

HP-UX System Administrator Manual



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

HP-UX System Administrator Manual

for the HP 9000 Series 500

Manual Part No. 97089-90047



© Copyright 1983, Hewlett-Packard Company.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this document is subject to change without notice.

Use of this manual and flexible disc(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs can be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

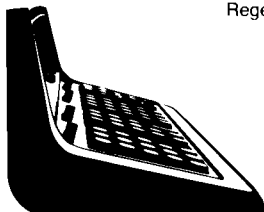
Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the Rights in Technical Data and Software clause in DAR 7-104.9(a).

© Copyright 1980, Bell Telephone Laboratories, Inc.

© Copyright 1979, 1980, The Regents of the University of California.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.



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 1983...First Edition

Warranty Statement

Hewlett-Packard products are warranted against defects in materials and workmanship. For Hewlett-Packard Fort Collins Systems Division products sold in the U.S.A. and Canada, this warranty applies for ninety (90) days from the date of delivery.* Hewlett-Packard will, at its option, repair or replace equipment which proves to be defective during the warranty period. This warranty includes labor, parts, and surface travel costs, if any. Equipment returned to Hewlett-Packard for repair must be shipped freight prepaid. Repairs necessitated by misuse of the equipment, or by hardware, software, or interfacing not provided by Hewlett-Packard are not covered by this warranty.

HP warrants that its software and firmware designated by HP for use with a CPU will execute its programming instructions when properly installed on that CPU. HP does not warrant that the operation of the CPU, software, or firmware will be uninterrupted or error free.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

* For other countries, contact your local Sales and Support Office to determine warranty terms.

Table of Contents

Chapter 1: Overview of the Job

The System Administrator's Responsibilities	1
Hardware Installation and Test	1
Installation of HP-UX	1
Evaluation of User Needs	1
Configuration of HP-UX	2
Allowing Access to Other Users	2
Monitoring File System Use and Growth	2
Periodic Updating	2
System Backup and Restoration	3
Error Detection and Correction in the File System	3
Assistance for Other Users	3
Provide a System Administrator "Back-up"	4
User Survey (Sample Form)	5

Chapter 2: Installing HP-UX

Before Installing HP-UX	9
Installation Procedure	13
What Now?	19
Verify the Installation	19
Set the System Clock	19
Onward to Success	21

Chapter 3: Concepts

Processes	23
Process Creation (Parent and Child Processes)	23
Process Termination	24
Process Groups	24
Terminal Affiliation	24
IDs	25
The Super-User	26
File System Implementation	26
The Volume Header	31
File Format and Compatibility	31
Compatibility Tips	32
File Protection	32
Protecting Directories	34
Setting Effective User and Group IDs	34
The Sticky Bit	36

Memory Management	36
Memory Allocation	36
Shared Code	39
Shared Code and the Sticky Bit	39
Shared Code: Benefits vs. Cost.	40
Demand Load	40
Demand Load: Benefits vs. Cost.	41
Virtual Memory	41
Virtual Memory: Benefits vs. Cost.	42
Paged External Data Segments	43
Paging: Benefits vs. Costs	44
Combining Memory Management Tools	45
The Buffer Cache	45
Read Ahead Level	46
The File System Buffer Cache: Benefits vs. Costs.	46
Device I/O.	47
Device Classes.	47
Drivers	47
Address	47

Chapter 4: System Boot and Login: a Detailed Examination

System Boot.	50
Login.	54
Changes You May Want to Make	55
Summary	60

Chapter 5: The System Administrator's Toolbox

Adding/Moving Peripheral Devices	63
Connect the Peripheral.	63
Drivers	65
Creating Special Files	69
Mass Storage Devices (Disc Drives and Magnetic Tape Drives).	70
Printers.	72
Graphic Output Devices (Plotters and CRT Graphics)	73
Graphic Input Devices	74
Memory "Discs"	74
Essential Special Files.	75
Examples	75
Terminals: Special Considerations	76
Printers: Special Considerations	77
Adding/Removing Users	77
Procedure.	77
Initializing Media	79
Examples	81
Backing Up and Restoring the File System.	82
System Backup and Restoration Procedures	83
Restoring a Backup	84
Special Considerations or "What if...?"	85

Booting HP-UX	86
Making Operating Systems “Loadable” or “Unloadable”	86
Searching of an Operating System	86
Changing the Contents of the Boot Area	87
Examining and Changing the Content of a Boot Area	87
Boot Procedure	88
Changing a Password	89
Communicating with System Users	90
Configuring HP-UX	91
Setting the Default Configuration Parameters	92
Controlling Disc Use	94
Creating a Group and Changing Group Membership	95
File System Check and Repair	96
fsck	96
fsck and Virtual Memory	100
Mounting and Unmounting Volumes	101
To Mount a Volume... ..	101
To Unmount a Volume... ..	104
Special Considerations	104
Option Product Installation and Updating	105
Setting the System Clock	110
Special Considerations	110
System Shutdown	111
stopsys	111
stopsys -r	112
chsys	112
killall	112
shutdown	113
Summary	114
Updating Your Operating System	115
Before You Begin	115
Phase 1 - Bring Down Current HP-UX System	118
Phase 2 - Preparing for the Update.	118
Phase 3 - Performing the Update	119

Appendix A: System Loader Messages

Messages	A-2
----------------	-----

Appendix B: Installation/Update Error Information

Error Numbers for Sysinstall, Sysupdate, Coreinstall, and Coreupdate	B-2
Error Numbers for Optinstall and Optupdate	B-4

Subject Index



Chapter 1

Overview of the Job

The system administrator is the person responsible for installing the HP-UX software, updating the software, tuning it (via configuration and other procedures) for optimum performance, maintaining it, and repairing it when something goes wrong. Additionally, the system administrator is a local HP-UX expert, a person to whom other HP-UX users go for help.

The System Administrator's Responsibilities

Hardware Installation and Test

As system administrator, you should make sure that your computer is installed and operating properly by using the available hardware tests (refer to your computer's installation manual). Attempts to install HP-UX succeed only if the computer is functioning properly.

Installation of HP-UX

The HP-UX Operating System is supplied on magnetic tape cartridge. As system administrator, you are responsible for ensuring that the operating system is installed. Instructions for installing HP-UX are provided in both the "Where Do I Start?" manual and this manual's chapter "Installing HP-UX".

Evaluation of User Needs

Once HP-UX is installed, you should 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, helps you configure HP-UX for optimum performance.

To help you in your task, 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 500000 bytes of data storage.". The survey is designed to help you identify the needs of the system users so that you can translate those needs into data relevant to system configuration.

Configuration of HP-UX

The operating system keeps a list of values that specifies system characteristics and the manner in which computer resources are allocated. By changing the values, you can adjust the operating system such that system resources are allocated in the most efficient way. This process is known as configuring the system. Altering the system configuration can improve response time (the time required for the system to complete a user's request) and decrease the number of disc accesses resulting in increased operating speed and longer disc life. Once you are familiar with the system, you can use the instructions in the "System Administrator's Toolbox" chapter of this manual to help you alter the system configuration.

Allowing Access to Other Users

Once HP-UX is installed, you are responsible for modifying the operating system to allow 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 located in the "System Administrator's Toolbox" chapter of this manual.

Monitoring File System Use and Growth

As your system is used, more and more files are added to the file system. If unused files are not removed from the system, the amount of space required to store the files eventually exceeds the system's capacity to store them. As system administrator, one of your responsibilities is to monitor the size of the file system, identifying the files that are not used. Such files should either be archived (written to a removable mass storage medium for future use) or removed from the system. Similarly, you should watch for files that continue to increase in size. Consult the file's owner to ensure that the file is needed and to see if its size can be reduced. Instructions for monitoring the file system use and growth are located in the "System Administrator's Toolbox" chapter of this manual.

Periodic Updating

If you have purchased HP product support services (CSS - Customer Support Service, SIS - System Information Service or SSS -Software Subscription Service) for the HP-UX Operating System, programming languages (such as C, FORTRAN and Pascal), and options (such as graphics), you will receive periodic updates for this software. These updates modify existing capabilities and can add new capabilities, ensuring that your system has the latest version of the software.

As system administrator, you are responsible for installing each software update. You also should ensure that the manuals for each software package are updated to include the documentation changes provided with each update. Keep a log showing the date when each update was installed and the reason that the update was issued. Since each update depends on changes made by the previous update, it is imperative that each update be installed when it arrives. Instructions for installing updates are located in the "System Administrator's Toolbox" chapter of this manual.

Additionally, you must notify all system users that the system has been updated, specifying any changes in system operation caused by the update. This is most easily accomplished by providing the user a message when he logs in (gains access to the system).

System Backup and Restoration

The HP-UX Operating System, programming languages, options and applications software represent a large investment of time and money. Each access to the system provides an opportunity for an error to occur; files can be unintentionally removed or corrupted. In HP-UX, such an error can easily cause additional errors in the system; when a corrupt system is used, the number of errors in the system can increase quickly.

In a similar vein, unforeseen circumstances (such as spilled coffee, smoke contamination, dust or fire) can cause the destruction of the mass storage device, its media, and/or the data it contains. This also results in the loss of the system.

As system administrator, you should make a **backup** - a copy of the HP-UX Operating System, file system, programming languages and options. Typically, at least one backup is performed daily. However, the frequency of your system backups should be based on the answer to the question "How much data can I afford to lose?"

In the event that your system is destroyed, you can quickly recover by restoring the latest version of your system backup. If a user accidentally removes a needed file, the file (or a previous version of the file) can be recovered by copying it back into the file system from the backup. The concepts of backup and restoration are discussed in the "Concepts" chapter of this manual. Instructions for using the supplied *backup* command are located in the "System Administrator's Toolbox" chapter of this manual.

Error Detection and Correction in the File System

Every day that the system is used, hundreds of 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, due to abnormal system shutdown or abnormal program termination). When an update fails, the file system can become corrupt.

HP-UX provides the *fsck* command - a program that checks the consistency of a file system and (optionally) repairs that system. You should check the file system's integrity on a periodic basis (for example, once a day) as well as each time HP-UX is booted. Continuing to use a corrupt file system only further corrupts the system. Instructions for verifying and repairing the file system are located in the "System Administrator's Toolbox" chapter.

Assistance for 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 CSS (Customer Support Service) or SIS (System Information Service), you receive PICS (Phone In Consulting Service). 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 difficulty, then you should call PICS.

Provide a System Administrator “Back-up”

At least one other person should be trained as 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. Later chapters in this manual show you how to run programs and perform tasks automatically at specified times. For example, by scheduling certain programs to run at a specified time, you can automatically back up the system or verify the integrity of the file system.

User Survey (Sample Form)

Name _____ Location _____

Phone _____

Location where you will be using the system. _____

User Category

- | | |
|---|---|
| ☐ 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 feature options will you use (such as graphics)? _____

6 Overview of the Job

What computer hardware or peripherals will you need to access?

☐ Thermal printer

☐ Plotter

☐ Impact printer

☐ Removable Mass Storage Device

☐ Graphics Terminal

☐ Other _____

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

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? _____

Chapter 2

Installing HP-UX

This section provides a step-by-step method for installing the HP-UX Operating System on an HP 9000 Series 500 Computer. If you have a working operating system and only want to update it, see the "Updating Your Operating System" and "Option Installation and Update" sections of the "System Administrator's Toolbox" chapter. The following terminology is used throughout this procedure:

- **CS/80** - a family of mass storage devices that communicate via the CS/80 (Command Set '80) command set. HP-UX can only be booted (loaded) from a CS/80 device.

CS/80 devices currently supported on the HP 9000 Series 500 are:

HP 7908P/R 16.5 Mbyte Disc/Tape Drive.

HP 7911P/R 28.1 Mbyte Disc/Tape Drive.

HP 7912P/R 65.6 Mbyte Disc/Tape Drive.

HP 7914P/R 132.1 Mbyte Disc/Tape Drive.

the built-in cartridge tape drive of the HP 7908P/R, HP 7911P/R, HP 7912P/R and HP 7914P/R Disc Drives.

HP 7933H 404 Mbyte Winchester Disc Drive.

HP 7935H 404 Mbyte Winchester Disc Drive with removable media.

HP 9130K - the built-in 0.25 Mbyte flexible disc drive of the HP 9000 Model 520 Computer.

HP 97093A - the built-in 10 Mbyte Winchester disc drive of the HP 9000 Model 520 Computer.

- **88140 cartridge tape drive** - the built in cartridge tape drive of a CS/80 disc drive (such as the HP 7908P/R, HP 7911P/R, HP 7912P/R, and HP 7914P/R).
- **source device** - the mass storage device from which HP-UX is installed. The source device must be a CS/80 device such as the built-in cartridge tape drive (88140L/S) of a CS/80 disc drive.
- **destination device** - the mass storage device on which HP-UX is to be installed. The destination device must be a CS/80 device other than the HP 9000 Model 520's built-in flexible disc drive or its built-in Winchester disc drive. The built-in cartridge tape drive of a CS/80 device should not be used as a destination device when installing the system.

- **kernel** - the core of the HP-UX operating system. The kernel is compiled code that is responsible for managing the computer's resources (such as allocating memory and scheduling programs for execution).
- **boot area** - a portion of a mass storage medium on which the operating system (such as the HP-UX kernel) resides. The boot area is created to be a specific size when the medium is initialized. Once created, the boot area is reserved for holding operating systems; it cannot be used to store your files.

The contents of a boot area can be marked "loadable" or "not loadable" with the *osmark* command. Marking a boot area as "loadable" or "not loadable" determines, in part, which operating system is loaded by the system loader.

- **system loader** - a program that permanently resides in the computer (in ROM); it is loaded and run each time the computer is powered-up. The function of the system loader is to search the boot areas of all CS/80 mass storage devices present on the system and load the first "loadable" operating system found. Boot areas marked "not loadable" are ignored by the system loader (whether the boot area contains a valid operating system or not). The order in which devices are searched is described in the System Administrator manual chapter, "System Boot and Login: a Detailed Examination".
- **system console** - a terminal whose location HP-UX always knows. System error messages, loader error messages and other system communications are sent to the system console. When HP-UX is installed on a Model 520, the system console is the computer's built-in keyboard and display (as described in the HP-UX: Selected Articles manual).

When HP-UX is installed on a Model 530 or Model 540 Computer, the system console is the terminal connected to the computer via select code zero (0). (If select code zero (0) is used for an HP 27130A Asynchronous 8-Channel Multiplex Interface, the terminal connected to port zero (0) of the HP 27130A is the system console.)

Before Installing HP-UX

Before HP-UX can be installed, a few procedures need to be followed.

1. Install your computer as described in its installation manual.
2. If you are installing HP-UX on a Model 520, this installation step does not apply to you. Skip to installation step 3.

If you are installing HP-UX on a Model 530/540, your HP Customer Engineer will initially install the computer hardware portion of your system. When all parts of your order have arrived, call your HP Customer Engineer. In case you want to re-install your system at a later time, instructions for installing HP-UX on your computer are included in this section. Read the remainder of this section (but leave the hardware connections to the HP Customer Engineer); it helps you identify much of the information you need to install the HP-UX Operating System.

The system console must next be connected. If you are connecting the system console via an HP 27128A ASI Card (Asynchronous Serial Interface Card), the card **must** be installed such that its select code is zero (0). Refer to the computer's installation manual to learn how to install the interface card and determine its select code. Before installing the card, locate the set of eight switches on the ASI Card. **Set the eight switches on the ASI Card such that switches 1 through 7 are open and switch 8 is closed.**

Alternately, an HP 27130A Asynchronous 8-Channel Multiplex Interface may be used when connecting to the system console. It must be installed such that its select code is zero (0). The terminal's interface cable must be connected to channel zero (0) of the Multiplex Interface.

- a. Next the terminal must be configured so that it can "talk" to HP-UX. The manual supplied with the terminal describes how to use the function keys to configure the terminal. First, press the appropriate key for terminal configuration; select the hardwired default values. If necessary, alter the appropriate fields such that the terminal's configuration parameters have the indicated values:

LocalEcho:	OFF
CapsLock:	OFF
InhHndShk(G):	YES
Inh DC2(H):	YES
Baud:	9600
Bits/Character:	8
Parity:	none
RecvPace:	Xon/Xoff
Full Duplex	
EnqAck:	YES

After the system is installed and running, you may change the configuration parameters to suit your own needs.

- b. Press the appropriate function key to select datacomm configuration; select the hardwired default values. Initially, the baud rate for the terminal should be 9600, to match the default `/etc/inittab` entry for the system console. After the system is installed and running, you may change the configuration parameters to suit your own needs.

3. Connect the source device and destination device to the computer. Your computer's installation manual describes the method for installing the interface cards. Before installing the card, refer to the installation manual supplied with the HP-IB Interface to:

- set the card for high speed operation.
- set the card as system controller.
- set the card's HP-IB address to 30.

Note

High speed devices (such as disc drives and tape drives) can share an HP-IB interface card with low speed devices (such as plotters and some printers), but they must be set to the slower speed.

CAUTION

DO NOT ATTEMPT TO INSTALL AND CONNECT A CS/80 DISC DRIVE (OTHER THAN THE HP 97093A, HP 9130K OR THE HP 7908P/R DISC DRIVES) YOURSELF. THE CS/80 DISC DEVICES ARE PACKED TO PREVENT DAMAGE DURING SHIPMENT. THEY MUST BE UNPACKED AND INSTALLED BY AN HP CUSTOMER ENGINEER TO PREVENT DAMAGE TO THE DEVICE.

The installation process involves loading a minimum working HP-UX system from the supplied source media. You must then run a program that installs the system on the destination device. For the installation program to operate correctly, the source device and destination device must be connected such that the select code, bus address and unit number of the device are within a range of acceptable limits, listed below.

The installation program assumes certain defaults for the select code and bus address of the source and destination devices. You can ease the installation task by installing and connecting the source and destination devices such that their select code, bus address and unit number are the default values for the installation program. The following table shows the acceptable ranges and the installation program's defaults for these values.

	Acceptable Range	Default Value
Source device:		
select code	0 - 23	5
bus address	0 - 7	0
unit number	0 - 1	1
Destination device:		
select code	0 - 23	5
bus address	0 - 7	0
unit number *	0	0

* Note that the unit number of the destination device is not controllable (it is always zero).

4. Set the select code, bus address and unit number of both the source device and the destination device. These values are needed by the installation program to properly install HP-UX.

The select code of the source device and destination device is determined by the slot number in which its interface card is installed (described in the computer's installation manual).

The bus address of the source device and destination device is controlled by a set of switches on the device. The manual supplied with the peripheral device describes the manner in which its bus address is set.

(If you are installing to an HP 7935H disc drive you must set the cartridge tape drive (source) to a lower bus address to ensure that the system will boot from the tape during initialization.)

The unit number of the source device is determined by the type of device. The unit number of the source device is zero (0) when the source device is the built-in cartridge tape drive of a dual controller HP 7911P/R (Opt. 001), a dual controller HP 7912P/R (Opt. 001), or a dual controller HP 7914P/R (Opt. 001).

The unit number of the source device is one (1) when the source device is the built-in cartridge tape drive of an HP 7908P/R, a single controller HP 7911P/R, a single controller HP 7912P/R, or a single controller HP 7914P/R.

The unit number of the destination device is always zero (0).

5. If you have a Model 520, verify that the computer, the disc drive, and its interface are installed and operating properly using the System Integrity Tests. These tests can be ordered from HP (part number 09020-10010). If you have a Model 530/540, these tests are incorporated into HP-UX. Thus, this step must be deferred until later.
6. Examine the media on which HP-UX is supplied. To properly install HP-UX, you must determine whether the source medium contains only HP-UX or whether the source medium contains HP-UX and other products (such as FORTRAN, Pascal, or the Device-independent Graphics Library).

The system is supplied on an HP 88140L/S cartridge tape for installation from the built-in cartridge tape drive of your HP 7908P/R, HP 7911P/R, HP 7912P/R, or HP 7914P/R Disc Drive. Installing to other types of CS/80 discs which do not have cartridge tape drives requires the temporary use of a CS/80 disc drive which does. Look for the product number(s) on the label of the cartridge tape. One of the following four numbers should be present:

- 97070A (Single User HP-UX for the HP 9000 Model 520).
- 97079A (Single User HP-UX for the HP 9000 Model 530/540).
- 97080A (Multi-user HP-UX for the HP 9000 Model 520).
- 97089A (Multi-user HP-UX for the HP 9000 Model 530/540).

The tape may contain other product numbers as well (such as HP 97071A - the product number of the FORTRAN programming language for a single-user HP-UX operating system). Thus, multiple products may be supplied on the same medium.

7. HP-UX is installed by running one of two supplied installation programs, *sysinstall* or *coreinstall*. Determine which of the two programs you will use to install your system.

Use the information gathered in step 6 to decide which program to use. If the supplied tape contains only HP-UX, you may use either installation program to install the system.

If the supplied tape contains HP-UX and one or more additional products, you must decide whether or not you want the additional products installed at the same time HP-UX is installed. If you want only HP-UX installed, use the *coreinstall* program. (The additional products can be installed at a later time, using the *optinstall* program, as described in this manual's chapter entitled "System Administrator's Toolbox"). If you want to install all products supplied on the cartridge tape, use the *sysinstall* program. **Optional boot area code of HP-UX (such as optional drivers) and optional boot area code of optional products are not installed by the *sysinstall* or *coreinstall* programs**, but are copied onto the destination disc under */system/product#*, and must be added as needed to the boot area. Appendix B contains error information for these commands.

8. As part of the installation process, the destination disc medium is initialized. Initializing the destination disc medium removes all its files and data, verifies that the destination medium is not corrupt and creates a boot area on the destination medium.

As a last step before installing the system, you should decide the size of the boot area that is created at initialization. There must be sufficient room in the boot area to install the boot area portion(s) of all products being installed. The minimum size allowable for the boot area is 500 000 bytes; the default size is 1 100 000 bytes (sufficiently large for either HP-UX or the BASIC Language Operating System; the BASIC Language Operating System is supported only on HP 9000 Model 520). The following table specifies the boot area portions of the currently available HP-UX products (and optional software). Use these values to calculate the size of the boot area you should create for your system:

Product	Recommended Boot Area Size for Product
HP-UX Operating System (including the HP-UX kernel and any optionally installed kernel code such as optional drivers ¹ .)	600 000 bytes
LAN 9000 (Local Area Network software)	300 000 bytes

You should create the boot area on the destination device such that it is large enough to contain the biggest system that you anticipate using. HP recommends that you "pad" the size of the boot area by 200 000 bytes to allow for future products that you may purchase and install. For example, if the largest system you anticipate using consists of HP-UX, its optional drivers and LAN 9000 (Local Area Network software), create a boot area that is at least 900 000 bytes (600 000 bytes + 300 000 bytes) in size; the recommended size would be 1 100 000 bytes (900 000 bytes required, plus 200 000 bytes for future products). Note that the size of the boot area cannot be changed except by re-initializing the disc.

If you plan to keep the BASIC Language Operating System on the same mass storage device as HP-UX (and plan to boot the BASIC Language Operating System from this device as well), use the manuals supplied with that system to determine the maximum size of the boot area required for that system. Then compare the boot area size required for each system; select the largest of the two values as the size of the boot area to be created by the installation program.

You are now ready to install HP-UX.

¹ Drivers for: CIPER protocol printers (such as the HP 2608S), HP 7970E Magnetic Tape Drive, HP 7971A 9-track Magnetic Tape Drive, HP 9885M/S Flexible Disc Drives, HP 82901 Flexible Disc Drive, HP 82902 Flexible Disc Drive, HP 9895 Flexible Disc Drive, and the Asynchronous Serial Interface driver for raw I/O transfers (used by the HP-UX Asynchronous Communications Software).

Installation Procedure

The system is supplied on an HP 88140L/S cartridge tape for installation from the built-in cartridge tape drive of your HP 7908P/R, HP 7911P/R, HP 7912P/R, or HP 7914P/R disc drive.

1. Examine the supplied cartridge tape containing HP-UX. Locate the write protect mechanism (labeled "SAFE") at the front, upper, right-hand corner of the medium. Using a coin or screwdriver, **turn the protect screw such that the arrow is pointed away from the word "SAFE"**.
2. Power up the source device. If the drive's busy light is on, wait for it to extinguish. Insert the HP 88140L/S cartridge into the disc's built-in cartridge tape drive. Wait for the drive's busy light to go out before continuing.
3. If the computer is not already powered down, switch off its power and wait a few seconds before proceeding. Switch on power to the computer. It first performs some hardware self-tests and then loads and runs the system loader. The system loader searches for an operating system, and, finding HP-UX on the supplied cartridge tape, begins to load HP-UX. As HP-UX loads, the following is displayed:

```
Loader Rev X
HP-UX Model YYYY Release X.X (970ZZA)
```

The message `Loader Rev X` identifies the current version of the computer's system loader. The characters `YYYY` stand for "520" or "530/540" which identify the model of HP 9000 Computer for which the software is intended. `970ZZA` identifies the product number.

Note

If any other message is displayed, refer to the appendix entitled "System Loader Messages" in the computer's installation manual. If no message is displayed, verify the operation of the computer and its peripherals by performing the tests described in the computer's installation manual.

4. The remainder of the installation procedure requires you to enter information from the keyboard of the system console. Before proceeding, you should understand how to correct errors made while typing the information. When typing information, use the backspace key (`BACK SPACE`) to move the cursor back to the position where the error was made. When `BACK SPACE` is pressed, the cursor moves one space backwards and erases the character that occupied the location. Only the backslash character (`\`) is not easily erased in this manner. Alternately, you can tell HP-UX to ignore everything typed on the current line by holding the `CTRL` key depressed as you press the `U` key. (After releasing both keys, press `RETURN`.)

Once `RETURN` is pressed, all information typed is sent to HP-UX; an error cannot be corrected once `RETURN` is pressed.

Note

When asked to enter numerical information (such as a select code or bus address), do not use leading zeros with the values you enter. For example, to specify the value 7, enter 7 not 07 or 007. Use of leading zeros can cause unexpected results.

When HP-UX is loaded and begins to execute, the message:

```
Load done.
```

is displayed. (On the Model 520, this message only appears briefly. On the Model 530/540, this message continues to exist on the system console.)

Some initial operations are performed and several messages are displayed (a copyright statement and statement specifying the version of the software). After about ten minutes, the super-user's prompt, #, is displayed. In response to the prompt, type the name of the installation program that you have decided to use (followed by **RETURN**). Execute either

```
sysinstall RETURN
```

to install all products supplied on the source medium or

```
coreinstall RETURN
```

to install only HP-UX from the source medium.

5. As the installation program executes, it displays the following:

```
HP-UX FILE SYSTEM INSTALLATION UTILITY
```

Then, depending on which installation program you specified, one of two messages is displayed. The following message is displayed when installing with the *sysinstall* program:

```
All products on this source media will be installed.
```

However, when installing HP-UX with the *coreinstall* program, the following message is displayed:

```
Only the HP-UX core product will be installed.
```

Next, the installation program displays the default values for the installation program's parameters:

```
Here are the default parameters for this installation:
```

```
* Source Device:          88140 Cartridge Tape Drive
                           select code      = 5
                           bus address      = 0
                           unit number     = 1

* Destination Device:    CS/80 Disc Drive
                           select code      = 5
                           bus address      = 0

* Initialization operation: WILL be done on destination disc.
                           bootsize        = 1100000 (bytes)
```

```
Are these values satisfactory for your system?
```

```
If they are not, you will be instructed to supply responses
for each parameter, interactively.
```

```
(Enter y or n, then RETURN; default is 'yes')
```

If the default installation parameters correctly specify the source and destination devices and if you are satisfied with program's stated intentions regarding initialization, enter y or Y (followed by **RETURN**). Then skip to step 10 of this procedure.

If one or more of the default installation parameters is not correct, enter n or N (followed by **RETURN**). Then using the following instructions, enter the correct values for the installation parameters.

6. Once you have specified that you do not want the default parameters used, you are prompted to identify the source device:

```
* Source Device:          88140 Cartridge Tape Drive
                           select code      = 5
                           bus address     = 0
                           unit number    = 1

Do these parameters accurately describe your source device?
(Enter y or n, then RETURN; default is 'yes')
```

If the source device is correctly identified, press **RETURN** to select the default response (yes) or enter:

```
y RETURN
```

Continue the installation procedure with step 7.

If the source device is not correctly identified, enter the following (then continue the installation procedure with step a):

```
n RETURN
```

- a. The system next prompts to find the select code of the source device:

```
Enter the select code of your 88140 tape drive.
The range for the select code is 0 through 23, inclusive.
```

Enter the select code of the source device, then press **RETURN**. Entering no value (simply pressing **RETURN**) causes the most recently entered value (or the default value, if it has never been changed) to be used.

- b. The system next prompts to find the bus address of the source device:

```
Enter the bus address of your 88140 tape drive.
The range for the bus address is 0 through 7, inclusive.
```

Enter the bus address of the source device, then press **RETURN**. Entering no value (simply pressing **RETURN**) causes the most recently entered value (or the default value, if it has never been changed) to be used.

- c. The system next prompts to find the unit number of the source device:

```
Enter the unit number of your 88140 tape drive.
The range of the unit number is 0 through 1, inclusive.
```

Enter the unit number of the source device, then press **RETURN**. Entering no value (simply pressing **RETURN**) causes the most recently entered value (or the default value, if it has never been changed) to be used.

7. Next, you are prompted to identify the destination device:

```
* Destination Device:      CS/80 Disc Drive
                           select code   = 5
                           bus address   = 0

Are the select code and bus address of the destination disc
correctly described?
(Enter y or n; then RETURN; the default is 'yes')
```

If the destination device is correctly identified, enter:

y

Continue the installation procedure with step 8.

If the destination device is not correctly identified, enter:

n

and identify the location of the destination device in a manner similar to that described in steps 6a, 6b, and 6c.

8. Next, the system asks about the size of the boot area to be created on the destination medium:

```
* Initialize operation: WILL be done on destination disc.
                           bootsize      = 1100000 (bytes)

Is the size of the boot area satisfactory?
(Enter y or n; then RETURN; default is 'yes')
```

If the boot area size displayed correctly identifies the amount of boot area space that you want reserved on the destination device, press (selecting the default answer, 'yes') or enter:

y

Continue this procedure at step 9.

If you want to change the size of the boot area created during initialization, enter:

n

You are next prompted to specify the size of the boot area to be created on the destination medium:

```
Please enter the number of bytes to be allocated for the
boot area.
The recommended size is 1100000 bytes.
The default size is 1100000 bytes.
The minimum size is 500000 bytes.
New value:
```

Enter a value specifying the size of the boot area that you want created on the destination medium, then press . Do not embed spaces or commas in the value you enter.

9. Next, the system displays the menu of default values shown in step 5, previously (the new values you entered during the prompting sequence replace the original values displayed). And again, you are prompted to specify whether or not the values displayed are valid for system installation. Continue the installation procedure from step 5. (This time, the default values for each installation parameter are the values you specified in steps 5 through 8).
10. Now that all of the parameters for the installation program have been entered and/or have been verified as correct, the system prompts you to re-examine the values and make sure that they are accurate:

```
Are you sure you are ready to begin the installation procedure
on the destination disc?
(Enter y or n, then RETURN; default is 'NO!')
```

If you are **not** satisfied with the installation parameters, either press **RETURN** (selecting the default answer, 'NO!') or enter:

```
n RETURN
```

The installation program terminates. Re-start the installation procedure.

If you are satisfied with the installation parameters and are ready to begin installation, enter:

```
y RETURN
```

The installation process begins with initialization of the destination device medium. The initialization process can require up to 60 minutes to complete.

CAUTION

DO NOT SWITCH OFF POWER TO THE COMPUTER OR DESTINATION DEVICE DURING DISC INITIALIZATION. TERMINATING THE INITIALIZATION PROCESS IN THIS FASHION MAY CORRUPT THE DISC MEDIUM. SHOULD THIS OCCUR, TRY RE-INSTALLING THE SYSTEM; IF THIS FAILS TO CORRECT THE PROBLEM, CALL YOUR HP CUSTOMER ENGINEER.

11. After the destination disc is initialized, the HP-UX kernel is copied to the boot area of the destination disc.

Next, the HP-UX files are copied from the source medium to the destination medium. The name of each file is displayed when it has been copied successfully. Up to one hour may be required to copy all of the HP-UX files to the destination medium.

If you are installing all products (such as FORTRAN) from the source medium, the products' files and non-optional boot area information are copied at this time.

12. When the system has been successfully installed, the following is briefly displayed:

```
INSTALLATION OF HP-UX ON THE DESTINATION DISC IS NOW  
COMPLETE,
```

```
The 88140 tape will now be automatically rewound for you.  
When you hear the buzzer, use the slide lever to remove the  
tape.
```

```
Once you have removed the tape, reboot the system.
```

When you hear the buzzer, remove the cartridge tape from the tape drive. Using a screwdriver or coin, **turn the write protect mechanism such that the arrow is pointed toward the word "SAFE"**.

Store the tape in a safe location (away from magnetic fields and harsh environments). If the disc should ever become corrupt such that HP-UX no longer boots from it, you can use the cartridge tape to re-install the system. To enhance security of your system, store the tape at a location accessible only to you and your back-up system administrator. Armed with the source tape, a determined individual can defeat the security of the system.

13. Finally, reboot the system. Power down the computer, wait a few seconds, then switch on power to the computer again. The system loader searches for and loads the HP-UX kernel (assuming that it is the first "loadable" operating system found). After some initial set-up (this may require several minutes), the login prompt is displayed:

```
login:
```

HP-UX is now installed.

What Now?

Note

Now that your system is installed, you have the ability to modify and customize the system. However, if you have purchased support services (CSS - Customer Support Services, SSS - Software Support Services, or SIS - Software Information Services) from Hewlett-Packard, only minor changes can be made without voiding your agreement. Consult your support plan or HP System Engineer regarding intended changes.

Verify the Installation

To verify that the HP-UX system has successfully been installed, you must first log in (gain access to the system). When the prompt `login:` is displayed, enter:

```
root 
```

(Note that does not erase characters during login, so any mistakes cause you to repeat the login procedure.)

Several messages are displayed followed by the shell prompt, `#`. Execute the `/etc/fsck` command by entering:

```
/etc/fsck -y /dev/rhd 
```

(If *fsck* reports that there is a superfluous process running, it is referring to the *cron* process. You need to find the *cron* process id (using `ps -e`), and then kill *cron* using the *kill* command.)

The file system verification routine (`/etc/fsck`) may take 20 minutes or more to complete. If the message `THIS FILE SYSTEM IS NOT COMPLETELY RESTORED` is displayed once `/etc/fsck` has completed, your HP-UX system has not been installed properly. Repeat the installation procedure.

If the message:

```
THIS FILE SYSTEM IS NOT COMPLETELY RESTORED
```

is **not** displayed, your file system is properly installed. (If you were forced to kill *cron* earlier, you may re-invoke it now.)

Set the System Clock

Check to see if the system clock is set correctly. Enter:

```
date 
```

The time displayed is in the format of a twenty-four hour clock (for example, 2:00 pm is 14:00). If the time displayed is not correct, you need to first set the correct time zone and then set the system clock.

As shipped to you, the system is set up to run in the Mountain Standard Time Zone. To change the time zone to your time zone, type two entries of the following 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. For example, in Denver, Colorado you would enter the following:

```
TZ=MST7MDT 
export TZ 
```

MST stands for Mountain Standard Time and MDT stands for Mountain Daylight Time. Until you change the entry in the file */etc/profile* (described in the System Administrator manual section "Setting the System Clock"), you will need to set the time zone in the fashion previously described each time the computer is powered up. Note, however, that setting the time zone yourself only sets the time zone for **you**, not for any other users. For the time zone setting to be effective for all users, the TZ variable must be set and exported in */etc/profile*.

Now that the time zone is set, you can set the correct time and date by typing an entry of the following form:

```
date MMddhhmmyy 
```

MM is a two digit integer representing the month (for example, March is represented as 03).

dd is a two digit integer representing the day of the month (for example, enter 02 for 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 a two digit integer specifying the last two digits of the current year. For example, 83 specifies 1983 as the current year.

The correct time and date are now set. Verify this by entering:

```
date 
```

If the time and date are not correct, repeat the procedure "Set the System Clock".

Note

Do not attempt to execute any other command 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 or the file system.

Log out by holding the **CTRL** key depressed as you press the **D** key.

Onward to Success

As shipped to you, your HP-UX system is set up to allow any of several users to log in. However, two in particular are of immediate interest: `root` and `guest`. Until you are familiar with HP-UX and understand its operation, you (or any other user) should avoid logging in as `root`; this is the user name supplied for the super-user. The user logging in with this user name has unlimited capabilities and can destroy system integrity if he is not careful.

Read the supplied tutorial text entitled “Introducing the UNIX System”. Try the examples on your system as you read. (Until you are familiar with the system, use the user name `guest` when logging in).

Next, read the chapters of the System Administrator manual entitled “Overview of the Job”, “Concepts”, and “System Boot and Login: a Detailed Examination”. These should give you a general understanding for the manner in which the system operates.

If you are the system administrator (and you have completed the recommended reading), log in to the system as the super-user (log in with user name `root`). Guided by the procedures in the chapter entitled “The System Administrator’s Tool Box”, consider performing the following operations:

- assign a password to the root user. Write down the password you assign; if it is lost or forgotten, no one can log in as the super-user. (A complete re-installation of the system may be required to recover from a forgotten super-user password. If the super-user password is lost, call your HP System Engineer for assistance.)
- add peripheral devices such as printers, plotters, discs, and 9-track tape drives to the system.
- add other users to the system.
- modify `/etc/motd`, `/etc/profile`, and `/etc/rc` to fit your installation’s requirements. The chapter “System Boot and Login: a Detailed Examination” describes the use of these programs.



Chapter 3

Concepts

In this section, the concepts essential to managing an HP-UX System are discussed. Initially, it is not necessary to understand all of these concepts in depth. However, you should at least be familiar with the terms.

Processes

A **process** is the 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 the *fork* system call. The terms **parent process** and **child process** refer to the original process and the process which it created, respectively.

When a child process is created, nearly all code and data (including virtual code and data) is copied from the parent to the child. Only shared code is not copied (the child process uses the same 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 memory 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. However, the identical call in the child's copy of the code always returns zero (0). Since the process IDs returned by the *fork* system call 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* statement 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.

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.

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 *setpgid* system call. The calling process becomes the **process group leader**; it and all of its 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. 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 (BREAK) 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.

IDs

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

A **real user ID** is an integer value which identifies the owner of the process. Similarly, a **real group ID** is an integer value which identifies the group to which the user belongs. The real group ID is a unique integer identifier that is shared by all members of a group. It is used to enable members of the same group to share files without allowing access to these files by non-group members. The real user ID and real group ID are specified by the file `/etc/passwd` and are assigned to the user at login.

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 “overlying” 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 students' 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 students' 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).

The Super-User

The term super-user describes those 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 (except those created and protected by other systems, such as the BASIC Language System).
- modify any system configuration/data files, add (and remove) users to the system.
- other system functions.

In short, a super-user has the ability to modify the system in any fashion he desires (for better or worse). Thus, super-user status should only be granted to those users who have a thorough understanding of the system and have a need for super-user capability. As system administrator, you must have super-user capability.

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.

By convention, the super-user is a member of the group "other".

File System Implementation

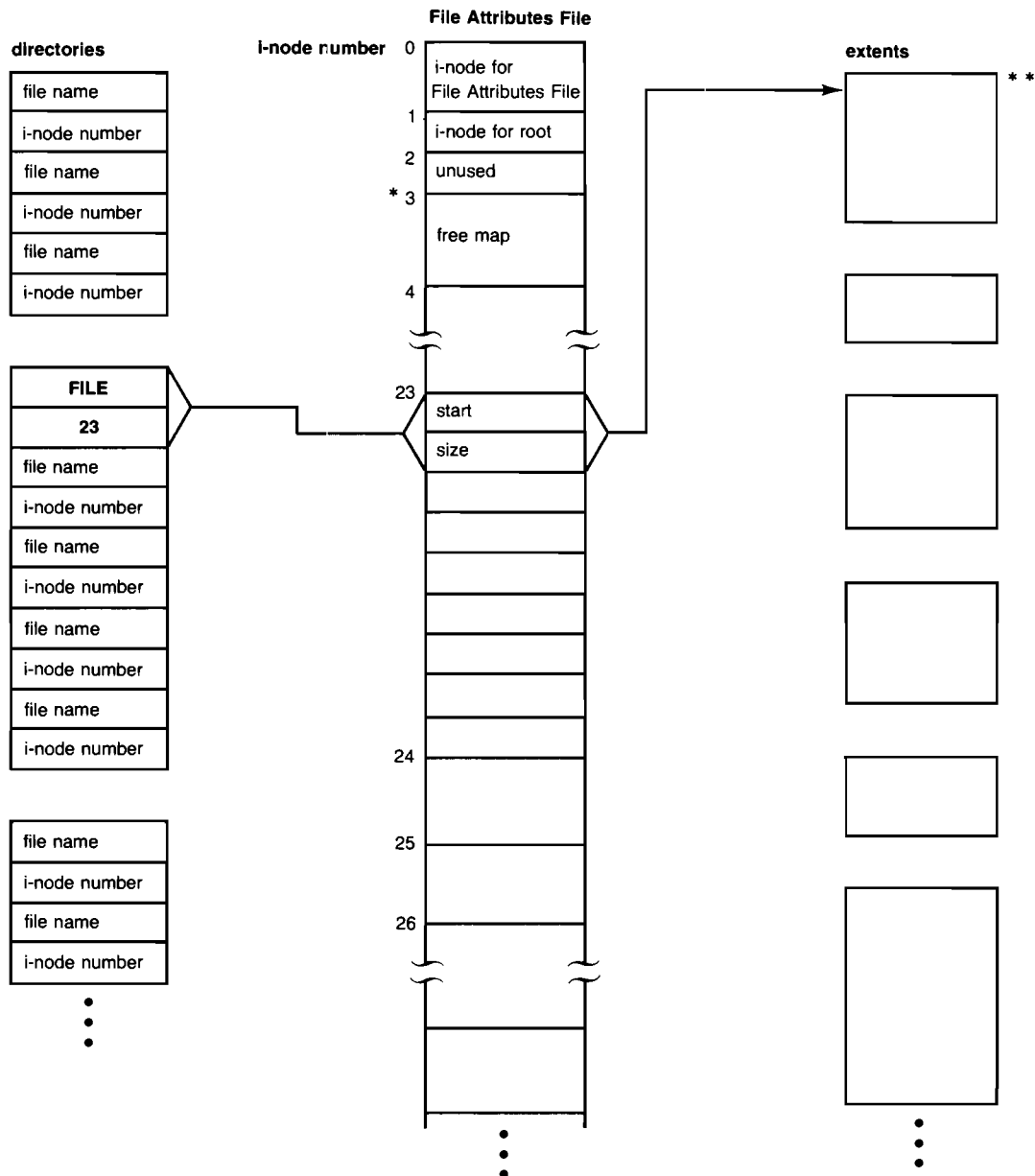
The supplied tutorial text for the operating system discusses the structure of the file system and introduces some basic concepts and terms. This section expands on those concepts and introduces new concepts which are unique to HP-UX. This information should aid you when verifying, maintaining and repairing the HP-UX file system(s).

The files of the HP-UX file system are stored on a formatted mass storage medium, usually a disc. A file is specified by a path name - a series of directory names separated by / characters and terminated with the file name. This method is easy to use and remember but difficult for the computer and disc to manage efficiently. The actual method in which files are stored is explained by the text and diagrams that follow.

The file system implementation is a concept best explained by example. Suppose that you have a file named FILE which contains a list of mailing addresses. When FILE is created, the system places its name in a directory (another HP-UX file). It then associates a unique integer value (in this case, 23) with the file name. This integer value (called an i-node value or number) points to a structure called an **i-node**. In turn, the i-node contains a pointer to one or more groups of contiguous disc blocks, called **extents**, in which FILE's data is actually stored. A pointer in the i-node consists of the starting address of the file's extent(s) and the size of the extent(s). This is shown in the illustration that follows. All i-nodes for a file system are kept in a file called the **File Attributes File**. Each disc has a single File Attributes File which describes the contents of the disc.

File System Implementation

File with a Single Extent



* The free map may actually occupy more than one i-node.

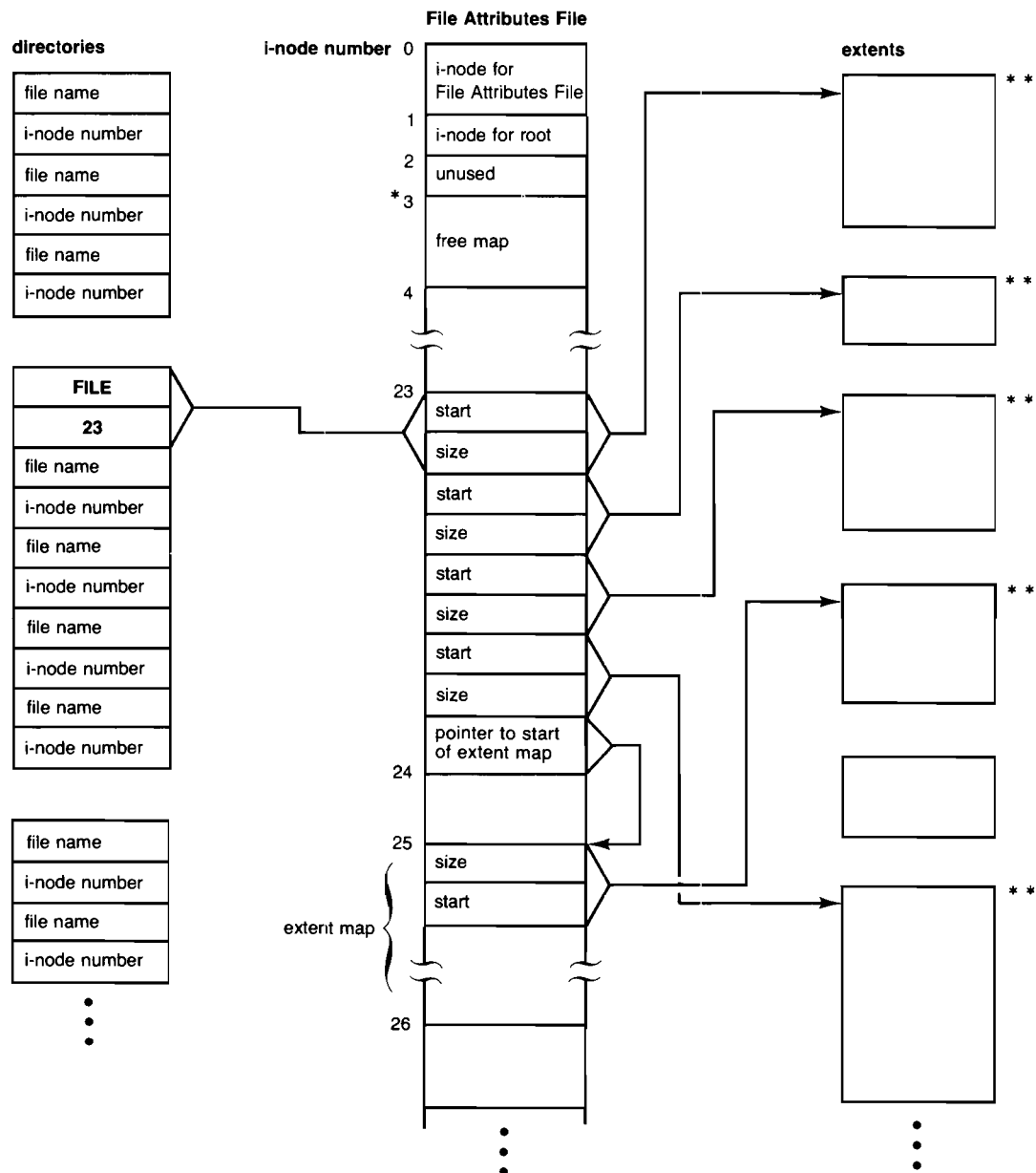
** Actual contents of FILE.

From the diagram, you can see that besides an i-node for each file stored on the disc, the File Attributes File contains an area called the **free map**. The free map is an array of binary values, one for each block on the disc. If a bit value is 0, the corresponding disc block is currently being used for storage. If a bit value is 1, then the corresponding disc block is available for storage.

When a file is created, its contents are placed in one large extent, if possible. If the file is later modified and enlarged, the system attempts to place the added information in blocks that are contiguous with the existing extent. If there are not enough free blocks that are contiguous to the existing extent, the system places the added information in a single new extent (if possible); it then adds to the file's i-node, a pointer to the additional extent. The i-node is capable of holding four pointers to the extents which comprise the file. If the system finds it necessary to create more than four extents for a file, an extent map is automatically added (chained) to the file's i-node. An extent map is capable of holding an additional 13 pointers; there is no limit to the number of extent maps which may be added to an i-node. You should note, however, that as the number of extents claimed by a file increases, the amount of time required to access the information also increases.

The following diagram shows the structure of the file, FILE (from the previous example) after it has been modified several times. When FILE was modified, there were not enough free blocks contiguous with the existing extent in which to place the added information. Thus, additional extents were created to hold the new information. When the number of extents claimed by FILE surpassed four, the system added an extent map to FILE's i-node.

File System Implementation File with Multiple Extents



* The free map may actually occupy more than one i-node.

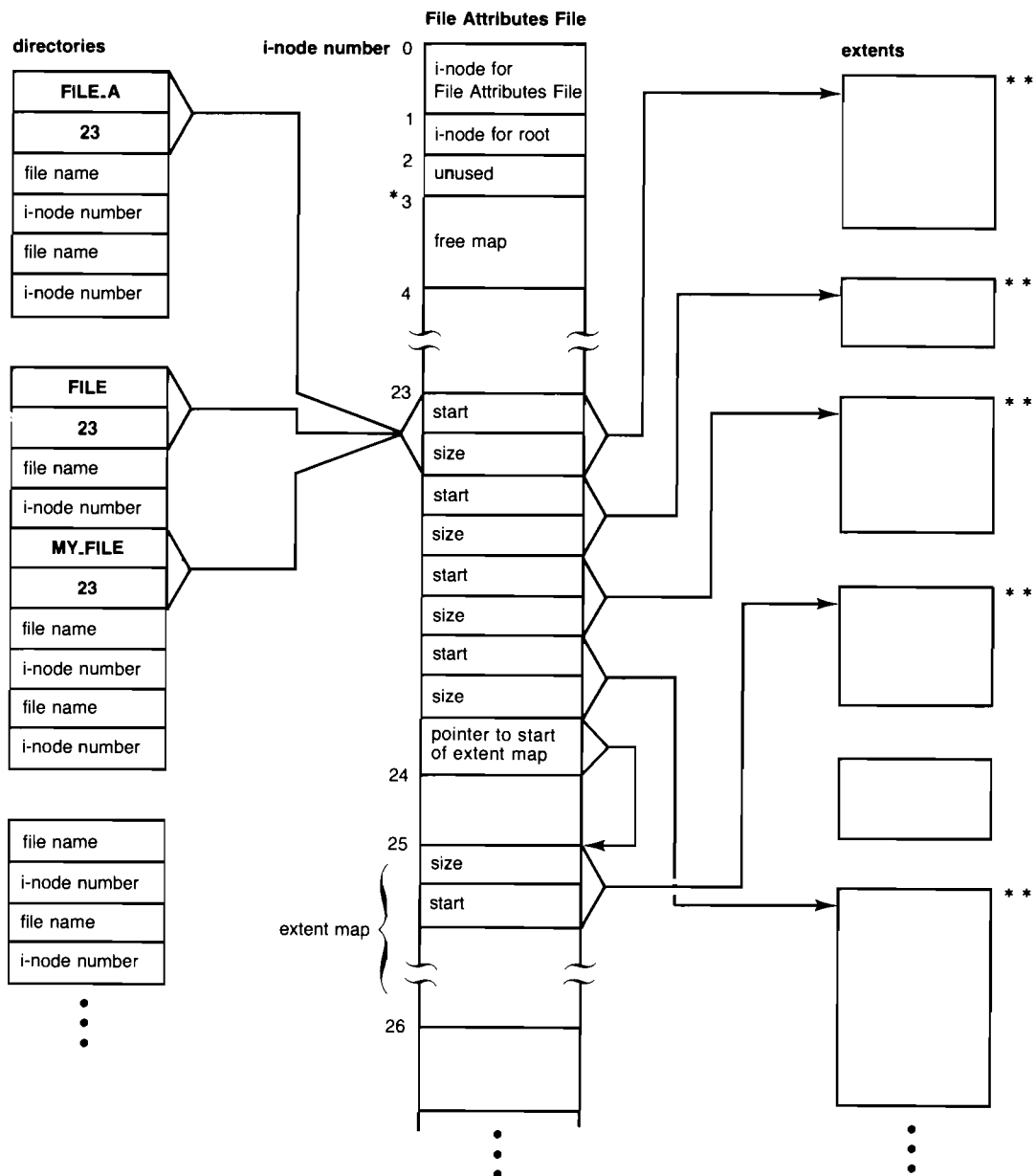
* * Actual contents of FILE.

In HP-UX, it is not unusual to have a single file on a disc which has several file names; the file is said to have multiple links. A file has multiple links when the file's i-node is pointed to by two or more directory entries. Each directory entry has its own name for the file but all share the same i-node value and thus, the same extents containing the file's data.

For example, suppose that three users all wish to have the file FILE (from the previous examples) in their login directory. Each user creates a link to FILE (via the *link* or *ln* command) from his login directory. From the viewpoint of each of these users, the file resides in their login directory.

A link is removed when a user removes the file from his directory (for example, with the *rm* command). When all links to an i-node have been removed, the disc blocks forming the file's extents are marked "free" in the free map. The file's i-node and extent maps (if it has extent maps) are marked as unused.

File System Implementation File with Multiple Links



* The free map may actually occupy more than one i-node.

** Actual contents of FILE.

The Volume Header

Each disc or tape medium available to your HP-UX system is called a **volume**. Each volume has a **volume header** located at block 0 of the volume. The volume header:

- identifies the volume format (always Structured Directory Format [SDF] for HP-UX files).
- specifies the size in bytes of each disc block.
- lists the maximum number of disc blocks available on the disc.
- lists the starting block number of the File Attributes File.
- lists the size and the starting location of the boot area.
- contains other system information (as described in *fs(5)* in the HP-UX Reference manual).

File Format and Compatibility

The format of the mass storage media on which HP-UX files are stored is Structured Directory Format (SDF). This is not necessarily the format for other operating systems patterned after the UNIX* operating systems.

If you are using both HP-UX and the BASIC Language System on your computer (supported on the Model 520 only), you should consider the following compatibility issues:

- BASIC recognizes many different file types (BDAT, BIN, ASCII, DATA and BCD).
- To BASIC, all HP-UX files are BDAT files.
- When BASIC files are copied with HP-UX commands, the file type of the copy is always BDAT (regardless of the file's original type).
- If a file or directory is used by both BASIC and HP-UX, be careful in assigning protections. A file protected in BASIC may not be accessible from HP-UX. Files with protection (access restrictions) in HP-UX may not be accessible from BASIC.

This last issue needs to be examined in more detail. When an HP-UX file is accessed from BASIC, the file's HP-UX mode corresponds to the file protection of a BASIC file. The HP-UX file's read and write permissions for the class of users "other system users" (not the file's owner nor the file's group) are interpreted as the the read and write protections of a BASIC file. The file's execute permission (as specified by its mode in HP-UX) is ignored when the file is accessed from BASIC.

* UNIX is a Trademark of Bell Telephone Laboratories, Inc.

The read, write and execute permissions of an HP-UX directory affect the manner in which the directory (and the files it contains) can be accessed from BASIC. The interpretations of the different protections for directories is shown in the following table:

HP-UX Directory Protection	BASIC Directory Protection
r - (read permission) allows the the user to list the contents of the directory. (read permission).	R - (read permission) allows user to list the contents of the directory with the BASIC CAT statement.
w - (write permission) allows the user to add and remove files from the directory.	W - (write permission) allows the user to create or delete entries from the directory.
x - (execute permission) allows the user to search the directory. This permission must be set in order to access a file below the directory in the file system hierarchy.	No equivalent BASIC protection.

For an HP-UX file to be accessed from BASIC, all directories in the file's path must have both the read and execute permissions enabled for "others". When an HP-UX file or directory is examined with the BASIC CAT statement, it is not possible to determine whether or not its HP-UX execute permission is enabled.

Compatibility Tips

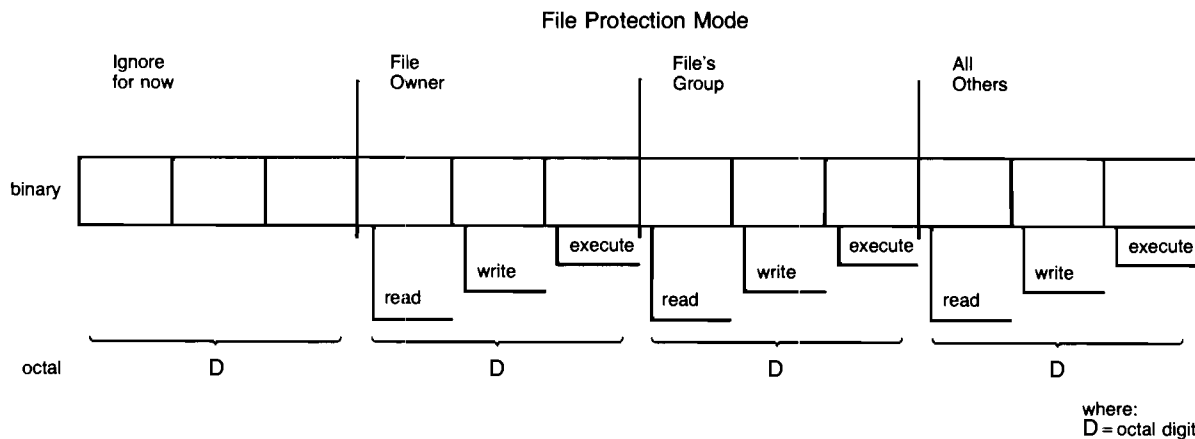
When creating files for use by both the BASIC Language System and HP-UX, assign no file protection to the file or to the directories containing the file. This ensures that both systems can access the file.

Use BASIC to back up volumes containing both BASIC Language System files and HP-UX files. This preserves the file types of all of the files.

File Protection

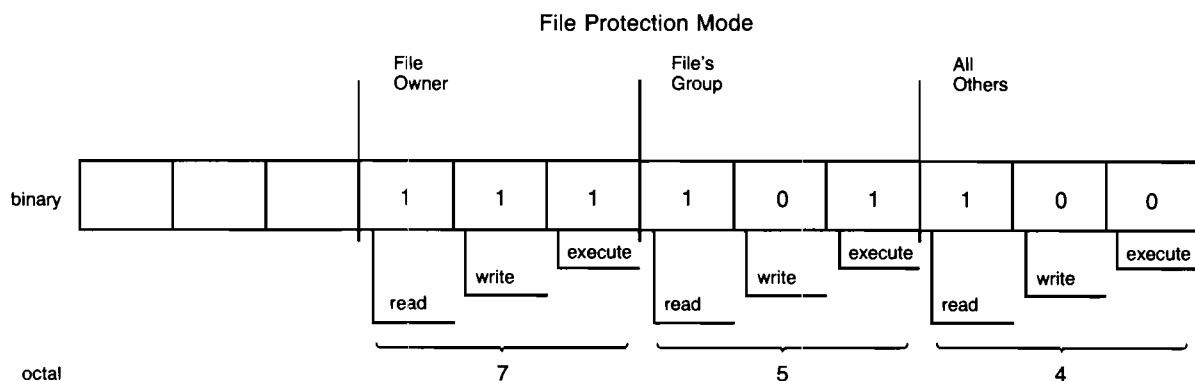
When each file in the file system is created, it is assigned a set of file protection bits, called the **mode** of the file. The mode determines which users may read from the file, write to the file or execute the program stored in the file. Read, write and execute permissions for a file can be set for the file's owner, all members of the file's group (other than the file's owner) and all other system users. The three permissions are mutually exclusive - no member of one permission group is included in any other permission group. 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 the *umask* command 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 the following diagram. 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).



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 following diagram, 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 may read and execute the file. It grants read permission to all other system users. The `ls (1)` command represents this mode as `rwxr-xr--`.



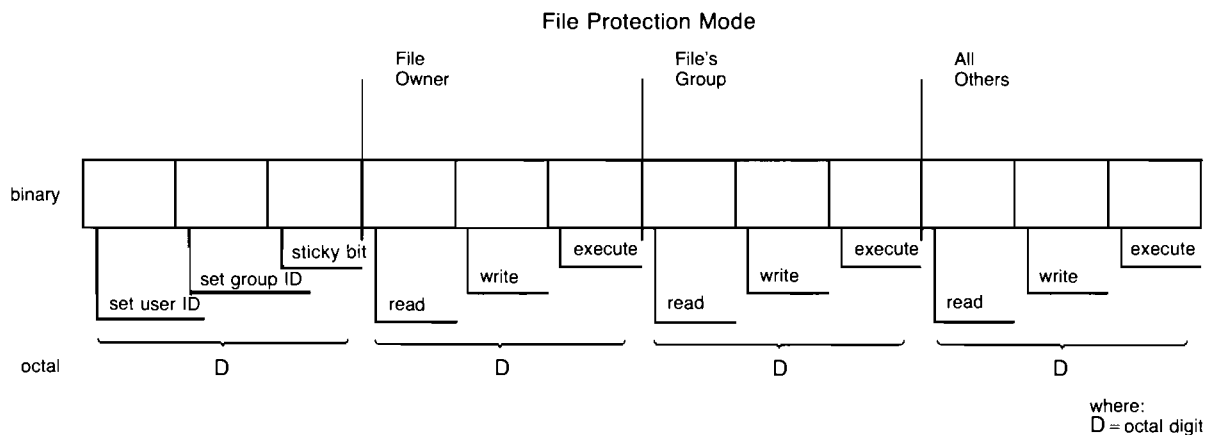
Protecting Directories

Directories, like all files in the HP-UX file system, have a mode. The directory's mode is identical to the mode of an ordinary file. 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 modify the contents of the directory itself. This capability is given 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.

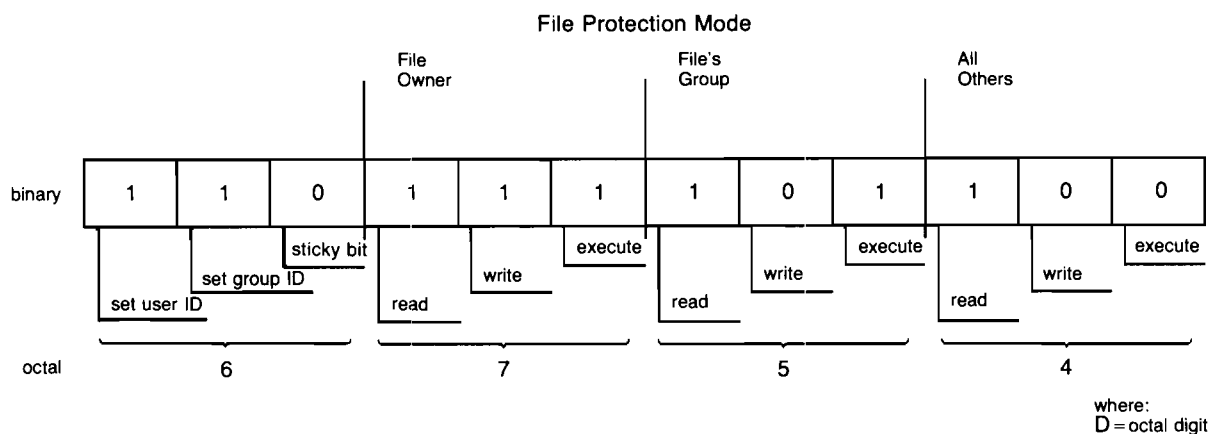
Setting Effective User and Group IDs

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).



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 the following table and diagram:

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



The Sticky Bit

Although the sticky bit can be set for all programs, **setting the sticky bit affects a program only if it is shared** (refer to 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**. When the sticky bit of a program file is set and the program is executed, the code segments of the file are loaded into the system partition (discussed with the memory management concepts later in this chapter), where they remain until you specify that they are to be removed. All users executing the file actually use the copy of the program that resides in the system partition. Setting the sticky bit reduces the amount of time needed for a number of users to access a program (since the code is transferred from disc only once). However, since the contents of the file remains in memory, the amount of memory available for other users decreases.

Once a program is in the system partition (via the sticky bit), it can only be removed by changing the file's mode such that the sticky bit is no longer set. When the file is executed again, the system recognizes the new mode and deletes the memory-resident copy of the file.

Only the super-user can set the sticky bit.

Memory Management

Your computer is equipped with a large but finite amount of Random Access Memory (RAM). It is important that you understand the manner in which this valuable system resource is used since you, in part, control its allocation through system configuration.

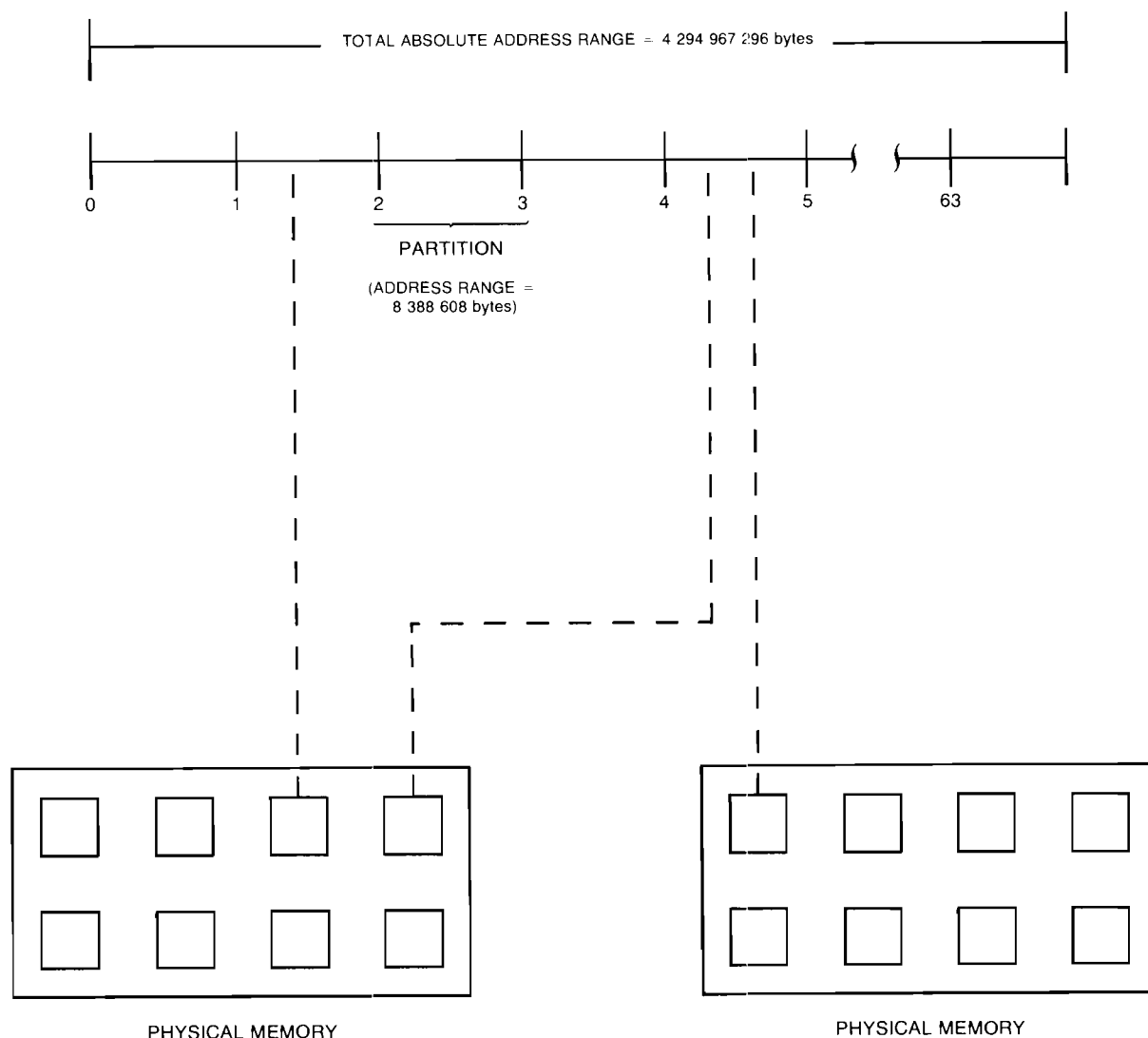
Memory Allocation

Before you consider the manner in which memory is allocated, you must understand some fundamental memory concepts. The basic concepts of memory allocation are determined by the computer's hardware architecture, not by the operating system. Thus, some terms and concepts that explain the computer's memory management are presented first. Then HP-UX's methods of memory allocation and management are presented.

Your Series 500 computer is a 32-bit computer. With 32 bits, a pointer can select (address) 4 294 967 296 different bytes of memory. Three of the 32 bits are used by the system for purposes other than addressing. Thus your computer has 29 bits with which to address memory. Thus, it can address 536 870 912 bytes (this range of addresses is called the **absolute address space**).

Although the computer can address this large amount of memory, it is equipped with a much smaller amount. The computer's memory is organized into groups of 16 Kbytes, called **memory blocks**. Each memory block can be **mapped** (assigned) to some 16 Kbyte portion of the absolute address space.

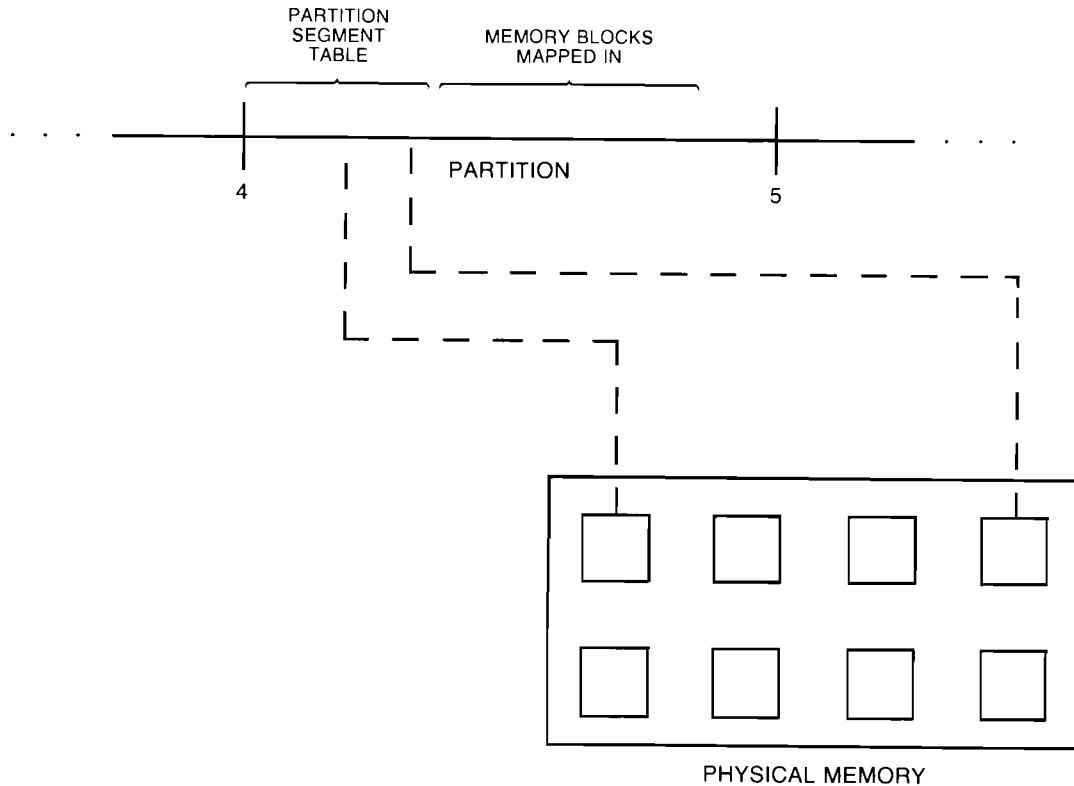
The Series 500 divides the absolute address space into sixty four 8 388 608 byte sub-ranges, called **partitions**.



A few of the partitions are used by the system, including a special partition, called the **system partition**. The system partition is used for code segments shared by many different users. The remaining partitions are available for user processes. A user process resides in the RAM that is mapped into a single partition. At most, one user process exists in a partition.

As memory blocks are needed by a process, they are mapped to the process' partition. Note that it is possible to map all of the computer's available memory into a single partition. When a block is no longer needed, it is freed for use by other partitions. Thus, the partition "grows" and "shrinks" as its need for memory changes. When a process terminates, the memory mapped to that process' partition is free to be mapped to any other partition.

At the beginning of the memory mapped into a partition is a table, called the **Segment Table**, that divides the remaining memory in the partition into segments. Each partition has only one segment table and that table describes only the memory in that partition. The only way that a user process can access the memory in its partition is through its partition's segment table. The segment table is managed by HP-UX; it is not accessible to the user process. Therefore, one user process cannot access or interfere with another user process' memory.



The system partition also has a segment table, known as the **system segment table**, that divides the system partition into segments. Any user process can access segments in the system partition through the system segment table.

Each user process uses:

- one or more **code segments (CS)** - segments containing the code that the process is executing.
- a **stack segment (SS)** - a segment that contains procedural parameters, the return addresses for procedure calls, and local variables for procedures.

The stack segment is used for variables which are allocated when a procedure is called and are de-allocated when the procedure returns. The size of this segment grows as needed by the executing program. It only shrinks during an *exec*. The stack is limited to a maximum size, which is determined by the system configuration.

- a **global data segment (GDS)** - a segment that holds the program's global data (for example, statically allocated data).

The global data segment is used for variables which are allocated when a program is loaded. These variables remain in memory for the duration of the program and are de-allocated when the program terminates. A GDS can be removed in response to memory demand.

- one or more **external data segments (EDS)**.

External data segments are general-purpose segments. They are used mainly for data. Because of attributes discussed later in this chapter, they can allow access to extremely large quantities of data.

Stack and global data are "fast access" segments. The system has mechanisms which allow the data they contain to be read and written very quickly.

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 process' partition dramatically decreases.

The term **shared code** describes user code which is loaded into the system partition. When a process executes shared code, it is directed to the copy of the code in the system partition. If the shared code is not yet loaded (no other process is currently accessing the code), the code is first loaded into the system partition before the process begins execution. Only one copy of the code exists in memory regardless of the number of processes running the program. When all processes accessing the shared code terminate, the memory in the system partition in which the shared code currently resides becomes available to satisfy requests for memory (unless the sticky bit was set on the file from which the code was loaded). The memory is not actually freed until a process requires it.

The system knows how many users 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 system partition, 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 system partition 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 the system partition, no additional memory is allocated. The new process simply executes the copy of *ed*'s code that resides in the system partition; 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, makes *ed*'s shared code segments available for use by other programs.

See *ld(1)* and *chattr(1)* for information about making programs shareable or non-shareable.

Shared Code and the Sticky Bit

Setting the sticky bit on a file containing shared code increments the shared code's use count by two instead of one when the shared code is loaded into the system partition. This keeps the use count from ever returning to zero, thus keeping the shared code segment in memory whether it is being accessed or not.

For example, suppose that the text processor program *ed* is marked “shared”. Also suppose that the sticky bit for the file in which *ed* is stored is set. When a process first executes *ed*, the shared code is loaded into the system partition and its use count is set to two. Suppose that while the first process is executing, another process executes *ed*. The new process executes the copy of *ed*’s code that resides in the system partition; the use count of the shared code is incremented from two to three. The first process now finishes editing and terminates its execution of the *ed* program. The use count of the shared code decrements from three to two as the second process continues to edit. When the second process finishes editing, it terminates execution of the *ed* program. The use count of the shared segment decrements from two to one. Note that even though no process is accessing the shared code, its use count does not decrement to zero. The shared code remains in memory; the memory is not freed for other uses.

To return the use count of a “shared”, “sticky” code segment to zero, the super-user must reprotect the file containing the shared segment such that the sticky bit is no longer set. When the program is again executed, the system recognizes that the sticky bit is no longer set and decrements the use count.

Shared Code: Benefits vs. Cost

Shared code significantly reduces the amount of memory required for process partitions when multiple processes are running the same code. There is no cost associated with sharing code unless the file containing the shared code has its sticky bit set. In this case, the code occupies memory in the system partition whether the code is being used or not. Consider making such code virtual (discussed later in this chapter) as well as shared and “sticky”. Since the number of shared code segments in the system is limited by the number of entries in the shared system segment table, overuse of sharing can result in system segment table overflow errors.

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. As mentioned previously, when a program is loaded, it is copied in its entirety into the user’s partition. If the unused code segments are a significant portion of the program, the memory allocated for that code is wasted.

With HP-UX, it is possible to mark code segments, data segments and data pages (discussed later) as **demand loadable**. When a demand loadable segment is requested to be loaded, no segments are actually loaded. Instead, a bit is set specifying that the segment is demand-loadable; the system is “fooled” into believing that the segment is actually present in memory even though it is not. No memory is allocated for the non-loaded segments until those segments are actually accessed. Only when the program attempts to access a demand-loadable segment is memory actually allocated and the segment loaded from disc.

To make code demand loadable, mark the segments that you want demand-loadable when the compiled code is linked (via the *-z* option of the *ld* command). Alternately, you can use the */bin/chatr* command to mark an existing executable file as demand-loadable. You can change a demand-loadable file such that it is no longer demand-loadable by using the *chatr* command.

Demand Load: Benefits vs. Cost

Making code segments demand-loadable reduces the amount of memory allocated for a user's partition. However, the program may actually take longer to execute than a program that is not demand-loadable if many of the demand-loadable segments are accessed.

Note that memory blocks are not allocated until the segment is actually accessed. Thus, if there is insufficient memory for the segment, an error will occur while the program is running; the program fails. If the demand-loadable attribute is not used, the lack of memory is detected when the program is loaded (because memory is allocated when a program is loaded).

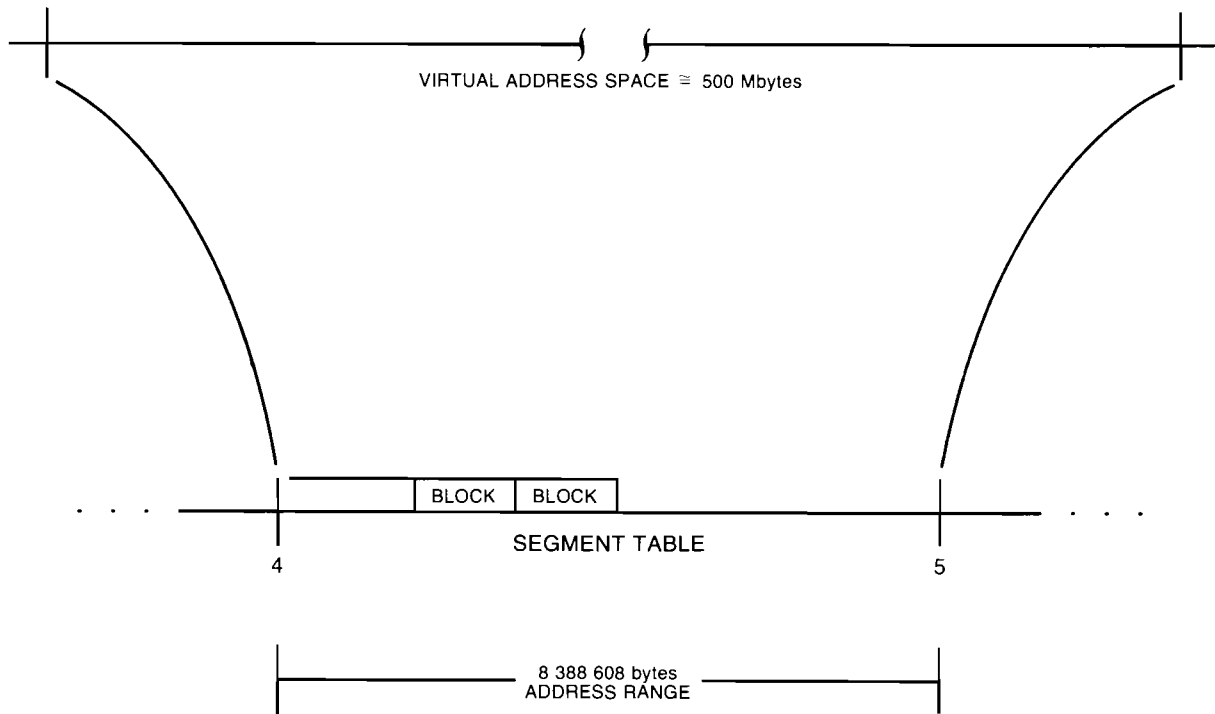
Virtual Memory

When a program executes, it typically performs one function (such as totalling an employee's hours) and passes its output on to another function within the same program (such as a wage calculating program or a paycheck program). Although the entire program is loaded into memory, only one such function is used at any one time. The remaining portions of the program sit idly by, occupying memory.

As you learned with the memory allocation concept, memory is mapped to a partition to hold the process' code and data segments. With the principles presented, if there is not enough memory available for a process' segments, the process cannot execute; it must wait until sufficient memory is free to be mapped to its partition. When large programs or programs that access large amounts of data are run, they must wait until few processes are running so that they may acquire the memory needed for their partition. Then, once running, other processes must wait for the large process to terminate before they can run (even though the large program may only be using a few segments and a small amount of memory at any given time). This also implies that no process can exceed the total memory size of the computer (since there is no way for sufficient memory to be mapped to the process' partition).

To avoid this restriction, HP-UX has implemented a memory management attribute, called **virtual memory**. Segments composing an executable file are marked virtual or non-virtual with the `/bin/chatr` command. The segments of the file are marked as virtual segments by default when the compiled objects are linked with the `ld` command. When you execute virtual code or access virtual data, the system places those segments that are currently needed into the available memory within the process' partition. If there are more segments in the object than memory available, the additional virtual segments are copied to files on the virtual memory device. You specify which device is the virtual memory device when you configure the system.

When the executing code attempts to access a segment that is not present in memory, the system attempts to map additional memory blocks to the process' partition. If the additional blocks of memory are not available, the system attempts to free memory that contains virtual segments no longer needed. These segments are "swapped out" (copied out) to files on the virtual memory device and the needed segments are "swapped in" to the newly freed memory. Because of the way that virtual memory is implemented, the "virtual address space" is much larger than the absolute address space of the process' partition.



In determining which segments to swap out, HP-UX uses the following criteria. No segments are swapped out until additional memory is needed. When memory is needed (to swap in a virtual segment being accessed), the system looks for any virtual segment that has been resident longer than a set length of time without being accessed. Segments meeting that criteria are “swapped out” until enough memory is free to load the needed segment. The set length of time referred to previously is specified with the system configuration. If no segment has been in memory without being accessed for the set length of time, the system swaps out the least recently used virtual segments to make room for the new segment(s).

Since only the segment currently needed is loaded at any one time, your programs and data may actually be larger than the physical memory within the computer. The maximum size of a virtual program is about 500 Mbytes. Similarly, the maximum size of virtual programs that utilize the system partition (such as shared code segments) is about 500 Mbytes.

Virtual Memory: Benefits vs. Cost

Making segments virtual dramatically reduces the amount of memory required by each user partition. However, the costs of virtual memory are extensive. Virtual code segments often occupy twice their original space on disc: one non-virtual copy (the original copy) and one copy of the file in the virtual address space. Additionally, each time the computer swaps a segment in or out, time is required to copy the program to/from disc. Thus the program runs more slowly. As more users utilize virtual memory, the computer must do more and more swapping.



Paged External Data Segments

A limitation of virtual memory is that all of an external data segment used by a program must be resident in memory when that segment is accessed. Suppose that you have written a program to analyze 30 Mbytes of data. (For example, finite element analysis programs and circuit analysis programs both use extremely large quantities of data.) That means that when your program runs, all 30 Mbytes of data need to be present in the process' partition. However, your system does not have enough memory to hold the data. Even if the system had 30 Mbytes of memory to map into the partition, your program would be using so much memory that other users would not be able to do anything. Typically, you only need to access a small portion of that data at any one time. The remainder of the data simply sits in memory, occupying system resources. To solve this problem, HP-UX allows virtual external data segments to be **paged**.

A **paged segment** is a segment that is divided into parts of a fixed size (such as 4096 bytes) called **pages**. The segment table, instead of storing the size and location of the entire segment (as is the case with non-paged virtual segments), stores a pointer to a page table. A **page table** is an ordered table indicating the address of each page in memory for the segment. If the page is currently on the disc, then the page table stores its disc location. The virtual memory system then moves virtual pages in and out of memory as needed, just as the virtual memory system moved entire segments in and out of memory in the previous discussion.

Not all pages of a segment have to be in memory at any given time. Thus, only a portion of a segment must be in memory at any given time, not the entire segment. This allows the segment to be larger than the size of memory, and still be accessed by the process; only the page being currently accessed must be resident in memory. Additionally, since each page is uniform in size, the virtual memory system can easily decide where to place the pages in memory. If one page is removed, then any other page can fit into its place.

The memory used for the pages of all the paged segments is allocated from a kind of memory "parking lot", called the **page frame pool**. When an executing program needs access to a paged external data segment, the system allocates the appropriate number of pages in the page frame pool and loads only that data into the pages. The memory pages are returned to a "free list" in the system's page frame pool when the process terminates.

When a new page is needed (a non-memory resident page is accessed) and there is not enough memory in the page frame pool, a page is swapped out to the virtual memory device. When a page is needed and is not resident, it is swapped in from the virtual memory device.

All paged segments in the system compete for the use of pages in the page frame pool. If no paged segments are active, then memory allocated to the page frame pool can be used for segments (for example, mapped to a process' partition) until the memory is needed by a paged segment. The maximum size of the page frame pool (and the size of each page) is determined by the system configuration. Page sizes are powers of two in the range 512 to 8192. Programs that access pages in sequential patterns usually benefit from larger page sizes, while those that access pages randomly benefit from smaller page sizes.

Pages that are currently in the page frame pool and are entered into the page table for a particular segment are known as **working set pages**. The working set pages are linked together in a per segment list by the virtual memory system. The system provides a method to guarantee that a certain number of pages (called the **guaranteed working set**) are always available for a specific segment. The system keeps track of the number of pages that are working set guaranteed for all paged segments and will not allow the sum of these pages to exceed the number of pages in the page frame pool.

All pages that are not in the working sets of allocated paged segments fall into three categories:

- **dirty pages** - pages that were removed from a working set and must be written back to the disc before they can be re-assigned to another working set.
- **used pages** - pages that were removed from a working set and were not modified since being read into memory (and thus do not have to be written back to the disc).
- **free pages** - pages that are available for use and have no valid data in them. All pages start out as free pages. If a segment is deallocated, then all the pages associated with it are placed in the free list.

All pages removed from some segment's working set are placed in either the list of used pages or the list of dirty pages, depending on whether they were modified. The dirty and used lists are maintained for the instance when a page is removed from a working set, is not written back to disc yet and then is requested again. The dirty or used page can be restored to a working set much faster than if it had to be read back from the disc.

External data segments are automatically specified to be paged when the data objects are linked via the *ld* command. Additionally, you may specify that data is to be paged with the Extended Memory System (via the *ems* system call). You can use the */lbin/chatr* command to specify that an existing paged external data segment is not to be paged.

Paging: Benefits vs. Costs

The chief benefit of paging is that it allows a process to use quantities of data that exceed the physical memory limits of the computer. The maximum amount of memory allocated to the page pool is specified by the system configuration. Memory allocated to the page pool cannot be allocated to process partition(s). If all existing pages are guaranteed to working sets, a new program attempting to use paged data fails.

A primary cost of paging data is the time required to read the file each time a page is read from disc. An additional cost of paging data is the time required to access the page. Each time a page is accessed, the process must first access the partition segment table, then access the page table (an external data segment) and then access the page (and possibly wait for it to be swapped in, if it is not already memory resident); access to paged external data segments is slower than accessing external data segments resident in the process' partition.

Combining Memory Management Tools

The memory management attributes previously discussed can be used in any combination, with some restrictions. These restrictions are:

- code segments cannot be paged.
- setting a file's sticky bit affects only shared code segments.
- global data segments are always virtual. They cannot be paged, or demand loaded.
- stack segments cannot be virtual, paged, or demand loaded.

Wise system administration calls for a mixture of the memory management attributes previously discussed. Most program files should be shared, virtual, paged, demand loadable or some combination of these. Your system is shipped with a proper mixture for the "average system". The system configuration, as shipped, is set up for the same "average system". The average system is based on a Series 500 computer equipped with 1.5 Mbytes of memory with an environment of the following four concurrently active tasks:

- compilation of a FORTRAN program.
- random and sequential read and write access to elements in a 2 Mbyte virtual array.
- execution of a computationally intensive program.
- invocation of interactive commands (such as editing or listing the contents of a file).

By changing the mixture, altering the system configuration and observing the change in system response, you can "tune" the system for the optimum response for your application mix.

The Buffer Cache

Program code and the data which it uses must be transferred from disc into physical 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. All code executed via the *exec* system call is transferred into the process' partition via a buffer called the **file system buffer cache**. Additionally, most data that is read from or written to disc is transferred through the file system buffer cache.

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 disc and physical memory. The size of each buffer and the number of buffers in the cache is specified when the system is configured. As the code and data are moved into the buffer cache, the system copies the information from the buffer cache into the user's partition.

The buffer cache can be thought of as a long tube (with sections of uniform size) that is shared by all processes. When code and data are loaded into the system, the system reads a portion of the file, equal in size to one buffer and places it in the tube. The system repeats this procedure until the entire program is transferred. When another program is loaded, it is placed into the tube behind the previously loaded program. This continues until there is not enough space in the file system buffer cache to load the next program. At that time, the least recently accessed program(s) is removed from the cache until there is enough room to load the new program.

If a user requests a program that is already in the buffer cache, the program is copied from the cache to the user's partition, eliminating the intermediate step of copying the file from disc to the buffer. The copy of the program in the buffer cache is then moved to the beginning of the cache, indicating that it is the most recently accessed program (and thus placing it last in the queue to be removed from the cache).

The primary advantage of the file system buffer cache is the speed with which large numbers of sequential reads are made to a data file. This advantage is largely due to an attribute of the file system buffer cache, called the **read ahead level**.

Read Ahead Level

The file system buffer cache has an attribute called the **read ahead level**. It is applicable primarily to users who are reading data from the disc with system calls. The read ahead level is the number of "buffers-full" of contiguous data the system should attempt to copy per disc access. For example, assume the read ahead level is four and a user is attempting to read 256 bytes of data from a file (via a system call). When the file is accessed, you would expect only the requested number of bytes to be copied into the buffer cache and then into the user's partition. However, because the read ahead level is four, the system attempts to copy enough data to fill four cache buffers. Only the requested 256 bytes is copied from the cache into the user's partition. The remaining data stays in the cache in anticipation of the user's next data read request. Since the system has to access the disc (a time consuming process) to copy the requested 256 bytes, it might as well copy some additional data (up to a maximum specified by the read ahead level) the user is likely to need.

Data accessed in this manner must be contiguous and accessed sequentially. For example, assume a user is attempting to access 256 bytes of data from a file consisting of two extents (the first extent contains 2 Kbytes, the second contains 3 Kbytes). Also assume that each cache buffer is 1 Kbyte in size and that the read ahead level is four. When the user reads the first extent to access the requested 256 bytes, the system examines the read ahead level and attempts to copy 4 Kbytes ($4 \times 1\text{K}/\text{buffer}$). However, after copying the first 2 Kbytes of data into the cache, it encounters the end of the extent and terminates the copy. Thus, only 2 Kbytes of data were copied into the buffer cache.

The File System Buffer Cache: Benefits vs. Cost

The primary benefit of the buffer cache is the increased speed with which large numbers of sequential data accesses are made. When the first of such transfers is made, the system recognizes that a sequential access has begun and attempts to read enough additional data to fill the read ahead level. Then when additional sequential accesses are made, the data is already present in memory, in the file system buffer cache. Thus less time is required to access the data after sequential access is detected.

Transferring a program from the buffer cache to a process' partition is much faster than transferring a program from disc, through the buffer cache, to a process' partition. Thus by increasing the size of the buffer cache, more programs and file data can be held in memory and the apparent system response time decreases. However, memory used by the system cache is unavailable for use in user partitions and thus decreases that precious system resource. When the file system buffer cache exceeds a certain size (approximately 50 - 100 Kbytes) system performance begins to decrease. The ideal size for the buffer cache depends on such things as the amount of sequential data access performed, the amount of physical memory installed on your computer and the speed of the disc drive.

As a rule of thumb, consider that as the size of the buffer cache increases, the probability of accessing a segment via the cache increases and the probability of accessing a segment via the disc decreases. However, as the size of the cache increases, the amount of physical memory available decreases (thus decreasing the performance of other processes).

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 either block devices or character devices. **Block devices** are devices which transmit and receive data in blocks (typically 1 block = 1024 bytes, but the size of a block is flexible). Typically, block devices are disc mass storage devices. However, a disc’s built-in cartridge tape drive (available with HP 7908, HP 7911 or HP 7912 disc drives) and other magnetic tape drives are occasionally used as block devices. **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 occasionally treated as character devices, even though they normally operate as block devices.

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 with HP-UX, use the *lsdev* command.

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.

`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 I/O slot in which the device's interface card is installed; refer to your computer's installation manual. (See the section describing the installation of interface cards for more information).

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 to specify a specific disc drive in a system with a multi-disc controller.

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.

Chapter 4

System Boot and Login: a Detailed Examination

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 the operating system, initializing the system, communicating messages to the user(s), and running scheduled routines. For you to administer your HP-UX installation 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. Through this tour, you can find features to make your task as system administrator easier. Several sample applications of important features are illustrated, including interactive examples that show how these features can be modified for your specific needs.

To help get you started, some fundamental terms are defined:

- **Boot or boot-up** - the process of searching for, loading, initializing and running the operating system.
- **Boot area** - a portion of a mass storage medium on which the operating system (such as HP BASIC Language System or the HP-UX kernel) resides. The boot area is created with a specific size when the medium is initialized. Once created, the boot area is reserved for holding operating systems; it cannot be used to store your files.

The contents of the boot area are examined with the *osck* command and are modified with the *oscp*, *osmark*, and *chsys* commands.

- **System loader** - a program that permanently resides in the computer (in ROM); it is loaded and run each time the computer is powered-up. The function of the system loader is to search the boot area of all mass storage devices present on the system and load the first "loadable" operating system found.

The boot area is marked **loadable** or **not loadable** with the */sbin/osmark* command. The system loader ignores the contents of the boot area if it is marked "not loadable"; it searches only those devices whose boot area is marked "loadable".

- **Login** - the process of gaining access to HP-UX.

- **System console** - the terminal whose location HP-UX always knows. System error messages, loader error messages, and other system communications are sent to the system console. When HP-UX is installed on a Model 520, the system console is the computer's keyboard and display. When HP-UX is installed on a Model 530/540, the system console is the terminal connected to the computer via select code 0 (if select code 0 is used for an HP 27130A Asynchronous 8-Channel Multiplex Interface, the terminal connected to port 0 of the HP 27130A is the system console).
- **CS/80** - a family of mass storage devices that communicate via the CS/80 (Command Set '80) command set. An operating system can only be booted (loaded) from a CS/80 device.

CS/80 devices currently supported on the Series 500 Computers are listed in this manual's chapter, "Installing HP-UX".

System Boot

As system administrator, you must understand the method in which the system loader determines which operating system to load. When more than one operating system is present on the system's mass storage devices, the location of the operating systems, the loadability of the operating systems (the boot area in which an operating system is stored was marked "loadable" or "not loadable" with the `/sbin/osmark` command) and the type of media on which they are stored determine which operating system is loaded.

Here is the sequence of events that occur from the moment power is switched on to the computer:

1. The computer performs a series of hardware self tests; refer to your computer's installation manual for a complete description of these tests and a guide for interpreting their results. The computer loads and runs the system loader. Then, the computer identifies and displays the current version of the system loader.

Note

Make sure that the mass storage device containing the operating system has completed its power-up cycle and is ready for use before attempting to boot; otherwise, the operating system you are attempting to load may not be found.

2. Next, the system loader searches for a "loadable" operating system; only CS/80 devices are searched. Boot areas marked "not loadable" are ignored by the system loader whether the boot area contains a valid operating system or not. (Boot areas are marked "loadable" or "not loadable" with the `/sbin/osmark` command.)

Beginning with select code 0 and continuing through the highest select code available to the computer, each removable media CS/80 mass storage device (such as a flexible disc drive or a cartridge tape drive) is searched for a "loadable" operating system. If more than one such device is found per select code, devices are searched in order of ascending bus addresses. If the devices have sub-addresses (such as unit numbers), the units are searched in order of ascending sub-addresses.

3. If no “loadable” operating system is found on a removable media mass storage device, the search pattern described in step 2 is repeated. This time, all CS/80 mass storage devices (fixed and removable media type devices) are searched for a “loadable” operating system.
4. Once a “loadable” operating system is located, the system loader displays a message (on the system console) identifying the name and version of the operating system that it is loading.

Note

If the computer cannot locate or cannot successfully load an operating system, one or more error messages are issued to the system console. These errors are listed and explained in the appendix entitled System Loader Messages. If no messages at all are displayed, suspect a hardware problem in your system. Use the hardware tests detailed in your computer's installation manual to verify that a hardware problem exists before calling your HP Customer Engineer for service.

5. Once the HP-UX Operating System has been found and loaded successfully, HP-UX initializes itself and searches for the root of the file system. The **root of the file system** is that portion of the file system that forms the base of the file system hierarchy. It is the portion of the file system below which other volumes are mounted. The file system root is important in that it contains the files required for HP-UX to properly initialize. Ideally, it should consist of the HP-UX files supplied with your system. The file system root can exist only on CS/80 mass storage devices.

A file system is identified as a file system root by marking its volume with the *rootmark* command. Only file systems marked “root” can be found when searching for the root of the file system. When you installed HP-UX, the file system placed on the destination device was marked as the root of the file system.

When searching for the file system root, the system uses the same search pattern as the system loader. Beginning with select code 0 and continuing through the highest select code available to the computer, each removable media CS/80 mass storage device (such as a flexible disc drive or a CS/80 cartridge tape drive) is searched for a file system marked “root”. If more than one such device is found per select code, devices are searched in order of ascending bus addresses. If the devices have sub-addresses (such as unit and volume numbers), the units are searched in order of ascending sub-addresses.

If no valid file system root is found on a removable media mass storage device, the search pattern previously described is repeated. This time, all CS/80 mass storage devices (fixed and removable media type devices) are searched for a file system root.

If no valid file system root is found, the system assumes that the root of the file system is located on a CS/80 mass storage device located at select code 5, bus address 0, unit number 0 and volume 0 (minor number 0x050000 - see *mknod* (8)).

6. Once the file system root is found, HP-UX sets up its first user process (the only process without a parent).

In this process, the file:

```
/etc/init
```

is executed. When *init* is executed from the first process, */etc/init 1* is implied (refer to *init(8)* in your HP-UX Reference manual).

Note

The documentation which follows describes the operation of the system as shipped to you. However, by altering the system software, any of the following procedures can change. For example, if you write your own */etc/rc* script, the paragraphs which follow are no longer valid. If you make changes to the software, document these changes and keep a copy of the documentation with this manual.

init then reads */etc/inittab*, finding all entries in this file whose state field is "1". The system retains the command fields of all state 1 entries for future execution.

7. After reading */etc/inittab* and signalling processes as required, *init* invokes */etc/rc* with three arguments: the current state, the number of times this state has been entered previously and the prior state. At power-up, the values of these arguments are 1, 0, and 0 respectively.

You should examine the */etc/rc* shell script. The comments within the script describe its actions and indicate the appropriate places to modify it. For example, the script includes a comment which states "put mounts here, if any". Add mount commands at this location to automatically mount different volumes under the HP-UX file system.

/etc/rc next sets and exports (via the *export* command) the environment variable TZ. Exporting TZ causes *rc* (and any child process of *rc*) to override the default time zone (for more information, refer to *ctime(3)* in your HP-UX Reference manual).

8. Next, */etc/rc* sets up and initializes the system console by using the */bin/stty* command to set such attributes as the baud rate, communications protocol and tab settings. The system then displays the current date and time. */etc/rc* then creates and initializes the system mount table, */etc/mnttab*. Finally, it sets the system node name.
9. The */etc/rc* shell script next executes the */etc/cron* program. *cron* executes commands at specified dates and times, according to the instructions in the file */usr/lib/crontab*. To run one or more commands on a scheduled basis, edit */usr/lib/crontab* and make one or more entries into the file. All commands executed by *crontab* run as if they were invoked by the super-user.

The system next displays the current system revision.

10. Finally, */etc/rc* declares that standard input, standard output and standard error for the *rc* process refer to the system console. Standard input is the console's keyboard, standard output is the console's display and standard error is the console's display.
11. Now that */etc/rc* has finished execution, */etc/init* executes the commands from the command field of all state 1 entries in */etc/inittab*. *init* does this by first opening the root of the file system (/) for reading on file descriptors 0, 1, and 2 (standard input, standard output, and standard error). Finally, *init* sets up a new process group and executes:

```
/bin/sh -c "exec command"
```

for each command in the command field of state 1 entries in */etc/inittab*. Typically, each command field consists of the */etc/getty* command, one for each terminal on which users are to log in. This sets up one process per terminal, the process that runs the login program and eventually runs the shell program once someone successfully logs in.

As shipped to you, */etc/getty* is invoked only for the system console. If your HP-UX Operating System is a multi-user system (HP product number HP 97080A or HP 97089A multi-user HP-UX), you need to add at least one entry to the file */etc/inittab* for each terminal on the system. You can find instructions for this in "The System Administrator's Toolbox" chapter of this manual or under *inittab* (5) in the HP-UX Reference manual.

12. 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.

When that process is terminated (when you log out), the *init* process wakes up. It identifies the terminated process and checks */etc/inittab* to see if the process is continuous. If the process is continuous, *init* again invokes */etc/getty* and repeats the procedures listed in steps 11 and 12. If the process is not continuous, the process terminates.

Login

The supplied tutorial text describes the procedures to log in (gain access to the system). This section describes the function of the operating system during that process.

1. The */etc/login* process begins when you supply a user name in response to the `login:` prompt. Once the user name has been entered, and */etc/getty* understands your terminal's defaults, *getty* executes *login* with the supplied user name; *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* again displays the prompt `login:` and waits for another user name to be entered.

3. Next *login* 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 (via the *exec* system call) whatever command is present in the command field of your */etc/passwd* entry. Typically, this is the shell, */bin/sh*. However, any command may be placed in the command field of */etc/passwd*. If no command is present in the command field, */bin/sh* is executed by default.
6. Next, the shell executes the shell script */etc/profile*. This script typically defines environment variables such as `PATH`, `TERM`, and `TZ`. Since */etc/profile* executes for each user as each user logs in and since the super-user (root) owns */etc/profile*, you (as system administrator) can modify */etc/profile* to change (and export) each user's default settings for the environment variables.
7. Additionally, */etc/profile* typically executes the *tabs* command to set the terminal's tab settings. It also lists the contents of */etc/motd* (message-of-the-day) on the display of the user logging in. It then uses *mail -e* to detect if the user has received any mail. If there is any mail, the message `You have mail.` is displayed. Finally, */etc/profile* lists to the display the names of all new files added to */usr/news* by executing *news -n*.
8. Finally, the shell executes the file *.profile*, if it exists in your home (login) directory. Typically, a *.profile* file is created by the system administrator for each user or is created by the user to customize 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.
9. Now that you have successfully logged in, the shell goes to sleep and waits for your first command.

Changes You May Want to Make

Since you now understand the method in which the operating system is loaded and initialized, and the procedures used to log in on the system, you should consider how you can modify the boot and login procedures to customize your HP-UX system. Where do you start?

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. The procedures for making the suggested modifications are detailed in the chapter entitled “The System Administrator’s Toolbox”.

1. The first “loadable” operating system found is loaded.

Changes: You can’t change the loader. However, by understanding the loader search algorithm previously described and by using supplied HP-UX commands, you can insure that the operating system you want loaded is found first (and thus is loaded). For example, you can use the *osmark* command to mark boot areas “loadable” or “not loadable” thus insuring that the desired system is found and loaded by the system loader.

The HP-UX kernel and any optional kernel software (such as optional drivers or boot area code of optional products) is shipped as ordinary files in the HP-UX file system. When the system is installed, these files are copied into the boot area of the destination device. Thus the information exists both in the boot area of the destination device and in the file system as ordinary files. The kernel BIN files (ordinary files containing boot area information) are placed in the following directories when the product is installed:

/system/970xxA - directory containing HP-UX kernel files. The characters *xx* depend on the product number of HP-UX. For example, if you ordered HP 97070 (single user HP-UX for the Series 500), the directory containing the HP-UX kernel files is */system/97070A*.

/system/xxxxxA - (where *xxxxxA* is the product number of optional software) directory containing kernel files for optional software.

The file name of kernel BIN files have one of two suffixes: *.req* or *.opt*. (Another type of file carries the suffix *.rev* and resides in directories with some of the kernel BIN files. It is discussed below.) File names ending with *.req* must be installed in the boot area for the product to be used. All files whose name ends with *.req* and exist in the directory */system/970xxA* must be installed in the boot area before HP-UX can load and run.

File names of kernel BIN files that have the suffix *.opt* identify files containing optional software. For example, several optionally installed drivers are supplied with HP-UX. The files containing the optional drivers are located in the directory */system/970xxA* and have file names with the suffix *.opt*. The name of the file identifies the peripheral associated with the driver. The drivers must be installed in the boot area before HP-UX can communicate with the associated peripheral device.

Files that reside in sub-directories of */system* and whose names have the suffix *.rev* are simply ASCII files containing a system ID string. Such files are never installed in the boot area. Typically, they are used as standard input when copying kernel BIN files into the boot area with the *oscp -m* command.

To find which kernel BIN files are installed in the boot area of a device, use the *osck -v* command. To copy a kernel BIN file into the boot area of a device, use the *oscp* command. For installations that have two different operating systems (such as the Model 520 which supports both HP-UX and the BASIC Language Operating System), you can use the *chsys* command to copy the kernel files of a system into the boot area of a device and then reboot. You should first edit the *chsys* script to specify the correct boot device and the correct kernel BIN files.

2. The first process is brought up and run; the shell script */etc/rc* is executed to run some commands.

Changes: You can place into */etc/rc* any command that you wish executed before anyone logs in. Simply edit the file. Environment variables set and exported at this time are inherited by other processes started by */etc/rc*. */etc/rc* is an ideal location for the */etc/cron* command (which runs programs at a scheduled time) since */etc/cron* should only be executed once and since */etc/rc* can distinguish between its first execution and any subsequent executions.

Other changes you may wish to make to */etc/rc* are: add one or more *mount* commands to mount other file systems before users log in, monitor and clean up logging files, and possibly run the *fsck* command to verify file system integrity (if you can afford the time for this check at each boot of HP-UX).

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

- back up the system.
- periodically execute the *sync* command (which flushes the system I/O buffers). This helps prevent corruption of the file system.
- perform file system integrity checks at more reasonable times (such as late at night when no one is using the system).
- monitor file system use and growth.

3. Execute commands from the command field of state 1 entries in */etc/inittab*.

Changes: Add one state 1 entry to */etc/inittab* for each terminal added to the system; the command field for that entry should be */etc/getty*. No other changes are recommended.

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

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

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

Changes: 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 *test_it* (a program written by your company that tests widgets). You might add an entry to */etc/passwd* of the form:

```
John::35:12:run test_it only:/users/John:/users/John/test_it
```

The name *John* is the user's login name and the values 35 and 12 are his user ID and group ID respectively. The entry *run test_it 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/test_it* is automatically run (instead of the shell). Note that an absolute path name must be specified.

The command field of */etc/passwd* is also useful for enabling a user to access information without logging onto the system. For example, a *passwd* entry with user name "who" could have */bin/who* in the command field. Supplying *who* in response to the login prompt would then cause a list of all system users (all users currently logged in) to be displayed on the terminal. The user would be logged in only for the duration of the *who* program and would be logged off when the program terminated.

6. The shell script */etc/profile* is automatically executed.

Changes: */etc/profile* allows you to force execution of one or more commands for each user when they log in, every time that they log in. This is ideal for communicating regularly with each system user and for forcing the execution of commands that each should execute at login. For example, */etc/profile* (as shipped to you) assumes that */etc/motd* contains one or more messages and sends the contents of that file to each user 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*.

Edit */etc/profile* when you want to alter its function. New commands can be added to the script; old commands can be removed or modified. As shipped to you, */etc/profile*:

- sets and exports (via the *export* command) several environment variables. Exporting the variables causes all children of the process to inherit the environment variables.
- sends the contents of */etc/motd* to each user as they log in.
- notifies each user of any mail received since they were last on the system.
- notifies each user of any news items.
- sets tabs on the user's terminal.
- sets reasonable erase and kill characters (via the *stty* command).

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

7. The shell script *.profile* is automatically executed, if it exists in the user's login directory.

Changes: Each user may have a personal version of *.profile* in his login directory: it is this personal copy that is executed once */etc/profile* has finished. *.profile* is used to execute those commands that only the user wants executed at login. For example, a user may want prompts from the shell that are different from the defaults. To change these, the user simply adds `PS1 = "new prompt"` and `PS2 = "new secondary prompt"` to (and exports them from) his *.profile*.

Edit `$HOME/.profile` (where `$HOME` is the user's login directory path name) 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 (via 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 view the files in the login directory).
- set terminal options.

Changing tab settings is not advised since many commands (such as the *vi* text editor) require the default tab settings to operate properly.

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

8. As previously mentioned, the system "comes up" executing the commands from the command field of state 1 entries in */etc/inittab*. However, once the system has come up, you can use the *init* command to take the system to another state (such as state 2 or state 5). For example, you may define one state for single user mode (only one *getty* command in */etc/inittab*) and another for multi-user mode (one *getty* command in */etc/inittab* for each login terminal).

Note

At any given time, a single user HP-UX system (HP 97070 or HP 97079) can have at most, a single open terminal and a single "incoming" uucp port. The single open terminal is the system console and is activated by a *getty* command in */etc/inittab*. The "incoming" uucp port is also activated by an entry in */etc/inittab*. (For more information, refer to "Uucp Implementation Description" in the HP-UX: Selected Articles manual and *uucp(1)* in the HP-UX Reference manual). There is no limit on the number of "outgoing" uucp ports.

At any given time, a multi user HP-UX system (HP 97080 or HP 97089) can have at most a total of 16 open ports, when, an open port is either an open terminal or an "incoming" uucp port. At least one of the open ports must be a login terminal activated by a *getty* command in the */etc/inittab* file. There is no limit on the number of "outgoing" uucp ports.

To change states, make entries in */etc/inittab* that define how you want the system to operate in its new state. For example, identify the state entry by a state number, identify the command you want executed for each state entry and list any flags that are to be considered. Once */etc/inittab* contains all of the entries you want for the new state, you may want to warn all users to log off before changing states. (Changing states while users are logged on can terminate their processes in the middle of execution if the new state explicitly kills them.) Clear all system I/O buffers by entering:

```
sync RETURN
```

Next, change states by making an entry of the form (where X is the number of the new state):

```
init X RETURN
```

When the system enters the new state, its actions are parallel to the actions described in system boot, step 5 except that the commands executed are those identified by the new state number. The system boot and login process proceed as previously described.

Summary

As you have seen, the system boot and login processes provides 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 configuration 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 the suggestions in conjunction with the suggestions listed above to determine which files to modify.

Configuration File	Description
<i>/etc/inittab</i>	Contains one state 1 entry for each terminal to be accessed as a login terminal at power up. Additionally, this file contains entries for the various states to which the system administrator may want to change the system. At any given time, a single user HP-UX system (HP 97070 or HP 97079) can have at most one open terminal and one “incoming” uucp port. The single open terminal is the system console and is activated by a <i>getty</i> command in <i>/etc/inittab</i>. The “incoming” uucp port is also activated by an entry in <i>/etc/inittab</i>. (For more information, refer to “Uucp Implementation Description” in the HP-UX: Selected Articles manual and <i>uucp(1)</i> in the HP-UX Reference manual). There is no limit on the number of “outgoing” uucp ports. At any given time, a multi user HP-UX system (HP 97080 or HP 97089) can have at most a total of 16 open ports. An open port is either an open terminal or an “incoming” uucp port. At least one of the open ports must be a login terminal activated by a <i>getty</i> command in the <i>/etc/inittab</i> file. There is no limit on the number of “outgoing” uucp ports.
<i>/etc/rc</i>	This shell script defines miscellaneous actions for each system state change. This script often contains commands to: ● mount file systems via the <i>mount</i> command. ● start the <i>cron</i> program. ● clean up any logging files. ● run the <i>fsck</i> command to verify file system integrity before using the system in the new state.
<i>/etc/passwd</i>	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.
<i>/etc/group</i>	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.
<i>/etc/motd</i>	Text file containing messages that are printed to each user when he logs in. If you have a message that you want every user to see (such as a message specifying a new system update), write the message in this file using your favorite text editor. As each user logs in, the message is printed assuming that the script <i>/etc/profile</i> is not modified to remove the command that prints <i>/etc/motd</i>.

Configuration File	Description
<i>/etc/profile</i>	Shell script that is automatically executed for each user, each time they successfully log into the shell. This is an ideal location to place commands that you require every user to execute. For example, you want every user to read the message of the day file (<i>/etc/motd</i>) since it contains information that each user should see before beginning work. This is accomplished by placing the following in this shell script: <pre>cat /etc/motd</pre> This script is also an ideal location to define and export default environment variables (such as <i>PATH</i> and <i>TZ</i>) just in case the user does not set them in his <i>.profile</i> shell script.
<i>\$HOME/.profile</i>	This shell script, when located in the user's home directory, is automatically executed each time the user successfully logs in. It is a good location to place commands that customize the user's environment and that perform functions the user wants executed each time he logs in. For example, a user's <i>.profile</i> script probably includes a definition of the default shell prompt (the <i>PS1</i> and <i>PS2</i> environment variables) and the default search path (the <i>PATH</i> environment variable). It probably also includes the execution of one or more commands (such as the <i>export</i> command to export environment variable definitions, the <i>who</i> command to identify who else is logged in on the system and the <i>mail</i> command to automatically display mail that has been received).
<i>/usr/lib/crontab</i>	This text file contains entries that identify programs that are run on a scheduled basis (at a specific time). For example, this file can contain an entry that backs up the system every night at midnight (when the system is not being used). Similarly, this file can contain an entry for automatically checking the file system integrity (and repairing any errors found) via the <i>fsck</i> command. By making entries in this file, you can execute programs at times when it is not convenient for you to be present at the keyboard (such as late at night).
<i>/etc/termcap</i>	This text file identifies terminal capabilities for programs such as the <i>vi</i> text editor. It defines terminal attributes. See <i>termcap</i> (5) in the HP-UX Reference manual for details.
<i>/etc/checklist</i>	This file is a text file that lists the default devices that the file system integrity check program (<i>fsck</i>) is to check. When no device is identified with the <i>fsck</i> command, <i>fsck</i> performs its checks on the devices listed in <i>/etc/checklist</i>.

Be sure that your configuration files are writable only by the super-user!



Chapter 5

The System Administrator's Toolbox

This chapter is designed to guide the inexperienced system administrator through his/her designated tasks. Each heading in the chapter identifies one or more tasks which you must perform. The procedure supplies at least one method for achieving the stated task. It does not cover all methods that you, as system administrator, may apply.

Adding/Moving Peripheral Devices

Adding a peripheral device to (or moving a peripheral device on) your system involves three steps:

- connect the peripheral to the computer and determine some initial information.
- determine if the driver required to communicate with the peripheral device is installed. Install the driver if it is not already installed.
- create a special file for the device. A special file is a file which defines the location of a peripheral and describes its mode of communication.

Connect the Peripheral

Connect the peripheral to your computer. The computer's installation manual 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 an address).

CAUTION

DO NOT ATTEMPT TO INSTALL AND CONNECT A CS/80 DISC DRIVE (OTHER THAN THE HP-7908P/R) YOURSELF. THE 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.

If you are connecting a supported terminal via an HP 27128A ASI Card (Asynchronous Serial Interface Card), several switches on the card must be set. Before installing the card, locate the set of eight switches on the ASI Card. Set the eight switches such that switches 1 through 7 are open and switch 8 is closed.

If you are connecting a supported terminal, you must next configure the terminal so that it can "talk" to HP-UX. The manual supplied with the terminal describes how to use the terminal's function keys to configure the terminal. First, press the appropriate key for terminal configuration; select the hardwired default values. If necessary, alter the appropriate fields such that the terminal's configuration parameters have the indicated values:

LocalEcho:	OFF
CapsLock:	OFF
InhHndShk(G):	YES
Inh DC2(H):	YES
Baud:	9600
Bits/Character:	8
Parity:	none
RecvPace:	Xon/Xoff
Full Duplex	
EnqAck:	YES

(After the terminal is installed and is operating properly, you may change the configuration parameters to suit your own needs.) If the terminal is to be used as a login terminal, change its datacomm parameters such that they agree with the */etc/inittab* entry you make for the terminal.

Next, there are several attributes of the peripheral which you must determine.

1. Identify the type of device interface (such as an HP-IB Interface or an Asynchronous Serial Interface) and its select code. The select code of the interface card is determined by the I/O slot in which it is installed (refer to your computer's installation manual for details).

If you are creating a special file for one of the Model 520's built-in devices (such as its thermal printer or Winchester disc drive), the device's select code (and other address information) is provided below:

- built-in thermal printer - select code is 6 (its bus address is 0 and its unit number is 0).
- built-in Winchester disc drive - select code is 7 (its bus address is 0 and its unit number is 0).
- built-in flexible disc drive - select code is 7 (its bus address is 0 and its unit number is 1).
- the computer's keyboard and display as system console - select code is 0 (its bus address is 0 and its unit number is 0).

Determine the bus address of the device, if it has one. Devices that connect to the computer via the HP-IB (IEEE 488-1978 interface standard) have bus addresses. The manual supplied with your peripheral can help you identify the device address and (optionally) change the device address. If the device is connected via the HP 8-Channel Multiplex Interface, determine the channel number by which the peripheral is connected.

If the device has sub-addresses (such as a unit number or a volume number), determine their values for the specific device for which you are creating a special file.

2. Determine whether you should create a **block special file** or a **character special file**. **Block special files** are used for communicating with disc mass storage devices that are to be used for mountable file systems. Block special files are also used for mounting a file system from the built-in cartridge tape drive of a supported disc drive. (Mounting a file system from a cartridge tape drive is not a recommended procedure due to excessive wear on the tape drive and the tape medium).

Character special files are used for communicating with terminals, printers, plotters, digitizers, magnetic tape drives, and, on occasion, disc mass storage. Communicating with a mass storage device (such as a disc device) via a character special file causes the system to treat the disc like a magnetic tape drive (refer to *disc(4)* in the HP-UX Reference manual).

3. Next, determine which **driver** is required to communicate with the peripheral. A **driver** is a compiled procedure that defines device-computer communication. The driver is identified by a driver number. To obtain a list of the drivers supplied with your system, enter:

```
lsdev RETURN
```

Drivers

To conserve memory, the following drivers are not installed when HP-UX is installed:

- driver for CIPER protocol printers (such as the HP 2608S).
- driver for the HP 7970 or HP 7971 9-track Tape Drive.
- driver for the HP 9885 M/S Flexible Disc Drive(s).
- driver for the HP 82901, 82902, and HP 9895 Flexible Disc Drives.
- driver for the monochromatic and high-performance color displays on the Model 520. (This driver is used only by the Device-independent Graphics Library software package.)
- driver for the standard color display on the Model 520, and for the HP 97062A Color Output Interface.
- driver for the HP 27112A GP-IO Interface Card.
- Asynchronous Serial Interface driver for raw I/O transfers (used by the HP-UX Asynchronous Communications Software).
- driver for accessing memory as a pseudo-mass storage device.

When HP-UX is installed, the drivers listed above are copied to the destination device and placed in the HP-UX file system, in the directory */system/970xxA* (where 970xxA is the product number of the HP-UX product you ordered). If you are creating a special file that needs one of these drivers, use the following command to see if it has already been installed on your system (assuming that the special file */dev/rhd* identifies the system disc):

```
osck -v /dev/rhd RETURN
```

A list identifying the segments installed in the boot area of the disc containing the HP-UX kernel is displayed. Use the information in the segment name to determine whether or not the optional driver that you need is installed. If it is not, use the following procedure to install the optional driver.

1. Determine the name of the file (in the `/system/970xxA` directory) that contains the driver you wish to install. The following list identifies the files and the drivers they contain:

`CIPERLP.opt` - this file contains the driver for CIPER protocol printers, such as the HP 2608S.

`HP7970.opt` - this file contains the driver for the HP 7970 and HP 7971 9-track Magnetic Tape Drive.

`HP9885.opt` - this file contains the driver for the HP 9885M/S Flexible Disc Drive.

`HP1B_FLEX.opt` - this file contains the driver for the HP 82901, HP 82902 and HP 9895 Flexible Disc Drives.

`SERIAL.opt` - serial interface driver for raw I/O transfers (used by the HP-UX Asynchronous Communications Software).

`HP27112A.opt` - driver for the HP 27112A GP-IO Interface Card.

`HP98770_GR.opt` - driver for the monochromatic and high-performance color displays on the Model 520 (used by Device-independent Graphics Library).

`HP97062_GR.opt` - driver for the standard color display on the Model 520, and for the HP 97062A Color Output Interface.

`MEMORY_VOL.opt` - driver for accessing memory as a pseudo-mass storage device.

2. To add an optional driver to the kernel in the boot area of the disc, type an entry of the following form:

```
oscp -a /system/970xxA/file_name /dev/rhd RETURN
```

`970xxA` is the product number of the HP-UX software that you ordered. (For example, use `97070A` if you ordered HP 97070A, HP-UX single user software for the Model 520 Computer.)

`file_name` is the name of the file containing the driver to be installed.

`/dev/rhd` is a character special file identifying the system disc.

3. Verify that the file has been copied into the boot area of the disc. Enter:

```
osck -v /dev/rhd RETURN
```

The name of the file (less the suffix, `.opt`) should be listed with the segment names that are displayed.

Repeat this procedure for each optional driver that you want to install.

Note

Installing an optional driver only copies the driver to the boot area of the disc. To access the installed driver, you must re-boot the system using *stopsys -r* (after taking steps to ensure that the loader will find the desired disc). This copies the information from the boot area of the disc into memory.

The following table identifies the drivers that are available and the devices which they control.

Driver Number	Use
1	Drives "CS-80" type mass storage devices: HP 7908P/R, HP 7911P/R, HP 7912P/R, HP 7914P/R, and HP 7933H Disc Drives (and the built-in tape drives of the HP 7908P/R, HP 7911P/R, HP 7912P/R and HP 7914P/R Disc Drives). Also drives the Model 520's built-in mass storage devices and other CS-80 devices.
6 ¹	Drives the HP 9895 8" Flexible Disc Drive.
8 ¹	Drives the HP 8290X 5 1/4" Flexible Disc Drives.
9 ¹	Drives the HP 9885M/S Flexible Disc Drives.
10 ¹	Driver to communicate with memory as if it were a disc device. Space is allocated once "memory disc" is initialized via the <i>sdfinit</i> command. See <i>sdfinit</i> (8).
11 ¹	Drives the HP 7971A and HP 7970E 9-track Tape Drives.
12	Drives HP-IB devices (except CS-80 mass storage devices and CIPER protocol printers, such as the HP 2608S printer). For example, use this driver with the HP 9872C/T Plotter, HP 9111A Data Tablet, and HP-IB printers requiring raw output.
14 ¹	Driver used for CIPER protocol printers (such as the HP 2608S).
15	Driver used by the system to talk to the NULL device. Data written to the special file with this driver is discarded. Reading a special file created with this driver returns an End Of File.
18 ¹	Driver to communicate with devices connected via the HP 27112A GP-IO Interface Card (except the HP 9885M/S Flexible disc drive - use driver 9 instead).
19 ¹	Driver to communicate with devices connected via the Asynchronous Serial Interface or 8-Channel Multiplex Interface. This driver is useful for raw transfers only and is not recommended for general use. <i>aterm</i> uses this driver. For terminals, use driver #31.
20	Special driver for <i>/dev/tty</i> file (this is its only use). More information is provided in section <i>tty</i> (4) of the HP-UX Reference manual.
22	Printer driver for formatted HP-IB output to the Model 520's built-in thermal printer and HP 2631-type printers. See <i>lp</i> (4) in the HP-UX Reference manual for more information.
26	Drives the Model 520's built-in thermal printer in raw mode. See <i>lp</i> (4) in the HP-UX Reference manual for information about raw printer mode.
28 ¹	Special driver for the Model 520's monochromatic and high-performance color displays. Used only by the HP GRAPHICS Software Package (AGP/DGL).
29	Driver for accessing the Model 520's keyboard and display as system console. It is currently used only for <i>/dev/console</i> . This driver has HP-UX terminal semantics (more information is provided in <i>tty</i> (4) of the HP-UX Reference manual).
31	Terminal driver for accessing terminals connected via the Asynchronous Serial Interface or the 8-Channel Multiplex Interface. This driver has HP-UX terminal semantics (more information is provided in <i>tty</i> (4) of the HP-UX Reference manual).
32 ¹	Driver for the Model 520's standard color display and for the HP 97062 Color Output Interface.
34 ¹	Network interface driver used for LAN 9000.

¹ Optionally installed driver.

Creating Special Files

Now that you have identified the necessary parameters, you are ready to create the special file for the peripheral. Special files are created with the *mknod* command. Alternately, you may want to consider using the *mkdev* command, which creates a standard set of special files with default names, parameters, and addresses. Before using *mkdev*, you must customize it (by editing the file) to select the special files to be created and to adjust their associated parameters. The remainder of this section teaches you how to create special files with the *mknod* command.

To create a special file with the *mknod* command, login as the super-user and make an entry of the following form:

```
mknod path_name file_type major minor
```

path_name is the pathname of the special file to be created. Select a file name for the special file which easily identifies the associated peripheral. (The section *intro(4)* in the HP-UX Reference manual describes a naming convention for special files. Using this naming convention makes your system easier to support and maintain.) Special files are usually kept in the directory */dev* for ease of “house-keeping” (keeping special files in the */dev* directory is a convention of HP-UX). However, many commands expect to find special files in */dev* and fail if the required special file is not there.

file_type is a single character; *b* specifies that the file is a block special file, *c* specifies that the file is a character special file.

major is the number of the driver used to communicate with the peripheral as described in the preceding table.

minor is a 6 digit hexadecimal value specifying the minor device. It has the form:

```
0xScAdUV
```

The meanings ascribed to the various components of *minor* are dependent on the driver and the device to which the special file will communicate. (For more information, refer to the tables that follow this discussion.) A description of the most common meanings ascribed to the components of *minor* are:

0x - specifies that the characters which follow represent hexadecimal values. These two characters (zero and x) must be entered as shown.

Sc - a two digit hexadecimal value specifying the select code of the interface card connecting the peripheral to the computer.

Ad - a two digit hexadecimal value specifying the bus address of a device. This value is also used to identify the channel number of the 8-Channel Multiplex Interface.

U - a single digit hexadecimal value specifying a secondary address, such as the unit number of the device.

V - a single digit hexadecimal value specifying a secondary address, such as the volume number in a multi-volume drive.

The meanings ascribed to the components of the minor address are determined by the device for which the special file is created and the driver used to communicate with the peripheral device. The following sets of tables identify for most supported peripheral devices:

- the product number and a description of the device. If the device is a disc drive with dual drives, a special file must be created for each drive.
- a recommended pathname for the special file (based on the naming conventions described in *intro(4)* in the HP-UX Reference manual). The supplied pathname assumes that the special file is not a raw file.
- major number (driver number).
- the required value or acceptable range of values for each component of the minor device number.

Mass Storage Devices (Disc Drives and Magnetic Tape Drives)

Product Number	Description	Suggested Pathname	Major (Driver)	Select Code	Bus Address	Unit Number	Volume Number
HP 7908P/R	16.5 Mbytes CS/80 Disc Drive	/dev/hd7908p	1	*	**	0	0
HP 7911P/R	28.1 Mbyte CS/80 Disc Drive	/dev/hd7911p	1	*	**	0	0
HP 7912P/R	65.6 Mbyte CS/80 Disc Drive	/dev/hd7912p	1	*	**	0	0
HP 7914P/R	132.1 Mbyte CS/80 Disc Drive	/dev/hd7914p	1	*	**	0	0
HP 7933H	404 Mbyte CS/80 Disc Drive	/dev/hd7933h	1	*	**	0	0
HP 7935H	404 Mbyte CS/80 Disc Drive	/dev/hd7935h	1	*	**	0	0
HP 97093A	HP 9000 Model 520's (opt. 610) built-in 10 Mbyte CS/80 Winchester Disc Drive	/dev/hd97093a	1	07	00	0	0
HP 9130K	HP 9000 Model 520's built-in 256 Kbyte CS/80 Flexible Disc Drive	/dev/fd9130k	1	07	00	1	0
HP 82901M	5 1/4" 540 Kbyte Flexible Disc Drive (dual drive, master)	/dev/fd82901m	8	*	**	**	0
HP 82902M	5 1/4" 270 Kbyte Flexible Disc Drive (single drive, master)	/dev/fd82902m	8	*	**	**	0
HP 9895A	8" 2.4 Mbyte Flexible Disc Drive (dual drive, master)	/dev/fd9895a	6	*	**	**	0

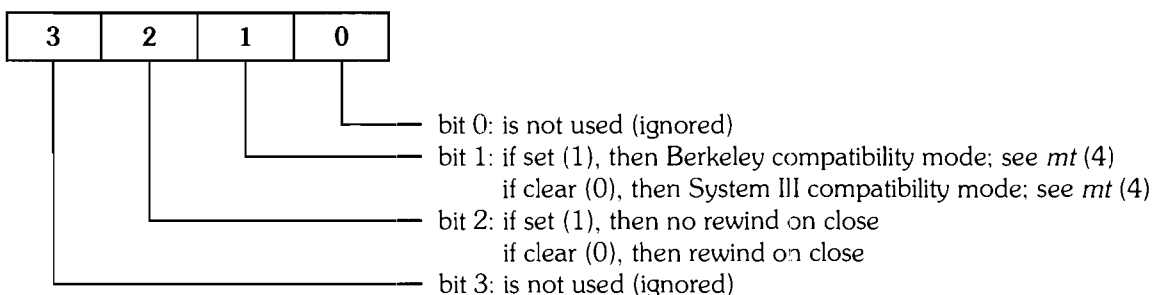
Product Number	Description	Suggested Pathname	Major (Driver)	Select Code	Bus Address	Unit Number	Volume Number
HP 9885M	8" 0.5 Mbyte Flexible Disc Drive (single drive, master)	/dev/fd9885m	9	*	00	**	0
HP 9885S	8" 0.5 Mbyte Flexible Disc Drive (single drive, slave)	/dev/fd9885s	9	*	00	**	0
HP 88140L	67 Mbyte built-in CS/80 cartridge tape drive of the HP 7911P/R, HP 7912P/R and HP 7914P/R Disc Drives.	/dev/ct88140l	1	*	**	1 (single controller drive) 0 (dual controller drive)	0
HP 88140S	16.7 Mbyte built-in CS/80 cartridge tape drive of the HP 7908P/R Disc Drive	/dev/ct88140s	1	*	**	1 (single controller drive) 0 (dual controller drive)	0
HP 7970E	9-track Tape Drive (1600 bpi, 1/2" tape, single 7970E master)	/dev/mt7970e	11	*	**	0	see below
HP 7971A	9-track Tape Drive (1600 bpi, 1/2" tape, single or dual 7970E masters)	/dev/mt7971a	11	*	**	0	see below

* The select code depends on the I/O slot of the computer in which the interface card is installed. Refer to the computer's installation manual for more information.

** Refer to the peripheral's manual for acceptable value or acceptable range of values.

The volume number field of the minor number has special meaning when creating special files for the HP 7970E or HP 7971A Tape Drives. The single hexadecimal volume number represents a four digit 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 drawing. For more information, refer to *mt(4)* in the HP-UX Reference manual.

Volume Number for a Tape Drive Special File



Printers

Product Number	Description	Suggested Pathname	Major (Driver)	Select Code	Bus Address	Unit Number	Volume Number
HP 2608S	400 line per minute, dot matrix, impact printer	/dev/lp2608s	14	*	**	1 (0 for raw mode)	see below
HP 2631B	180 character per second, dot matrix, impact printer	/dev/lp2631b	22 ¹ , 12 ¹	*	**	0	see below
HP 2631G	180 character per second, dot matrix, impact printer	/dev/lp2631g	22 ¹ or 12 ¹	*	**	0	see below
HP 97090A	HP 9000 Model 520's built-in thermal printer	/dev/lp97090a	22 ² or 26 ²	06	00	0	see below

* The select code depends on the I/O slot of the computer in which the interface card is installed. Refer to the computer's installation manual for more information.

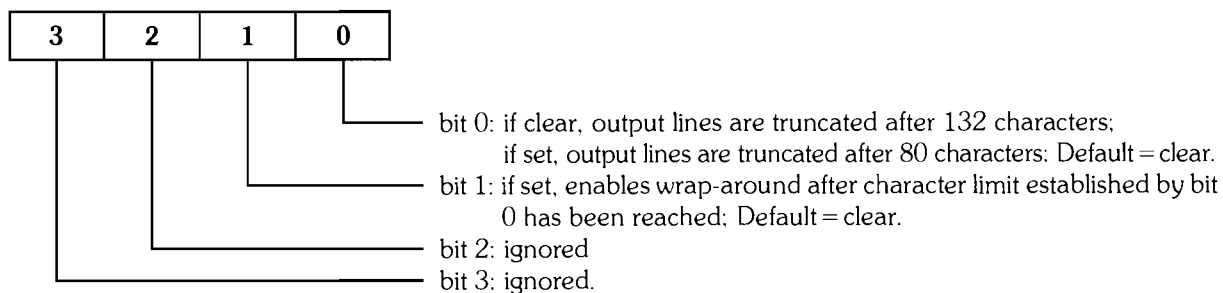
** Refer to the peripheral's manual for acceptable value or acceptable range of values.

¹ Use driver 22 when the printer is connected via HP-IB and you are using cooked (non-raw) mode. For more information, see *lp(4)*.
Use driver 12 when the printer is connected via HP-IB and you are using raw mode. For more information, see *lp(4)*.

² Use driver 22 when you are using cooked (non-raw) mode. For more information, see *lp(4)*.
Use driver 26 when you are using raw mode. For more information, see *lp(4)*.

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, two of which control the wrap-around and character-per-line characteristics of the printer. The diagram below shows and describes the bits involved. For more information, refer to *lp(4)* in the HP-UX Reference manual.

Volume Number for a Printer Special File



Note

The system requires that a special file named `/dev/lp` exist in addition to any other special files for printers that exist on the system. You can create an individual special file for your favorite printer and give it the pathname `/dev/lp`. Alternatively, you can take your favorite printer and create a link from its special file to the pathname `/dev/lp`.

Graphic Output Devices (Plotters and CRT Graphics)

Product Number	Description	Suggested Pathname	Major (Driver)	Select Code	Bus Address (Port Number)	Unit Number	Volume Number
HP 7470A	A-size, single pen plotter	<code>/dev/plt7470a</code>	12 ¹	*	**	0	0
HP 7580B	D-size, eight pen plotter	<code>/dev/plt7580b</code>	12 ¹	*	**	0	0
HP 7585B	E-size, eight pen plotter	<code>/dev/plt7585b</code>	12 ¹	*	**	0	0
HP 9872C	B-size, eight pen plotter	<code>/dev/plt9872c</code>	12 ¹	*	**	0	0
HP 9872T	B-size, eight pen plotter	<code>/dev/plt9872t</code>	12 ¹	*	**	0	0
HP 98760A	Standard color display on the HP 9000 Model 520.	<code>/dev/plt98760a</code>	32	01	00	0	0
HP 98770B	High-performance color display on the HP 9000 Model 520.	<code>/dev/plt98770b</code>	28	01	00	0	0
HP 98780B	Monocromatic display on the HP 9000 Model 520.	<code>/dev/plt98780a</code>	28	01	00	0	0
HP 97062A	Color Output Interface of the HP 9000 Model 520.	<code>/dev/plt97062a</code>	32	*	00	0	0

* The select code depends on the I/O slot of the computer in which the interface card is installed. Refer to the computer's installation manual for more information.

** Refer to the peripheral's manual for acceptable value or acceptable range of values.

¹ Use driver 12 if the plotter is connected via HP-IB.

Graphic Input Devices

Product Number	Description	Suggested Pathname	Major (Driver)	Select Code	Bus Address	Unit Number	Volume Number
HP 9111A	Data Tablet	/dev/dig9111a	12	*	**	0	0

* The select code depends on the I/O slot of the computer in which the interface card is installed. Refer to the computer's installation manual for more information.

** Refer to the peripheral's manual for acceptable value or acceptable range of values.

Memory "Discs"

Creating a special file using driver number 10 enables you to treat a portion of your computer's RAM as if it were a disc device. Once a memory "disc" has been created, it may be treated just as any disc device. It can be mounted, for instance, or used to contain a boot area. It can also be used to test procedures which access a disc frequently, enabling you to avoid disc delays during testing. One obvious difference is, since a memory "disc" is made up of RAM, it cannot survive a system power-down.

Description	Suggested Pathname	Major (Driver)	Select Code	Bus Address	Unit Number	Volume Number
Treating internal RAM as a disc device	/dev/[r]mem[0-f]	10	XX	XX	0-f	X

X's indicate a "don't care" condition — these values are ignored.

When creating a special file for a memory "disc", the select code, bus address, and volume number parameters are ignored. The hexadecimal unit number is used to create multiple memory "discs" (a maximum of 16 "discs" are allowed). For example, the command

```
mknod /dev/rmemc c 10 0x0000c0
```

creates a character special file called `/dev/rmemc` which can now be used to communicate with a portion of your computer's RAM as if it were a disc. The amount of RAM allocated to a particular memory "disc" is set by the *interleave* parameter of *sdfinit*(8). Note that memory allocated to a memory "disc" is **not** virtual, and is thus unavailable for any other use.

To de-allocate RAM from a memory "disc" and return it to the system, re-initialize the memory "disc" (using *sdfinit*(8)) with a *blocksize*, *bootsize*, and *interleave* of zero, as in

```
sdfinit /dev/rmemc 0 0 0
```

Essential Special Files

Each HP-UX installation must have the special files `/dev/null`, `/dev/console` 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; HP-UX cannot boot if a special file for the system console does not exist. The special file `/dev/tty` is a synonym for the control terminal associated with a process group. It is described in section four of the HP-UX Reference manual. When HP-UX is installed, these special files are created for you. If one or more are accidentally deleted, log in as the super-user and create the required special file with the appropriate entry:

```
mknod /dev/null c 15 0x000000 (null file or “bit bucket”)

mknod /dev/console c 29 0x000000 (system console for HP 9000 Model 520)

mknod /dev/console c 31 0x000000 (system console for HP 9000 Model 530/
540)

mknod /dev/tty c 20 0x010000 (control terminal of a process group)
```

Examples

Suppose that you want to create a special file to drive an HP 9895 Flexible Disc Drive. First, you connect the drive and determine the select code of the HP-IB interface connecting the drive to the computer (assume that the select code is 4). Next you determine the HP-IB address of the drive (assume that it is 0). Next, you log in as the super-user and check to see if the driver for the HP 9895 Disc Drive had been installed in the boot area of the system disc. To do this, you enter:

```
osck -v /dev/rhd RETURN
```

By examining the segment names listed on the display, you discover that the HP9895 driver has not yet been installed. You install the driver by entering (assume that the HP-UX software product number is 97070A):

```
oscp -a /system/97070A/HPIB_FLEX.opt /dev/rhd RETURN
```

You verify that the driver has been installed by again entering:

```
osck -v /dev/rhd
```

By examining the segment names listed, you determine that the driver is now installed in the boot area of the system disc. To access the newly installed driver, you must re-boot the system. Make sure that no one else is using the system and, using the *shutdown* command, take the system down. Once the shutdown process is completed, shut off power to the system, wait a few seconds, and then reboot the system.

After once again logging in as the super-user, you can proceed with the creation of the special file. Since you are creating a special file for a disc drive, you know the file should be a block special file. The select code and bus address of the HP 9895A are assumed to be 4 and 0, respectively. By examining the HP 9895's manual, you determine that the unit number of the left-hand drive is zero (0). Thus, to create the special file for the left-hand drive, you enter:

```
mknod /dev/fd9895a b 6 0x040000 RETURN
```

The special file for the left-hand drive of an HP 9895A is created; its name is `/dev/fd9895a`. To communicate with the right-hand drive of the HP 9895A, create another special file (all parameters are the same as the previous example except for the name of the special file and the unit number of the drive).

Terminals: Special Considerations

To create a special file for a terminal (to be used for logging in), make an entry of the form:

```
mknod /dev/ttyxx c 31 0xScAd00
```

The characters `xx` in the special file name `/dev/ttyxx` specify a numerical value to differentiate between different terminal special files (for example, `tty02` and `tty03`). The characters `Sc` identify the 2 digit hexadecimal select code of the terminal's interface. If the terminal is connected via an HP 27130A Asynchronous 8-Channel Multiplex Interface, the characters `Ad` specify the 2 digit hexadecimal port number of the interface. (If the terminal is connected via an HP 27128A Asynchronous Serial Interface Card, the characters `Ad` should both be zero.)

When a terminal to be used for login is added to the system, an entry must be made to the `/etc/inittab` file (in addition to the steps described above). This allows a user to login from the terminal. The entry into `/etc/inittab` has the form:

```
state:id:flags:/etc/getty special_file_name H 0
```

The use of the first three fields (represented here as `state`, `id`, and `flags`) is discussed both in `inittab(5)` in the HP-UX Reference manual and in this manual's chapter entitled "System Boot and Login: a Detailed Examination". The normal values for these fields are: `state = 1`, `id =` a unique two character value, and `flags` should include the `c` flag. The `c` flag specifies that the command in the command field (in this case, `/etc/getty`) is to be re-invoked each time the process terminates (such as at logout).

The fourth field must contain the `/etc/getty` command; it is immediately followed by three parameters. The first parameter, represented by `special_file_name`, is the file name (not the pathname) of the terminal's character special file. The named file must reside in the `/dev` directory. The other two parameters define the type of terminal as an HP terminal (represented by `H`) and specify a zero delay time (represented by `0`); a non-zero value is relevant only for dial-up modem connections. For more information, refer to the section `getty(8)` in the HP-UX Reference manual.

HP terminals, as delineated by the `H` parameter, should be configured to have the following characteristics:

LocalEcho:	OFF
CapsLock:	OFF
InhHndShk(G):	YES
INh DC2(H):	YES
Baud:	9600
Bits/Character:	8
Parity:	none
RecvPace:	Xon/Xoff
Enq/Ack	YES

Printers: Special Considerations

When using an HP 2631 Printer, you must align the perforations of the printer paper with the top of the printer head (not at the indicated location on the printer). Aligning the paper perforations with the location specified on the device causes occasional unexpected formfeeds to occur.

Note

It is wise to set the mode of the */dev* directory such that it has permissions 755 or 555 octal. This prevents a user from accidentally creating an ordinary file in */dev* when he/she tries to output data to a device file that doesn't exist.

Adding/Removing Users

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

Procedure

1. Log in to the system as the super-user.
2. Modify the file */etc/passwd*. However, if someone attempts to login or change their password while you are modifying the file, unexpected results can occur. Therefore, you may want to copy the existing */etc/passwd* file to another file, such as */etc/newpasswd*. Then edit */etc/newpasswd*. When all changes to the file have been made, move the */etc/newpasswd* file to */etc/passwd* with the *mv* command (this overwrites the existing */etc/passwd* file). Alternately, you can modify */etc/passwd* when you are certain that no one else is using the system.

Using the text editor of your choice (such as *ed*, *vi*, or *ex*), edit the file */etc/passwd*. Add a line describing the new user to the file. The new line should have the form:

```
user_name::user_id:group_id:comment:login_directory:command
```

Each field in the line is separated from the next by a colon.

user_name is the user's login name, consisting of 1 to 8 lowercase characters.

:: is the password field. Since the delimiting colons have nothing between them, this field is empty, so that user *user_name* has no password initially. This field will be filled in later with the user's encrypted password when he creates one for himself using the *passwd* command.

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. By convention, the values 1 through 99 are reserved for system use.

There should be only one entry per user ID; the one user whose 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*.

comment is a word or phrase that uniquely identifies the user or specifies the reason for the entry. Typically, this field holds the user's full name and optionally, his location.

`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.

`command` is the name of a single command to be executed for the user at login. Typically, `/bin/sh` is placed in this field as it invokes the shell for the user. However, the name of any executable program may be placed in this field. No arguments can be supplied with the command. The command must be a compiled program, not a shell script. If the command field is left blank, the command `/bin/sh` is executed by default. When the user logs in, the command listed in this field is executed and control is passed to that program. Once the program terminates, the user is logged out.

Once you are satisfied with the contents of `/etc/passwd`, write the modified file to the disc and terminate the edit session.

3. Create a login directory for the user with the `mkdir` command. The path name of this login directory must match the login directory name supplied in `/etc/passwd`.
4. As an optional step, create a `.profile` file for the user. If a file named `.profile` exists in a user's login directory, the shell attempts to execute that file during initialization. Typically, this file contains a set of shell commands and environment variable definitions which customize the user's environment and/or automatically run one or more application programs.
5. Since you created the user's login directory and `.profile`, you own these files. Use the `chown` and `chgrp` commands to change the ownership and group association of these files from you and your group to the user and his/her group (if any). Note that the user's group must exist in `/etc/group` and the user must be a member of the group before using the `chgrp` command to change the files' group.

To add the user to a group, refer to the section "Creating a Group and Changing Group Membership" presented later in this chapter.

6. Use the `chmod` command to change the protection mode of the user's login directory and `.profile` to an appropriate value, such as 0755. The mode 0755 provides read, write, and execute permission for the file's owner while providing only read and execute permission for all others.

The user may now log in. You should encourage him to create a password for himself with the `passwd` command.

To remove a user from the system. simply delete his entry from the `/etc/passwd` file and from the `/etc/group` file. Then remove the user's files to free valuable disc space.



Initializing Media

The Series 500 computers recognize SDF (Structured Directory Format) on initialized discs and tape. The `sdfinit(8)` command is used to initialize media in this format.

The `sdfinit` command performs five functions:

- It initializes your media in SDF, which is recognized by all HP 9000 Series 500 computers.
- It optionally certifies your media. Certification verifies that the media accurately reproduces the data written on it. If any portion of the media fails certification, that portion is “spared” - that is, excluded from any further I/O operations;

The certification process can be **disabled** by specifying the `-i` option on the `sdfinit` command line. Note that use of the `-i` option is strongly discouraged. Although skipping the certification step saves time, you are left with initialized media whose reliability is unsure at best.

- It selects an *interleave factor* for the media. The interleave factor is an integer specifying how many (if any) sectors are skipped each time an I/O operation occurs (a *sector* is the smallest amount of media space that can be read or written at one time).

For example, an interleave factor of two (2) causes sectors 1, 3, 5, 7, ... to be written on or read from. When the end of the track is reached, the media wraps around and begins using sectors 2, 4, 6, and so on. Thus, two passes are required to fill the media. If an interleave factor of four (4) is specified, the media uses sectors 1, 5, 9, 13, ... on the first pass, sectors 2, 6, 10, 14 ... on the second pass, sectors 3, 7, 11, 15, ... on the third pass, and finally sectors 4, 8, 12, 16, ... on the final pass.

The purpose of the interleave factor is to facilitate maximum data transfer to/from the media. Faster devices (such as hard discs) do well with an interleave factor of one (i.e. no sectors are skipped). This puts data closer together on the media, enabling data to be provided as fast as the device can read it. Slower devices (such as external flexible disc drives) need a larger interleave factor to provide data slowly enough for the device to process it.

If no interleave factor is specified, `sdfinit` assigns a default of one. This works well with most media. The exceptions are the internal 10 megabyte Winchester disc available on the Model 520, which has a recommended interleave factor of four, and external flexible disc drives. Note that the built-in flexible disc drive on the Model 520 has been optimized such that an interleave factor of one is appropriate.

- It assigns a *block size* to be used in data transfers to/from the media. A block size is an integer specifying the number of bytes per block of data. When data is then written to or read from the media, it is saved up until one block's worth of data is accumulated, and then the entire block is transferred.

The block size is used to simultaneously determine the amount of media space used, and the speed at which data transfers are made. Smaller block sizes save media space but cause inefficient data transfers. Larger block sizes cause data to be transferred more efficiently and quickly, but waste media space.

To illustrate the block size's effect on media space, you need to understand how a block size is used by a particular medium. Whenever a block size is defined, the medium for which it is defined automatically allocates one block for every file yet to be created on that medium. Thus, if the block size is 2048 bytes, for example, a file containing 32 bytes is still allocated 2048 bytes, leaving 2016 bytes that are unused yet unavailable for any other use. If the file happens to grow such that it now contains 2049 bytes, another 2048-byte block is allocated for it, leaving 2047 bytes unused and therefore wasted. When a medium becomes riddled with these chunks of unused space, it is said to be *internally fragmented*.

Large block sizes, however, permit much more efficient data transfer. For example, consider the two block sizes, 256 bytes and 2048 bytes, which are to be used to read a 1 megabyte file. If the medium on which the file resides was initialized with a 256-byte block size, it would take 3907 read operations from the medium before the entire file is read. With the 2048-byte block size, however, only 489 read operations are needed. You can see the larger block size greatly reduces the number of device accesses needed, thereby reducing wear and tear on the device. Also, the data can be transferred in much less time.

Obviously, a compromise must be made. The recommended block size for hard discs is 1024 bytes. For flexible discs, block sizes of 256 bytes or 512 bytes work well. Depending on the application, block sizes should not exceed 4096 bytes (sometimes useful for serial data access from hard discs). If the block size you specify is not a power of two, *sdfinit* rounds it up to the next power of two. If you do not specify a block size, *sdfinit* uses a default block size of 512 bytes.

One final point to keep in mind involves the *cache buffer size* (set by the *uconfig(8)* command) and its relationship to the block size. You should never set the cache buffer size smaller than the largest block size you are supporting on your system. For example, if you have a hard disc with a block size of 2048 bytes, a hard disc with a block size of 1024 bytes, and two flexible disc drives with block sizes of 256 and 512 bytes, respectively, your cache buffer size should be at least 2048 bytes. If in this example the cache buffer size is 1024 bytes, the number of cache buffers would effectively be halved, degrading system performance. If the cache buffer size is 4096 bytes, half of every cache buffer would be wasted.

- It optionally creates a *boot area* on the medium being initialized. A boot area is a section of the volume which is set apart to contain the various code segments needed to boot up HP-UX and any optional products you may have purchased. If you want to use a particular medium as the boot-up device for your system, then specify the boot area size, in bytes, on the *sdfinit* command line. A minimum boot area size of 500,000 bytes is recommended. If you omit the boot area size, *sdfinit* uses a default size of zero bytes (no boot area is created).

When is it appropriate to use *sdfinit*, and on what devices? You should always use *sdfinit* on hard discs (HP 7908, 7911, 7912, 7914, 7933, 7935). Do not use *sdfinit* on nine-track magnetic tape.

For pre-certified cartridge tapes, do not use *sdfinit* if you are using the cartridge tape with *backup(8)*, *tcio(1)*, or other raw applications (i.e. those applications using character special files). If you want a boot area on the pre-certified cartridge tape, you should use *sdfinit* (with the *-i* option to inhibit certification).

For cartridge tapes that have not been pre-certified, *sdfinit* should be used to certify them (i.e. do **not** use the **-i** option).

A helpful hint: if you have any medium that you would like to certify, but you do not want the Structured Directory Format written on the medium, specify a *bootsize* which is larger than the total capacity of the medium. For example, for a DC600 cartridge tape, specify a *bootsize* greater than 67 megabytes. *Sdfinit* completes the certification process, and then aborts when it determines that the specified *bootsize* is too large for the medium.

Examples

```
sdfinit /dev/hd 1024 1000000
```

This example initializes the hard disc addressed by `/dev/hd` with a block size of 1024 bytes, and a boot area of 1 megabyte. The default interleave factor of one is selected.

```
sdfinit /dev/rfd,0 256 0 3
```

This example initializes the flexible disc addressed by `/dev/rfd,0` with a block size of 256 bytes, a boot area size of zero, and an interleave factor of three.

```
sdfinit /dev/ct881401 512 100000000
```

This example certifies the DC600 cartridge tape addressed by `/dev/ct881401`. *Sdfinit* aborts after certification because the specified boot area size (100 megabytes) is larger than the capacity of the cartridge tape.

Backing Up and Restoring the File System

One of your responsibilities as system administrator is to make periodic backups (copies) of the file system. You should also make at least one backup of the current configuration of the operating system, programming languages and options. To minimize the chance of loss, these backups should be stored at a different location from the system. "Data safes", specially designed air-tight, water-proof containers for mass storage media, are available from many computer accessory manufacturers. In the event that a file is lost or that the entire system is lost, you can easily recover by restoring the latest version of your system backup.

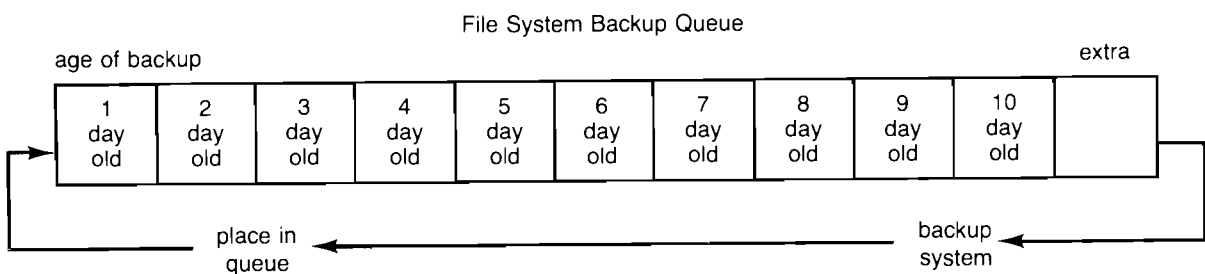
The method, frequency and extent of a backup depends on the amount of use the system receives and the amount of data you feel you can afford to lose. There are different philosophies dealing with the method and frequency of backups. The ideal method of backup would be to back up the entire system several times a day, keeping each backup permanently. Then, in case one or more files are lost (regardless of when they are discovered missing), you could restore the system to a point just prior to when the file(s) was lost. The only data lost would be that data entered since the last backup.

However, performing a backup requires substantial use of system resources. Additionally, the cost for backup media can be significant.

Some users may find it cost effective to back up (archive) the entire file system once every week, supplementing this with a daily incremental backup. An **incremental backup** is one that is comprised only of files that have changed since the last complete file system archive. These can be easily identified with the *find* command. Incremental backups usually require fewer system resources and less backup media than complete daily file system backups.

For example, suppose that you make a backup of the entire file system on Monday and make incremental backups Tuesday and Wednesday. Each incremental backup encompasses only those files that have changed since Monday. If only a few files have changed, the incremental backups may take only minutes. Then Thursday, the file system is destroyed. You may reconstruct the file system by first restoring the Monday backup of the file system, and then restoring the files from the Wednesday incremental backup. As you can see, reconstructing the file system from several backups is more time consuming than restoring a single backup of the entire file system; however, less time was involved making those backups in the first place. Pick the method and extent of backup that makes the most sense for your installation.

HP recommends that you make daily backups of the file system. Keep the backup at least two weeks (10 working days for most facilities). At the end of two weeks, the "oldest" backup media can be re-used to make another backup. By keeping a "queue" of backup media, each new backup can be added to the front of the queue. The backup medium needed for the backup is taken from the back of the queue. This limits the expense of backup media while providing protection against loss of data during the previous two weeks. Additionally, once every month you should take one of the complete file system backups created above and place it in a permanent archive.



System Backup and Restoration Procedures

HP-UX can be backed up in any one of a number of ways:

- interactively use the *cpio* or *tar* commands to copy selected files to a backup volume.
- write your own backup program or shell script using *cpio*, *tar*, or *cp*.
- use the supplied *backup* command to create a complete file system backup (also called an archive) or an incremental backup (also called a delta).

The remainder of this section provides a tutorial for using the supplied *backup* command. Before this command is run the first time, ensure that a character special file (raw mode) named */dev/rmt79xx* exists.

The optimum scheme for backing up your system with the *backup* command is presented in the following procedure.

Note

Before the *backup* command is used, you must modify (by editing) the command. Consider modifying the command's parameters to customize it for your installation (see *backup(8)* in the HP-UX Reference manual for details). For example, you must modify the backup command to dictate whether the backup is to be created with only a *cpio* command or by piping the output of *cpio* through *tcio* to a raw special file. This latter method is preferred when the backup device is the built-in cartridge tape drive of a CS/80 mass storage device. If the *tcio* command is not used when backing up to the built-in cartridge tape drive of a CS/80 mass storage device, damage to the drive may result with repeated use.

/etc/backup assumes that the name of the special file for the backup is */dev/79xx*. */etc/backup* is set up initially for backing up to the built-in cartridge tape drive of a CS/80 mass storage device.

Make sure that you edit and examine */etc/backup* before using it. Otherwise, unexpected results may occur!

1. Log in as the super-user. If you are not the super-user, you will have problems copying files that you do not own or have permission to access. Similarly, when restoring files, the ownership of the files is changed when the backup is restored.
2. Create a complete file system backup (archive) by entering:

```
/etc/backup -archive RETURN
```

This copies the entire file system to the backup device identified by */dev/rmt79xx*. The file */etc/archivedate* is "touched" (with the *touch* command) to remember the time and date of this backup.

backup logs 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. This file continues to grow with each backup. Use this file to help you determine when a new archive is needed. By keeping an eye on the size of the backup, you can estimate when it is necessary to create a new archive.

3. When it's time to perform the next backup, create an incremental backup (delta) by entering:

```
/etc/backup RETURN
```

backup determines the date of the last archive (by examining */etc/archivedate*) and backs up all files that have been modified since that time and date.

As before, *backup* logs 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.

4. Create incremental backups until their size becomes unwieldy (for example, larger than one tape) or until you otherwise feel it is necessary to create a new archive.
5. When it's time to create a new backup of the entire file system, repeat step 1 above.

If at any time the *backup* command runs out of room on the backup medium, you are prompted to insert a new volume (unless *backup* was invoked by *cron* (8), in which case it simply terminates).

Restoring a Backup

Before you can restore a backup, you must know how the backup was made. (When you customized the *backup* command, you specified how the backup was created: only with the *cpio* command, or with the output of the *cpio* command piped to the *tcio* command.) If the backup was created with only the *cpio* command, you restore the backup by using the *cpio* command and its *-i* option, *-c* option and *-B* option (include the *-x* option if you want special files restored; include the *-v* option for additional feedback during the restoration). If the backup was created with both *cpio* and *tcio*, you restore the file system by piping the output of *tcio -i* to *cpio -ic* (again, include the *-x* option with *cpio* if you want special files restored; include the *-v* option for additional feedback during the restoration.)

To restore the file system, log in as the super-user and place the backup medium in the mass storage device. Enter a command of the following form (select the appropriate command form based on the information in the previous paragraph):

```
cpio -icxvB < special_file
```

or

```
tcio -i raw_special_file | cpio -icxv
```

If the entire file system is lost (or enough of the system is lost such that *cpio -i* cannot function), you must first re-install the system from the original supplied media before restoring the file system. Use the *coreinstall* command (or optionally, the *sysinstall* command) with the original HP-UX media; instructions are provided in this manual's chapter "Installing HP-UX". This installs the HP-UX kernel in the boot area of the system disc and places the original HP-UX file system on the disc. Use the *osck* and *oscp* commands to re-install any HP-UX optional boot area code (such as optionally installed drivers). If you re-installed the system with the *coreinstall* command, use the *optinstall* command to re-install any optional products. (Again use the *osck* and *oscp* commands to install any optional boot area code in the boot area of the system disc.)

Once the boot area of the system disc is restored, you can use that system to restore the “lost” system from the most recent archive and the most recent incremental backup. There should be no need to re-install any updates to HP-UX. These should be included in the archive. Once you have an operational system, you can use its version of *cpio -i* and *tcio -i* to copy in the backups.

Note

If you are recovering from a total loss of the file system, be careful when restoring the most recent backups. Since the backup is a copy of the system, it may contain the same error that caused the system to “crash” in the first place.

Special Considerations or “What if ... ?”

If you are in the vicinity of the backup device, you can determine when the end of media is reached because the backup device stops. If you are using a tape drive for the backup device, the tape may or may not be rewound at completion of the backup. This attribute is controlled by the control bits in the magnetic tape drive’s special file (refer to this manual section “Adding/Moving Peripheral Devices”). When this occurs, change the media and enter the following:

```
/dev/special_file_name RETURN
```

If you are running the backup from *cron*, it may be difficult to detect when the end of media is reached, since no message is displayed on the terminal. Should this occur, your backup media contains only a partial backup. (You are not notified to change media. Since backup is being run by *cron*, it is probably running when no one is around, thus there is a good chance that no one would be around to change the media anyway.) Examine */etc/backuplog* to determine if and when an error has occurred.

If you are restoring files from a tape, Bell System III semantics are selected (see *mt* (4)), and there is no auto rewind capability enabled, pressing the **DEL** key causes the tape to seek EOF (End Of File). The restore process suspends until EOF is found or the drive is reset.

Archives are typically done interactively from a terminal. However, incremental backups (deltas) are often performed by placing the *backup* command in the */usr/lib/crontab* file. If you wish to perform incremental backups automatically, add an entry to */usr/lib/crontab*. For example, one such entry might be:

```
0 0 * * * /etc/backup -fsck
```

This causes an incremental backup to be performed each night at midnight. The *-fsck* option causes the file system to be verified after the backup. The results of the file system check are written to */dev/lp*. (This can be changed by editing the *backup* script.)

In general, the system cannot write to a busy file (for example, if the file is open or is sharable and is being executed via *exec*). Thus, files being used during a recovery (in particular, */bin/cpio*) should not be sharable. If they are sharable, they may not be recovered from the backup.

Booting HP-UX

When your computer is powered up, it performs a series of hardware self tests and then loads and runs the system loader. The system loader is a program that permanently resides in the computer. Its function is to search the boot area of all CS/80 mass storage devices and load the first "loadable" operating system found. The mass storage device from which the operating system is loaded is called the **boot device**. The mass storage device containing the root of the file system (the portion of the file system that forms the base of the file system hierarchy) is called the **root device**.

Several HP Computers support more than one operating system (for example, the Model 520 supports both the BASIC Language Operating System and the HP-UX Operating System). To ensure that the proper operating system is loaded, you must understand:

- the difference between a "loadable" and a "not loadable" system (and how to mark a system "loadable" or "not loadable").
- the manner in which the loader searches for a "loadable" operating system.
- the manner in which the contents of the boot area of a mass storage device is examined and altered.

Making Operating Systems "Loadable" or "Not Loadable"

The contents of a mass storage device's boot area is marked "loadable" or "not loadable" with the `/sbin/osmark` command. The system loader ignores the contents of the boot area if it is marked "not loadable"; it considers loading only those operating system that are marked "loadable".

Searching for an Operating System

When the computer is powered-up, the system loader begins searching the boot areas of all removable media CS/80 mass storage devices (such as flexible disc drives) for a "loadable" operating system. It begins searching devices at the lowest select code available and continues through devices located at the highest select code. If more than one removable media mass storage device is present at a particular select code, the search order is determined by bus address. The search begins at the lowest bus address and continues through the highest bus address.

If no operating system is found on a removable media mass storage device, the system loader repeats the search pattern described above, this time checking all mass storage devices, not just removable media type devices.

If for any reason, the system loader is unable to find and load a "loadable" operating system, one or more messages are displayed on the system console. These messages are described in the appendix entitled "System Loader Messages".

Changing the Contents of the Boot Area

The HP-UX kernel and any optional kernel BIN (such as optional drivers or boot area code of optional products) is shipped as ordinary files in the HP-UX file system. When the system is installed, these files are copied into the boot area of the destination device. Thus the information exists both in the boot area of the destination device and the file system as ordinary files. The kernel BIN files (ordinary files containing boot area information) are placed in the following directories when the product is installed:

`/system/970xxA` - directory containing HP-UX kernel files. The characters `xx` depend on the product number of HP-UX. For example, if you ordered HP 97070A (single user HP-UX software for the Model 520 Computer), the directory containing the HP-UX kernel files is `/system/97070A`. This directory is located on the root device.

`/system/xxxxxA` - (where `xxxxA` is the product number of optional software) directory containing kernel files for optional software. Such directories are located on the root device.

The file names of kernel BIN files have one of two suffixes: `.req` or `.opt`. File names ending with `.req` must be installed in the boot area of the boot device for the product to be used. All files whose name ends with `.req` and exist in the directory `/system/970xxA` must be installed in the boot area before HP-UX can load and run (This is true only for `xx = 70, 79, 80, 89`).

File names of kernel BIN files that have the suffix `.opt` identify files containing optional software. For example, several optionally installed drivers are supplied with HP-UX. The files containing the optional drivers are located in the directory `/system/970xxA` and have file names with the suffix `.opt`. The name of the file identifies the peripheral(s) associated with the driver. The drivers must be in the boot area of the boot device (and have been loaded by the system loader) before HP-UX can communicate with the associated peripheral device.

Another type of file that resides in directories with the kernel BIN files has the suffix `.rev`. They are ASCII files containing a system ID string. Such files are never installed in a boot area. Typically, they are used as standard input when copying kernel BIN files into a boot area with the `oscp -m` command.

Examining and Changing the Content of a Boot Area

To find which kernel BIN files are installed in the boot area of a device, use the `osck -v` command.

To copy a kernel BIN file into a boot area of a device, use the `oscp` command. You can replace the current contents of a boot area with the `-m` option of the `oscp` command. You can append a kernel BIN file to the current contents of a boot area with the `-a` option of the `oscp` command. You can copy the contents of one device's boot area to the boot area of another device with the `oscp` command (no options).

For installations that have two different operating systems (such as the Model 520 Computer which supports both HP-UX and the BASIC Language Operating System), you can use the *chsys* command to copy the kernel files of a system into the boot area of a device and then reboot. You should first edit the *chsys* script to specify the correct boot device and the correct kernel BIN files.

Changes made to a boot area of a device do not take effect until the system is rebooted (and the altered contents are loaded by the system loader).

Boot Procedure

To load the HP-UX Operating System:

1. Turn off power to your computer.
2. If HP-UX is installed on an external mass storage device (any device other than the computer's built-in mass storage devices) power up that mass storage device.
3. Power up all other peripheral devices, except those mass storage devices containing an operating system that the loader could find before HP-UX. Use the search pattern described earlier to determine which operating system the system loader will find and load. Any disc that is not powered-up before system boot can only be mounted after you login. This implies that the disc cannot be mounted automatically from */etc/rc*.

Some disc drives require several minutes to complete their power-up sequences. Refer to the manual supplied with the disc drive to determine the exact time requirement. Wait for these devices to complete their power-up sequence before continuing.

4. Power up the computer. The following prompt is displayed when HP-UX has finished loading:

login:

5. Power up any peripheral not yet powered-up.
6. If */etc/rc* does not include the execution of *fsck*, the file system consistency check program, consider manually executing this command before allowing other users to access the system. This process may take 30 minutes or longer to complete.

On small systems, it is often desirable to allow any user to power up the system. You should write a short document that describes in detail the procedure for loading HP-UX and distribute the document to all users. As system administrator you know the locations of all other operating systems stored on your system. Thus, you should be able to eliminate the search pattern description earlier; this leads to a concise, easy-to-follow procedure.

Changing a Password

Users often create passwords for themselves which they cannot remember. Once a user has forgotten his/her password, he/she cannot log in. Since only the encrypted form of the password exists in */etc/passwd*, even you cannot determine the user's password. This situation can be corrected in one of two ways.

First, you can edit */etc/passwd* and remove the existing characters from the password field. Then, with the */bin/passwd* command, you or the user can assign a new password (hopefully, one that is more easily remembered).

Alternately, you can enter:

```
/bin/passwd user_name RETURN
```

where *user_name* is the user's login name. The old password is removed and the system prompts you to enter the new password. Only the super-user may use either of these methods for changing a password.

To protect the security of the system, */etc/passwd* should be owned by root (the super-user) and should be writable by only the super-user. As a general rule, this is true of all files in */etc*.

If the super-user password is forgotten, call your HP System Engineer for assistance.

Communicating with System Users

The directory */usr/news* and the *news* command provide a way to get brief announcements to the system users. The system, as shipped, notifies each user at login that there is news when */usr/news* contains entries. More pressing items can be entered in the */etc/motd* file, as long as the items are reasonably brief. HP-UX, as shipped, sends the contents of */etc/motd* to each user at login.

To write to users who are already logged in, use the *write* or *wall* command. Don't use *wall* while logged-in as the super-user, except in emergencies. *Wall* sends its message immediately to all users, regardless of the tasks that they are performing. Receiving an unimportant message while in the middle of an edit session can be a frustrating experience.

Any user can send a message to any other user via the *mail* command. The message is sent to the designated user and stored in a file. For the content of the message to be seen, the message receiver must execute the *mail* command.

Configuring HP-UX

HP-UX allows certain memory related attributes to be configured. These attributes determine the manner in which the system allocates memory and thus, the performance of your system. The attributes that you can specify (configure) are:

- the virtual memory device - the mass storage device that is used by virtual memory.
- cache buffer size - the size (in bytes) of an individual buffer in the file system buffer cache.
- number of cache buffers - the number of buffers that form the file system buffer cache.
- the read ahead level - the number of "buffers-full" of information that is read into the buffer cache when a sequential data access is detected.
- the swap time - the criterion used by the virtual memory system when determining which segment(s) to swap out of memory. When a memory block is needed by the virtual memory system, it examines all virtual segments in memory. Any virtual segment that has been in memory longer than the "swap time" is eligible to be swapped out. If no virtual segment has been in memory longer than the "swap time", the system swaps out the least recently accessed virtual segment.

The swap time is measured in units called "ticks". One tick equals 10 milliseconds.

- the page size - the size (in bytes) of each page, when data is paged in the virtual memory system.
- the virtual memory page frame pool size - the maximum size (in bytes) of the memory allocated to the page frame pool (no matter how many memory blocks are free) for the virtual memory system.
- the display memory size - the size, in pages (one page of display memory equals 24 displayed lines) of the display memory of the Model 520's display.

When HP-UX is installed on a Model 520, the computer's display and keyboard act as a "terminal". As a terminal, it needs memory to hold the information to be displayed. Increasing the size of the display memory increases the amount of data that can be scrolled on the display. Each page of display memory allocated reduces the amount of memory available to the system by about 6 Kbytes.

If at power-up the system does not have enough memory available to allocate the requested amount of display memory, the system notifies you and allocates one page of display memory.

The primary status of the Model 520 "terminal" identifies the amount of display memory in the "terminal". In the "terminal"'s primary status byte, each page of display memory allocated appears to be 2 Kbytes, instead of 6 Kbytes, as indicated in the preceding paragraph.

Note

The display memory size attribute is ignored when HP-UX is running on a Model 530/540 Computer.

- the local time zone - the number of minutes between your local time and Greenwich Mean Time. This is the one configurable attribute that does not deal directly with memory allocation in your system.
- the stack size - the maximum size (in bytes) of a partition's stack segment.
- the maximum number of segments - an integer value specifying the maximum number of entries in the shared system segment table. All shared code and data segments are allocated from this segment table.
- the working set ratio - a decimal value (representing a percentage) specifying the minimum number of virtual pages that are guaranteed to any process in memory.

Suppose that you are running a program that, if completely loaded into the system, required 1000 pages to be allocated to your partition. Also suppose that the working set ratio is .04. This guarantees that $(.04) \times (1000 \text{ pages})$ or 40 pages are mapped to your process' partition. If more pages are available in the page frame pool, they may be mapped to your process' partition as well. However, when pages are needed by another partition (and none are available in the page frame pool), you are guaranteed that your process' partition will have at least 4% of its maximum need (40 pages).

The system is configured with the *uconfig* (8) command. This command writes the new configuration values to the boot area of the boot device (the mass storage device from which the system booted). This means that **to see the effect of the new configuration, you must shut the system down and re-boot** (thus loading the kernel and the new configuration values from the boot area).

The operating system being used to modify configuration parameters in a boot area must be the same version as the operating system residing in the boot area being modified.

To configure the system (with other than the default configuration attribute values) you must create a text file. Each line in the text file must have the form:

```
id value [# comment]
```

where *id* is a fixed name identifying the attribute you wish to change and *value* is the new value of the attribute. You can attach comments to any line in the file (and thus make a note to yourself the reason for the value change) by entering the comment after a # character.

Once the file is created, it can be copied to the boot area of a device with the *uconfig -f* command. See *uconfig* (8) in the HP-UX Reference manual.

Setting the Default Configuration Parameters

The HP-UX Reference manual describes each *id* and their range of acceptable values for each attribute. Should you accidentally set values that reduce the performance of your system, you can restore the system's default configuration values by simply entering (where *boot_device* is the name of the special file of the boot device):

```
uconfig -d <boot_device> RETURN
```

This command sets the following default values:

- virtual memory device default: (0 0 0 0 0). This specifies that the default virtual memory device is the root device.

- cache buffer size default: 1024 (bytes).
- number of cache buffers default: 50 (buffers).
- virtual memory page frame pool size default: 0 (bytes). This specifies the maximum amount of memory that is mapped to the virtual memory page frame pool. If the default value is 0, a new value is calculated to optimize the page frame pool. This value then becomes the default. When you ask for default values, the system sets the default = 0.
- read ahead level default: 4 (buffers-full).
- swap time default: 20 “ticks” (where 1 tick equals 10 msec).
- page size default: 4096 (bytes).
- display memory size default: 2 pages or 48 lines.
- local time zone default: +420 (minutes), specifying Mountain Time Zone.
- stack size default: 2 097 152 (bytes).
- segment number default: 700 (segments).
- working set ratio default: .00375 (0.375%).



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 your future disc needs.

1. The *du* command should be executed regularly (for example, weekly) and the output kept in an accessible file for later comparison. In this way you can spot users who are rapidly increasing their disc usage. (*du | sort -nr* works nicely for this.)
2. Use the *find* command to locate inactive or large files. For example, the following entry records in the file *somefile* the names of files neither written nor accessed in the last 90 days:

```
find / -mtime +90 -atime +90 -print > somefile
```

3. Use the *df* command to list the amount of free disc space on a volume.

Some files, if present, are automatically written to by certain HP-UX commands to monitor use and for general house-keeping. Such files are called logging files. If not periodically checked and cleared, these files can grow without bound. HP-UX, as shipped, includes no logging files. The following is a list of some logging files that get created:

- */etc/utmp* (binary) - current login status.
- */usr/adm/wtmp* (binary) - history of logins, logouts and date changes.
- */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/cronlog* (ASCII) - history of actions by *cron*.
- */usr/adm/errfile* (ASCII) - file where system error messages are logged (also displayed on the system console).

Creating a Group and 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 form of the entry and the meaning of each field is shown below:

```
group_name:password:group_id:member1, member2, ..., memberN
```

`group_name` - is the name of the group.

`password` - currently, this field is not used. To prevent non-group members from switching to this group (via the *newgrp* command), place an asterisk (*) in this field.

`group_id` - is the unique integer ID shared by all group members. Each group member's entry in */etc/passwd* must have this integer value in the group ID field.

`member1, member2, ..., memberN` - list consisting of the user name of each group member. User names in this list are separated by commas.

To alter group membership, simply modify the membership field for the group entry in */etc/group*, and modify the `group_id` field for each member's entry in */etc/passwd*.

When you are satisfied with the group definition, write the modified file to disc and terminate the edit.

Note that a user may belong to many different groups. By using the *newgrp* command, the user can change his effective group ID to that of another group. A user's default group is specified by his group ID entry in */etc/passwd*.

File System Check and Repair

Every working day, hundreds of files are created, modified, and removed. Every time a file is modified, HP-UX performs a series of file system updates. These updates, when written on disc, yield a consistent file system. When one or more updates fail, the file system may become corrupt. An update might fail for any number of reasons. The most common reasons are:

- improper system shutdown (such as taking a mounted file system off-line or forgetting to execute *sync* prior to halting the system).
- physically write-protecting a mounted file system.
- computer or disc device power failure.
- hardware failure (such as discovering a bad block on a disc or the failure of a disc drive).

The file system is further corrupted if the inconsistency is not quickly identified and repaired. Using a corrupt file system and further modifying it can be disastrous.

HP-UX provides a tool, the */etc/fsck* command, that checks the structural integrity of a static file system (the file system should not be accessed during the integrity check and repair). Except for the root file system, the file system should not be mounted when checked. It is imperative that the file system be static because the command makes multiple passes at the file system when checking its consistency. If a file is modified between passes, it is impossible to determine whether or not the file system is consistent.

When *fsck* discovers an inconsistency, the error is reported and the system asks whether it should take a corrective action. Failing to correct the inconsistency results in a corrupt file system. However, you are not required to make the correction immediately. Often, it is beneficial to examine the files in question before correcting the problem by re-running the *fsck* command.

To ensure system integrity, this command should at least be run each time HP-UX is loaded.

fsck

This document describes only the function of the *fsck* command. To find the syntax for its use, refer to *fsck(8)* in the HP-UX Reference manual. Note the use of the file */etc/checklist* to specify the default file systems to be checked.

When *fsck* is executed:

1. The following messages are displayed to indicate that *fsck* has begun execution:

```
Checking <file_system>.  
** Checking Directories.
```

It first creates a list of all extents (groups of contiguous disc blocks,) that the system believes are claimed by a file or that are free (as identified by the free map). Each entry in the list consists of the starting address of an extent and the size of the extent.

During this phase of its execution, *fsck* also traverses all directories in the file system, creating a map of all i-nodes pointed to by directory entries (a map of all i-nodes in the file system).

2. When the following message is displayed, *fsck* begins looking for inconsistencies:

```
** Checking Blocks.
```

It examines the list of extents built previously in step 1. The list identifies all disc allocation. In a consistent file system, this list includes every block on the disc. Any extents not claimed by either a file or the free map are reported at the end of this process along with the total number of unclaimed blocks in the file system.

Similarly, any extents claimed by more than one i-node are reported at this time. The system issues a report of the form:

```
MULTIPLE FILES CLAIM THE SAME SPACE ON THIS DISC!!
Multiply claimed extents:
XXXX to XXXX, claimed by inodes YYYY and ZZZZ
```

XXXX identifies the disc block(s) in the extent claimed by inodes YYYY and ZZZZ. When an i-node value equals -1, it specifies that the free map is one of the i-nodes claiming the extent. If one of the i-node values reported is the free map, re-running *fsck* with the *-s* option corrects the situation by re-building the free map. However, if neither i-node is the free map, then two valid files claim the same extent. In this instance you should examine the file associated with each i-node and remove one of the files (usually the oldest file is removed). To find the name of each file associated with each inode, enter the following command:

```
/bin/find / -inum YYYY -print RETURN
```

Once you know the name of each file claiming a common extent, you can examine each file and remove the file(s) that are improperly claiming the extent. When one of the files is removed, its claim on the extent is transferred to the free map. Then, when *fsck* is re-run with the *-s* option, the situation is corrected. *fsck* with the *-s* option must be re-run immediately after removing the file to prevent the operating system from reallocating its space to yet another file. A consistent file system contains no multiply-claimed extents.

3. When the following is displayed, *fsck* attempts to locate orphaned files.

```
** Checking the File Attribute File.
```

First, the list of i-nodes created in step 1 above is examined. Any i-node not included in the list is examined to identify the disc space it claims. An i-node can claim space and yet not be recognized as a file if the i-node exists but has no links from a directory. If the space claimed by the i-node is part of the unclaimed blocks list, the file is termed an "orphan". An orphan is an i-node that represents a valid file but has no link from a directory.

The system next examines the size of the "orphaned file" (i-node). If the size is 0, the system asks if it may clear the i-node and restore its extents to the free map. An affirmative reply clears up the inconsistency. A negative response causes no action and leaves the file system corrupt. If the size of the orphan is non-zero, the system asks if it should attempt to restore the file. An affirmative response causes the orphan's i-node to be linked to the directory */lost + found*. The associated disc space is re-marked as "claimed" and is removed from the list of unclaimed blocks. The name of the "adopted" file is its i-node number. A negative response causes the orphan to be treated as if its size were zero; the system asks if it may clear the i-node and restore its extents to the free map.

4. *fsck* next checks for "orphaned" extent maps. This is signified by the display of the message:

```
** Checking Extent Maps.
```

When an i-node claims more than four extents, it must use one or more extent maps to identify the additional extents. Occasionally, when a file becomes corrupt, the i-node may be cleared but the extent map(s) claimed by the i-node is not cleared. When this happens, an orphaned extent map is created.

If the orphaned extent map claims blocks listed in the unclaimed block list (created in step 1 above) and if all orphaned files have been restored, the system asks if you wish to add the blocks claimed by the orphaned extent map to the free map. An affirmative response causes the claimed blocks to be added to the free map. A negative response causes no action; the file system remains corrupt.

5. When the following message is displayed, the system begins verifying the number of links to each i-node:

```
** Checking Link Counts.
```

Each i-node keeps track of the number of directory entries linked to it. Additionally, as *fsck* executes it creates a separate list with which to cross check the i-node link count. If the link counts disagree, the system displays the messages:

```
Wrong link count for inode XXXX.  
It's currently Y but should be Z.  
Shall I fix it?
```

where Y represents the link count maintained by the i-node and Z represents the link count created by *fsck*. An affirmative response to the prompt resets the i-node link count such that it agrees with the count maintained by *fsck*. A negative response prohibits corrective action from taking place; the file system remains corrupt.

6. If, during any of the previous steps, you have failed to take a suggested corrective action (thus leaving the file system corrupt), the following message is displayed and the checks described in this step are not performed:

```
** Unable To Check Free Map.
```

This message specifies that the file system still has some uncorrected problems.

If the file system is consistent and no further corrections are needed, the following message is displayed and the checks described in this step are not performed:

```
** No Need To Check Free Map.  
There are no unclaimed blocks.
```

This is an informational message stating that nothing is wrong.

If all of the corrective actions specified by *fsck* have been taken and if the free map is not known to be current, *fsck* displays:

```
** Checking The Free Map.
```

Next, the current state of the file system is compared to the disc's free map. If the system finds an inconsistency between the two, a message is displayed specifying that the free map is out of date. You are then asked if you wish to update it (the free map). An affirmative response causes the system to re-build the free map such that it is consistent with the current state of the file system. A negative response causes no corrective action to be taken; the file system remains corrupt.

7. Finally, the system updates the internal information it keeps about the file system root or virtual memory volume, as indicated by the display:

```
** Updating kernel data structures.
```

If this update fails (the update always fails when checking an unmounted volume), the system displays the message:

```
Kernel failed to update its data structures for <file_system>.
```

If the volume being checked is the file system root or the virtual memory device, you should re-boot the system at this time.

If the update succeeds (or the volume being checked is not mounted), *fsck* next reports:

- the total number of files on the volume.
- the total number of blocks used by the file system.
- the total number of free blocks available on the disc.
- the percent of the disc that is unused.

If the file system has not been fully repaired, the following message is displayed:

```
THIS FILE SYSTEM IS NOT COMPLETELY RESTORED.
```

You should fully repair the system before allowing access to it. However, some errors in the file system are more dangerous than others. Unless the file system contains multiply-claimed extents, you can probably use the file system without further corrupting the system. If the file system contains multiply-claimed extents, you must correct the situation (by re-running *fsck*) before using the file system.

Occasionally, a file system is too corrupt to be repaired with *fsck*. In this instance, you must initialize the medium and then restore the file system from its last backup. If the HP-UX file system is too corrupt to be repaired, you may have to re-install the system (initializing the disc) and then restore the file system from your last backup.

It is often useful to run *fsck -n* through *cron*, the program scheduler, when no one is using the system (for example, late at night). Redirect the standard output and standard error from this command to the system printer so that no information is lost. (Do not redirect the standard output or standard error to a file on the device being checked, as that yields a non-static file system). Even though no corrective action is performed, the output from the command identifies any inconsistencies in the file system and thus shows whether or not you need to manually run *fsck* to correct the situation.

fsck and Virtual Memory

One of the devices that *fsck* can be called upon to check is the virtual memory device. Unfortunately, *fsck* requires a great deal of memory for its own use and this means that the file system is often not quiescent when checking the virtual memory device. However, consider the following points:

- *fsck* reads the free map fairly early in the checking sequence before it gets the space to keep track of file system extents. This means that *fsck* thinks that any existing virtual memory extents are unclaimed (and treats them as such). A number of these extents may be associated with *fsck*'s encompassing shell, existing processes, and what little memory *fsck* has already gotten.
- As *fsck* grabs more memory, it is not reflected in its internal copy of the free map.
- When *fsck* rebuilds the free map, all virtual memory extents are marked as "free". Although this is a potentially dangerous situation, *fsck* immediately notifies the operating system, which restores the true status of the virtual memory extents in the free map.

Thus, under normal circumstances, it should be safe to run *fsck* on any virtual memory volume. However, if there is any other virtual memory activity occurring on the volume, especially while the operating system is updating its virtual memory data structures, the volume could be corrupted or destroyed.

Mounting and Unmounting Volumes

When HP-UX is installed, only one file system (the file system root) exists. You may create, modify, and delete files from this system. By default, the HP-UX file system exists on one volume and thus, on one mass storage medium. It is possible to have other file systems on other volumes; any other disc device can be used for an independent file system. The independent file system(s) are then appended to the root file system. The process of appending independent file systems to the file system root is called **mounting** and is achieved with the *mount* command. The process of removing independent file systems from the file system root is called **unmounting** and is achieved with the *umount* command.

Once a special file exists for the disc device, the volume it contains can be mounted. Mounting a volume adds its files to the file system hierarchy. Files can then be created, modified and deleted on this new volume. When you are finished with the files on that volume, it can be unmounted. Unmounting a volume removes its files from the file system hierarchy. The files remain on disc and can be accessed at any time by simply re-mounting the volume.

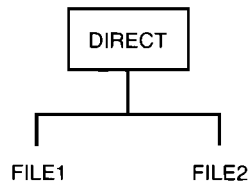
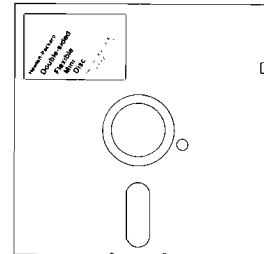
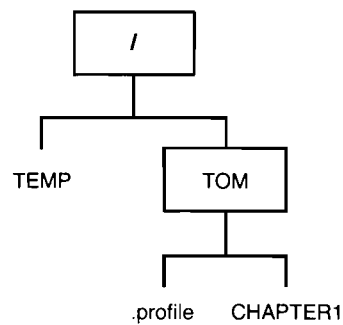
Typically, this feature is used to provide personal mass storage via removable media mass storage devices. For example, suppose that a user wished to keep his programs and data on a flexible disc in his desk drawer (for example, the disc contains a secret project or backups of his files). The user obtains and initializes a flexible disc, mounts the volume as a file system, does his work, moves the modified files to the removable file system and finally, after unmounting the volume, places it in his desk.

As system administrator, you may wish to set the "set-user-ID bit" in the protection modes of the */etc/mount* and */etc/umount* commands. If this is done, also use the *chown* command to make "root" the owner of these commands. Making these changes allows any user to execute these commands and thus provides them the ability to mount and unmount their own volumes.

To Mount a Volume ...

1. The volume must have been initialized at some point in its past. Its format must be SDF. Initializing the volume with the HP-UX command *sdfinit* (8) causes the volume to be initialized in SDF format. If the volume is initialized with the HP 9000 BASIC *INITIALIZE* command, be careful to specify SDF format, as many different formats are allowed.
2. Next, you must decide the location of the mounted volume within the hierarchical file system. When a volume is mounted, the hierarchical file system it contains is attached to a directory in the existing file system. Any files that the directory contained previously are temporarily replaced by the file system contained on the mounted volume.

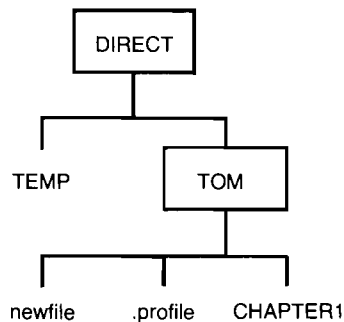
For example, suppose that the directory *DIRECT* contains two files, *FILE1* and *FILE2*. Additionally, suppose that you wish to mount an SDF format disc, containing a hierarchical file system (as shown in the following illustration), with *DIRECT* as the destination. Once the files on the mounted volume are modified, the volume is unmounted. This process is shown in the illustrations that follow.

FILE SYSTEM HIERARCHY
OF MOUNTED FILE SYSTEMFILE SYSTEM HIERARCHY OF
UNMOUNTED FLEXIBLE DISC VOLUME

= Directory

The preceding diagram shows the directory **DIRECT** of the existing mounted file system. It also shows the file system hierarchy on the as yet unmounted flexible disc volume.

FILE SYSTEM HIERARCHY AFTER MOUNTING FLEXIBLE DISC VOLUME

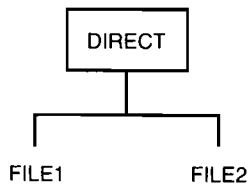
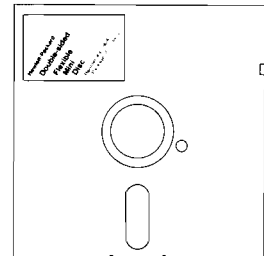
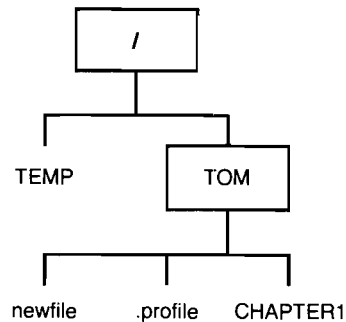


= Directory

The preceding diagram shows the file system hierarchy once the flexible disc volume is mounted on the directory **DIRECT**. The file **newfile** has been added to the flexible disc volume after mounting. Notice that the files **FILE1** and **FILE2** that previously existed under **DIRECT** are no longer accessible (the files have not been destroyed, they are just inaccessible until the flexible disc volume is unmounted).

The flexible disc volume was mounted with a command of the following form:

```
mount /dev/fd9130k /DIRECT RETURN
```

FILE SYSTEM HIERARCHY
OF MOUNTED FILE SYSTEMFILE SYSTEM HIERARCHY OF
UNMOUNTED FLEXIBLE DISC VOLUME

 = Directory

The preceding diagram shows the directory **DIRECT** of the existing mounted file system once the flexible disc volume is unmounted. Notice that the two files **FILE1** and **FILE2** are once again accessible. The diagram also shows the file system hierarchy on the unmounted flexible disc volume.

The flexible disc volume was unmounted with a command of the following form:

```
umount /dev/fd9130k RETURN
```

3. Create a special file, using *mknod*, for the mass storage device from which the volume is to be mounted, if one does not yet exist. The special file must be a block special file. `/dev/fd9130k` is used in the preceding example.
4. Make sure that the mass storage device is powered up and is on-line. If the volume is on removable media (such as a flexible disc), insert the volume in the mass storage device at this time. Enter the following:

```
mount special_file destination_directory RETURN
```

where `special_file` is the pathname of the mount device's special file and `destination_directory` is the pathname of the directory where the volume's file system is to be mounted.

Always unmount a volume before removing it from its mass storage device. Removing a mounted volume from its mass storage device before unmounting it often corrupts the volume. Use the following procedure to unmount the volume.

To Unmount a Volume ...

1. Make sure that all files on the volume are static; that is, no one should be accessing any file on the volume. Attempting to unmount a volume that has open files (including your current working directory) causes the *umount* command to fail without unmounting the volume.
2. Enter the following:

```
umount    special_file  RETURN
```

where *special_file* is the pathname of the block special file of the device on which the volume is mounted.

3. When the volume is unmounted, the shell prompt is again displayed. If the volume is on a removable medium (such as a flexible disc), the medium can now be removed safely.

Special Considerations

You can't unmount a volume that has open files. The following situations are the most frequent cause of open files on a volume:

- having your current working directory on a volume causes an open file on the volume.
- if a program stored on a volume 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. (To un-stick a file, use the *chmod* (1) command to clear the sticky bit, and re-execute the file (even if it executes erroneously).)

Option Product Installation and Updating

If you decide to purchase an optional software product for your HP-UX system, you need a way to install the software. If you elected not to install all software products when you installed the system (for example, you used the *coreinstall* command to install HP-UX, not the *sysinstall* command), you need a way to install the additional software products. HP-UX provides the *optinstall* command to install optional HP-UX software products on existing HP-UX systems.

If you order support services for optional HP-UX software products, you periodically receive updates to the software. If you do not install the optional product updates with the *sysupdate* command, as described in this chapter's section "Updating Your Operating System", you can use the *optupdate* command to update any optional software product. (All products are optional except 97070A, 97079A, 97080A, and 97089A.) Appendix B contains error information for these commands.

Note

The *optinstall* and *optupdate* scripts append all files in */system/product_number* which end in *.req* to the boot area, regardless of whether or not these files already exist in the boot area. For more information, refer to step 7 under "Before You Begin" in this manual's section "Updating Your Operating System".

Instructions for installing and updating optional software products are provided in the following procedure.

1. The option or update is supplied on an HP 88140L/S cartridge tape for installation from the built-in cartridge tape drive of your HP 7908P/R, HP7911P/R, HP 7912P/R, or HP 7914P/R disc drive.

Examine the supplied cartridge tape containing HP-UX. Locate the write-protect mechanism (labeled "SAFE") at the front, upper, right-hand corner of the medium. Make sure the arrow is pointed to "SAFE". If it is not, using a coin or screwdriver, **turn the protect screw such that the arrow is pointed toward the word "SAFE"**. Note that this is exactly opposite to the instruction given for *sysinstall*.

2. Determine from which device you want to load the tape's data. This is the "source device".
3. Power up the source device. After the drive's buzzer has sounded, insert the HP 88140L/S cartridge tape into the disc's built-in cartridge tape drive. Wait for the drive's busy light to extinguish before continuing.
4. Log in as the super-user and execute one of the following commands.

```
optinstall product_number [unload] [debug]
```

or

```
optupdate product_number
```

where *product_number* is the product number of the option to install or update. If the **unload** keyword is specified, the cartridge tape is automatically rewound and unloaded at the end of the update. Refer to *optinstall* (8) in the HP-UX Reference manual for additional information. The following are some typical examples.

```
optinstall 97071A
```

```
optupdate 97074A
```

```
optinstall 97059A
```

The exact entry is the same as that listed on the spine of the tape cartridge for that product.

The system responds by printing the following on the system console.

```
HP-UX FEATURE PRODUCT INSTALLATION UTILITY
```

or

```
HP-UX FEATURE PRODUCT UPDATE UTILITY
```

5. The system then lists the default parameters for the utility.

```
Here are the default parameters for this installation/update:
```

```
* Source device:      88140 L/S tape cartridge
                      select code      = 5
                      bus address      = 0
                      unit number      = 1
```

```
* Destination device: HP-UX root volume
```

```
Are these values satisfactory for your system?
```

```
If they are not, you will be instructed to supply
responses for each parameter, interactively.
(Enter y or n, then RETURN; default is 'yes')
```

When an option or update is incorporated into the operating system, either the boot area or the file system or both are altered. For example, the graphics options are just system calls, libraries and routines. They do not contain any boot area code. Thus, only the file system is updated.

The destination device specifies which boot area and file system are altered. The default is the current root volume.

6. If the displayed parameters are correct, execute (then continue with step 10):

```
y 
```

If you want to change one or more of the parameters, enter (and continue with step 7):

```
n 
```

7. If you typed a negative answer, you are prompted to identify the source device:

```
* Source device:      88140 Cartridge Tape Drive
                      select code      = 5
                      bus address      = 0
                      unit number      = 1
```

```
Do these parameters accurately describe your source device?
```

```
(Enter y or n, then RETURN; default is 'yes')
```

If the source device is correctly identified, press to select the default response (yes) or enter:

```
y 
```

Continue the installation procedure with step 8.

If the source device is not correctly identified, enter the following (then continue the installation procedure with step a):

```
n 
```

- a. The system next prompts to find the select code of the source device:

```
Enter the select code of your 88140 tape drive.
The range for the select code is 0 through 23, inclusive.
```

Enter the select code of the source device, then press **RETURN**. Entering no value (simply pressing **RETURN**) causes the most recently entered value (or the default value, if it has never been changed) to be used.

Note

When asked to enter numerical information (such as a select code or bus address), do not use leading zeros with the values you enter. For example, to specify the value 7, enter 7, not 07 or 007. Use of leading zeros can cause unexpected results.

- b. The system next prompts to find the bus address of the source device:

```
Enter the bus address of your 88140 tape drive.
The range for the bus address is 0 through 7, inclusive.
```

Enter the bus address of the source device, then press **RETURN**. Entering no value (simply pressing **RETURN**) causes the most recently entered value (or the default value, if it has never been changed) to be used.

- c. The system next prompts to find the unit number of the source device:

```
Enter the unit number of your 88140 tape drive.
The range of the unit number is 0 through 1, inclusive.
```

Enter the unit number of the source device, then press **RETURN**. Entering no value (simply pressing **RETURN**) causes the most recently entered value (or the default value, if it has never been changed) to be used.

8. Next, you are prompted to identify the destination device:

```
* Destination device:      HP-UX root volume

Is your destination device correctly described?
(Enter y or n, then RETURN; default is 'yes')
```

If the destination device is correctly identified, enter:

y **RETURN**

Continue the installation/update procedure with step 9.

If the destination device is not correctly identified, enter the following (then continue the installation procedure with step a):

n **RETURN**

- a. The system next prompts to find the select code of the destination device:

```
Enter the select code of the destination device.
The range for the select code is 0 through 23, inclusive.
```

Enter the select code of the destination device, then press . Entering no value (simply pressing) causes the most recently entered value (or the default value, if it has never been changed) to be used.

- b. The system next prompts to find the bus address of the destination device:

```
Enter the bus address of the destination device.
The range for the bus address is 0 through 7, inclusive.
```

Enter the bus address of the destination device, then press . Entering no value (simply pressing) causes the most recently entered value (or the default value, if it has never been changed) to be used.

Since the unit number of the destination device must be zero (0), you are not prompted for this value.

9. Next the system displays the menu of default values shown in step 5, previously (the new values you entered during the prompting sequence replace the original values displayed). And again, you are prompted to specify whether or not the values displayed are valid for system installation. Continue the installation procedure from step 5. (This time, the values shown for each installation parameter are the values you specified in steps 5 through 8).
10. Now that all of the parameters for the installation program are correct, the system prompts you to re-examine the values and make sure that they are accurate:

```
Are you sure you are ready to begin the installation/update procedure on the
destination disc?
```

```
(You should have a current backup of your file system!)
```

```
(Enter y or n, then RETURN; default is 'NO'))
```

If you are NOT satisfied with the installation parameters, either press (selecting the default answer, 'no') or enter:

```
n 
```

The program terminates. Re-start the procedure at step 4.

If you are satisfied with the parameters and are ready to begin the installation or update, enter:

```
y 
```

The system begins the installation or update and supplies the following comments as it proceeds.

```
Mounting the source device.
Checking the source device for product 97xxxA
Installing/Updating the 97xxxA product.
```

The system then lists files for the product that are being installed or updated. Once all the files have been installed or updated the following message appears:

```
THE UPDATE/INSTALLATION OF HP-UX FEATURE PRODUCT ON THE
DESTINATION DISC IS NOW COMPLETE.
```

To install another optional product, go back to step 4 of this procedure.

If the **unload** keyword was not specified when you invoked *optinstall*, the tape is not automatically rewound after an option installation. To rewind and unload the tape manually, execute:

```
tcio -urV /dev/rmt
```

The above example assumes that */dev/rmt* is the special file for the cartridge tape drive. To perform *tcio* to rewind your tape, you may have to create a special file for the drive using *mknod*. See the “Creating Special Files” section of this chapter.

This may take a few minutes to execute, so it is useful to do it in the background.

If the installation of the update affected the boot area, you must re-boot your system to access the new features.

Setting the System Clock

Occasionally, the system clock needs to be set or reset. Only the effective super-user can change the system clock. To set the current time and date:

1. Login as the super user.
2. Insure that the time zone environment variable TZ is set properly (for more information refer to *tzset* under *ctime(3)* in the HP-UX Reference manual). TZ should be assigned the seven character value XXXHYYY, where XXX and YYY are the 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. For example, in Denver, Colorado TZ should be set:

```
TZ=MST7MDT
export TZ
```

MST identifies Mountain Standard Time and MDT identifies Mountain Daylight Time. Exporting the time zone environment variable (with the *export* command) ensures that all future child processes also receive the variable definition.

Typically, TZ's value is set with a variable declaration in the shell script */etc/profile* (refer to the chapter entitled "System Boot and Login: a Detailed Examination" for more details). It can also be set from an application program with the *tzset* library routine.

3. To set the correct time and date, enter:

```
date MMddhhmmyy 
```

MM is a two digit integer representing the month (for example March is represented as 03).

dd is a two digit integer representing the 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; 14 specifies 2:00 pm.

mm is a two digit integer specifying the number of minutes past the stated hour.

yy is an **optional** two digit integer specifying the last two digits of the current year. For example, 82 specifies 1982 as the current year.

Special Considerations

Altering the system clock can cause some unexpected results for routines scheduled by *cron*. 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, *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* re-executes continually until it catches up (not just once a minute).

If you are only changing the clock by a small amount (20 - 30 minutes), this should present little problem. No corrective action should be needed. However, if you are changing the clock by a larger increment, you may want to perform the following steps:

1. Log in as the super-user.

2. Enter the following and locate the *cron* process:

```
ps -ef RETURN
```

3. Terminate the *cron* process by entering:

```
kill -9 process_number RETURN
```

where `process_number` is the process number of the *cron* process.

4. Change the time and date with the *date* command.
5. Start the *cron* process running again by entering:

```
crontab RETURN
```

System Shutdown

If you power down the computer or an “on-line” mass storage device without proper preparation, the file system can become corrupt. In addition, you may ruin work in progress being done by others. There are a number of ways to shut down your system in a more orderly fashion; each has advantages and disadvantages. There are five commands used to shut down or reboot a system:

- *stopsys*;
- *stopsys* -r;
- *chsys*;
- *killall*;
- *shutdown*.

Note

If you have a multi-user system, use the *shutdown* command (discussed at the end of this section) to stop your system until you are very familiar with your system and shut down procedures.

stopsys

The *stopsys* command shuts down the system. *stopsys* dumps all system I/O buffers to mass storage volumes; that is, performs a *sync*, and shuts down all virtual memory activity. Then, *stopsys* stops the operating system so that the hardware may be powered down.

Just before it stops the system, *stopsys* writes a message to */dev/console* indicating that the system is stopped and can be safely powered down.

Executing *stopsys* directly is only safe for a single-user system because *stopsys* does not ensure that the system is idle. If any user processes are running, the *sync* may be ineffective. On multi-user systems, you should execute *shutdown* or *killall* prior to running *stopsys*.

To execute *stopsys*, type:

```
stopsys RETURN
```

stopsys -r

If you want to reboot your system after shutting it down, execute *stopsys -r* instead of *stopsys*. *stopsys -r* does everything *stopsys* does and then automatically reboots the system (invokes the boot loader) without the hardware being powered down.

Type:

```
stopsys -r RETURN
```

chsys

To alter the operating system and reboot, use *chsys*. *chsys* is a shell script that enables you to boot a different operating system, or a different version of the same operating system, using only one boot area on one disc. The synopsis is:

```
/sbin/chsys sysname
```

where *sysname* is one of a number of operating system names defined within *chsys*. *chsys* uses *oscp* to rebuild the boot area on */dev/rhd* with the selected system, reading from ordinary files containing operating system code. *chsys* then invokes *osck* to confirm that the new system is “healthy”. (Note that the check performed by *osck* is redundant, so its invocation in *chsys* may be removed if you want to save time.)

Chsys works for a single-user system if the system is idle. Before you execute a *chsys* on a multi-user system, you should execute *shutdown* or *killall*.

Chsys invokes *oscp* as quietly as possible. *chsys* causes *oscp* to read the new system ID string from a file selected by the *sysname* given, and redirects the output from *oscp* to */dev/null*. If *oscp* and *osck* are successful, *chsys* calls *stopsys -r* to switch to the new operating system. Note that *oscp* and *osck* together can take longer than a minute to run. During this time, *chsys* keeps you informed as to what actions are being taken.

You should modify *chsys* for your system. You may want to add or delete available *sysnames* (system names), change the names or meanings of *sysnames*, change the name of the character special file (*/dev/rhd*) which points to the boot volume, and so on. *chsys* recognizes four default *sysnames*. They stand for:

- HP-UX Model 520 single-user minimal system;
- HP-UX Model 520 single-user complete system;
- BASIC minimal system;
- BASIC complete system.

To see the list, execute *chsys* without any arguments.

killall

Using the *init 9* and *killall* commands is a quick way to bring your system to a single-user state. They are useful when you have a small multi-user system. When you want to shut down your system, you just tell all the other users to finish up their current work.

Executing *init 9* causes the system to enter state 9 which, under default system configuration, sets the system to a single-user state. This prevents the *init* process from restarting *getty* for each terminal after *killall*.

The default form of *killall* sends signal 9 (kill) to all existing processes in the system except:

- the *init* process;
- all processes (including background processes) associated with the terminal from which *killall* was invoked;
- any *ps -ef* process, if owned by *root*;
- any *sed -e* process, if owned by *root*;
- any *shutdown* process;
- any *killall* process;
- any */etc/rc* process.

killall doesn't really shut down the system, but puts it in a single-user state with the terminal executing the *killall* as the only active terminal.

After executing *killall*, you can continue working in a single-user mode or execute *stopsys*, *stopsys -r* or *chsys* to truly shut down (and optionally reboot) the system.

A good way to use *killall* is as follows.

1. Log in as the super-user.
2. Execute:

```
init 9
```

Wait a moment after executing *init* to ensure that it has taken effect.

3. Execute:

```
killall
```

shutdown

shutdown terminates in an orderly and cautious manner all processes currently running on the system. It notifies users to log off, *syncs* each file system, executes *init 9* and *killall*, and optionally executes *stopsys*.

This means you can power down the system hardware without adversely affecting the file system. It is slower than invoking *init 9* and *killall* directly, but is safer to use.

shutdown is especially suited for large multi-user systems where you need a formal method of bringing down the system.

To run the *shutdown* command, perform the following steps.

1. Log in as the super user.
2. Enter:

```
shutdown grace_Period RETURN
```

where *grace_Period* is the number of seconds you want *shutdown* to wait between notifying all system users of the impending *shutdown* (via *wall*) and terminating all processes.

3. *shutdown* next prompts to see whether you wish to send the standard file save 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**.
4. After waiting the specified amount of time, *shutdown* asks if you want to continue. An affirmative response causes the command to *umount* all file systems and then prompts to see if you want to run the file system integrity check routine. Supply the appropriate response.
5. *shutdown* next prompts to see whether you wish to power off the system or simply stay in the single user mode. An affirmative response causes *shutdown* to stop the operating system and display a message on the console when it completes its task. You may then power off the system.

If you answer negatively, you remain in single user mode. You can then stop the system using one of the following methods:

- executing *stopsys*;
- stopping the system and then rebooting it using *stopsys -r*;
- changing the system and rebooting it by using *chsys*.

Summary

There are five basic commands you may want to use when shutting down your system.

- *stopsys* is a quick way to shut down a single-user system.
- *stopsys -r* is a quick way to stop and reboot a single-user system.
- *chsys* is a quick way to reconfigure and reboot a single-user system.
- *init 9*, *killall*, and then *stopsys*, *stopsys -r* or *chsys* is a faster way to shut down a system and optionally reboot it, but requires more care. It is useful for a small multi-user system.
- *shutdown* is an interactive program which gracefully shuts down your system. It is slow, but is useful for large multi-user systems. You can optionally execute a *stopsys* from the program or execute *stopsys*, *stopsys -r* or *chsys* yourself.

Updating Your Operating System

If you order support services for HP-UX and its optional software products, you periodically receive one or more cartridge tapes containing updates to the software. These updates correct problems in the software and can add features or capabilities.

Before You Begin

Before you begin the process of updating your HP-UX operating system, there are several conditions that must be met to ensure that the update procedure is successful.

1. The first condition involves your system configuration (set/modified by the *uconfig(8)* command) as it relates to your virtual memory disc. If you have ever changed any part of your configuration (regardless of whether or not you have changed the virtual memory disc), you should follow the following procedure: (if not, then skip to step 2 below)
 - a. Log in as *root* on your system.
 - b. Execute the following two commands:

```
# uconfig >newfile
# vi newfile
```

You should see something similar to this:

```
vm_device          1 5 0 0 0
vm_pool_size       393216
cache_buf_size     1024
cache_buf_num      50
page_size          4096
swap_time          20
read_ahead_level   4
scroll_pages       2
max_proc_per_usr   25
stack_size         2097152
segment_num        700
work_set_ratio     0.00375
```

- c. Use the *vi* editor to change the top line (containing *vm_device*) to:

```
vm_device          0 0 0 0 0
```

- d. Terminate your editing session by typing **ZZ**.
 - e. Execute the command:

```
# uconfig -f newfile /dev/rhd
```

where */dev/rhd* is the character special file which refers to your boot disc.

The purpose of the above procedure is to ensure that the kernel chooses the root volume as the virtual memory disc. This is necessary because, during the system update, the cartridge tape is the root volume. Unless the cartridge tape is also the virtual memory device, your disc is not unmounted correctly at the end of the update.

2. This step is relevant only for those users who are using an HP 7935 (or a 7933 converted to a 7935). If this does not apply to you, skip to step 3 below.

Either or both of the conditions listed below must be true for your update to succeed (i.e. satisfying either condition is sufficient):

- a. The address of the cartridge tape drive must be lower than the address of the HP 7935. Either a lower select code, or a lower HP-IB address on the same select code, suffices.
- b. The 7935 volume must be unmarked using

```
rootmark -u /dev/rhd
```

so that the "findroot" algorithm in the operating system (not to be confused with the boot loader) identifies the cartridge tape drive as the root volume. This step must be done *before* re-booting the system with the cartridge tape as root volume.

The *sysupdate*, *coreupdate*, *sysinstall*, and *coreinstall* scripts all re-mark the disc automatically during their execution.

Note that you should **not** unmark the 7935's boot area with *osmark*. *Sysupdate* requires that the kernel operating system comes from the hard disc.

3. All optional products being updated must have previously been installed using *optinstall*.
4. All previous HP-UX updates must have been performed on all products prior to the current update. For example, the 3.0 update must have been performed prior to 4.0, etc.
5. The portion of your file system which is to be updated must not be split across more than one volume. If this is the case, *sysupdate* will fail, since it puts everything on one disc. You may, however, put files on secondary volumes if they are contained in directories that are not used by the updated file system.

For example, if you have a secondary volume mounted on */users* which contains files for certain users, the update will fail. This is because files are supplied in the update for */users/root*. If, however, your users' files are contained on a secondary volume mounted on */specialusers*, then the update will succeed, since no files are updated in */specialusers*.

6. You must not have excessively modified the existing file system. The following notes list the types of things to watch:
 - a. Adding links to files. This is fine, as long as you realize that a link could reference completely different file contents after the update. For example, if you have a link to */bin/ls* called **listit**, and */bin/ls* is then updated, **listit** now references the updated version.
 - b. Renaming files. **Do not rename the standard HP-UX files.** The update process works by file name, so renamed files will not get updated.
 - c. Removing files and directories. Use caution in removing any of the HP-UX files and directories. The update process should replace removed files with the newer version, but no attempt is made to ensure that this takes place. Some files are required to run HP-UX, and replacing them could require a service call. If you are in doubt about the safety of removing a particular file or directory, then do not remove it.
 - d. Modifying the contents, ownership, and/or mode of a file. Use caution in modifying any of these things, as the update procedure could depend on them.

- e. Modifying configuration files (refer to “System Boot and Login: a Detailed Examination” in this manual for a list of the configuration files). It is almost impossible to avoid modifying at least one of the configuration files. Thus, to protect your modifications, the update procedure creates a new directory, called */etc/newconfig*, and places the new version of the file there. For example, if */etc/passwd* is updated, the new version is placed in */etc/newconfig* to avoid destroying the previous contents of */etc/passwd*. Thus, at your leisure, you may examine the contents of */etc/newconfig/passwd* to see if any features have been added to **passwd** which might be useful to you. If so, you may incorporate them into your current **passwd** file; if not, remove the new **passwd** file.

A list of some of the files updated in this manner for the HP-UX and C language products is provided in the file */users/root/Update_info*, which is created by the update procedure.

- f. Modifying directory ownership and modes. Two problems can occur from doing this. First, some commands may no longer execute. Second, the update procedure is not able to restore the ownership and modes of directories modified in this manner. Thus, it is a good idea to leave directories which contain HP-UX standard files as they are.
7. This step applies to the *sysupdate*, *optupdate*, and *optinstall* scripts only, and is included to help you avoid a potential problem with your boot area.

These scripts look in the directory */system/product_number*, where *product_number* is the product number of an optional product you are updating or installing. All files in this directory which end in **.req** are appended to the boot area, **regardless of whether or not these files already exist in the boot area!** Thus, if you update a particular product 10 times, that product's code segment(s) are produced in the boot area 10 times. Not only does this consume space in the boot area, but it also consumes RAM when the boot loader brings the boot area into memory at power-up.

There are several things you can do to avoid this. First, before performing the installation/update, perform an *osck* command to see what is actually in the boot area. Second, you can rename all **.req** files in their */system/product_number* directories which already appear in the boot area such that their names end in **.opt**. This stops them from being added to the boot area. Third, you can rebuild the boot area manually using *oscp -m*. In this way you can explicitly choose which kernel code segments are in the boot area, in anticipation of what these update/installation scripts do.

Note that this is a consideration only when installing or updating a product for the second (or subsequent) time. This cannot happen when installing/updating the HP-UX Operating System, since the boot area is completely rebuilt during the process.

Phase 1 - Bring Down Current HP-UX System

The first step in performing the HP-UX update is to bring down your current HP-UX system. To do this, follow the steps given below.

1. If you are running in single-user mode, skip directly to step 2 below. If you are running in multi-user mode, you must first bring your system to single-user mode. To do this, follow these steps:
 - a. Notify your users that the system is about to be brought down. You can use the *wall* command to do this.
 - b. Change to single-user state by using the *init* command and specifying your system's single-user state number.
 - c. Execute the *killall* command to kill all user processes still active.

Steps a through c above are all performed by the *shutdown* command (if you do not want to perform these steps individually).

- d. If your system is part of a Local Area Network (LAN), you need to bring the network down. Refer to the *LAN/9000 User's Guide* for information on how to do this.
2. If you do not have a current backup of your current file system, **do not continue until you do!** This is the single most important safeguard you can perform before you update your system.
3. From single-user mode, execute *fsck* to check the integrity of your file system. Fix any problems you may encounter.
4. Bring down your current HP-UX system by executing the *stopsys* command with no arguments.
5. Turn off power to your computer.

Phase 2 - Preparing for the Update

1. Remove all removable media (except the update tape cartridge and the 7935 disc drive media module) from your peripheral mass storage devices.
2. If you have cleaned the head on your cartridge tape drive recently, skip to step 3. Otherwise, clean the cartridge tape drive head according to the instructions located in the compartment immediately to the right of the cartridge tape drive. The tape drive head cleaner is also located in this compartment.
3. Make sure the write protect screw on the cartridge tape containing the system update is turned such that its slot points **away** from the word "SAFE" (i.e. **not** "SAFE").
4. Power up your hard disc drive (if you powered it down anytime prior to this step).
5. Once the hard disc drive's buzzer has ceased, insert the cartridge tape containing the update into the cartridge tape drive. You are now ready to proceed with the HP-UX update.

Phase 3 - Performing the Update

At this point, your computer should be powered-down, the hard disc drive should be powered-up, and the cartridge tape containing the update should be inserted in the cartridge tape drive on your hard disc drive. To complete your HP-UX update, wait for the cartridge tape drive's busy light to go out, then follow the steps given below.

1. Power-up your computer. HP-UX finds the kernel on the hard disc, locates the root volume on the cartridge tape, and begins reading from the cartridge tape. HP-UX will read from the cartridge tape for approximately 5 minutes. When the tape has been read, the super-user prompt (#) is displayed.
2. Included on your HP-UX update tape are two commands, one of which must be used to update your system. Their names are *coreupdate* and *sysupdate*. These commands are included only to perform the update procedure, and do not become part of the HP-UX file system. If your update tape contains an update only for the HP-UX kernel, file system, and the C language, or if you only want to update these three items regardless of what else is on the tape, then execute *coreupdate* at this time. If there are updates for optional products included on the tape with the HP-UX kernel, file system, and the C language update, and you want to update these products at the same time you update HP-UX, then execute *sysupdate*. Neither command requires any arguments. Appendix B contains error information for these commands.

Note that if optional product updates are included on your update tape, but you only update the HP-UX operating system, you must later use the *optupdate* command if you want to update these products - see *optupdate(8)* in the HP-UX Reference manual. Be sure **not** to use *sysinstall* or *coreinstall*.

3. Once *coreupdate* or *sysupdate* has begun to execute, make sure that the message

HP-UX FILE SYSTEM UPDATE UTILITY

is displayed. If any other message appears, or if, in particular, the word "INSTALL" appears instead of "UPDATE", abort the utility as soon as possible by pressing the BREAK key.

4. The utility now identifies the locations of the source and destination devices. It displays the select code, bus address, and unit number for the source device, and the select code and bus address for the destination device. If any of these values is wrong, enter

n RETURN

and then enter the correct value(s) by responding to the prompts displayed by the program.

If all the values displayed initially are correct, simply press RETURN.

5. Once the locations of the source and destination devices are established, the utility gives you one last chance to abort before the update proceeds. If you do not have a current backup of your file system, or if anything else is wrong, then abort the procedure, boot up your current version of HP-UX, make your backup (or other corrections), and re-start the update procedure. A backup is **very** important; you should not proceed without it.

If you are sure that everything is ready to begin the update, then enter

y RETURN

The utility then proceeds with the update. At this point, no further interaction is needed. The utility prints out information showing the progress of the update (lists actions being taken, files being copied, etc.). This process can take up to an hour.

6. When the update is complete, the message

THE UPDATE OF HP-UX ONTO THE DESTINATION DISC IS NOW COMPLETE

is displayed. The utility automatically rewinds the cartridge tape for you, and releases the tape. When you hear the buzzer, remove the tape using the slide lever underneath the tape slot. **Turn the write protect screw such that its slot points to the word "SAFE"**, and store the tape in a safe place away from harsh temperatures.

7. Power down your computer, and then power it back up again to boot the new version of HP-UX that now resides on your system disc.
8. If you require any optional drivers (such as Local Area Net, etc.) you should install these drivers at this time. (See the section called "Drivers" in the chapter entitled "The System Administrator's Toolbox".) Once they are installed, repeat step 7 above.
9. Execute *fsck* to make sure that your file system is intact.

The update of your HP-UX Operating System is now complete.

Appendix A

System Loader Messages

The **system loader** is a program which permanently resides in the computer and causes the computer to search for and load an operating system. If the computer is unable to locate and load the TEST System, (or any operating system), a message is displayed. Each of these messages is explained below. Possible causes for many of the messages are provided. If the message begins with **ERROR :**, the system halts after issuing the message. If the message begins with **NOTE :**, the message provides information and the computer continues operating.

If the message you receive indicates a hardware failure, run the Module Self Test or, if present, the System Integrity Test before calling your HP Customer Engineer for service.

Often, the computer attempts to identify the device to which it was "talking" when the message was generated. A value (represented as NNNNN in the messages below) is displayed with the message. You may identify the device with the following procedure.

1. Divide NNNNN by 8; drop the fractional portion to yield an integer result. This value identifies the I/O Processor (IOP) which controls the device to which the computer was "talking" when the message was generated: 2 identifies the first IOP, 3 identifies the second IOP, and 4 identifies the third IOP.
2. Find the I/O card cage controlled by the IOP identified in step 1 above. The first IOP always controls the I/O card cage located on the right side of the computer. The second and third IOPs control I/O expanders.
3. Find the result of NNNNN modulo 8. This value specifies a slot number of I/O card cage identified in step 2 above. Find the interface card occupying this slot. (The top slot in the computer's built-in I/O card cage is slot 2; the first slot in an I/O expander is slot 0.) The computer was talking to this interface card or to a device connected via this interface card when the message was generated.

For example, if NNNNN is 29 then the second IOP is the controlling I/O processor (since the integer portion of $29/8$ is 3 and since 3 identifies the second IOP). The slot number of the I/O card cage is 5 (since $29/8$ is 3 with remainder 5; the result of modulo division is defined to be the integer remainder of the division).

Messages

Loader XXX - informational message identifying the revision of the system loader. This message is usually followed by a single line message identifying the operating system the computer is attempting to load.

Please mount next volume. - informational message. The loader is ready to load another portion of the operating system. Mount the volume containing an unloaded portion of the operating system. Volumes may be mounted in any order without affecting the loading process.

SYSTEM NOT FOUND; WILL RETRY: XXX - unable to find an operating system on any mass storage device. The loader will attempt to find an operating system again in XXX seconds. Possible causes: mass storage device not powered up, no media in mass storage device, wrong disc in disc drive, computer or mass storage device hardware failure, media failure, incompatible loader/system revision numbers, etc.

BAD SYSTEM FILE: NNNNN - operating system loaded; however, an error has been detected in the operating system code during loading. Possible causes: corrupt system, media failure, mass storage hardware failure or computer hardware failure.

INSUFFICIENT USABLE MEMORY: XXXX - The amount of usable memory is too small to load the operating system. The total amount of usable memory available is: XXXX - 32Kbytes. Possible causes: corrupt system or hardware (memory) failure.

BAD CARD OR DEVICE: NNNNN - informational message. A hardware failure has been detected (interface card or mass storage device did not pass Module Self Test). The loader continues searching for an operating system.

DEVICE NOT READY: NNNNN - while loading, the device containing the operating system went "off line".

VOLUME NOT MOUNTED: NNNNN - while loading, the volume containing the operating system was removed from its mass storage device.

DMA FAILED: NNNNN - data did not transfer properly from the mass storage device to the computer. Possible causes: mass storage device hardware failure or computer hardware failure.

UNRECOVERABLE DATA: NNNNN - part of the operating system is not readable. Possible causes: media failure or mass storage hardware failure.

END OF VOLUME: NNNNN - attempt to address or read past the end of a volume. Possible causes: corrupt system, media failure or mass storage device hardware failure.

CTRLR/UNIT FAULT: NNNNN - hardware passed initial self test; however, it failed while being used to load the operating system. Possible causes: computer (interface card) hardware failure or mass storage device hardware failure.

ID TIMEOUT: NNNNN - mass storage device failed to respond fast enough while attempting to load from it. Possible causes: computer hardware failure or mass storage device hardware failure.

CS80 DEVICE: NNNNN - indicates a mass storage device hardware failure.

HPIB CARD: NNNNN - transaction to the indicated HPIB interface card was terminated due to a probable interface card failure.

KBD/SCM NOT FOUND, - indicates a computer hardware failure (keyboard) on a Model 520 Computer; computer hardware failure (system control module) on a Model 530/540 Computer.

BAD ID BUS: NNNNN - indicates a computer hardware failure on the computer's first I/O Processor.

BAD NVM: NNNNN - indicates that NVM (Non-Volatile Memory) failed self test. Possible cause: computer hardware error.

BAD RTC: NNNNN - indicates that the computer's built-in real time clock is not functioning.

BAD SP: NNNNN - indicates that the Model 530/540 Computer's service processor failed self test.

A-4 System Loader Messages

Appendix B

Installation/Update Error Information

This appendix contains a list, by error number, of the possible error messages generated by the *sysinstall*, *sysupdate*, *coreinstall*, *coreupdate*, *optinstall*, and *optupdate* routines.

Whenever an error message is generated by one of these routines, the routine's name, an error number, and some descriptive text appear on your screen. Use the supplied error number as an index into this appendix to find a description of the problem. Note that there are some gaps between defined error numbers denoting error numbers which are unused. Do not confuse this error number with the error indicators *errno(2)* and *errinfo(2)*. There is no correlation between them.



Error Numbers for Sysinstall, Sysupdate, Coreinstall, and Coreupdate

Error Number	Description
2	<i>Mknod</i> (8) failed while attempting to create the block or character special file for the cartridge tape drive or the destination disc. The only possible reason for this is that the special file already exists. If it does not, call your local HP Sales and Support Office.
6	The destination disc could not be initialized. Check the address parameters (select code, HP-IB address) for the destination disc. Also, check the <i>bootsize</i> you asked for - it must be much less than the total size of the disc. Another possibility is that the disc is being used as the virtual memory disc. Finally, the disc could be corrupt. If this is the case, call your local HP Sales and Support Office.
7	The destination disc could not be marked as the HP-UX root volume. Check the address parameters (select code, HP-IB address) for the destination disc. If these values are correct, call your local HP Sales and Support Office for assistance.
8	The destination disc could not be mounted. Either the disc is already mounted, or the disc is not an SDF volume. If neither of these things is true, call your local HP Sales and Support Office.
9	The <i>cd</i> (1) command failed in attempting to change directories to the destination disc after it was mounted. This could indicate a problem with the disc media. Call your local HP Sales and Support Office for assistance.
10	The select code you specified is out of range. The select code must be of the form 0 - 9 (or 00 - 09), or 10 - 23 decimal.
11	The <i>cd</i> (1) command failed in attempting to change directories to the destination disc's <i>/system</i> directory. This should not happen. Call your local HP Sales and Support Office for assistance.
13	The <i>cd</i> (1) command failed in attempting to change directories to the destination disc for kernel unpacking. This should not happen. Call your local HP Sales and Support Office for assistance.
14	The <i>upm</i> (1) command failed in its attempt to unpack the kernel operating system. Clean the cartridge tape drive head and try the update or installation procedure again. If it still does not work, call your local HP Sales and Support Office for assistance.
15	The <i>oscp</i> (8) command failed in its attempt to copy the kernel operating system into the boot area. One possible explanation is that there is no more space in the boot area. If this is not the problem, contact your local HP Sales and Support Office for assistance.
17	The <i>mknod</i> (8) command failed in its attempt to create the character special device file <i>/dev/rhd</i> on the destination disc. This should not happen. Call your local HP Sales and Support Office for assistance.
18	The <i>chmod</i> (1) command failed in its attempt to change the permission bits on the special file(s) <i>/dev/hd</i> and/or <i>/dev/rhd</i> . This should not happen. Call your local HP Sales and Support Office for assistance.

Error Number	Description
19	The <i>mknod</i> (8) command failed in its attempt to create the block special device file <i>/dev/hd</i> on the destination disc. This should not happen. Call your local HP Sales and Support Office for assistance.
23	The HP-UX and C language products were not found on your cartridge tape. You probably received a defective tape. Call your local HP Sales and Support Office to report the problem.
25	The <i>upm</i> (1) command failed in its attempt to copy information from the tape onto the destination disc. Clean the cartridge tape drive head and try the update or installation procedure again. If it still does not work, call your local HP Sales and Support Office for assistance.
26	The <i>cd</i> (1) command failed in its attempt to change directories to the destination disc. This should not happen. Call your local HP Sales and Support Office for assistance.
27	The customization of the specified product has failed. One possible cause is lack of disc space. If this is not the problem, call your local HP Sales and Support Office for help.
29	The <i>umount</i> (1) command failed in its attempt to unmount the destination disc. Review step 1 under "Before You Begin" in this manual's section "Updating Your Operating System". Otherwise, this should not happen. Call your local HP Sales and Support Office for assistance.
30	The <i>oscp</i> (8) command failed in its attempt to copy the kernel operating system for the HP-UX core product. The boot area on the root volume is either too small or non-existent. The minimum boot area size required is 500,000 bytes, but more space may be required if optional products are involved.
31	The update or installation of HP-UX was terminated. The most probable cause of this error is hitting RETURN to select an assumed default answer of 'YES' when the actual default answer is 'NO'. Restart the update or installation procedure.
32	The <i>oscp</i> (8) command failed in its attempt to copy a product's kernel operating system onto the destination disc's boot area. The probable causes are a non-existent boot area, an insufficient boot area size, or a full boot area. Re-initialize the boot area to a larger size (the minimum boot area size is 500,000 bytes). A complete backup of all files that you want to keep is an absolute necessity!
33	The <i>oscp</i> (8) command failed in its attempt to copy a product's kernel operating system onto the destination disc's boot area. This failure is due to some cause other than those listed above for error 32. This error should not happen. Call your local HP Sales and Support Office for assistance.
34	The source and destination device addresses are the same. Check that the addresses you entered are correct; you could have entered an incorrect value by simply pressing RETURN , which selects the default or previous value for that address. Restart the procedure.
35	The <i>cd</i> (1) command failed in its attempt to change directories to the destination device. This should not happen. Call your local HP Sales and Support Office for assistance.
36	A particular script failed for the current product. The only probable cause is a lack of disc space. If this is not the problem, call your local HP Sales and Support Office.
37	A particular product file is not found on the destination disc. This indicates the absence of a file on the cartridge tape that should be there. Call your local HP Sales and Support Office and report the problem.

Error Numbers for Optinstall and Optupdate

Error Number	Description
1	The installation or update procedure was terminated by the user. A possible cause is using an inappropriate affirmative response (the accepted affirmative responses are 'y', 'Y', 'yes', or 'YES'). Also, you could have pressed RETURN thinking that the default was 'YES' when the default was actually 'NO'. Restart the procedure.
2	<i>Mknod</i> (8) failed while attempting to create the block or character special file for the cartridge tape drive or the destination disc. The only probable reason for this is that the special file already exists. If it does not, call your local HP Sales and Support Office.
3	The source and destination device addresses are the same. Check that the addresses you entered are correct; you could have entered an incorrect value by simply pressing RETURN , which selects the default or previous value for that address. Restart the procedure.
4	The <i>oscp</i> (8) command failed while attempting to copy the kernel operating system to the destination disc's boot area. The cause for this is an inadequate boot area size, or a non-existent boot area. One solution is to rebuild the boot area (using <i>osck</i> and <i>oscp</i>) to remove code that is no longer needed. Otherwise, you can re-initialize the disc with a larger boot area (the minimum boot area size is 500,000 bytes). Be sure you have an adequate backup before re-initializing!
5	The <i>oscp</i> (8) command failed while attempting to copy the kernel operating system to the destination disc's boot area. This error number is given when a reason other than that given for error 4 above caused the termination. One possible cause is that the character special file <i>/dev/rhd</i> may not address the root volume. If this is not the problem, call your local HP Sales and Support Office.
8	The destination disc could not be mounted. This error only applies to those situations when the destination disc is not the root volume. Possible causes are: invalid HP-IB address and/or select code for the destination disc, or the disc is busy (i.e. the disc is already mounted). If these causes do not apply, call your local HP Sales and Support Office.
9	The <i>cd</i> (1) command failed in attempting to change directories to the destination disc after it was mounted. This should not happen. Call your local HP Sales and Support Office.
10	The select code you specified is out of range. The select code must be in the range 0 - 9 (or 00 - 09), or 10 - 23 decimal.
13	The <i>cd</i> (1) command failed while attempting to change directories to the destination disc to unpack the product's files. This should not happen. Call your local HP Sales and Support Office.
23	HP-UX and C comprise the core product and cannot be optionally updated or installed by either of these utilities. Use <i>sysinstall</i> , <i>sysupdate</i> , <i>coreinstall</i> , or <i>coreupdate</i> instead.

Error Number	Description
24	The specified product number cannot be found on the tape. Check to make sure that you specified the correct product number (for example, multi-user FORTRAN is entered as 97081A and separated from the command name <i>optinstall</i> or <i>optupdate</i> by a space), and/or check that you have inserted the correct tape into the drive. If this does not solve the problem, call your local HP Sales and Support Office for assistance.
25	The <i>oscp</i> (8) command failed while attempting to copy the product's kernel code segments from the tape to the disc. Clean the cartridge tape drive head and re-start the procedure. If it still fails, call your local HP Sales and Support Office for assistance.
26	The <i>cd</i> (1) command failed while attempting to change directories to the destination disc for customization purposes. This should not happen. Call your local HP Sales and Support Office for assistance.
27	The customization process failed for the specified product number. Check to make sure that you have sufficient disc space. If this is not the problem, call your local HP Sales and Support Office for assistance.
29	The <i>umount</i> (1) command failed while attempting to unmount the destination disc. This error can occur when the destination disc is the virtual memory device. Reconfigure your system such that your destination disc is a device other than the virtual memory device.
30	The <i>umount</i> (1) command failed while attempting to unmount the cartridge tape. This should not happen. Call your local HP Sales and Support Office for assistance.
31	The source mount directory (<i>/optinstall.src</i>) could not be created. One problem may be that this directory already exists. Also, check to see if there is sufficient disc space to create the directory. Finally, you could have a permission problem. You should be logged in as the super-user while running either of these utilities. If the problem persists, call your local HP Sales and Support Office.
32	The cartridge tape could not be mounted. Check to make sure that the source address (HP-IB address, select code) is correct. Also, the tape could already be mounted, or be otherwise busy. If the problem continues, call your local HP Sales and Support Office for assistance.
33	The destination mount directory (<i>/optinstall.dst</i>) could not be created. One problem may be that this directory already exists. Also, check to see if there is sufficient disc space to create the directory. Finally, you could have a permission problem. You should be logged in as the super-user while running either of these utilities. If the problem persists, call your local HP Sales and Support Office.
34	The source mount directory (<i>/optinstall.src</i>) could not be removed. This could be caused by other active users on the system. Single-user mode is recommended. Otherwise, this should not happen. Call your local HP Sales and Support Office for assistance.
35	The destination mount directory (<i>/optinstall.dst</i>) could not be removed. This could be caused by other active users on the system. Single-user mode is recommended. Otherwise, this should not happen. Call your local HP Sales and Support Office for assistance.
36	The cartridge tape block special device file could not be removed. This should not happen. Call your local HP Sales and Support Office for assistance.
37	The destination block special device file could not be removed. This should not happen. Call your local HP Sales and Support Office for assistance.

Error Number	Description
38	The cartridge tape could not be released. This applies only when the unload parameter is specified. This could be caused by other active users on the system. Single-user mode is recommended. Otherwise, this should not happen. Call your local HP Sales and Support Office for assistance.
39	The cartridge tape character special device file could not be removed. This should not happen. Call your local HP Sales and Support Office for assistance.
40	The destination disc's character special device file could not be removed. This should not happen. Call your local HP Sales and Support Office for assistance.
41	The directory <i>/optinstall.src</i> is already present on the tape and should not be. You should either rename or remove it.
42	The directory <i>/optinstall.dst</i> is already present on the destination disc and should not be. You should either rename or remove it.
43	A particular script failed for the specified product. The only probable cause is a lack of disc space. If this is not the problem, call your local HP Sales and Support Office for assistance.
44	The <i>cd(1)</i> command failed while attempting to change directories to the destination disc for execution of a particular script. This should not happen. Call your local HP Sales and Support Office for assistance.

Subject Index

a

absolute address space 36
adding users 77
address 47
ASCII 31
ASI card 9, 63
Asynchronous 8-Channel Multiplex Interface 8,
9
Asynchronous Serial Interface Card 63

b

backing up a file system 3, 82
BASIC 12, 31
BCD 31
BDAT 31
BIN 31
block devices 47
block size 79
block special file 65
boot 49
boot area 8, 12, 16, 49, 80, 87
boot device 86
boot-up 49
booting 86
bus address 47, 48

c

cache buffer size 80
cache buffer size default 93
cache buffers 45, 93
changing a password 89
character devices 47
character special file 65
cartridge tape drives 7
chatr 40, 41
child process 23
chmod 32
chsys 112
CIPERLP.opt 66
code segment 38
communication 90
configuring your system 2, 91
coreinstall 11, 12, 14, 84

coreinstall/coreupdate error messages . . . B-1
coreupdate 119
cpio 83, 84
cron 19, 56
CS/80 discs 7, 10, 50

d

DATA 31
date 19
demand loadable 40
destination device 7, 10, 14
device I/O 47
df 94
dirty pages 44
discs 94
display memory size default 93
drivers 47, 65, 68
du 94

e

effective group ID 25, 33, 34
effective user ID 25, 26, 34
error messages A-1, B-1
execute permission 31
export TZ 20
extent map 28
extents 26
external data segments 38, 43

f

file attributes file 26
file compatibility (HP-UX/BASIC) 31
file format 31
file protection 32
file system 2, 26
file system buffer cache 45
find 94
fork 23, 24
free list 43
free map 27
free pages 44
fsck 3, 19, 56, 61, 96

g

getty	58
global data segment	38
graphics devices	73
group ID	54
guaranteed working set	44

h

hardware installation	1, 9
HP27112A.opt	66
HP7970.opt	66
HP97062A_GR.opt	66
HP98770B_GR.opt	66
HP9885.opt	66
HPIB_FLEX.opt	66

i

i-node	26
ID	23, 25, 34
incremental backup	82
init	24
initialization	12, 17, 79
installing HP-UX	7, 9, 13, 19
installing option products	105
interleave factor	79

k

kernel	8, 18, 55
killall	112

l

ld	40, 41
local time zone default	93
logging in	19
logging out	21
login	18, 49, 54, 57
lsdev	65

m

mass storage devices	70
memory allocation	36
memory blocks	36
memory "discs"	74
MEMORY_VOL.opt	66
mkdev	69
mknod	47, 69, 74, 75
mounting volumes	101
multiple links	29

n

number of cache buffers default	93
---------------------------------------	----

o

optinstall	12, 81, 105
optinstall/optupdate error messages	B-1
option products	87, 105
optupdate	105
osck	65, 66
oscp	66, 87
"other" group	26

p

page	43
page frame pool	43
page size default	93
paged external data segments	43
paged segment	43
page table	43
parent process	23
partitions	36
passwords	89
peripherals	63
printers	72, 77
process	23
process group leader	24
process groups	24
process ID	23
process termination	24
protecting directories	34

r

read ahead level	46
read ahead level default	93
read permission	31
real group ID	25
real user ID	25
removing users	77
restoring a file system	82
root	26, 51
root device	86

s

sample user survey form	1, 5
SDF	31, 79
sdfinit	79
segment number default	93
segment table	37
select code	47, 48
SERIAL.opt	66
set group ID bit	34
set user ID bit	34
setgid	25
setpgrp	24
setuid	25
shared code	39
shutdown	111, 113
source device	7, 10, 13, 14
special files	47, 69
stack segment	38
stack size default	93
sticky bit	35, 36, 39
stopsys	111, 112
Structured Directory Format	31, 79
super-user	26
swap time default	93
sysinstall	11, 14
sysinstall/sysupdate error messages	B-1
system administrator	1, 26
system backup	3
system boot	50
system clock	19, 110
system configuration	2
system console	8, 9, 50
system loader	8, 13, 49
system loader messages	13, A-1
system partition	37
system restoration	3
system segment table	38
system shutdown	111
sysupdate	119

t

tar	83
tcio	84
terminal affiliation	24
terminals	76
time zone	20
time zone default	93
TZ	20

u

uconfig	92
umask	32
unit number	47, 48
unmounting volumes	101, 104
updating	2
updating option products	105
use count	39
used pages	44
user ID	54
user survey form	1, 5
uucp	58

v

virtual memory	41, 42, 100
virtual memory device default	92
virtual memory page frame pool size default	93
volume	31
volume header	31
volume number	47, 48

w

wall	90
working set pages	44
working set ratio default	93
write permission	31

Manual Comment Sheet Instruction

If you have any comments or questions regarding this manual, write them on the enclosed comment sheets and place them in the mail. Include page numbers with your comments wherever possible.

If there is a revision number, (found on the Printing History page), include it on the comment sheet. Also include a return address so that we can respond as soon as possible.

The sheets are designed to be folded into thirds along the dotted lines and taped closed. Do not use staples.

Thank you for your time and interest.