/help/rc
usr/lib/help/term
usr/lib/help/text
usr/lib/help/un
usr/lib/help/ut
usr/lib/makekey
usr/lib/nmf
usr/lib/term
usr/lib/term/tab2631
usr/lib/term/tab2631-c
usr/lib/term/tab2631-e
usr/lib/term/tab300
usr/lib/term/tab300-12
usr/lib/term/tab300S
usr/lib/term/tab300S-12
usr/lib/term/tab37
usr/lib/term/tab382
usr/lib/term/tab4000A
usr/lib/term/tab450
usr/lib/term/tab450-12
usr/lib/term/tab832
usr/lib/term/tabX
usr/lib/term/tabai
usr/lib/term/tablp
usr/lib/term/tabtn300
usr/mail
usr/news
usr/spool/cron
usr/spool/cron/atjobs
usr/spool/cron/crontabs

FILE SET: AFTN_CORE

Description: Fortran files neede for DGL/GP

lib/frt0.o
usr/lib/libF77.a
usr/lib/libI77.a
system/FTN_CORE/revlist

FILE SET: APAS_CORE

Description: Pascal files needed for DGL/AGP

lib/libpc.a
usr/lib/libheap2.a
system/PAS_CORE/APAS_CORE
system/PAS_CORE/revlist

PARTITION: SYS_TOOLS

This partition resides on the 4 red labeled micro discs. This partition contains editors and useful programs.

FILE SET: ACMD

Description: This file set resides on micro disc 1 and contains the vi editor (shown as bin/ed) and helpful programs such as help and sort. If you plan to do text editing with the vi editor described in Chapter 6, you should load this file set.

bin
bin/cmp
bin/clri
bin/csh
bin/du
bin/ed
bin/false
bin/nice
bin/pr
bin/sort
bin/tail
bin/tee
bin/touch
bin/true
bin/uname
bin/wc
etc
etc/chroot
etc/newconfig
etc/newconfig/csh.login
etc/newconfig/d.Autost
etc/newconfig/d.environ
etc/newconfig/d.exrc

```
etc/newconfig/mkdev
system/ACMD/customize
usr
usr/bin
usr/bin/bfs
usr/bin/ex
usr/bin/fold
usr/bin/help
usr/lib
usr/lib/builtins/@
usr/lib/builtins/alias
usr/lib/builtins/alloc
usr/lib/builtins/case
usr/lib/builtins/cd
usr/lib/builtins/chdir
usr/lib/builtins/dirs
usr/lib/builtins/eval
usr/lib/builtins/exec
usr/lib/builtins/foreach
usr/lib/builtins/glob
usr/lib/builtins/hashstat
usr/lib/builtins/history
usr/lib/builtins/if
usr/lib/builtins/jobs
usr/lib/builtins/kill
usr/lib/builtins/login
usr/lib/builtins/logout
usr/lib/builtins/newgrp
usr/lib/builtins/nice
usr/lib/builtins/nohup
usr/lib/builtins/notify
usr/lib/builtins/onintr
usr/lib/builtins/popd
usr/lib/builtins/pushd
usr/lib/builtins/rehash
usr/lib/builtins/repeat
usr/lib/builtins/set
usr/lib/builtins/setenv
usr/lib/builtins/source
usr/lib/builtins/stop
usr/lib/builtins/switch
usr/lib/builtins/time
usr/lib/builtins/umask
usr/lib/builtins/unalias
usr/lib/builtins/unhash
usr/lib/builtins/unset
usr/lib/builtins/unsetenv
usr/lib/builtins/wait
usr/lib/builtins/while

usr/lib/exrecover
usr/lib/help
usr/lib/help/default
usr/lib/help/he
```

FILE SET: ABCMD

Description: This file set on micro disc 2 contains lif and bif utilities. The lif utilities allow you to see what is on tapes or micro discs, and provides an interface between HP 9000 Series 200 or 500 computers and your computer. The bif utilities are used to transfer data or script files in a specific format from the Series 200 or 500 systems to your new system.

```
usr
usr/bin
usr/bin/bifchgrp
usr/bin/bifchmod
usr/bin/bifchown
usr/bin/bifcp
usr/bin/biffind
usr/bin/biffsck
usr/bin/bifls
usr/bin/bifmkdir
usr/bin/bifmkfs
usr/bin/bifrm
usr/bin/bifrmdir
usr/bin/hp
usr/bin/lifinit
usr/bin/lifls
usr/bin/lifrename
usr/bin/lifrm
usr/bin/man
usr/bin/tr
usr/bin/whereis
usr/bin/whoami
system
system/ABCMD
system/ABCMD/customize
```

FILE SET: AWINDOW

Description: This file set on micro disc 3 contains the utilities necessary for you to run HP Windows/9000. You will need to load this file set along with the ASTARBAS file set (a dependency) to run Windows on your system (the computer will prompt you to insert the 4th disc when you need to).

```
system
system/AWINDOW
```

system/AWINDOW/customize
system/AWINDOW/revlist
users
users/window
users/window/.wmsh
usr
usr/bin
usr/bin/wborder
usr/bin/wmstart
usr/lib
usr/lib/raster
usr/lib/raster/12x20
usr/lib/raster/12x20/cour.0U
usr/lib/raster/12x20/cour.b.0U
usr/lib/raster/18x30
usr/lib/raster/18x30/math.0M
usr/lib/raster/18x30/pica.8U
usr/lib/raster/6x8
usr/lib/raster/6x8/lp.8U
usr/lib/raster/6x8/lp.b.8I
usr/lib/raster/6x8/math.8M
usr/lib/raster/7x10
usr/lib/raster/7x10/lp.8U
usr/lib/raster/8x16
usr/lib/raster/8x16/kana.8K
usr/lib/raster/8x16/linedraw.0L
usr/lib/raster/8x16/lp.8U
usr/lib/raster/8x16/lp.b.8U
usr/lib/raster/8x16/lp.i.8U
usr/lib/raster/8x16/math.0M
usr/lib/raster/L6x15
usr/lib/raster/L6x15/lp.8U
usr/lib/wm
usr/man
usr/man/cat1
usr/man/cat1/wborder.1
usr/man/cat1/wcreate.1
usr/man/cat1/wdestroy.1
usr/man/cat1/wdisp.1
usr/man/cat1/wfont.1
usr/man/cat1/windows.1
usr/man/cat1/wlist.1
usr/man/cat1/wmove.1
usr/man/cat1/wmready.1
usr/man/cat1/wmstart.1
usr/man/cat1/wmstop.1
usr/man/cat1/wselect.1
usr/man/cat1/wsh.1
usr/man/cat1/wsize.1

FILE SET: ASTARBAS

Description: This file set on micro disc 4 is a dependency for the AWINDOW file set.

usr
usr/lib
usr/lib/starbase
usr/lib/starbase/hp98710
usr/lib/starbase/hp98710/te_data
usr/lib/starbase/hp98710/te_mcode
usr/lib/starbase/defaults
usr/lib/starbase/char_sets
usr/lib/starbase/sb_daemon_1.1
usr/lib/starbase/stroke
usr/lib/starbase/stroke/markers
usr/lib/starbase/stroke/usascii
usr/lib/starbase/stroke/usascii/1
usr/lib/starbase/stroke/usascii/2
usr/lib/starbase/stroke/katakana
usr/lib/starbase/stroke/katakana/1
usr/lib/starbase/stroke/katakana/2
usr/lib/starbase/stroke/jisascii
usr/lib/starbase/stroke/jisascii/1
usr/lib/starbase/stroke/jisascii/2
usr/lib/starbase/stroke/hproman
usr/lib/starbase/stroke/hproman/1
usr/lib/starbase/stroke/hproman/2
usr/lib/starbase/stroke/font_info
usr/lib/starbase/stroke/font_info/1
usr/lib/starbase/stroke/font_info/2
usr/lib/starbase/errors
usr/lib/starbase/errors/american
usr/lib/starbase/errors/american/sb.errors
usr/lib/starbase/errors/n-computer
usr/lib/starbase/errors/n-computer/sb.errors
system/ASTARBAS/customize
system/ASTARBAS/revlist

FILE SET: ADSTARBAS

Description: Demos for Starbase

usr
usr/lib
usr/lib/starbase
usr/lib/starbase/demos
usr/lib/starbase/demos/SBUTILS
usr/lib/starbase/demos/SBUTILS/README
usr/lib/starbase/demos/SBUTILS/makefile
usr/lib/starbase/demos/SBUTILS/sb.3d.c
usr/lib/starbase/demos/SBUTILS/sb.dbuf.c
usr/lib/starbase/demos/SBUTILS/sb.glob.c
usr/lib/starbase/demos/SBUTILS/sb.glob.h
usr/lib/starbase/demos/SBUTILS/sb.test1.c
usr/lib/starbase/demos/SBUTILS/sbutils.c.h
usr/lib/starbase/demos/SBUTILS/sbutils.h
usr/lib/starbase/demos/SBUTILS/sb.prim.c
usr/lib/starbase/demos/DATA
usr/lib/starbase/demos/DATA/fighterdata

usr/lib/starbase/demos/DATA/frame
usr/lib/starbase/demos/DATA/glow
usr/lib/starbase/demos/DATA/pcmap
usr/lib/starbase/demos/DATA/planet
usr/lib/starbase/demos/DATA/tracmat
usr/lib/starbase/demos/DATA/tractor
usr/lib/starbase/demos/DATA/xwing_data
usr/lib/starbase/demos/boxes.c
usr/lib/starbase/demos/bspline.c
usr/lib/starbase/demos/example.c
usr/lib/starbase/demos/dumpgraphics.c
usr/lib/starbase/demos/fighter.c
usr/lib/starbase/demos/logo.c
usr/lib/starbase/demos/shuttle.c
usr/lib/starbase/demos/trac.c
usr/lib/starbase/demos/xwing.c
usr/lib/starbase/demos/makefile
system/DSTARBAS/customize
system/DSTARBAS/revlist

FILE SET: APSTARBAS

Description: Files needed for Starbase in PE

usr
usr/include
usr/include/starbase.c.h
usr/include/starbase.f1.h
usr/include/starbase.f2.h
usr/include/starbase.p1.h
usr/include/starbase.p2.h
usr/lib
usr/lib/libdd262x.a
usr/lib/libdd300h.a
usr/lib/libdd300l.a
usr/lib/libdd9836a.a
usr/lib/libdd9836c.a
usr/lib/libdd9837.a
usr/lib/libdd98700.a
usr/lib/libdd98710.a
usr/lib/libddhil.a
usr/lib/starbase
usr/lib/starbase/libddhil_d.a
usr/lib/starbase/libddhpgl_d.a
usr/lib/starbase/libddterm_d.a
usr/lib/starbase/sb_daemon.o
usr/lib/starbase/demo
usr/lib/starbase/demo/CC
usr/lib/starbase/demo/FC
usr/lib/starbase/demo/PC
usr/lib/starbase/demo/color1c.c
usr/lib/starbase/demo/color2c.c
usr/lib/starbase/demo/color3c.c
usr/lib/starbase/demo/color4c.c
usr/lib/starbase/demo/color5c.c
usr/lib/starbase/demo/color6c.c
usr/lib/starbase/demo/data
usr/lib/starbase/demo/example1c.c
usr/lib/starbase/demo/example1f.f
usr/lib/starbase/demo/example1p.p
usr/lib/starbase/demo/example2c.c
usr/lib/starbase/demo/example2f.f
usr/lib/starbase/demo/example2p.p
usr/lib/starbase/demo/example3c.c
usr/lib/starbase/demo/example3f.f
usr/lib/starbase/demo/example3p.p
usr/lib/starbase/demo/example4c.c
usr/lib/starbase/demo/example4f.f
usr/lib/starbase/demo/example4p.p
usr/lib/starbase/demo/example5c.c
usr/lib/starbase/demo/example5f.f
usr/lib/starbase/demo/example5p.p
usr/lib/starbase/demo/example6c.c
usr/lib/starbase/demo/example6f.f
usr/lib/starbase/demo/example6p.p
usr/lib/starbase/demo/text0c.c
usr/lib/starbase/demo/text10c.c
usr/lib/starbase/demo/text11c.c
usr/lib/starbase/demo/text12c.c
usr/lib/starbase/demo/text13c.c
usr/lib/starbase/demo/text14c.c
usr/lib/starbase/demo/text15c.c
usr/lib/starbase/demo/text16c.c
usr/lib/starbase/demo/text17c.c
usr/lib/starbase/demo/text18c.c
usr/lib/starbase/demo/text19c.c
usr/lib/starbase/demo/text1c.c
usr/lib/starbase/demo/text20c.c
usr/lib/starbase/demo/text21c.c
usr/lib/starbase/demo/text22c.c
usr/lib/starbase/demo/text23c.c
usr/lib/starbase/demo/text24c.c
usr/lib/starbase/demo/text25c.c
usr/lib/starbase/demo/text26c.c
usr/lib/starbase/demo/text27c.c
usr/lib/starbase/demo/text2c.c
usr/lib/starbase/demo/text3c.c
usr/lib/starbase/demo/text4c.c
usr/lib/starbase/demo/text5c.c
usr/lib/starbase/demo/text6c.c
usr/lib/starbase/demo/text7c.c
usr/lib/starbase/demo/text8c.c
usr/lib/starbase/demo/text9c.c
usr/lib/starbase/demo/final1c.c
usr/lib/starbase/demo/final1f.f
usr/lib/starbase/demo/final1p.p
usr/lib/libddhpgl.a
usr/lib/libddkbd.a
usr/lib/libddhpterm.a
usr/lib/libsb1.a

usr/lib/libsb2.a
system/PSTARBAS/customize
system/PSTARBAS/revlist

AMSTARBAS

Description: Starbase manual pages

usr
usr/man
usr/man/man3
usr/man/man3/await_event.3g
usr/man/man3/await_retra.3g
usr/man/man3/background_.3g
usr/man/man3/block_move.3g
usr/man/man3/block_read.3g
usr/man/man3/block_write.3g
usr/man/man3/buffer_mode.3g
usr/man/man3/character_e.3g
usr/man/man3/character_h.3g
usr/man/man3/character_s.3g
usr/man/man3/character_w.3g
usr/man/man3/clear_contr.3g
usr/man/man3/clear_view_.3g
usr/man/man3/clip_depth.3g
usr/man/man3/clip_indica.3g
usr/man/man3/clip_rectan.3g
usr/man/man3/concat_matr.3g
usr/man/man3/concat_tran.3g
usr/man/man3/dc_to_vdc.3g
usr/man/man3/define_colo.3g
usr/man/man3/define_rast.3g
usr/man/man3/depth_indic.3g
usr/man/man3/designate_c.3g
usr/man/man3/disable_eve.3g
usr/man/man3/display_ena.3g
usr/man/man3/draw.3g
usr/man/man3/drawing_mod.3g
usr/man/man3/echo_type.3g
usr/man/man3/echo_update.3g
usr/man/man3/enable_even.3g
usr/man/man3/fill_color.3g
usr/man/man3/flush_matri.3g
usr/man/man3/gclose.3g
usr/man/man3/gerr_contro.3g
usr/man/man3/gescape.3g
usr/man/man3/gopen.3g
usr/man/man3/initiate_re.3g
usr/man/man3/inquire_col.3g
usr/man/man3/inquire_ger.3g
usr/man/man3/inquire_id.3g
usr/man/man3/inquire_inp.3g
usr/man/man3/inquire_req.3g
usr/man/man3/inquire_siz.3g
usr/man/man3/inquire_tex.3g
usr/man/man3/interior_st.3g
usr/man/man3/intra_chara.3g
usr/man/man3/line_color.3g
usr/man/man3/line_repeat.3g
usr/man/man3/line_type.3g
usr/man/man3/make_pictur.3g
usr/man/man3/mapping_mod.3g
usr/man/man3/marker_colo.3g
usr/man/man3/marker_orie.3g
usr/man/man3/marker_size.3g
usr/man/man3/marker_type.3g
usr/man/man3/move.3g
usr/man/man3/partial_pol.3g
usr/man/man3/perimeter_c.3g
usr/man/man3/perimeter_r.3g
usr/man/man3/perimeter_t.3g
usr/man/man3/polygon.3g
usr/man/man3/polyline.3g
usr/man/man3/polymarker.3g
usr/man/man3/pop_matrix.3g
usr/man/man3/push_matrix.3g
usr/man/man3/push_vdc.3g
usr/man/man3/read_choice.3g
usr/man/man3/read_locato.3g
usr/man/man3/rectangle.3g
usr/man/man3/replace_mat.3g
usr/man/man3/request_cho.3g
usr/man/man3/request_loc.3g
usr/man/man3/sample_choi.3g
usr/man/man3/sample_loca.3g
usr/man/man3/set_locator.3g
usr/man/man3/set_p1_p2.3g
usr/man/man3/set_signals.3g
usr/man/man3/text.3g
usr/man/man3/text_alignm.3g
usr/man/man3/text_color.3g
usr/man/man3/text_font_i.3g
usr/man/man3/text_line_p.3g
usr/man/man3/text_line_s.3g
usr/man/man3/text_orient.3g
usr/man/man3/text_path.3g
usr/man/man3/text_precis.3g
usr/man/man3/text_switch.3g
usr/man/man3/track.3g
usr/man/man3/track_off.3g
usr/man/man3/transform_p.3g
usr/man/man3/vdc_extent.3g
usr/man/man3/vdc_to_dc.3g
usr/man/man3/vdc_to_wc.3g
usr/man/man3/viewport_ju.3g
usr/man/man3/wc_to_vdc.3g
usr/man/man3/write_enabl.3g
usr/man/man3/starbase.3g
system/MSTARBAS/customize
system/MSTARBAS/revlist

FILE SET: ADWINDOW

Description: Demos for HPWINDOWS

```
system
system/DWINDOW
system/DWINDOW/customize
system/DWINDOW/revlist
usr
usr/lib
usr/lib/hpwindows
usr/lib/hpwindows/demo
usr/lib/hpwindows/demosrc
usr/lib/hpwindows/demosrc/README
usr/lib/hpwindows/demosrc/Makefile
usr/lib/hpwindows/demosrc/Wfont
usr/lib/hpwindows/demosrc/chcolor
usr/lib/hpwindows/demosrc/crabs.c
usr/lib/hpwindows/demosrc/domenu.c
usr/lib/hpwindows/demosrc/editicon.c
usr/lib/hpwindows/demosrc/fclock.f
usr/lib/hpwindows/demosrc/fontbyte.c
usr/lib/hpwindows/demosrc/fontcurs.c
usr/lib/hpwindows/demosrc/fontdump.c
usr/lib/hpwindows/demosrc/fontedit.c
usr/lib/hpwindows/demosrc/fontinfo.c
usr/lib/hpwindows/demosrc/fontopt.c
usr/lib/hpwindows/demosrc/fontprop.c
usr/lib/hpwindows/demosrc/fontview.c
usr/lib/hpwindows/demosrc/lens.c
usr/lib/hpwindows/demosrc/life.c
usr/lib/hpwindows/demosrc/objonly/.HPCalc.hlp
usr/lib/hpwindows/demosrc/objonly/.HPCalc.msg
usr/lib/hpwindows/demosrc/objonly/calc
usr/lib/hpwindows/demosrc/objonly/fontdiag
usr/lib/hpwindows/demosrc/repaint.c
usr/lib/hpwindows/demosrc/seticon.c
usr/lib/hpwindows/demosrc/sprite.c
usr/lib/hpwindows/demosrc/wcal
usr/lib/hpwindows/demosrc/wclock.c
usr/lib/hpwindows/demosrc/wmdiag.c
usr/lib/hpwindows/demosrc/wqix.c
usr/lib/raster
usr/lib/raster/6x8
usr/lib/raster/6x8/calcfont
usr/lib/raster/icons
usr/lib/raster/icons/calculator
usr/lib/raster/icons/mailbox_empty
usr/lib/raster/icons/mailbox_full
```

FILE SET: AMWINDOW

Description: HPWINDOWS manaul pages

```
system
system/MWINDOW
system/MWINDOW/customize
system/MWINDOW/revlist
usr
usr/lib
usr/lib/hpwindows
usr/lib/hpwindows/man_examples
usr/lib/hpwindows/man_examples/basalt.c
usr/lib/hpwindows/man_examples/base_load.c
usr/lib/hpwindows/man_examples/btest.c
usr/lib/hpwindows/man_examples/build_icon.c
usr/lib/hpwindows/man_examples/clear_gr.c
usr/lib/hpwindows/man_examples/conceal_t0.c
usr/lib/hpwindows/man_examples/create_gr.c
usr/lib/hpwindows/man_examples/create_t0.c
usr/lib/hpwindows/man_examples/echo_band.c
usr/lib/hpwindows/man_examples/est_gr.c
usr/lib/hpwindows/man_examples/est_t0.c
usr/lib/hpwindows/man_examples/est_wm_com.c
usr/lib/hpwindows/man_examples/getfontinfo.c
usr/lib/hpwindows/man_examples/invert_bc.c
usr/lib/hpwindows/man_examples/kill_wm.c
usr/lib/hpwindows/man_examples/loc_in_user.c
usr/lib/hpwindows/man_examples/pan_gr.c
usr/lib/hpwindows/man_examples/pause_resume.c
usr/lib/hpwindows/man_examples/poll_events.c
usr/lib/hpwindows/man_examples/quarter_clip.c
usr/lib/hpwindows/man_examples/replace_icon.c
usr/lib/hpwindows/man_examples/reset_loc.c
usr/lib/hpwindows/man_examples/rm_window.c
usr/lib/hpwindows/man_examples/set_gr_labs.c
usr/lib/hpwindows/man_examples/setlabel_gr.c
usr/lib/hpwindows/man_examples/shrink_it.c
usr/lib/hpwindows/man_examples/shrink_t0.c
usr/lib/hpwindows/man_examples/shuffle_dn.c
usr/lib/hpwindows/man_examples/stair_step.c
usr/lib/hpwindows/man_examples/stretch_gr.c
usr/lib/hpwindows/man_examples/term_gr.c
usr/lib/hpwindows/man_examples/term_t0.c
usr/lib/hpwindows/man_examples/term_wm_com.c
usr/lib/hpwindows/man_examples/toggle_icon.c
usr/lib/hpwindows/man_examples/toggle_sel.c
usr/lib/hpwindows/man_examples/wbanner_sub.c
usr/lib/hpwindows/man_examples/wrepaint.c
usr/lib/hpwindows/man_examples/write_dn.c
usr/man
usr/man/man1
usr/man/man1/wborder.1
usr/man/man1/wcreate.1
usr/man/man1/wdestroy.1
usr/man/man1/wdisp.1
usr/man/man1/wfont.1
usr/man/man1/windows.1
usr/man/man1/wlist.1
usr/man/man1/wmove.1
```

```
usr/man/man1/wmready.1
usr/man/man1/wmstart.1
usr/man/man1/wmstop.1
usr/man/man1/wselect.1
usr/man/man1/wsh.1
usr/man/man1/wsize.1
usr/man/man3
usr/man/man3/altfont_ter.3w
usr/man/man3/basefont_te.3w
usr/man/man3/faclear.3w
usr/man/man3/facolors.3w
usr/man/man3/facursor.3w
usr/man/man3/fafontactiv.3w
usr/man/man3/fafontload.3w
usr/man/man3/fafontremov.3w
usr/man/man3/fagetinfo.3w
usr/man/man3/fainit.3w
usr/man/man3/farectwrite.3w
usr/man/man3/faroll.3w
usr/man/man3/fasetinfo.3w
usr/man/man3/faterminate.3w
usr/man/man3/fawrite.3w
usr/man/man3/fm_activate.3w
usr/man/man3/fm_clipflag.3w
usr/man/man3/fm_cliplim.3w
usr/man/man3/fm_colors.3w
usr/man/man3/fm_fileinfo.3w
usr/man/man3/fm_fontdir.3w
usr/man/man3/fm_getname.3w
usr/man/man3/fm_load.3w
usr/man/man3/fm_opt.3w
usr/man/man3/fm_rasterin.3w
usr/man/man3/fm_remove.3w
usr/man/man3/fm_str_len.3w
usr/man/man3/fm_styleinf.3w
usr/man/man3/fm_write.3w
usr/man/man3/fontgetid_t.3w
usr/man/man3/fontgetname.3w
usr/man/man3/fontload_te.3w
usr/man/man3/fontreplace.3w
usr/man/man3/fontsize_te.3w
usr/man/man3/fontswap_te.3w
usr/man/man3/fromxy_term.3w
usr/man/man3/toxy_term0.3w
usr/man/man3/wautodestro.3w
usr/man/man3/wautoselect.3w
usr/man/man3/wautotop.3w
usr/man/man3/wbanner.3w
usr/man/man3/wbottom.3w
usr/man/man3/wconceal.3w
usr/man/man3/wcreate_gra.3w
usr/man/man3/wcreate_ter.3w
usr/man/man3/wdestroy.3w
usr/man/man3/wdfltpos.3w
usr/man/man3/weventclear.3w
usr/man/man3/weventpoll.3w
usr/man/man3/wgetbcolor.3w
usr/man/man3/wgetbcoords.3w
usr/man/man3/wgetcoords.3w
usr/man/man3/wgetecho.3w
usr/man/man3/wgeticonpos.3w
usr/man/man3/wgetlocator.3w
usr/man/man3/wgetname.3w
usr/man/man3/wgetrastere.3w
usr/man/man3/wgetscreen.3w
usr/man/man3/wgetsigmask.3w
usr/man/man3/wiconic.3w
usr/man/man3/winit.3w
usr/man/man3/wmenu_activ.3w
usr/man/man3/wmenu_creat.3w
usr/man/man3/wmenu_delet.3w
usr/man/man3/wmenu_event.3w
usr/man/man3/wmenu_item.3w
usr/man/man3/wminquire.3w
usr/man/man3/wmkill.3w
usr/man/man3/wmove.3w
usr/man/man3/wmpathmake.3w
usr/man/man3/wmrepaint.3w
usr/man/man3/wpan.3w
usr/man/man3/wpauseoutpu.3w
usr/man/man3/wrecover.3w
usr/man/man3/wselect.3w
usr/man/man3/wsetbcolor.3w
usr/man/man3/wsetecho.3w
usr/man/man3/wseticon.3w
usr/man/man3/wseticonpos.3w
usr/man/man3/wsetlabel.3w
usr/man/man3/wsetlocator.3w
usr/man/man3/wsetrastere.3w
usr/man/man3/wsetsigmask.3w
usr/man/man3/wsfk_mode.3w
usr/man/man3/wsfk_prog.3w
usr/man/man3/wshuffle.3w
usr/man/man3/wsize.3w
usr/man/man3/wterminate.3w
usr/man/man3/wtop.3w
```

FILE SET: APWINDOW

Description: HPWINDOWS for PE

```
system
system/PWINDOW
system/PWINDOW/customize
system/PWINDOW/revlist
usr
usr/include
usr/include/fa.h
usr/include/fonticon.h
usr/include/window.h
```

usr/lib
usr/lib/libfa.a
usr/lib/libfontm.a
usr/lib/libwindow.a

FILE SET: APSYSCOM

Description: Optional commands for system control

etc/getx25
etc/newconfig/mailx.rc
usr/bin/cu
usr/bin/kermit
usr/bin/mailx
usr/bin/umodem
usr/bin/uucp
usr/bin/uulog
usr/bin/uuls
usr/bin/uuname
usr/bin/uupick
usr/bin/uusnap
usr/bin/uustat
usr/bin/uuto
usr/bin/uux

FILE SET: APSYSCOM2

Description: Second set of optional commands for system control

usr/lib/mailx
usr/lib/mailx/mailx.help
usr/lib/mailx/mailx.help.~
usr/lib/uucp
usr/lib/uucp/.OLD
usr/lib/uucp/.XQTDIR
usr/lib/uucp/X25
usr/lib/uucp/X25/HP2334A.clr
usr/lib/uucp/X25/HP2334A.in
usr/lib/uucp/X25/HP2334A.out
usr/lib/uucp/X25/HP2334A.outf
usr/lib/uucp/X25/HP2334A.outg
usr/lib/uucp/X25/HP2334A.outi
usr/lib/uucp/X25/clrsvc
usr/lib/uucp/X25/opx25
usr/lib/uucp/X25/ventel.out
usr/lib/uucp/newconfig/L-devices
usr/lib/uucp/newconfig/L-dialcodes
usr/lib/uucp/newconfig/L.cmds
usr/lib/uucp/newconfig/L.sys
usr/lib/uucp/newconfig/USERFILE
usr/lib/uucp/newconfig/dialit
usr/lib/uucp/newconfig/dialit.c
usr/lib/uucp/newconfig/uudemon.day
usr/lib/uucp/newconfig/uudemon.hr
usr/lib/uucp/newconfig/uudemon.wk
usr/lib/uucp/uucico
usr/lib/uucp/uuclean
usr/lib/uucp/uusub
usr/lib/uucp/uuxqt
usr/spool/uucp
usr/spool/uucppublic

FILE SET: APEDITOR

Description: Optional editors for the PE

usr/bin/ex8

FILE SET: APFILEMGR

Description: Optional commands used for file management

usr/bin/pack
usr/bin/unpack
usr/bin/cpset

FILE SET: APFILTER

Description: Optional commands for filtering files

bin/head
bin/nm
bin/od
bin/size
system/PFILTER/customize
usr/bin/adjust
usr/bin/ccat
usr/bin/comm
usr/bin/compact
usr/bin/dircmp
usr/bin/join
usr/bin/lorder
usr/bin/nl
usr/bin/paste
usr/bin/rev
usr/bin/rmnl
usr/bin/split
usr/bin/ssp
usr/bin/strings
usr/bin/tsort
usr/bin/uncompact
usr/bin/uniq
usr/lib/diff3prog
usr/lib/diffh
usr/lib/libndir.a

FILE SET: APNLS

Description: Optional files for Native Language Support

usr/bin/dumpmsg
usr/bin/findmsg
usr/bin/findstr
usr/bin/insertmsg
usr/bin/inv
usr/bin/vis
usr/lib/nls/american
usr/lib/nls/c-french
usr/lib/nls/config
usr/lib/nls/danish
usr/lib/nls/dutch
usr/lib/nls/english
usr/lib/nls/finnish
usr/lib/nls/french
usr/lib/nls/german
usr/lib/nls/italian
usr/lib/nls/kanji
usr/lib/nls/katakana
usr/lib/nls/norwegian
usr/lib/nls/portuguese
usr/lib/nls/spanish
usr/lib/nls/swedish
usr/bin/gencat

FILE SET: APCEUTIL

Description: CE utilities

usr/CE.utilities
usr/CE.utilities/CS80
usr/CE.utilities/CS80/exerciser
usr/CE.utilities/Crtadjust
usr/CE.utilities/Crtadjust/adjust
usr/CE.utilities/Crtadjust/adjustdevmenu
usr/CE.utilities/Crtadjust/adjustmenu
usr/CE.utilities/Crtadjust/adjusttest
usr/CE.utilities/Crtadjust/align.c
usr/CE.utilities/Crtadjust/bar.c
usr/CE.utilities/Crtadjust/clear.c
usr/CE.utilities/Crtadjust/get4digits
usr/CE.utilities/Crtadjust/help
usr/CE.utilities/Crtadjust/helptext
usr/CE.utilities/Crtadjust/runtest
usr/CE.utilities/Crtadjust/white.c
usr/CE.utilities/Gsft
usr/CE.utilities/Gsft/function
usr/CE.utilities/Gsft/help
usr/CE.utilities/Gsft/helptext
usr/CE.utilities/Gsft/mc_98710
usr/CE.utilities/Gsft/mc_98710/alu.o
usr/CE.utilities/Gsft/mc_98710/dram.o
usr/CE.utilities/Gsft/mc_98710/fpchips.o
usr/CE.utilities/Gsft/mc_98710/kernal.o
usr/CE.utilities/Gsft/mc_98710/pram.o
usr/CE.utilities/Gsft/mc_98710/scan.o
usr/CE.utilities/Gsft/mc_98710/seq.o
usr/CE.utilities/Gsft/pattern
usr/CE.utilities/Gsft/pattern98700
usr/CE.utilities/Gsft/pattern98700me
usr/CE.utilities/Gsft/test98700
usr/CE.utilities/Gsft/test98700menu
usr/CE.utilities/Sft
usr/CE.utilities/Sft/discmenu
usr/CE.utilities/Sft/disctest
usr/CE.utilities/Sft/disctest2
usr/CE.utilities/Sft/get4digits
usr/CE.utilities/Sft/get6digits
usr/CE.utilities/Sft/graphics.c
usr/CE.utilities/Sft/graphicsmenu
usr/CE.utilities/Sft/graphicstest
usr/CE.utilities/Sft/help
usr/CE.utilities/Sft/helptext
usr/CE.utilities/Sft/printermenu
usr/CE.utilities/Sft/printertest
usr/CE.utilities/Sft/readterm
usr/CE.utilities/Sft/readterm.c
usr/CE.utilities/Sft/rs232test
usr/CE.utilities/Sft/rungrafxtest
usr/CE.utilities/Sft/runtest
usr/CE.utilities/Sft/sft
usr/CE.utilities/Sft/tablet.c
usr/CE.utilities/Sft/tablettest
usr/CE.utilities/Sft/tapemenu
usr/CE.utilities/Sft/tapetest
usr/CE.utilities/Sft/termtest
usr/CE.utilities/Sft/testpattern
usr/CE.utilities/Sft/topmenu
usr/CE.utilities/help
usr/CE.utilities/helptext

PARTITION: PROG_LANGS

This partition is contained on the blue labeled micro disc. Both file sets contained in this partition are used for configuring your operating system and making a recovery system.

FILE SET: AC

bin
bin/cc
lib
lib/c2
lib/ccom
lib/cpp
lib/libc.a

FILE SET: APROG

bin
bin/ar
bin/as
bin/ld
bin/make
bin/strip

FILE SET: APC

Description: C compiler files

bin/prof
bin/time
usr/bin/cb
usr/bin/cflow
usr/bin/cxref
usr/bin/lint
usr/lib/dag
usr/lib/flip
usr/lib/lint1
usr/lib/lint2
usr/lib/llib-lc
usr/lib/llib-lc.ln
usr/lib/llib-port
usr/lib/llib-port.ln
usr/lib/xcpp
usr/lib/xpass

FILE SET: APC2

Description: C compiler include files

usr/include/a.exec.h
usr/include/a.out.h
usr/include/alarm.h
usr/include/ar.h
usr/include/assert.h
usr/include/b.out.h
usr/include/checklist.h
usr/include/core.h
usr/include/ctype.h
usr/include/curses.h
usr/include/dial.h
usr/include/disktab.h
usr/include/dumprestor.h
usr/include/dvio.h
usr/include/errnet.h
usr/include/errno.h
usr/include/execargs.h
usr/include/fatal.h
usr/include/fcntl.h
usr/include/ftw.h
usr/include/globaldefs.h
usr/include/grp.h
usr/include/iovalid.h
usr/include/kern_prof.h
usr/include/langinfo.h
usr/include/machine/clock.h
usr/include/machine/param.h
usr/include/machine/pcb.h
usr/include/machine/psl.h
usr/include/machine/pte.h
usr/include/machine/reg.h
usr/include/machine/sendsig.h
usr/include/machine/trap.h
usr/include/machine/vmparam.h
usr/include/macros.h
usr/include/magic.h
usr/include/malloc.h
usr/include/math.h
usr/include/memory.h
usr/include/misc.h
usr/include/mnttab.h
usr/include/model.h
usr/include/mon.h
usr/include/msgbuf.h
usr/include/msgcat.h
usr/include/ndir.h
usr/include/nl_ctype.h
usr/include/nlist.h
usr/include/oldfatal.h
usr/include/oldmacros.h
usr/include/port.h
usr/include/pwd.h
usr/include/ranlib.h
usr/include/regexp.h
usr/include/rje.h
usr/include/search.h

```
usr/include/setjmp.h
usr/include/sgtty.h
usr/include/signal.h
usr/include/siovalid.h
usr/include/stand.h
usr/include/stdio.h
usr/include/string.h
usr/include/symbol.h
usr/include/sys/acct.h
usr/include/sys/amigo.h
usr/include/sys/bootrom.h
usr/include/sys/buf.h
usr/include/sys/callout.h
usr/include/sys/ciper.h
usr/include/sys/clock.h
usr/include/sys/cmap.h
usr/include/sys/conf.h
usr/include/sys/cons.h
usr/include/sys/cs80.h
usr/include/sys/dbg.h
usr/include/sys/dil.h
usr/include/sys/dilio.h
usr/include/sys/dir.h
usr/include/sys/dk.h
usr/include/sys/dma.h
usr/include/sys/dmap.h
usr/include/sys/errno.h
usr/include/sys/fblk.h
usr/include/sys/file.h
usr/include/sys/filsys.h
usr/include/sys/fs.h
usr/include/sys/gpio.h
usr/include/sys/graf.h
usr/include/sys/graphics.h
usr/include/sys/hil.h
usr/include/sys/hilioctl.h
usr/include/sys/hpib.h
usr/include/sys/hpibio.h
usr/include/sys/ino.h
usr/include/sys/inode.h
usr/include/sys/intrpt.h
usr/include/sys/iobuf.h
usr/include/sys/ioctl.h
usr/include/sys/iomap.h
usr/include/sys/ipc.h
usr/include/sys/ite.h
usr/include/sys/kbd.h
usr/include/sys/kbd_chars.h
usr/include/sys/kernel.h
usr/include/sys/lanfunc.h
usr/include/sys/lock.h
usr/include/sys/lp.h
usr/include/sys/lprio.h
usr/include/sys/map.h
usr/include/sys/mbuf.h
usr/include/sys/mf.h
usr/include/sys/mknod.h
usr/include/sys/mman.h
usr/include/sys/modem.h
usr/include/sys/mount.h
usr/include/sys/msg.h
usr/include/sys/msgbuf.h
usr/include/sys/mtio.h
usr/include/sys/mux.h
usr/include/sys/nami.h
usr/include/sys/opt.h
usr/include/sys/optflag.h
usr/include/sys/param.h
usr/include/sys/pcb.h
usr/include/sys/pdi.h
usr/include/sys/privgrp.h
usr/include/sys/proc.h
usr/include/sys/psl.h
usr/include/sys/pt.h
usr/include/sys/ptrace.h
usr/include/sys/pty.h
usr/include/sys/ptyio.h
usr/include/sys/reboot.h
usr/include/sys/reg.h
usr/include/sys/resource.h
usr/include/sys/rfa_user.h
usr/include/sys/rje_ioctl.h
usr/include/sys/rjedriver.h
usr/include/sys/rtprio.h
usr/include/sys/seg.h
usr/include/sys/sem.h
usr/include/sys/shm.h
usr/include/sys/signal.h
usr/include/sys/simon.h
usr/include/sys/space.h
usr/include/sys/stat.h
usr/include/sys/sysmacros.h
usr/include/sys/systm.h
usr/include/sys/termio.h
usr/include/sys/text.h
usr/include/sys/ti9914.h
usr/include/sys/timeb.h
usr/include/sys/timeout.h
usr/include/sys/times.h
usr/include/sys/trace.h
usr/include/sys/trap.h
usr/include/sys/tryrec.h
usr/include/sys/ttold.h
usr/include/sys/tty.h
usr/include/sys/types.h
usr/include/sys/uio.h
usr/include/sys/user.h
usr/include/sys/utsname.h
usr/include/sys/vm.h
usr/include/sys/vmmac.h
```

usr/include/sys/vmmeter.h
usr/include/sys/vmparam.h
usr/include/sys/vmsystm.h
usr/include/sys/vtimes.h
usr/include/tcio.h
usr/include/term.h
usr/include/termio.h
usr/include/time.h
usr/include/tp_defs.h
usr/include/tryrec.h
usr/include/unctrl.h
usr/include/unistd.h
usr/include/ustat.h
usr/include/utmp.h
usr/include/values.h
usr/include/varargs.h
usr/include/volhdr.h

FILE SET: APFORTRAN

Description: Optional fortran compilers

usr/bin/ratfor

FILE SET: APPROG

Description: Optional commands for programming support

bin/adb
bin/arcv
bin/atrans
bin/cdb
bin/m4
lib/crt0.o
lib/lib881.a
lib/libPW.a
lib/libm.a
lib/mcrt0.o

FILE SET: APPROG2

Description: Second set of optional commands for programming support

usr/bin/chatr
usr/bin/lex
usr/bin/mkstr
usr/bin/yacc
usr/lib/ar.ranlib
usr/lib/cdb.errors
usr/lib/cdb.help
usr/lib/cdb.help.man
usr/lib/end.o
usr/lib/lex
usr/lib/lex/ncform
usr/lib/lex/nrform
usr/lib/libBSD.a
usr/lib/libcurses.a
usr/lib/libdvio.a
usr/lib/libhp.a
usr/lib/libl.a
usr/lib/libmalloc.a
usr/lib/liby.a
usr/lib/llib-lm
usr/lib/llib-lm.ln
usr/lib/llib-lmalloc.l
usr/lib/yaccpar

FILE SET: A98519A

Description: Single-user Pascal compiler

bin/pc
lib/libpc.a
usr/lib/pascomp
usr/lib/libheap2.a
usr/lib/escerrs
usr/lib/ioerrs
usr/lib/paserrs
usr/lib/syserrs
system/98519A/revlist
system/98519A/A98519A

FILE SET: A98599A

Description: Multi-user Pascal

bin/pc ✓
lib/libpc.a ✓
usr/lib/pascomp ✓
usr/lib/libheap2.a ✓
usr/lib/escerrs ✓
usr/lib/ioerrs ✓
usr/lib/paserrs ✓
usr/lib/syserrs ✓
system/98599A/A98599A -
system/98599A/revlist ✓

FILE SET: A98518A

Description: Single-user Fortran copiler

lib/f1
lib/frt0.o
lib/mfrt0.o
usr/bin/f77
usr/lib/f77pass1
usr/lib/libF77.a
usr/lib/libI77.a
system/98518A/revlist
system/98518A/customize

system/98518A/A98518A

FILE SET: A98598A

Description: Multi-user Fortran

lib/f1
lib/frt0.o
lib/mfrt0.o
usr/bin/f77
usr/lib/f77pass1
usr/lib/libF77.a
usr/lib/libI77.a
system/98598A/A98598A
system/98598A/revlist
system/98598A/customize

PARTITION: MISC_UTILS

This partition is contained on the 2 green labeled micro discs. The two file sets in this partition are used for reconfiguring the operating system and for making a recovery system.

FILE SET: ACONFIG

etc
etc/master
etc/mkrs
etc/mkrs.devs
etc/mkrs.swap
etc/config
etc/conf
etc/conf/h
etc/conf/h/buf.h
etc/conf/h/callout.h
etc/conf/h/cmap.h
etc/conf/h/conf.h
etc/conf/h/dir.h
etc/conf/h/dmap.h
etc/conf/h/errno.h
etc/conf/h/ino.h
etc/conf/h/inode.h
etc/conf/h/ioctl.h
etc/conf/h/ipc.h
etc/conf/h/kernel.h
etc/conf/h/magic.h
etc/conf/h/map.h
etc/conf/h/msg.h
etc/conf/h/opt.h
etc/conf/h/param.h
etc/conf/h/privgrp.h
etc/conf/h/proc.h
etc/conf/h/pty.h
etc/conf/h/ptyio.h
etc/conf/h/resource.h
etc/conf/h/sem.h
etc/conf/h/shm.h
etc/conf/h/signal.h
etc/conf/h/space.h
etc/conf/h/sysmacros.h
etc/conf/h/systm.h
etc/conf/h/text.h
etc/conf/h/time.h
etc/conf/h/tty.h
etc/conf/h/types.h
etc/conf/h/user.h
etc/conf/libdil_srm.a
etc/conf/libdreq.a
etc/conf/liblan.a
etc/conf/libmin.a
etc/conf/machine
etc/conf/machine/a.out.h
etc/conf/machine/param.h
etc/conf/machine/pcb.h
etc/conf/machine/pte.h
etc/conf/machine/timeout.h

FILE SET: ACONFIG2

etc
etc/conf
etc/conf/dfile.full
etc/conf/dfile.full.lan
etc/conf/dfile.min
etc/conf/dfile.support
etc/conf/libdevelop.a
etc/conf/libsysV.a
etc/conf/libdreq.a
etc/conf/libkreq.a

FILE SET: APAPPLIC

Description: Special application files

bin/bs
bin/chsh
etc/newconfig/lib.b
lib/libld.a
usr/bin/bc

usr/bin/cal
usr/bin/calendar
usr/bin/captoinfo
usr/bin/clear
usr/bin/dc
usr/bin/factor
usr/bin/leave
usr/bin/lock
usr/bin/primes
usr/bin/tic
usr/bin/units
usr/bin/untic
usr/lib/builtins
usr/lib/builtins/@
usr/lib/builtins/alias
usr/lib/builtins/alloc
usr/lib/builtins/case
usr/lib/builtins/cd
usr/lib/builtins/chdir
usr/lib/builtins/dirs
usr/lib/builtins/eval
usr/lib/builtins/exec
usr/lib/builtins/foreach
usr/lib/builtins/glob
usr/lib/builtins/hashstat
usr/lib/builtins/history
usr/lib/builtins/if
usr/lib/builtins/jobs
usr/lib/builtins/kill
usr/lib/builtins/login
usr/lib/builtins/logout
usr/lib/builtins/newgrp
usr/lib/builtins/nice
usr/lib/builtins/nohup
usr/lib/builtins/notify
usr/lib/builtins/onintr
usr/lib/builtins/popd
usr/lib/builtins/pushd
usr/lib/builtins/rehash
usr/lib/builtins/repeat
usr/lib/builtins/set
usr/lib/builtins/setenv
usr/lib/builtins/source
usr/lib/builtins/stop
usr/lib/builtins/switch
usr/lib/builtins/time
usr/lib/builtins/umask
usr/lib/builtins/unalias
usr/lib/builtins/unhash
usr/lib/builtins/unset
usr/lib/builtins/unsetenv
usr/lib/builtins/wait
usr/lib/builtins/while
usr/lib/calprog
usr/lib/unittab

FILE SET: APPERIPH

Description: Optional commands for peripheral support

bin/mt
etc/lsdev
usr/bin/slp
usr/bin/reset

FILE SET: APSCCS

Description: Optional commands for SCCS support

usr/bin/admin
usr/bin/delta
usr/bin/get
usr/bin/prs
usr/bin/rmdel
usr/bin/sccsdiff
usr/bin/unget
usr/bin/val

FILE SET: APACCT

Description: System accounting files

usr/adm/acct
usr/adm/acct/fiscal
usr/adm/acct/nite
usr/adm/acct/sum
usr/lib/acct/acctcms
usr/lib/acct/acctcom
usr/lib/acct/acctcon1
usr/lib/acct/acctcon2
usr/lib/acct/acctdisk
usr/lib/acct/acctdusg
usr/lib/acct/acctmerg
usr/lib/acct/accton
usr/lib/acct/acctprc1
usr/lib/acct/acctprc2
usr/lib/acct/acctwtmp
usr/lib/acct/chargefee
usr/lib/acct/ckpacct
usr/lib/acct/diskusg
usr/lib/acct/dodisk
usr/lib/acct/fwtmp
usr/lib/acct/holidays
usr/lib/acct/lastlogin
usr/lib/acct/monacct
usr/lib/acct/nulladm
usr/lib/acct/prctmp
usr/lib/acct/prdaily

usr/lib/acct/prtacct
usr/lib/acct/ptecms.awk
usr/lib/acct/ptelus.awk
usr/lib/acct/remove
usr/lib/acct/runacct
usr/lib/acct/shutacct
usr/lib/acct/startup
usr/lib/acct/turnacct
usr/lib/acct/wtmpfix

FILE SET: A50951A

Description: Single user LAN on the S300

etc/conf/liblan.a
etc/rfadaemon
etc/rfaserver
etc/nftdaemon
etc/nftserver
etc/rlbdaemon
etc/rlbserver
etc/npowerup
etc/linkloop
system/50951A/revlist
system/50951A/customize
usr/bin/nusers
usr/bin/landiag
usr/bin/dscopy
usr/lib/nls/n-computer/lan.cat
usr/lib/libn.a
usr/include/netio.h
usr/netdemo/LLA/nget.c
usr/netdemo/LLA/nput.c
usr/netdemo/LLA/nserver.c
usr/netdemo/LLA/ncopy.h
usr/man/man4/lan.4
usr/man/man3/net_aton.3
usr/man/man2/netunam.2
usr/man/man2/errnet.2
usr/man/man1/netunam.1
usr/man/man1/dscopy.1
usr/man/man1/landiag.1
usr/man/man1/npowerup.1
usr/man/man1/nusers.1
usr/man/man1m/linkloop.1m

FILE SET: A50952A

Description: Multi-user LAN on the S300

etc/conf/liblan.a
etc/rfadaemon
etc/rfaserver
etc/nftdaemon
etc/nftserver
etc/rlbdaemon
etc/rlbserver
etc/npowerup
etc/linkloop
system/50952A/revlist
system/50952A/customize
usr/bin/nusers
usr/bin/landiag
usr/bin/dscopy
usr/lib/nls/n-computer/lan.cat
usr/lib/libn.a
usr/include/netio.h
usr/netdemo/LLA/nget.c
usr/netdemo/LLA/nput.c
usr/netdemo/LLA/nserver.c
usr/netdemo/LLA/ncopy.h
usr/man/man4/lan.4
usr/man/man3/net_aton.3
usr/man/man2/netunam.2
usr/man/man2/errnet.2
usr/man/man1/netunam.1
usr/man/man1/dscopy.1
usr/man/man1/landiag.1
usr/man/man1/npowerup.1
usr/man/man1/nusers.1
usr/man/man1m/linkloop.1m

FILE SET: A50956A

Description: Single-user LAN on the S200

etc/conf/liblan.a
etc/rfadaemon
etc/rfaserver
etc/nftdaemon
etc/nftserver
etc/rlbdaemon
etc/rlbserver
etc/npowerup
etc/linkloop
system/50956A/revlist
system/50956A/customize
usr/bin/nusers
usr/bin/landiag
usr/bin/dscopy
usr/lib/nls/n-computer/lan.cat
usr/lib/libn.a
usr/include/netio.h
usr/netdemo/LLA/nget.c
usr/netdemo/LLA/nput.c
usr/netdemo/LLA/nserver.c
usr/netdemo/LLA/ncopy.h
usr/man/man4/lan.4
usr/man/man3/net_aton.3
usr/man/man2/netunam.2
usr/man/man2/errnet.2

```
usr/man/man1/netunam.1
usr/man/man1/dscopy.1
usr/man/man1/landiag.1
usr/man/man1/npowerup.1
usr/man/man1/nusers.1
usr/man/man1m/linkloop.1m
```

FILE SET: A50957A

Description: Multi-user LAN on the S200

```
etc/conf/liblan.a
etc/rfadaemon
etc/rfaserver
etc/nftdaemon
etc/nftserver
etc/rlbdaemon
etc/rlbserver
etc/npowerup
etc/linkloop
system/50957A/revlist
system/50957A/customize
usr/bin/nusers
usr/bin/landiag
usr/bin/dscopy
usr/lib/nls/n-computer/lan.cat
usr/lib/libn.a
usr/include/netio.h
usr/netdemo/LLA/nget.c
usr/netdemo/LLA/nput.c
usr/netdemo/LLA/nserver.c
usr/netdemo/LLA/ncopy.h
usr/man/man4/lan.4
usr/man/man3/net_aton.3
usr/man/man2/netunam.2
usr/man/man2/errnet.2
usr/man/man1/netunam.1
usr/man/man1/dscopy.1
usr/man/man1/landiag.1
usr/man/man1/npowerup.1
usr/man/man1/nusers.1
usr/man/man1m/linkloop.1m
```

FILE SET: A98693A

Description: SRM

```
usr/bin/srmclean
usr/bin/srmcp
usr/bin/srmls
usr/bin/srmmkdir
usr/bin/srmmv
usr/bin/srmprotect
usr/bin/srmrm
usr/man/man1/srm.1
usr/man/man1/srmclean.1
usr/man/man1/srmcp.1
usr/man/man1/srmls.1
usr/man/man1/srmmkdir.1
usr/man/man1/srmmv.1
usr/man/man1/srmprotect.1
usr/man/man1/srmrm.1
system/98693A/customize
system/98693A/revlist
```

PARTITION: TEXT

This partition is contained on the orange labeled micro disc.

FILE SET: AMANUAL

Description: This file set contains formatted manual pages which give you a description of the commands available to you. If you load this file set, you can use the man command.

```
usr
usr/man
usr/man/cat1
usr/man/cat1/asa.1
usr/man/cat1/awk.1
usr/man/cat1/banner.1
usr/man/cat1/basename.1
usr/man/cat1/bfs.1
usr/man/cat1/bifchmod.1
usr/man/cat1/bifchown.1
usr/man/cat1/bifcp.1
usr/man/cat1/biffind.1
usr/man/cat1/biffsck.1
usr/man/cat1/bifls.1
usr/man/cat1/bifmkdir.1
usr/man/cat1/bifmkfs.1
usr/man/cat1/bifrm.1
usr/man/cat1/cat.1
usr/man/cat1/cd.1
usr/man/cat1/chmod.1
usr/man/cat1/chown.1
usr/man/cat1/cmp.1
usr/man/cat1/cp.1
usr/man/cat1/cpio.1
usr/man/cat1/csh.1
```

usr/man/cat1/cut.1
usr/man/cat1/date.1
usr/man/cat1/dd.1
usr/man/cat1/du.1
usr/man/cat1/echo.1
usr/man/cat1/ed.1
usr/man/cat1/edit.1
usr/man/cat1/enable.1
usr/man/cat1/ex.1
usr/man/cat1/expr.1
usr/man/cat1/find.1
usr/man/cat1/fold.1
usr/man/cat1/grep.1
usr/man/cat1/help.1
usr/man/cat1/hostname.1
usr/man/cat1/hp.1
usr/man/cat1/ipcrm.1
usr/man/cat1/ipcs.1
usr/man/cat1/intro.1
usr/man/cat1/kill.1
usr/man/cat1/lifcp.1
usr/man/cat1/lifinit.1
usr/man/cat1/lifls.1
usr/man/cat1/lifrename.1
usr/man/cat1/lifrm.1
usr/man/cat1/line.1
usr/man/cat1/login.1
usr/man/cat1/lp.1
usr/man/cat1/lpstat.1
usr/man/cat1/ls.1
usr/man/cat1/man.1
usr/man/cat1/mediainit.1
usr/man/cat1/mkdir.1
usr/man/cat1/more.1
usr/man/cat1/nice.1
usr/man/cat1/pam.1
usr/man/cat1/passwd.1
usr/man/cat1/pr.1
usr/man/cat1/ps.1
usr/man/cat1/pwd.1
usr/man/cat1/rm.1
usr/man/cat1/sh.1
usr/man/cat1/sleep.1
usr/man/cat1/sort.1
usr/man/cat1/stty.1
usr/man/cat1/su.1
usr/man/cat1/sync.1
usr/man/cat1/tabs.1
usr/man/cat1/tail.1
usr/man/cat1/tar.1
usr/man/cat1/tcio.1
usr/man/cat1/tee.1
usr/man/cat1/touch.1
usr/man/cat1/tr.1
usr/man/cat1/true.1
usr/man/cat1/tset.1
usr/man/cat1/umask.1
usr/man/cat1/uname.1
usr/man/cat1/vi.1
usr/man/cat1/wc.1
usr/man/cat1/what.1
usr/man/cat1/whereis.1
usr/man/cat1/who.1
usr/man/cat1/whoami.1
usr/man/cat1m
usr/man/cat1m/accept.1m
usr/man/cat1m/backup.1m
usr/man/cat1m/brc.1m
usr/man/cat1m/chroot.1m
usr/man/cat1m/clri.1m
usr/man/cat1m/config.1m
usr/man/cat1m/devnm.1m
usr/man/cat1m/fsck.1m
usr/man/cat1m/fsclean.1m
usr/man/cat1m/getty.1m
usr/man/cat1m/init.1m
usr/man/cat1m/lpadmin.1m
usr/man/cat1m/lpsched.1m
usr/man/cat1m/mkdev.1m
usr/man/cat1m/mkfs.1m
usr/man/cat1m/mknod.1m
usr/man/cat1m/mount.1m
usr/man/cat1m/mvdir.1m
usr/man/cat1m/newfs.1m
usr/man/cat1m/reboot.1m
usr/man/cat1m/reconfig.1m
usr/man/cat1m/setmnt.1m
usr/man/cat1m/shutdown.1m
usr/man/cat1m/syncer.1m
usr/man/cat5
usr/man/cat5/bif.5
usr/man/cat5/lif.5
system
system/AMANUAL
system/AMANUAL/customize

FILE SET: AHPUX_MAN

Description: First set of Manual pages

usr
usr/man
usr/man/man1
usr/man/man1/acctcom.1
usr/man/man1/as.1
usr/man/man1/adjust.1
usr/man/man1/admin.1
usr/man/man1/ar.1
usr/man/man1/arcv.1
usr/man/man1/atrans.1

usr/man/man1/asa.1
usr/man/man1/at.1
usr/man/man1/yacc.1
usr/man/man1/chatr.1
usr/man/man1/awk.1
usr/man/man1/banner.1
usr/man/man1/basename.1
usr/man/man1/bc.1
usr/man/man1/bdiff.1
usr/man/man1/bfs.1
usr/man/man1/bifchmod.1
usr/man/man1/bifchown.1
usr/man/man1/bifcp.1
usr/man/man1/bifdf.1
usr/man/man1/biffind.1
usr/man/man1/biffsck.1
usr/man/man1/biffsdb.1
usr/man/man1/bifls.1
usr/man/man1/bifmkdir.1
usr/man/man1/bifmkfs.1
usr/man/man1/bifrm.1
usr/man/man1/bs.1
usr/man/man1/cal.1
usr/man/man1/calendar.1
usr/man/man1/cat.1
usr/man/man1/cb.1
usr/man/man1/cc.1
usr/man/man1/cd.1
usr/man/man1/cdb.1
usr/man/man1/cdc.1
usr/man/man1/cflow.1
usr/man/man1/lsdev.1
usr/man/man1/xargs.1
usr/man/man1/chmod.1
usr/man/man1/chown.1
usr/man/man1/chsh.1
usr/man/man1/clear.1
usr/man/man1/cmp.1
usr/man/man1/col.1
usr/man/man1/comm.1
usr/man/man1/compact.1
usr/man/man1/cp.1
usr/man/man1/cpio.1
usr/man/man1/cpp.1
usr/man/man1/crontab.1
usr/man/man1/csh.1
usr/man/man1/ctags.1
usr/man/man1/cu.1c
usr/man/man1/cut.1
usr/man/man1/cxref.1
usr/man/man1/date.1
usr/man/man1/dc.1
usr/man/man1/dd.1
usr/man/man1/delta.1
usr/man/man1/deroff.1
usr/man/man1/diff.1
usr/man/man1/diff3.1
usr/man/man1/diffmk.1
usr/man/man1/dircmp.1
usr/man/man1/du.1
usr/man/man1/echo.1
usr/man/man1/ed.1
usr/man/man1/edit.1
usr/man/man1/enable.1
usr/man/man1/env.1
usr/man/man1/write.1
usr/man/man1/ex.1
usr/man/man1/expand.1
usr/man/man1/expr.1
usr/man/man1/factor.1
usr/man/man1/fc.1
usr/man/man1/file.1
usr/man/man1/find.1
usr/man/man1/findmsg.1
usr/man/man1/findstr.1
usr/man/man1/fixman.1
usr/man/man1/fold.1
usr/man/man1/gencat.1
usr/man/man1/get.1
usr/man/man1/getopt.1
usr/man/man1/getprivgrp.1
usr/man/man1/grep.1
usr/man/man1/groups.1
usr/man/man1/head.1
usr/man/man1/help.1
usr/man/man1/hostname.1
usr/man/man1/hp.1
usr/man/man1/hyphen.1
usr/man/man1/id.1
usr/man/man1/insertmsg.1
usr/man/man1/intro.1
usr/man/man1/ipcrm.1
usr/man/man1/ipcs.1
usr/man/man1/join.1
usr/man/man1/kill.1
usr/man/man1/last.1
usr/man/man1/ld.1
usr/man/man1/leave.1

FILE SET: AHPUX_MAN2

Description: Second set of Manual pages

usr/man/man1/lex.1
usr/man/man1/lifcp.1
usr/man/man1/lifinit.1
usr/man/man1/lifls.1
usr/man/man1/lifrename.1
usr/man/man1/lifrm.1
usr/man/man1/line.1

usr/man/man1/linkinfo.1
usr/man/man1/sed.1
usr/man/man1/lock.1
usr/man/man1/login.1
usr/man/man1/logname.1
usr/man/man1/lorder.1
usr/man/man1/lp.1
usr/man/man1/lpstat.1
usr/man/man1/ls.1
usr/man/man1/mediainit.1
usr/man/man1/whoami.1
usr/man/man1/m4.1
usr/man/man1/machid.1
usr/man/man1/mail.1
usr/man/man1/mailx.1
usr/man/man1/make.1
usr/man/man1/man.1
usr/man/man1/nm.1
usr/man/man1/mesg.1
usr/man/man1/mkdir.1
usr/man/man1/mkstr.1
usr/man/man1/mm.1
usr/man/man1/more.1
usr/man/man1/mt.1
usr/man/man1/newgrp.1
usr/man/man1/news.1
usr/man/man1/nice.1
usr/man/man1/nl.1
usr/man/man1/prof.1
usr/man/man1/who.1
usr/man/man1/nohup.1
usr/man/man1/nroff.1
usr/man/man1/od.1
usr/man/man1/pack.1
usr/man/man1/pam.1
usr/man/man1/passwd.1
usr/man/man1/paste.1
usr/man/man1/pc.1
usr/man/man1/pr.1
usr/man/man1/prealloc.1
usr/man/man1/slp.1
usr/man/man1/prs.1
usr/man/man1/ps.1
usr/man/man1/vis.1
usr/man/man1/pwd.1
usr/man/man1/whereis.1
usr/man/man1/ratfor.1
usr/man/man1/rev.1
usr/man/man1/what.1
usr/man/man1/rm.1
usr/man/man1/rmdel.1
usr/man/man1/rmnl.1
usr/man/man1/rtprio.1
usr/man/man1/sact.1
usr/man/man1/sccsdiff.1
usr/man/man1/ptx.1
usr/man/man1/sh.1
usr/man/man1/size.1
usr/man/man1/sleep.1
usr/man/man1/tcio.1
usr/man/man1/sort.1
usr/man/man1/spell.1
usr/man/man1/split.1
usr/man/man1/ssp.1
usr/man/man1/strings.1
usr/man/man1/strip.1
usr/man/man1/stty.1
usr/man/man1/su.1
usr/man/man1/sum.1
usr/man/man1/sync.1
usr/man/man1/tabs.1
usr/man/man1/tail.1
usr/man/man1/tar.1
usr/man/man1/tbl.1
usr/man/man1/adb.1
usr/man/man1/wc.1
usr/man/man1/tee.1
usr/man/man1/test.1
usr/man/man1/time.1
usr/man/man1/touch.1
usr/man/man1/tput.1
usr/man/man1/tr.1
usr/man/man1/true.1
usr/man/man1/lint.1
usr/man/man1/tsort.1
usr/man/man1/tty.1
usr/man/man1/ul.1
usr/man/man1/umask.1
usr/man/man1/uname.1
usr/man/man1/unget.1
usr/man/man1/uniq.1
usr/man/man1/units.1
usr/man/man1/wait.1
usr/man/man1/uucp.1c
usr/man/man1/uuls.1c
usr/man/man1/uusnap.1c
usr/man/man1/uustat.1c
usr/man/man1/uuto.1c
usr/man/man1/uux.1c
usr/man/man1/val.1
usr/man/man1/vi.1
usr/man/man1/tset.1

FILE SET: AHPUX_MAN3

Description: Third set of Manual pages

usr/man/man2
usr/man/man2/access.2
usr/man/man2/acct.2

usr/man/man2/alarm.2
usr/man/man2/brk.2
usr/man/man2/chdir.2
usr/man/man2/chmod.2
usr/man/man2/chown.2
usr/man/man2/chroot.2
usr/man/man2/close.2
usr/man/man2/creat.2
usr/man/man2/dup.2
usr/man/man2/ftime.2
usr/man/man2/write.2
usr/man/man2/wait.2
usr/man/man2/errno.2
usr/man/man2/exec.2
usr/man/man2/exit.2
usr/man/man2/fcntl.2
usr/man/man2/fork.2
usr/man/man2/fsync.2
usr/man/man2/getgroups.2
usr/man/man2/getprivgrp.2
usr/man/man2/gethostname.2
usr/man/man2/getitimer.2
usr/man/man2/getpid.2
usr/man/man2/profil.2
usr/man/man2/gettimeofday.2
usr/man/man2/getuid.2
usr/man/man2/intro.2
usr/man/man2/ioctl.2
usr/man/man2/kill.2
usr/man/man2/link.2
usr/man/man2/lockf.2
usr/man/man2/lseek.2
usr/man/man2/umount.2
usr/man/man2/uname.2
usr/man/man2/vfork.2
usr/man/man2/utime.2
usr/man/man2/ustat.2
usr/man/man2/mkdir.2
usr/man/man2/mknod.2
usr/man/man2/mount.2
usr/man/man2/msgctl.2
usr/man/man2/msgget.2
usr/man/man2/msgop.2
usr/man/man2/nice.2
usr/man/man2/open.2
usr/man/man2/pause.2
usr/man/man2/pipe.2
usr/man/man2/plock.2
usr/man/man2/prealloc.2
usr/man/man2/ptrace.2
usr/man/man2/reboot.2
usr/man/man2/read.2
usr/man/man2/setgroups.2
usr/man/man2/rmdir.2
usr/man/man2/rtprio.2
usr/man/man2/select.2
usr/man/man2/semctl.2
usr/man/man2/semget.2
usr/man/man2/semop.2
usr/man/man2/swapon.2
usr/man/man2/sethostname.2
usr/man/man2/setpgrp.2
usr/man/man2/setuid.2
usr/man/man2/shmctl.2
usr/man/man2/shmget.2
usr/man/man2/shmop.2
usr/man/man2/sigblock.2
usr/man/man2/signal.2
usr/man/man2/sigpause.2
usr/man/man2/sigsetmask.2
usr/man/man2/sigspace.2
usr/man/man2/sigvector.2
usr/man/man2/stat.2
usr/man/man2/stime.2
usr/man/man2/stty.2
usr/man/man2/dup2.2
usr/man/man2/sync.2
usr/man/man2/time.2
usr/man/man2/times.2
usr/man/man2/unlink.2
usr/man/man2/truncate.2
usr/man/man2/ulimit.2
usr/man/man2/umask.2
usr/man/man3
usr/man/man3/a64l.3c
usr/man/man3/abort.3c
usr/man/man3/abs.3c
usr/man/man3/assert.3x
usr/man/man3/bessel.3m
usr/man/man3/bsearch.3c
usr/man/man3/catread.3c
usr/man/man3/clock.3c
usr/man/man3/conv.3c
usr/man/man3/crypt.3c
usr/man/man3/ctermid.3s
usr/man/man3/ctime.3c
usr/man/man3/ctype.3c
usr/man/man3/curses.3x
usr/man/man3/cuserid.3s
usr/man/man3/dial.3c
usr/man/man3/directory.3c
usr/man/man3/drand48.3c
usr/man/man3/ecvt.3c
usr/man/man3/end.3c
usr/man/man3/erf.3m
usr/man/man3/exp.3m
usr/man/man3/fclose.3s
usr/man/man3/ferror.3s
usr/man/man3/floor.3m
usr/man/man3/fopen.3s

usr/man/man3/fread.3s
usr/man/man3/frexp.3c
usr/man/man3/fseek.3s
usr/man/man3/ftw.3c
usr/man/man3/gamma.3m
usr/man/man3/getc.3s
usr/man/man3/getcwd.3c
usr/man/man3/getenv.3c
usr/man/man3/getfsent.3x
usr/man/man3/getgrent.3c
usr/man/man3/getlogin.3c
usr/man/man3/getmsg.3c
usr/man/man3/getopt.3c
usr/man/man3/getpass.3c
usr/man/man3/getpw.3c
usr/man/man3/getpwent.3c
usr/man/man3/gets.3s
usr/man/man3/getut.3c
usr/man/man3/gpio_get_st.3i
usr/man/man3/gpio_set_ct.3i
usr/man/man3/hpib_abort.3i
usr/man/man3/hpib_bus_st.3i
usr/man/man3/hpib_card_p.3i
usr/man/man3/hpib_eoi_ct.3i
usr/man/man3/hpib_io.3i
usr/man/man3/hpib_pass_c.3i
usr/man/man3/hpib_ppoll.3i
usr/man/man3/hpib_ppoll_.3i
usr/man/man3/hpib_ren_ct.3i
usr/man/man3/hpib_rqst_s.3i
usr/man/man3/hpib_send_c.3i
usr/man/man3/hpib_spoll.3i
usr/man/man3/hpib_status.3i
usr/man/man3/hpib_wait_o.3i
usr/man/man3/hsearch.3c
usr/man/man3/hypot.3m
usr/man/man3/monitor.3c
usr/man/man3/intrapoff.3m
usr/man/man3/intro.3
usr/man/man3/io_burst.3i
usr/man/man3/io_eol_ctl.3i
usr/man/man3/io_get_term.3i
usr/man/man3/io_interrup.3i
usr/man/man3/io_on_inter.3i
usr/man/man3/io_reset.3i
usr/man/man3/io_speed_ct.3i
usr/man/man3/io_timeout_.3i
usr/man/man3/io_width_ct.3i
usr/man/man3/l3tol.3c
usr/man/man3/langinfo.3c
usr/man/man3/logname.3x
usr/man/man3/lsearch.3c
usr/man/man3/malloc.3c
usr/man/man3/malloc.3x
usr/man/man3/matherr.3m
usr/man/man3/memory.3c
usr/man/man3/mktemp.3c
usr/man/man3/initgroups.3c
usr/man/man3/nl_conv.3c
usr/man/man3/nl_ctype.3c
usr/man/man3/nl_string.3c
usr/man/man3/nlist.3c
usr/man/man3/perror.3c
usr/man/man3/popen.3s
usr/man/man3/printf.3s
usr/man/man3/printmsg.3c
usr/man/man3/putc.3s
usr/man/man3/putenv.3c
usr/man/man3/putpwent.3c
usr/man/man3/puts.3s
usr/man/man3/qsort.3c
usr/man/man3/rand.3c
usr/man/man3/regcmp.3x
usr/man/man3/scanf.3s
usr/man/man3/setbuf.3s
usr/man/man3/setjmp.3c
usr/man/man3/sinh.3m
usr/man/man3/sleep.3c
usr/man/man3/sputl.3x
usr/man/man3/ssignal.3c
usr/man/man3/stdio.3s
usr/man/man3/stdipc.3c
usr/man/man3/string.3c
usr/man/man3/strtod.3c
usr/man/man3/strtol.3c
usr/man/man3/swab.3c
usr/man/man3/system.3s
usr/man/man3/termcap.3x
usr/man/man3/tmpfile.3s
usr/man/man3/tmpnam.3s
usr/man/man3/trig.3m
usr/man/man3/tsearch.3c
usr/man/man3/ttyname.3c
usr/man/man3/ttyslot.3c
usr/man/man3/ungetc.3s
usr/man/man3/vprintf.3s

FILE SET: AHPUX_MAN4

Description: Fourth set of Manual pages

usr/man/man4
usr/man/man4/ct.4
usr/man/man4/disc.4
usr/man/man4/iomap.4
usr/man/man4/hpib.4
usr/man/man4/intro.4
usr/man/man4/graphics.4
usr/man/man4/lp.4
usr/man/man4/mem.4

usr/man/man4/modem.4
usr/man/man4/mt.4
usr/man/man4/null.4
usr/man/man4/pty.4
usr/man/man4/sttyv6.4
usr/man/man4/termio.4
usr/man/man4/tty.4
usr/man/man5
usr/man/man5/core.5
usr/man/man5/a.out.5
usr/man/man5/passwd.5
usr/man/man5/dir.5
usr/man/man5/ar.5
usr/man/man5/bif.5
usr/man/man5/checklist.5
usr/man/man5/col_seq_16.5
usr/man/man5/col_seq_8.5
usr/man/man5/fs.5
usr/man/man5/cpio.5
usr/man/man5/dialups.5
usr/man/man5/inode.5
usr/man/man5/master.5
usr/man/man5/privgrp.5
usr/man/man5/disktab.5
usr/man/man5/utmp.5
usr/man/man5/ttytype.5
usr/man/man5/terminfo.5
usr/man/man5/fspec.5
usr/man/man5/gettydefs.5
usr/man/man5/group.5
usr/man/man5/inittab.5
usr/man/man5/term.5
usr/man/man5/sccsfile.5
usr/man/man5/intro.5
usr/man/man5/issue.5
usr/man/man5/lif.5
usr/man/man5/magic.5
usr/man/man5/ranlib.5
usr/man/man5/mknod.5
usr/man/man5/mnttab.5
usr/man/man5/model.5
usr/man/man5/profile.5
usr/man/man5/acct.5
usr/man/man6
usr/man/man7
usr/man/man7/ascii.7
usr/man/man7/environ.7
usr/man/man7/fcntl.7
usr/man/man7/hier.7
usr/man/man7/hpnls.7
usr/man/man7/intro.7
usr/man/man7/kana8.7
usr/man/man7/langid.7
usr/man/man7/man.7
usr/man/man7/math.7
usr/man/man7/mm.7
usr/man/man7/regexp.7
usr/man/man7/roman8.7
usr/man/man7/stat.7
usr/man/man7/term.7
usr/man/man7/types.7
usr/man/man7/values.7
usr/man/man7/varargs.7
usr/man/man8
usr/man/man8/intro.8
usr/man/man9
usr/man/man9/intro.9
usr/man/man1m
usr/man/man1m/accept.1m
usr/man/man1m/acct.1m
usr/man/man1m/acctcms.1m
usr/man/man1m/acctcon.1m
usr/man/man1m/acctmerg.1m
usr/man/man1m/acctprc.1m
usr/man/man1m/acctsh.1m
usr/man/man1m/clri.1m
usr/man/man1m/backup.1m
usr/man/man1m/brc.1m
usr/man/man1m/sysrm.1m
usr/man/man1m/config.1m
usr/man/man1m/catman.1m
usr/man/man1m/chroot.1m
usr/man/man1m/tunefs.1m
usr/man/man1m/swapon.1m
usr/man/man1m/clrsvc.1m
usr/man/man1m/fsck.1m
usr/man/man1m/cpset.1m
usr/man/man1m/cron.1m
usr/man/man1m/devnm.1m
usr/man/man1m/df.1m
usr/man/man1m/diskusg.1m
usr/man/man1m/fsclean.1m
usr/man/man1m/setprivgrp.1m
usr/man/man1m/fsdb.1m
usr/man/man1m/mkfs.1m
usr/man/man1m/captoinfo.1m
usr/man/man1m/fwtmp.1m
usr/man/man1m/getty.1m
usr/man/man1m/ncheck.1m
usr/man/man1m/init.1m
usr/man/man1m/install.1m
usr/man/man1m/intro.1m
usr/man/man1m/kermit.1m
usr/man/man1m/killall.1m
usr/man/man1m/link.1m
usr/man/man1m/lpadmin.1m
usr/man/man1m/lpsched.1m
usr/man/man1m/makekey.1m
usr/man/man1m/mkdev.1m
usr/man/man1m/newfs.1m

usr/man/man1m/mklp.1m
usr/man/man1m/mknod.1m
usr/man/man1m/mkrs.1m
usr/man/man1m/mount.1m
usr/man/man1m/mvdir.1m
usr/man/man1m/reboot.1m
usr/man/man1m/whodo.1m
usr/man/man1m/wall.1m
usr/man/man1m/uuxqt.1m
usr/man/man1m/uusub.1m
usr/man/man1m/uuclean.1m
usr/man/man1m/uucico.1m
usr/man/man1m/pwck.1m
usr/man/man1m/umodem.1m
usr/man/man1m/reconfig.1m
usr/man/man1m/revck.1m
usr/man/man1m/rootmark.1m
usr/man/man1m/runacct.1m
usr/man/man1m/shutdown.1m
usr/man/man1m/setmnt.1m
usr/man/man1m/syncer.1m
usr/man/man1m/update.1m
usr/man/man1m/untic.1m
usr/man/man1m/tic.1m
usr/man/man1m/termconv.1m
system
system/HPUX_MAN4
system/HPUX_MAN4/customize

FILE SET: APTEXT

Description: Text editing commands

bin/hyphen
etc/newconfig/hlist
etc/newconfig/hlista
etc/newconfig/hlistb
etc/newconfig/hstop
usr/bin/deroff
usr/bin/diffmk
usr/bin/mm
usr/bin/mmchek
usr/bin/neqn
usr/bin/nroff
usr/bin/osdd
usr/bin/ptx
usr/bin/spell
usr/bin/tbl
usr/bin/ul

FILE SET: APTEXT2

Description: Second set of text editing commands

usr/lib/eign
usr/lib/macros
usr/lib/macros/an
usr/lib/macros/an6
usr/lib/macros/cmp.n.d.an
usr/lib/macros/cmp.n.d.m
usr/lib/macros/cmp.n.t.an
usr/lib/macros/cmp.n.t.m
usr/lib/macros/mmn
usr/lib/macros/osdd
usr/lib/macros/ucmp.n.an
usr/lib/macros/ucmp.n.m
usr/lib/macros/vmca
usr/lib/ms
usr/lib/ms/README
usr/lib/ms/end.awk
usr/lib/ms/endnote
usr/lib/ms/s.acc
usr/lib/ms/s.cov
usr/lib/ms/s.eqn
usr/lib/ms/s.ref
usr/lib/ms/s.tbl
usr/lib/ms/s.ths
usr/lib/ms/s.toc
usr/lib/spell
usr/lib/spell/compress
usr/lib/spell/hlista
usr/lib/spell/hlistb
usr/lib/spell/hstop
usr/lib/spell/spellin
usr/lib/spell/spellout
usr/lib/spell/spellprog
usr/lib/suftab
usr/lib/tmac
usr/lib/tmac/tmac.an
usr/lib/tmac/tmac.an6
usr/lib/tmac/tmac.m
usr/lib/tmac/tmac.osd
usr/lib/tmac/tmac.s
usr/lib/tmac/tmac.v

PARTITION: GRAPHICS

This partition is contained on the purple labeled micro disc.

FILE SET: A98520A

Description: Single-user DGL/AGP graphics

usr/lib/graphics/demos/README_agp
usr/lib/graphics/demos/view_agp.f
usr/lib/graphics/demos/house_agp.f
usr/lib/graphics/demos/house_agp.p
usr/lib/graphics/demos/chrt_agp.f
usr/lib/graphics/demos/linkwsp
usr/lib/graphics/demos/README_dgl
usr/lib/graphics/demos/graph_dgl.f
usr/lib/graphics/demos/chrt_dgl.f
usr/lib/graphics/pascal/pdgl1.h
usr/lib/graphics/pascal/pagp1.h
usr/lib/graphics/pascal/pdgl2.h
usr/lib/graphics/pascal/pagp2.h
usr/lib/graphics/fortran/k1fil.f
usr/lib/graphics/fortran/t1int.f
usr/lib/graphics/fortran/z1ctb.f
usr/lib/graphics/fortran/fdgl1.h
usr/lib/graphics/fortran/fagp1.h
usr/lib/graphics/c/cdgl1.h
usr/lib/graphics/c/cagp1.h
usr/lib/libUPLIB.a
system/98520A/revlist
system/98520A/customize

FILE SET: A98520A2

Description: Single-user DGL/AGP graphics

usr/lib/libWSPLIB.a
usr/lib/wsp.o
usr/lib/k2sdf.f
lib/libA0000.a
lib/libA0001.a
lib/libB0000.a
lib/libB0001.a
lib/libD0001.a
lib/libD0019.a
lib/libD0020.a
lib/libD0036.a
lib/libK0000.a
lib/libK0001.a
lib/libL0001.a
lib/libL0019.a
lib/libP0001.a
lib/libP0019.a
lib/libV0001.a
lib/libV0019.a
lib/libL0004.a
lib/libB0004.a
lib/libP0004.a
lib/libV0004.a
lib/libA0047.a
lib/libB0047.a
lib/libD0006.a
lib/libD0015.a
lib/libD0016.a
lib/libD0031.a
lib/libD0046.a
system/98520A2/revlist
system/98520A2/customize

FILE SET: A98520A3

Description: Single-user DGL/AGP graphics

lib/libD0047.a
lib/libD0048.a
lib/libL0002.a
lib/libL0006.a
lib/libL0031.a
lib/libL0046.a
lib/libP0002.a
lib/libP0006.a
lib/libD0053.a
lib/libD0054.a
lib/libB0056.a
lib/libD0056.a
lib/libL0056.a
lib/libP0056.a
lib/libV0056.a
lib/libd.262x.a
lib/libd.2623.a
lib/libd.2627.a
lib/libd.hpgl.a
lib/libd.hil.a
lib/libd.kbd.a
lib/libd.9836a.a
lib/libd.9836c.a
lib/libd.9837.a
lib/libd.300h.a
lib/libd.3001.a
lib/libd.98700.a
lib/libd.98710.a
lib/libD0042.a

lib/libL0042.a
lib/libP0042.a
lib/libV0042.a
system/98520A3/revlist
system/98520A3/customize

FILE SET: A98520A4

Description: Single-user DGL/AGP graphics

lib/libDIDD.a
system/98520A4/revlist
system/98520A4/customize

FILE SET: A98600A

Description: Multi-user DGL/AGP

usr/lib/graphics/demos/README_agp
usr/lib/graphics/demos/view_agp.f
usr/lib/graphics/demos/house_agp.f
usr/lib/graphics/demos/house_agp.p
usr/lib/graphics/demos/chrt_agp.f
usr/lib/graphics/demos/linkwsp
usr/lib/graphics/demos/README_dgl
usr/lib/graphics/demos/graph_dgl.f
usr/lib/graphics/demos/chrt_dgl.f
usr/lib/graphics/pascal/pdgl1.h
usr/lib/graphics/pascal/pagp1.h
usr/lib/graphics/pascal/pdgl2.h
usr/lib/graphics/pascal/pagp2.h
usr/lib/graphics/fortran/k1fil.f
usr/lib/graphics/fortran/t1int.f
usr/lib/graphics/fortran/z1ctb.f
usr/lib/graphics/fortran/fdgl1.h
usr/lib/graphics/fortran/fagp1.h
usr/lib/graphics/c/cdgl1.h
usr/lib/graphics/c/cagp1.h
usr/lib/libUPLIB.a
system/98600A1/revlist
system/98600A1/customize

FILE SET: A98600A2

Description: Multi-user DGL/AGP

usr/lib/libWSPLIB.a
usr/lib/wsp.o
usr/lib/k2sdf.f
lib/libA0000.a
lib/libA0001.a
lib/libB0000.a
lib/libB0001.a
lib/libD0001.a
lib/libD0019.a
lib/libD0020.a
lib/libD0036.a
lib/libK0000.a
lib/libK0001.a
lib/libL0001.a
lib/libL0019.a
lib/libP0001.a
lib/libP0019.a
lib/libV0001.a
lib/libV0019.a
lib/libL0004.a
lib/libB0004.a
lib/libP0004.a
lib/libV0004.a
lib/libA0047.a
lib/libB0047.a
lib/libD0006.a
lib/libD0015.a
lib/libD0016.a
lib/libD0031.a
lib/libD0046.a
system/98600A2/revlist
system/98600A2/customize

FILE SET: A98600A3

Description: Multi-user DGL/AGP

lib/libD0047.a
lib/libD0048.a
lib/libL0002.a
lib/libL0006.a
lib/libL0031.a
lib/libL0046.a
lib/libP0002.a
lib/libP0006.a
lib/libD0053.a
lib/libD0054.a
lib/libB0056.a
lib/libD0056.a
lib/libL0056.a
lib/libP0056.a
lib/libV0056.a
lib/libd.262x.a
lib/libd.2623.a
lib/libd.2627.a
lib/libd.hpgl.a
lib/libd.hil.a
lib/libd.kbd.a
lib/libd.9836a.a
lib/libd.9836c.a
lib/libd.9837.a
lib/libd.300h.a
lib/libd.3001.a
lib/libd.98700.a

lib/libd.98710.a
lib/libD0042.a
lib/libL0042.a
lib/libP0042.a
lib/libV0042.a
system/98600A3/revlist
system/98600A3/customize

FILE SET: A98600A4

Description: Multi-user DGL/AGP

lib/libDIDD.a
system/98600A4/revlist
system/98600A4/customize

FILE SET: A98683X

Description: DGL/AGP skelton

usr/lib/graphics/skeleton/sk/Axxsk.make
usr/lib/graphics/skeleton/sk/Bxxsk.make
usr/lib/graphics/skeleton/sk/DIDDxx.make
usr/lib/graphics/skeleton/sk/Dxxsk.make
usr/lib/graphics/skeleton/sk/Kxxsk.make
usr/lib/graphics/skeleton/sk/Lxxsk.make
usr/lib/graphics/skeleton/sk/Makefile
usr/lib/graphics/skeleton/sk/Pxxsk.make
usr/lib/graphics/skeleton/sk/Vxxsk.make
usr/lib/graphics/skeleton/sk/axxsk.f
usr/lib/graphics/skeleton/sk/bxxsk.f
usr/lib/graphics/skeleton/sk/dxxsk.f
usr/lib/graphics/skeleton/sk/kxxsk.f
usr/lib/graphics/skeleton/sk/lxxsk.f
usr/lib/graphics/skeleton/sk/m0cop.i
usr/lib/graphics/skeleton/sk/m0iot.i
usr/lib/graphics/skeleton/sk/midxx.f
usr/lib/graphics/skeleton/sk/msred.f
usr/lib/graphics/skeleton/sk/mxyxx.f
usr/lib/graphics/skeleton/sk/pxxsk.f
usr/lib/graphics/skeleton/sk/tbege.f
usr/lib/graphics/skeleton/sk/tcmap.f
usr/lib/graphics/skeleton/sk/techo.f
usr/lib/graphics/skeleton/sk/tedrw.f
usr/lib/graphics/skeleton/sk/tende.f
usr/lib/graphics/skeleton/sk/thclp.f
usr/lib/graphics/skeleton/sk/tictb.f
usr/lib/graphics/skeleton/sk/vxxsk.f
usr/lib/graphics/skeleton/sk/z0acd.f
usr/lib/graphics/skeleton/sk/z0acd.i
usr/lib/graphics/skeleton/sk/z0adv.f
usr/lib/graphics/skeleton/sk/z0adv.i
usr/lib/graphics/skeleton/sk/z0ain.f
usr/lib/graphics/skeleton/sk/z0ain.i
usr/lib/graphics/skeleton/sk/z0bcd.f
usr/lib/graphics/skeleton/sk/z0bcd.i
usr/lib/graphics/skeleton/sk/z0bdv.f
usr/lib/graphics/skeleton/sk/z0bdv.i
usr/lib/graphics/skeleton/sk/z0bfi.i
usr/lib/graphics/skeleton/sk/z0bin.f
usr/lib/graphics/skeleton/sk/z0bin.i
usr/lib/graphics/skeleton/sk/z0buf.i
usr/lib/graphics/skeleton/sk/z0cat.i
usr/lib/graphics/skeleton/sk/z0con.i
usr/lib/graphics/skeleton/sk/z0cor.i
usr/lib/graphics/skeleton/sk/z0cpa.i
usr/lib/graphics/skeleton/sk/z0ctb.f
usr/lib/graphics/skeleton/sk/z0ctb.i
usr/lib/graphics/skeleton/sk/z0dcd.f
usr/lib/graphics/skeleton/sk/z0dcd.i
usr/lib/graphics/skeleton/sk/z0dct.f
usr/lib/graphics/skeleton/sk/z0dct.i
usr/lib/graphics/skeleton/sk/z0ddv.f
usr/lib/graphics/skeleton/sk/z0ddv.i
usr/lib/graphics/skeleton/sk/z0din.f
usr/lib/graphics/skeleton/sk/z0din.i
usr/lib/graphics/skeleton/sk/z0dlm.f
usr/lib/graphics/skeleton/sk/z0dlm.i
usr/lib/graphics/skeleton/sk/z0ext.f
usr/lib/graphics/skeleton/sk/z0ext.i
usr/lib/graphics/skeleton/sk/z0ixx.f
usr/lib/graphics/skeleton/sk/z0ixx.i
usr/lib/graphics/skeleton/sk/z0kcd.f
usr/lib/graphics/skeleton/sk/z0kcd.i
usr/lib/graphics/skeleton/sk/z0kdv.f
usr/lib/graphics/skeleton/sk/z0kdv.i
usr/lib/graphics/skeleton/sk/z0kin.f
usr/lib/graphics/skeleton/sk/z0kin.i
usr/lib/graphics/skeleton/sk/z0lcd.f
usr/lib/graphics/skeleton/sk/z0lcd.i
usr/lib/graphics/skeleton/sk/z0ldv.f
usr/lib/graphics/skeleton/sk/z0ldv.i
usr/lib/graphics/skeleton/sk/z0lin.f
usr/lib/graphics/skeleton/sk/z0lin.i
usr/lib/graphics/skeleton/sk/z0llm.f
usr/lib/graphics/skeleton/sk/z0llm.i
usr/lib/graphics/skeleton/sk/z0nat.f
usr/lib/graphics/skeleton/sk/z0nat.i
usr/lib/graphics/skeleton/sk/z0nca.f
usr/lib/graphics/skeleton/sk/z0nca.i
usr/lib/graphics/skeleton/sk/z0npa.f
usr/lib/graphics/skeleton/sk/z0npa.i
usr/lib/graphics/skeleton/sk/z0pcd.f
usr/lib/graphics/skeleton/sk/z0pcd.i
usr/lib/graphics/skeleton/sk/z0pdv.f
usr/lib/graphics/skeleton/sk/z0pdv.i
usr/lib/graphics/skeleton/sk/z0pin.f
usr/lib/graphics/skeleton/sk/z0pin.i
usr/lib/graphics/skeleton/sk/z0plm.f
usr/lib/graphics/skeleton/sk/z0plm.i
usr/lib/graphics/skeleton/sk/z0ptb.i

usr/lib/graphics/skeleton/sk/z0sys.i
usr/lib/graphics/skeleton/sk/z0vcd.f
usr/lib/graphics/skeleton/sk/z0vcd.i
usr/lib/graphics/skeleton/sk/z0vdv.f
usr/lib/graphics/skeleton/sk/z0vdv.i
usr/lib/graphics/skeleton/sk/z0vin.f
usr/lib/graphics/skeleton/sk/z0vin.i
usr/lib/graphics/skeleton/sk/z0wrk.i
usr/lib/graphics/skeleton/sk/zaend.f
usr/lib/graphics/skeleton/sk/zaint.f
usr/lib/graphics/skeleton/sk/zalph.f
usr/lib/graphics/skeleton/sk/zbend.f
usr/lib/graphics/skeleton/sk/zbint.f
usr/lib/graphics/skeleton/sk/zbutn.f
usr/lib/graphics/skeleton/sk/zcolr.f
usr/lib/graphics/skeleton/sk/zcsiz.f
usr/lib/graphics/skeleton/sk/zdcol.f
usr/lib/graphics/skeleton/sk/zdend.f
usr/lib/graphics/skeleton/sk/zdint.f
usr/lib/graphics/skeleton/sk/zdraw.f
usr/lib/graphics/skeleton/sk/zhigh.f
usr/lib/graphics/skeleton/sk/ziacs.f
usr/lib/graphics/skeleton/sk/zicol.f
usr/lib/graphics/skeleton/sk/ziesc.f
usr/lib/graphics/skeleton/sk/zkend.f
usr/lib/graphics/skeleton/sk/zkint.f
usr/lib/graphics/skeleton/sk/zkybd.f
usr/lib/graphics/skeleton/sk/zlend.f
usr/lib/graphics/skeleton/sk/zlint.f
usr/lib/graphics/skeleton/sk/zlstl.f
usr/lib/graphics/skeleton/sk/zlwid.f
usr/lib/graphics/skeleton/sk/zmark.f
usr/lib/graphics/skeleton/sk/zmove.f
usr/lib/graphics/skeleton/sk/znewf.f
usr/lib/graphics/skeleton/sk/zoesc.f
usr/lib/graphics/skeleton/sk/zpend.f
usr/lib/graphics/skeleton/sk/zpgdd.f
usr/lib/graphics/skeleton/sk/zpick.f
usr/lib/graphics/skeleton/sk/zpint.f
usr/lib/graphics/skeleton/sk/zpoly.f
usr/lib/graphics/skeleton/sk/zsloc.f
usr/lib/graphics/skeleton/sk/zsval.f
usr/lib/graphics/skeleton/sk/ztext.f
usr/lib/graphics/skeleton/sk/zvend.f
usr/lib/graphics/skeleton/sk/zvint.f
usr/lib/graphics/skeleton/sk/zwloc.f
usr/lib/graphics/skeleton/sk/zwval.f
usr/lib/graphics/skeleton/sk/tdpat.o
usr/lib/graphics/skeleton/sk/tdred.o
usr/lib/graphics/skeleton/sk/tfill.o
usr/lib/graphics/skeleton/sk/tpgcp.o
usr/lib/graphics/skeleton/sk/zidrw.o
usr/lib/graphics/skeleton/sk/zimov.o
usr/lib/graphics/skeleton/sk/zipgd.o
usr/lib/graphics/skeleton/sk/zipgi.o
usr/lib/graphics/skeleton/sk/ziply.o
usr/lib/graphics/skeleton/sk/zpgdi.o
usr/lib/graphics/skeleton/README
usr/lib/graphics/skeleton/dev.group
usr/lib/graphics/skeleton/files
usr/lib/graphics/skeleton/manual
system/98683X/revlist

Notes

System Parameters

D

If you need to customize your HP-UX kernel, you can use these tunable parameters with the *config* command. *config* is described in the "Toolbox" chapter, the section called "Configuring HP-UX".

NOTE

You can adversely affect the operation of HP-UX by changing these parameters. BE CAREFUL, and make sure you know all the implications of using the parameters before you configure your system.

---

acctresume

Name

acctresume - resume accounting due to disk usage

Range

-100 -> 101

Default

4

Use

The system automatically disables process accounting when the available space on the file system where the accounting file resides falls below a certain threshold. The threshold is described under *acctsuspend*. The system also automatically re-enables process accounting when sufficient space becomes available. The parameter, *acctresume*, is the threshold (percentage of free space) which the system must have to re-enable process accounting. This percentage is added to minimum free percentage (minfree) for the file system.

A value of zero re-enables accounting when the free space reaches minfree. A value less than zero allows process accounting to use the space which is reserved for super-user use. A value greater than 100 prevents process accounting from ever being re-enabled for because of available space.

When accounting is re-enabled in this way, the message

```
Accounting resumed
```

is printed on the system console.

acctresume is only relevant to systems which turn on process accounting.

Cost:

None

Dependencies (interactions with other system values)

acctsuspend < acctresume

acctsuspend

Name
acctsuspend - suspend accounting due to disk usage

Range
-100 -> 100

Default
2

Use
The system automatically disables process accounting when the available space on the file system where the accounting file resides falls below a certain threshold. The parameter, *acctsuspend* (specified as a percentage of free space), is the threshold. This percentage is added to minimum free percentage (minfree) for the file system.

A value of zero disables accounting when the free space falls below minfree. A value less than zero allows process accounting to use the space which is reserved for super-user use. If the sum of acctsuspend and minfree is less than zero, process accounting can never be disabled for this reason.

When accounting is disabled in this way, the message

```
Accounting suspended
```

is printed on the system console.

acctsuspend is only relevant to systems which turn on process accounting.

Cost
None

Dependencies (interactions with other system values)
acctsuspend < acctresume

argdevnblk

Name

argdevnblk—limit the size of the area reserved for argdev

Range

0, 48 disc blocks -> 1024 disc blocks (in multiples of 12)
The series 200/300 have 1 Kbyte disc blocks.

Default

0 (configured dynamically)

Use

exec uses *argdevnblk* as a scratch pad. The bigger the *argdevnblk*, the greater the number of concurrent *execs* allowed. Each *exec* call takes approximately 12 disc blocks.

Cost

Since *argdev* occupies the first part of the swap device, the greater the *argdevnblk*, the smaller the swap space actually available to the user's process.

dst

Name
dst—daylight savings time

Range
0 -> 5

Default
1

Use
This variable defines the daylight savings time correction.

```
( from file ../h/time.h )

#define DST_NONE      0     /* not on dst */
#define DST_USA       1     /* USA style dst */
#define DST_AUST      2     /* Australian style dst */
#define DST_WET       3     /* Western European dst */
#define DST_MET       4     /* Middle European dst */
#define DST_EET       5     /* Eastern European dst */
```

Cost
None.

Dependencies (interactions with other system values)
It is used in conjunction with time zone.

maxdsiz

Name

maxdsiz - maximum data size

Range

256 Kbytes -> 20 Mbytes[1]

Default

0x01000000 (16 Megabytes)

Use

This value is entered in bytes.

maxdsiz defines the maximum size that the data segment of an executing process can have.

The default value is large enough for the data used by most processes. *maxdsiz* should only be increased if you know that you have one or more processes which use huge amounts of data.

Each time the system loads a process, or an executing process attempts to expand its data segment, the system checks the size of the process's data segment.

If the *maxdsiz* is exceeded, the process will be terminated or returned with an appropriate error.

Cost

None.

Dependencies (interactions with other system values)

process text + process data + process stack <= 16 MB (Model 310 and Series 200)

[1] To increase beyond 20 Mbytes, see Table D.1 at the end of this Appendix.

maxssiz

Name

maxssiz - maximum stack size

Range

256 Kbytes -> 20 Mbytes[1]

Default

0x00200000 (2 Megabytes)

Use

This value is entered in bytes. *maxssiz* defines the maximum size that the stack segment of an executing process can have.

The default is large enough for the stack of most processes. *maxsiz* should only be increased if you have one or more processes which require a huge stack.

The stack grows dynamically. As the stack grows, the system checks the size of the process's stack segment. If the *maxssiz* is exceeded the process will be killed.

Cost

none.

Dependencies (interactions with other system values)

process text + process data + process stack <= 16 MB (Model 310 and Series 200)

[1] To increase beyond 20 Mbytes, see table D.1 at the end of this appendix.

maxtsiz

Name
maxtsiz - maximum text size

Range
256 Kbytes -> 24 Mbytes[1]

Default
0x01000000 (16 Megabytes)

Use
This value is entered in bytes.

maxtsiz defines the maximum size that the shared text segment of an executing process can have.

The current default will accomodate the text segments of most processes. Unless you run a processwith a text segment larger than 16 Megabytes, *maxtsiz* should not be modified.

Cost
none.

Dependencies (interactions with other system values)
Each time the system loads a process with shared text, the system checks the size of its shared text segment. *exec* issues an error message and aborts the process if the process's text segment is larger than *maxtsiz*.

process text + process data + process stack <= 16 MB (Model 310 or Series 200)

[1] To increase beyond 24 Mbytes, see Table D.1 at the end of this appendix.

maxuprc

Name

maxuprc - maximum number of user processes

Range

3 -> (procs - 3)

Default

25 processes

Use

Maxuprc defines the maximum number of simultaneous processes that a user can have. A user is identified by the UID (user ID) number, not by the number of login instances. Each user will need at least one process for the shell, plus an adequate number to be able to do useful work (25 is usually more than enough).

The super user is exempt from this limit.

Pipelines need at least one simultaneous process for each side of a '|'. Some commands, such as *cc*, *fc*, and *pc*, use more than one process per invocation.

When the total number of processes for a given user is larger than *maxuprc*, the following message is printed to the user tty:

`no more processes`	if from shell, or
`EAGAIN error`	if a fork call from a program.

Cost

none

Dependencies (interactions with other system values)

If *maxuprc* is set to a value greater than or equal to *nproc* (maximum number of processes in the system) then *maxuprc* is no longer a limit, and the system can be hoarded by a single user.

maxusers

Name

maxusers - limiter for system resource allocation

Range

0 - 8

Default

8

Use

maxusers is used as a limiter for system resource allocation. By itself, *maxusers* does not determine the size of any structures in the system. The value of other global system parameters depend on *maxusers*. If you tune the parameters that use *maxusers*, then the effect of *maxusers* on kernel size is proportionately smaller.

It is NOT used as a limit for the number of users in the system, therefore its name (inherited from System V) is very confusing.

Cost:

Size of cblock structure is 32 bytes
(the system allocates nclist(NCLIST) # of cblock structures)

Size of proc structure is 164 bytes
(the system allocates nproc(NPROC) # of proc structures)

Size of callout structure is 16 bytes
(the system allocates ncallout(NCALLOUT) # of callout structures)

Size of inode structure is 184 bytes
(the system allocates ninode(NINODE) # of inode structures)

Size of file structure is 26 bytes
(the system allocates nfile(NFILE) # of file structures)

Size of text structure is 76 bytes
(the system allocates ntext(NTEXT) # of text structures)

Dependencies (interactions with other system values)

```
#define     NETSLOP      0
#define     NCLIST       (100+16*MAXUSERS)
#define     NPROC        (20+8*MAXUSERS)
#define     NCALLOUT     (16+NPROC)
#define     NINODE       ((NPROC+16+MAXUSERS)+32)
#define     NFILE        (16*(NPROC+16+MAXUSERS)/10+32+2*NETSLOP)
#define     NTEXT        (24+MAXUSERS+NETSLOP)
```

mesg

Name
mesg - System V messages

Range
0 -> 1

Default
1

Use
mesg determines whether the code for System V IPC messages is included in the kernel. If *mesg*=1, the code is included; if *mesg*=0, then the code is not included. If *mesg*=0, all programs using the system calls (*msgget(2), msgbp(2),* and *msgctl(2)*) will receive a SIGSYS signal.

Cost
The size of the message code is 2 680 bytes.

Dependencies (interactions with other system values)
All message parameters depend on mesg.

If mesg = 0, then the message parameters are not declared and no message structures are allocated.
If mesg = 1, then the message parameters are declared and can can be tuned and all other message structures are allocated.

msgmap

Name
msgmap - message map

Range
3 -> memory limited

Default
(msgtql+2)

Use
Each set of messages allocated per identifier occupies 1 or more contiguous slots in the msg array. As messages are allocated and deallocated the msg array may become fragmented.

msgmap dimensions the resource map used to allocate the buffer space for messages. This map shows the free holes in the msg array. An entry in the map is used to point to each set of contiguous unallocated slots, and it consists of a pointer to the set, plus the size of the set.

If message usage is heavy, and a request for a message set cannot be accomodated, the message:

```
danger:  mfree map overflow
```

will appear. If you get the error, you should make a new kernel with a larger msgmap.

There will be less fragmentation of the msg array if all message identifiers have the same number of messages. *msgmap* can then be smaller.

3 is the lower limit: 1 slot is overhead for the map and the second slot is always needed at system initialization to indicate that the msg array is completely free.

Cost
4 bytes * msgmap.

Dependencies (interactions with other system values)
msgmap <= (msgtql+2)

msgmap <= (msgseg+2)

If *mesg* = 0, then the message code is not included in the kernel and the value of *msgmap* is irrelevant.
If *mesg* = 1, then the message code is included and *msgmap* can then be tuned.

msgmax

Name
msgmax - message maximum size

Range
0 -> memory limited

Default
8192 bytes

Use
msgmax limits the size, in bytes, of a single message.

It should be increased only if you have applications which require larger messages. Its main value is to keep malicious or poorly written programs from using all the message buffer space.

A *msgsnd* system call which attempts to send a message larger than *msgmax* bytes returns the EINVAL error.

Cost
none

Dependencies (interactions with other system values)
msgmax <= msgmnb

If *mesg* = 0, then the message code is not included in the kernel and the value of *msgmax* is irrelevant.
If *mesg* = 1, then the message code is included and *msgmax* can then be tuned.

msgmnb

Name
msgmnb - maximum number of bytes on message queue

Range
0 -> memory limited

Default
16,384 bytes

Use
msgmnb is the maximum total size, in bytes, of all messages that can be queued on a message queue at the same time.

A *msgsnd* system call which attempts to exceed this limit either:

- blocks or,
- returns EAGAIN error

depending on whether the IPC_NOWAIT flag is set with the call.

Cost
none

Dependencies (interactions with other system values)
msgmnb >= msgmax

msgmnb <= (msgssz×msgseg)

If *mesg* = 0, then the message code is not included in the kernel and the value of *msgmnb* is irrelevant.
If *mesg* = 1, then the message code is included and *msgmnb* can then be tuned.

msgmni

Name

msgmni - number of message queue identifiers

Range

1 -> memory limited

Default

50

Use

msgmni dimensions the array of message queue identifiers.

A message queue identifier is needed for each message queue in the system.

An attempt to allocate a new message queue with the *msgget* system call when there already exist *msgmni* message queues returns a ENOSPC error.

If a message queue is not deallocated, it will stay around in the system even after the process(es) using it have died.

All users of messages should deallocate them when through with them, as running against the msgmni limit usually means that users have not freed them up.

Cost

46 bytes * msgmni.

Dependencies (interactions with other system values)

If *mesg* = 0, then the message code is not included in the kernel and the value of *msgmni* is irrelevant.
If *mesg* = 1, then the message code is included and *msgmni* can then be tuned.

msgseg

Name

msgseg - message segments

Range

1 -> memory limited

Default

16,384

Use

msgseg together with *msgssz* determines the size of the buffer available for queueing messages.

msgssz determines the size, in bytes, of the units in which messages are allocated space. When a message is allocated, its size is rounded up to the nearest multiple of *msgssz*.

msgseg is the number of these units available.

In most cases the product of *msgseg* × *msgssz* is of most interest since it determines the total amount of space available for messages. Different *msgseg:msgssz* ratios which yield the same product will just cause this space to be fragmented differently for the same usage.

Cost

msgseg×msgssz bytes

Dependencies (interactions with other system values)

(msgseg×msgssz) >= msgmnb

(msgseg×msgssz) >= msgmax

If *mesg* = 0, then the message code is not included in the kernel and the value of *msgseg* is irrelevant.
If *mesg* = 1, then the message code is included and *msgseg* can then be tuned.

msgssz

Name
msgssz - message segment size

Range
1 -> memory limited

Default
1 page

Use
msgssz together with *msgseg* determines the size of the buffer available for queueing messages.

msgssz determines the size, in bytes, of the units in which messages are allocated space. When a message is allocated, its size is rounded up to the nearest multiple of *msgssz*.

msgseg is the number of these units available.

In most cases, the product of *msgseg* × *msgssz* is of most interest since it determines the total amount of space available for messages. Different *msgseg:msgssz* ratios which yield the same product will just cause this space to be fragmented differently for the same usage.

Cost
msgseg * msgssz bytes

Dependencies (interactions with other system values)
(msgseg×msgssz) >= msgmnb

(msgseg×msgssz) >= msgmax

If *mesg* = 0, then the message code is not included in the kernel and the value of *msgssz* is irrelevant.
If *mesg* = 1, then the message code is included and *msgssz* can then be tuned.

msgtql

Name
msgtql - message number of headers

Range
1 -> memory limited

Default
40

Use
msgtql dimensions an array of message headers. A message header is used for each message queued in the system.

A *msgsnd* system call which attempts to exceed this limit either:
- blocks or,
- returns EAGAIN error

depending on whether the IPC_NOWAIT flag is set with the call.

Cost
12 bytes * msgtql.

Dependencies (interactions with other system values)
msgmap <= msgtql + 2

If *mesg* = 0, then the message code is not included in the kernel and the value of *msgtql* is irrelevant.
If *mesg* = 1, then the message code is included and *msgtql* can then be tuned.

nbuf

Name
nbuf - number of buffer headers

Range
0, 16 -> (≈10% of available memory)÷page size
page size = 4096 on Series 300, 2048 on Series 200.

Default
0 (configured dynamically)

if at boot time nbuf is = 0, then the amount of buffer space is set to 10% of available memory The buffer space is distributed among the buffer headers.

Use
nbuf defines the number of file system buffer-cache buffer headers.

At boot time, the system loads the kernel and then assigns to the system buffer-cache (bufpages) 10% of the available memory, in pages. Then *nbuf* is computed, and these *nbuf* headers will have bufpages of space available to them.

The 10% of available memory is a ceiling. If *nbuf* is greater than this ceiling, the effect is, not to increase the size of the system buffer-cache, but rather to vary the amount of memory assigned to each buffpage. However, *nbuf* can be used to decrease the size of the system buffer-cache.

These buffers are used for all file system I/O operations, plus all other block I/O operations in the system (*exec, mount,* i-node reading, some device drivers, etc).

NOTE

If you set *nbuf* to a number less than 16, the system will not have enough buffers to boot.

Cost
Size of buf structure is 104 bytes.

Dependencies (interactions with other system values)
nswbuf (number os swap buffer headers) = max(256, (*nbuf*÷2))

ncallout

Name
ncallout - number of callouts

Range
6 -> memory limited

Default
(16+NPROC)

Use
ncallout is the maximum number of timeouts which can be scheduled by the kernel at any one time. Timeouts are used by :

- *alarm* (system call)
- *setitimer* (system call)
- drivers
- *uucp* processes
- process scheduling

When the system runs out of callouts, it prints the following fatal error to the console:

```
panic: timeout table overflow
```

Cost
Size of callout structure is 16 bytes

(the system allocates *ncallout*(NCALLOUT) # of callout structures)

Dependencies (interactions with other system values)
The larger procs is, the larger *ncallout* should be.
A rough guideline of 1 callout per process should be used unless one has processes which use lots of callouts.

nfile

Name
nfile - number of files

Range
14 -> memory-limited

Default
(16×(NPROC+16×MAXUSERS)÷10+32+2×NETSLOP)

Use
nfile defines the maximum number of open files at any one time in the system.

It is the number of slots in the file descriptor table. Be generous with this number since the cost is low, and not having enough slots would cut down in the amount of work that can be done simultaneously in the system.

Cost
Size of file structure is 26 bytes.

Dependencies (interactions with other system values)

Windows/9000	3 or 4 file descriptors per window
Processes	At least 3 file desc. per process (stdin, stdout, stderr)
Pipes	2 per pipe (1 per side)

Shell scripts and background processes are treated like any other processes.

NETSLOP = 0

NPROC = (20+8*MAXUSERS)

MAXUSERS = 8

nflock

Name
nflocks - number of file locks

Range
2 -> 2000

Default
200

Use
nflocks gives the possible number of file/record locks in the system. When choosing this number it should be noted that one file may have several locks and that especially databases may need a large number of locks (if they use *lockf* at all).

Cost
Each lock entry costs 20 bytes.

Dependencies
none

ninode

Name
ninode - number of inodes

Range
14 -> memory-limited

Default
((NPROC+16+MAXUSERS)+32)

Use
ninode defines the maximum number of open inodes which can be in-core.

It is the number of slots in the inode table. The inode table is used as a cache memory. For efficiency reasons, the last *ninode* (number of) open inodes is kept in main memory. The table is hashed.

Cost
Size of inode structure is 184 bytes.

Dependencies (interactions with other system values):
Each unique open file has an open inode associated with it, therefore the larger the number of unique open files, the larger *ninode* should be.

MAXUSERS=8

NPROC=(20+8×MAXUSERS)

nproc

Name

nproc - number of processes

Range

6 -> memory limited

Default

(20+8×MAXUSERS)

Use

nproc specifies the maximum total number of processes that can exist simultaneously in the system.

There have to be at least 3 system overhead processes at all times (*swapper*, *init*, and *pageout* demon), and there is always one entry reserved for the super user.

Other features and the number of processes are:

LAN network	3 processes
Printer scheduler	1 process
Cron	1 process

When the total number of processes in the system is larger than *nproc*, the following messages are printed:

to the console:

```
proc: table is full
```

to the tty of the user who tries to start the last process(es):

`no more processes`	if from shell, or
`EAGAIN error`	if a fork call from a program.

Cost

Size of proc structure is 164 bytes

Dependencies (interactions with other system values)

ninode	NINODE	((NPROC+16+MAXUSERS)+32)
nfile	NFILE	(16*(NPROC+16+MAXUSERS)/10+32+2*NETSLOP)
ncallout	NCALLOUT	(16+NPROC)

maxuprc <= (nproc - 4)

npty

Name

npty - number of pseudo-teletypes

Range

1 -> 82

Default

82

Use

npty limits the number of the following structures can be used by the pseudo-teletype driver:

```
struct  tty       pt_tty[npty];
struct  tty       *pt_line[npty];
struct  pty_info  pty_info[npty];
```

npty is used primarily by the window subsystem. If windows are not used, or the pty driver is not explicitly used, no pty structures are necessary.

Cost

It costs 450 bytes per pty.

The size of of the tty structure is 260 bytes.
The size of of the pty_info structure is 186 bytes.

Dependencies (interactions with other system values)

None.

ntext

Name
ntext - number of texts

Range
10 -> 32

Default
(24+MAXUSERS)

Use
ntext defines the maximum number of shared text (or code) descriptors (data structures describing the text) which can be active at any one time. Note that this does not limit the number of processes sharing the same shared text program.

Attempting to start a new process that may require a new text descriptor (when no more text descriptors are available) will get "text table is full" message on the console and the new process will be killed.

Cost
The size of the data structure describing a shared data object is about 80 bytes.

Dependencies (interactions with other system values)
If there are no more text descriptors when starting the execution of a process (at load time), the console gets a message "text table is full", and the process will be killed.

parity_option

Name
parity_option - used to handle parity errors

Range
0 -> 2

Default
1

Use
parity_option selects the kind of action that the system takes if it encounters a parity error.

The actions are as follows:

0 Print a 'Parity error' message to the console.

1 Print a 'Parity error' message to console, plus:

- if user state it kills the current process (which may not always be the process which caused the error, as with a DMA card) and prints an error message to its tty.
- if supervisor state it panics with a 'parity error' message to the console.

2 Always panics with a 'parity error' message to console.

Cost
4 bytes.

Dependencies (interactions with other system values):
None.

sema

Name

sema - System V semaphores

Range

0 -> 1

Default

1

Use

sema determines whether the code for System V IPC semaphores will be included in the kernel.

If *sema* = 1, the code is included;
if *sema* = 0, then the code is not included.

HP Windows/9000 and the Starbase graphics library both require the semaphore code.

If *sema*=0, and you have programs which use the *semget(2)*, *semop(2)*, or *semop(2)* system calls, you will receive a SIGSYS signal.

Cost

The size of the semaphore code is 4 240 bytes.

Dependencies (interactions with other system values)

All semaphore parameters depend on the value of *sema*.

If *sema* = 0, then the semaphore parameters are not declared and no semaphore structures allocated.
If *sema* = 1, then the semaphore parameters are declared and can be tuned and the semaphore structures allocated.

If *sema* = 0, then HP Windows/9000 and Starbase graphics will be unable to run.

semaem

Name
semaem - "adjust on exit" maximum value for semaphores

Range
0 -> min (semvmx, 32767)

Default
16,384

Use
An undo is a special, optional, flag in a semaphore operation which causes that operation to be undone if the process which invoked it dies.

semaem is the maximum value by which a semaphore can be undone.

This value is cumulative per process, so if one process has more than one undo operations on a semaphore, the value of each undo operation is added up in the variable *semadj*. *semadj* is the number by which the semaphore will be incremented or decremented if the process dies.

Read the manual page for *semop(2)* for more detailed information on semaphore undos.

Any *semop* calls which attempt to set $|semadj| > semaem$ result in an ERANGE error.

Cost
none

Dependencies (interactions with other system values)
$|semaem| <= semvmx$

If *sema* = 0, then the semaphore parameters are not declared and no semaphore structures allocated.
If *sema* = 1, then the semaphore parameters are declared and can be tuned and the semaphore structures allocated.

semmap

Name

semmap - semaphore map

Range

4 -> memory limited

Default

(SEMMNI+2)

Use

Each set of semaphores allocated per identifier occupies 1 or more contiguous slots in the sem array. As semaphores are allocated and deallocated the sem array may become fragmented.

semmap dimensions the resource map which shows the free holes in the sem array. An entry in this map is used to point to each set of contiguous unallocated slots, and it consists of a pointer to the set, plus the size of the set.

If semaphore usage is heavy, and a request for a semaphore set cannot be accomodated, the message:

```
|Mdanger: mfree map overflow
```

will appear. It will then be helpful to make a new kernel with a larger *semmap*.

Fragmentation of the sem array is reduced if in usage of the system all semaphore identifiers have the same number of semaphores. *semmap* can then be somewhat smaller.

4 is the lower limit: 1 slot is overhead for the map and the second slot is always needed at system initialization to indicate that the sem array is completely free.

Cost

4 bytes * semmap

Dependencies (interactions with other system values)

(*semmap*-2) = maximum # of contiguous unallocated pieces of the sem array.

semmap <= (*semmni*-2)

If *sema* = 0, then the semaphore code is not included in the kernel and the value of *semmap* is irrelevant.
If *sema* = 1, then the semaphore code is included and *semmap* can then be tuned.

semmni

Name

semmni - semaphore number of identifiers

Range

2 -> memory limited

Default

64

Use

semmni defines the number of sets (identifiers) of semaphores available to the users.

When the system runs out of semaphore sets, the *semget* system call will return a ENOSPC error message.

Cost

30 bytes×*semmni*

Dependencies (interactions with other system values)

semmni <= *semmns*

SEMMSL = 25 /* maximum number of semaphores per id */

semmns <= (*semmni*×*semmsl*)

semmap <= (*semmni*+2)

If *sema* = 0, then the semaphore code is not included in the kernel and the value of *semmni* is irrelevant.
If *sema* = 1, then the semaphore code is included and *semmni* can then be tuned.

semmns

Name

semmns - semaphore number in system

Range

2 -> memory limited

Default

128

Use

semmns defines the total number of semaphores available to the users of the system.

When the system does not have enough contiguous semaphores in the sem array to satisfy a *semget* request, the call returns a ENOSPC error. This error may occur even though there may be enough free semaphores, but they are not contiguous.

Cost

8 bytes×*semmns*

Dependencies (interactions with other system values)

semmni <= *semmns*

SEMMSL = 25   /* maximum number of semaphores per id */

semmns <= (*semmni*×*semmsl*)

If *sema* = 0, the semaphore code is not included in the kernel and the value of *semmns* is irrelevant.
If *sema* =1, the semaphore code is included and then *semmns* can be tuned.

semmnu

Name

semmnu - semaphore number of undo structures

Range

1 -> (procs - 3)

Default

30

Use

An undo is a special, optional, flag in a semaphore operation which causes that operation to be undone if the process which invoked it dies.

semmnu is the number of processes which can have undo's pending on a given semaphore. It determines the size of the sem_undo structure.

Read the manual page for *semop(2)* for a more detailed explanation of undos.

You should increase *semume* if the user gets ENOSPC errors on *semop* calls using the SEM_UNDO flag.

Cost

(6+(8×*semume*)) bytes×*semmnu*

Dependencies (interactions with other system values)

- *semmnu* determines the size of the structure sem_undo, which in turn contains the sub-structure dimensioned by *semume*.
- There is no point in having *semmnu* > (nproc-3) since it is the largest number of processes in the system which could use semaphores simultaneously.

If *sema* = 0, then the semaphore parameters are not declared and no semaphore structures allocated.
If *sema* = 1, then the semaphore parameters are declared and can be tuned and the semaphore structures allocated.

semume

Name
semume - semaphore undo entries per process

Range
1 -> semmns

Default
10

Use
An undo is a special, optional, flag in a semaphore operation which causes that operation to be undone if the process which invoked it dies.

semume limits the number of semaphores that each process can have undos pending on.

Read the manual page for *semop(2)* for a more detailed explanation of undos.

When the user gets EINVAL errors on *semop* calls with the SEM_UNDO flag, then you should increase the value of *semume*.

Cost
(6+(8×*semume*)) bytes×*semmnu*

Dependencies (interactions with other system values)
semume <= *semmns*

semume is the size of the substructure undo, which is part of the sem_undo structure. The size of sem_undo is determined by *semume*.

If *sema* = 0, then the semaphore parameters are not declared and no semaphore structures allocated.
If *sema* = 1, then the semaphore parameters are declared and can be tuned and the semaphore structures allocated.

semvmx

Name
semvmx - semaphore maximum value

Range
1 -> 65,535

Default
32,767

Use
semvmx is the maximum value that a semaphore is allowed to reach. This limit is imposed by the largest number that can be stored in a 16-bit unsigned integer (65,535).

semop system calls which try to increment a semaphores value > *semvmx* result in ERANGE errors. If semvmx > 65,535 then semaphore values ca overflow and these errors will not be caught.

Cost
none

Dependencies (interactions with other system values):
semaem <= *semvmx*

If *sema* = 0, then the semaphore parameters are not declared and no semaphore structures allocated.
If *sema* = 1, then the semaphore parameters are declared and can be tuned and the semaphore structures allocated.

shmall

Name

shmall - total size of all shared memory segments

shmall is present just for compatibility reasons with System V.

shmbrk

Name
shmbrk - shared memory break

Range
0 -> maximum number of pages in the data segment.

Default
16 pages

Use
shmbrk defines the size of the gap, in pages, between the top of the current data segment and the first default shared memory address.

Cost
none

Dependencies (interactions with other system values)
If *shmem* = 0, then the code for shared memory is not included in the kernel and *shmbrk* has no effect.
If *shmem* = 1, then the code for shared memory is included in the kernel and *shmbrk* is tunable.

shmem

Name

shmem - System V shared memory enable/disable

Range

0, 1

Default

1

Use

shmem determines whether the code for System V IPC shared memory is included in the kernel. If *shmem* = 1 then the code is included, if *shmem* = 0 it is not.

Cost

The size of the *shmem* code is about 11 Kbytes.

Dependencies (interactions with other system values)

HP Windows/9000 and the Starbase graphics library both require the presence of shared memory code.

shmmax

Name
shmmax - shared memory maximum

Range
2 Mbytes -> 20 Mbytes[1]

Default
0x00400000 (4 Mbytes)

Use
shmmax defines the maximum shared memory segment size.

Cost
none

Dependencies (interactions with other system values)
shmmin <= *shmmax*

If *shmem* = 0, then the code for shared memory is not included in the kernel and the value of *shmmax* is irrelevant.
If *shmem* = 1, then the code for shared memory is included and *shmmax* is tunable.

For Windows/9000 to run, *shmmax* must be >= 2 Mbytes.

[1] To increase beyond 20 Mbytes, see Table D.1 at the end of this appendix.

shmmaxaddr

Name
shmmaxaddr - shared memory maximum address

Range
positive integer

Default
0x01000000 (16 Mbytes)

Use
shmmaxaddr specifies the highest address allowed for a shared memory segment in user address space.

Cost
none

Dependencies (interactions with other system values)
If *shmem* = 0, then the code for shared memory is not included in the kernel and *shmmaxaddr* has no effect.
If *shmem* = 1, then the code for shared memory is included and *shmmaxaddr* is tunable.

If you allow a shared memory segment to attach at a high address location, you will require more system overhead for the translation table data structure. **Do not reconfigure this unless absolutely necessary**.

shmmin

Name
shmmin - shared memory minimum

Range
all positive integers

Default
1 byte

Use
shmmin defines the minimum shared memory segment size.

Cost
none

Dependencies (interactions with other system values)
shmmin < *shmmax*

If *shmem* = 0, then the code for shared memory is not included in the kernel and the value of *shmmin* is irrelevant.
If *shmem* = 1, then the code for shared memory is included and *shmmin* is tunable.

If it is reconfigured other subsystems (such as Windows/9000) may not work.

shmmni

Name
shmmni - shared memory maximum number of identifiers

Range
positive integers

Default
30 identifiers

Use
shmmni defines the maximum number of shared memory segments systemwide.

shmmni should be large enough to hold as many shared memory segments as will be used simultaneously.

Cost
The data structure associated at each shared memory segment is about 100 bytes.

Dependencies (interactions with other system values)
If *shmem* = 0, then the code for shared memory is not included in the kernel and *shmmni* has no effect.
If *shmem* = 1, then the code for shared memory is included and *shmmni* can be tuned.

Windows/9000 and Starbase graphics require *shmmni* to be >= 4.

shmseg

Name
shmseg - shared memory segments

Range
all positive integers

Default
10

Use
shmseg defines the maximum number of shared memory segments that can be attached to a process at any given time.

Cost
about 12 bytes × *nproc* × *shmseg*

Dependencies (interactions with other system values)
If *shmem* = 0, then the code for shared memory is not included in the kernel and *shmseg* has no effect.
If *shmem* = 1, then the code for shared memory is included and *shmseg* can be tuned.

Windows/9000 and Starbase graphics require that *smseg* be >= 4.

timeslice

Name
timeslice - scheduling timeslice interval

Range
-1 -> 2147483647

Default
0

Use
The system performs round-robin scheduling among processes at a given priority. The timeslice interval is the amount of time one process is allowed to run before the CPU is given to the next process at the same priority.

The value of timeslice is specified in units of 20 millisecond clock ticks. There are two special values:

0	use the system default value (currently 5 ticks, or 100 milliseconds)
-1	disable round-robin scheduling completely

One side-effect of this parameter is that a process always checks for pending signals when its timeslice expires. This guarantees that a process which does not make any system calls (including a runaway process in an infinite loop) can be terminated. Thus setting timeslice to a very large value, or to -1, can prevent such processes from seeing signals.

It is anticipated that this parameter will only be changed on systems dedicated to applications with specific real-time needs.

Cost
There is no memory allocation related to this parameter. There is some amount of CPU time spent at each timeslice interval, but this time has not been precisely measured.

Dependencies (interactions with other system values)
None.

timezone

Name
timezone - minutes west of Greenwich

Range
0 -> no real upper limit

Default
420

Use
Timezone information, indicates the minutes west of Greenwich.

```
struct  timezone tz = { TIMEZONE, DST };
struct timezone {
int tz_minuteswest; /* minutes west of Greenwich */
int tz_dsttime;     /* type of dst correction */
};
#define DST_NONE      0     /* not on dst */
#define DST_USA       1     /* USA style dst */
#define DST_AUST      2     /* Australian style dst */
#define DST_WET       3     /* Western European dst */
#define DST_MET       4     /* Middle European dst */
#define DST_EET       5     /* Eastern European dst */
```

Cost
None.

Dependencies (interactions with other system values)
It is used in conjunction with dst (daylight savings time).

unlockable_mem

Name

unlockable_mem - unlockable memory

Range

0 -> (the available memory indicated at power-up)

Default

102,400 bytes

Use

unlockable_mem defines the minimum amount of memory which is guaranteed to be available for virtual memory and/or system overhead at any one time.

It limits the amount of memory which can be locked (lockable memory) to (the available memory indicated at powerup - *unlockabl_mem*).

If *unlockable_mem* is greater than available memory, then the system will set *unlockable_mem* ot zero.

Locakble memory is used for:

- process images and overhead locked with *plock(2)*
- shared memory segments locked with the SHM_LOC command of the *shmctl(2)* system call
- kernel memory allocated for LAN by the *npowerup* command
- miscellaneous dynamic kernel data structures used by the shared memory system and some drivers.

Any call which needs to use lockable memory may fail if its value is too small. Note that lockable memory is a limit on the amount of memory which can be locked, but that this memory is available for virtual memory except when it is locked.

Cost

none

Dependencies (interactions with other system values)

unlockable_mem <= physical memory

If you change *maxdsiz, maxtsiz, maxssiz,* or *shmmax*, you must also change *dmmin, dmmax, dmshm*, and *dmtext*. These parameters interact, and a wrong value might make your virtual memory system unworkable.

Also, as process size increases, the total amount of non-pageable physical memory needed increases. This is because page tables are non-pageable. For example, on a Model 320, a 4096 Mbyte process requires 4 Mbytes of main memory to hold its page table.

Table D.1: Virtual Memory Parameter Table

if maxdsiz, maxtsiz, maxssiz, or shmmax is (in bytes):	**dmmin should be:**	**dmmax should be:**	**dmshm should be:**	**dmtext should be:**
	(in 1KB disc blocks)			
256KB < X <= 16MB[1]	16	2048	2048	2048
16MB < X <= 20MB[2]	16	2048	2048	2048
20MB < X <= 40MB	32	4096	4096	4096
40MB < X <= 80MB	64	8192	8192	8192
80MB < X <= 160MB	128	16384	16384	16384
160MB < X <= 320MB	256	32768	32768	32768
320MB < X <= 640MB	512	65536	65536	65536
640MB < X <= 1280MB	1024	131072	131072	131072
1280MB < X <= 2560MB	2048	262144	262144	262144
2560MB < X <= 4096MB	2048	524288	524288	524288

1 On the Series 300, Model 310, and the Series 200 computers, the process data + process stack + process text <= 16 Mbytes.
2 default setting.

Notes

Glossary

The following terms and definitions are defined as they apply to the Series 200/300 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.

block
The primary logical unit of information in the HFS file system. 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. Blocks can be divided into 1 Kbyte, 2 Kbyte, or 4 Kbyte fragments.

block mode
Buffered I/O: data is 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 block mode is defined as BLKDEV_IOSIZE in */usr/include/sys/param.h*.

boot or boot-up
The process of loading, initializing and running an operating system.

boot area
The first 8 Kbytes of the disc 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 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.

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.

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.

cylinder
One or more vertical collections of tracks in a disc pack (see Figure G.1).

cylinder group
One or more consecutive cylinders (see Figure G.1). 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.

destination device
The mass storage device on which HP-UX is to be installed. The destination device must be a hard disc drive.

driver number
A pointer to the part of the kernel needed to use the device. The driver number, for a particular device, can be found in the tables in "Adding/Moving Peripheral Devices" in the "Toolbox" chapter.

disc
The term used for a collection of recording platters contained in a single disc drive. Disc is synonomous with disc pack.

/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(8)* 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(8)* for details on this shell.

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.

file system
The organization of files and directories, associated with a block special file, on a hard disc. 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 disc space allowed. Once the file system capacity reaches this threshold, only the super-user is allowed to allocate disc blocks. The default is 10%; if it is less, file system performance degrades.

HFS file system
High performance File System. This is the file system implemented on your Series 200/300 HP-UX system, version 5.x. Other versions of HP-UX, and other models of Hewlett-Packard computers running HP-UX, may have a different file system.

HP-UX directory structure
The hierarchical grouping of directories and files on HP-UX.

inode
A data structure containing information about a file such as file type, pointers to data, owner, group, and protection information.

ITE
The Internal Terminal Emulator program which allows the display provided with some of the supported consoles 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 (see the *LIF(5)* entries in the *HP-UX Reference*.

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).

major number

Same as driver number.

minor number

A hexidecimal number made up of a select code, bus address, unit number, and volume number of the device.

multi-user state

A state of HP-UX when terminals in addition to the system console allow communication between the system and its users. The multi-user state (not to be confused with a multi-user system) is usually state 2.

MUX

MUX is an abbreviation for Asynchronous Multiplexer. The HP 98642 four channel MUX is available for the Series 200/300. Each channel is an RS-232C port which is normally associated with a */dev/ttyXX* file.

partition

A part of the hard disc that contains a file system. Currently you can only have one partition per disc.

path name
A series of directory names separated by / characters, and ending in a directory name or a file name.

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. 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 requestioning the I/O, rather than going through the file system buffer cache.

root
Root refers to the highest level directory (root directory or /).

select code
Part of an address used for devices; a number determined by switch settings on the interface card. Each interface card is in turn connected to a peripheral. Multiple peripherals connected to the same interface card share the same select code.

shell
A program that interfaces between the user and the operating system. HP-supported shells are: /etc/sh /etc/csh /etc/rsh /etc/pam

single-user state
A state of HP-UX when the system console provides the only communication mechanism between the system and its users.

source device
The mass storage device from which HP-UX is installed. The source device must be a cartridge tape drive or flexible disc drive. It cannot be the internal flexible disc drive on the models 226 or 236.

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.

superblock

A data structure containing global information about the file system such as file system size, disc information, and cylinder group parameters. The superblock is created at the same time as the file system and is replicated into each cylinder group.

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.

track

One of several concentric circles on the surface of a disc upon which data is recorded (see Figure G.1).

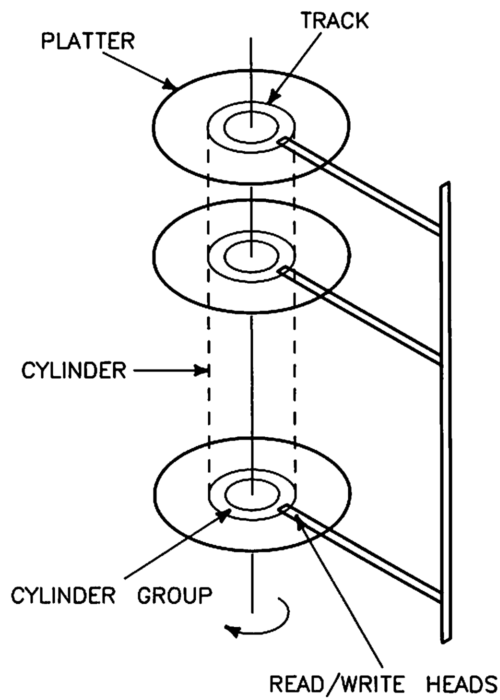


Figure G.1: Track, Cylinder, and Cylinder Group on a Disc

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 disc drive in a device with a multi-drive controller. When referring to single-controller integrated disc/tape or disc/flexible disc drive, a unit is used to distinguish between disc and cartridge tape drives or hard disc and flexible disc drive.

The unit number also selects a single partition on the 913x series.

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 disc drive. The volume number is also used to inform the device driver of special handling semantics (such as printer drivers skipping over perferations).

Index

a

b

C

d

e

f

g

h

i

k

l

m

n

o

p

S

t

U

V

W

MANUAL COMMENT CARD

HP-UX System Administrator Manual
for HP 9000 Series 200/300 Computers
Manual Reorder No. 97033-90049

Name: ______________________________

Company: ______________________________

Address: ______________________________

____

Phone No: ________________

Please note the latest printing date from the Printing History (page ii) of this manual and any applicable update(s); so we know which material you are commenting on ________________.

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
Fort Collins Systems Division
Attn: Customer Documentation
3404 East Harmony Road
Fort Collins, Colorado 80525

HEWLETT
PACKARD

Reorder Number
97033-90049
Printed in U.S.A. 12/85

97033-90606
Mfg. No. Only