MOTOROLA, INC.
Computer Group
Computer Systems Division
100 Passaic Avenue
One Greenbrock Corporate Center
Fairfield, New Jersey 07006

UNIX® System V/68 and V/88 Release 4

System Administrator's Guide

Part 2: Chapters 8 through 14

R40GUSAG/D2

Preliminary Documentation Packaging

This software was derived by Motorola, Inc. from the UNIX System V, Release 4.0 product owned by UNIX System Laboratories, Inc.

Portions of this software were derived from STREAMware TCP developed by INTERACTIVE Systems Corporation and Convergent Technologies, Inc.

IMPORTANT NOTE TO USERS

While every effort has been made to ensure the accuracy of all information in this document, UNIX System Laboratories, Inc. (USL) assumes no liability to any party for any loss or damage caused by errors or omissions or by statements of any kind in this document, its updates, supplements, or special editions, whether such errors are omissions or statements resulting from negligence, accident, or any other cause. USL further assumes no liability arising out of the application or use of any product or system described herein; nor any liability for incidental or consequential damages arising from the use of this document. USL disclaims all warranties regarding the information contained herein, whether expressed, implied or statutory, *including implied warranties of merchantability or fitness for a particular purpose.* USL makes no representation that the interconnection of products in the manner described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting or license to make, use or sell equipment constructed in accordance with this description.

USL reserves the right to make changes without further notice to any products herein to improve reliability, function, or design.

TRADEMARKS

Delta Series, DeltaSERVER, and M88000 are trademarks of Motorola, Inc.
Epson FX-86e is a trademark of Epson America, Inc.
HP and LaserJet are registered trademarks of Hewlett-Packard Company.
IBM is a registered trademark of International Business Machines.
Motorola and the Motorola symbol are registered trademarks of Motorola, Inc.
PostScript is a registered trademark of Adobe Systems, Inc.
Proprinter is a trademark of International Business Machines.
STREAMware is a trademark of INTERACTIVE Systems Corporation.
Teletype is a registered trademark of AT&T.
UNIX is a registered trademark of UNIX System Laboratories, Inc. in the U.S.A. and other countries.

8 Multi-Processing Terminology and Basic Commands

DRAFT COPY
February 1, 1992
File: Cmultiproc

Introduction

UNIX System V Release 4 is an interactive, multi-user, multi-tasking operating system. It allows multiple users to be active at the same time, while executing multiple applications concurrently. The operating system serializes execution of these tasks to achieve an effective, concurrent, multi-tasking environment on a single processor system. UNIX System V Release 4 Multi-Processing continues the evolution of this multi-tasking technology to take advantage of shared-memory base, multi-processing systems. It allows multiple tasks to execute in parallel on the processors comprising the multi-processing system.

This chapter introduces you to the basic terminology and commands you will use to set up, operate, and adjust the multi-processor.

Terminology

This section describes terminology and concepts you will be seeing throughout this chapter. Some of the terms may be familiar and some may be new to you as a system administrator for a UNIX System V Multi-Processing system. The terminology is listed in alphabetical order and is identified as predominant characteristics of the UNIX System V Release 4 Multi-processor, and of multi-processing in general. Some of them extend beyond the boundaries of use for a system administrator, but are included to provide a full picture of multi-processing technology.

concurrent process — UNIX System V Release 4 provides the environment for multiple processes to execute concurrently. Logically, the processes run independently and asynchronously. In reality, the operating system time slices the processor between them and offers mechanisms to synchronize their execution. With UNIX System V Release 4 Multi-Processing, concurrent processes can execute in parallel.

DDI/DKI — Acronyms identifying the Device Driver Interface and the Driver-Kernel Interface. These interfaces specify the interactions between a device driver or STREAMS module and the rest of the UNIX System V kernel.

granularity — Multi-Processing applications can be structured into functions or tasks for concurrent processing. The degree of decomposition or separation is regarded as the granularity of the application.

coarse-grained - A form of multi-processing that generally executes tens of thousands of instructions between synchronization points.

medium-grained - A form of multi-processing that generally executes thousands of instructions between synchronization points.

fine-grained - A form of multi-processing that generally executes hundreds of instructions between synchronization points.

kernel — The UNIX operating system kernel manages system resources and controls user processes and commands. The kernel, a permanent part of the memory, manages the computer's resources (processor time, memory and I/O devices) so that multiple programs can execute simultaneously.

multi-processing

The UNIX system is a multi-tasking operating system. This means that multiple programs can execute concurrently with each other, time-sharing the compute and input/output resources in the system. On a uniprocessor system, there is only one compute element: the central processing unit (CPU) or processor. The processor executes one task at a time. With multi-processing systems, there are multiple compute elements in the system. Each processor executes a different task, thus making the System V Release 4 Multi-Processor a multi-processing operating system.

multi-processor

A computer with one processor executing applications is a uniprocessor. A computer with two or more processors that share common memory and peripherals to execute multiple programs simultaneously is a multi-processor.

multi-tasking

Multitasking allows you to share both resources and data while simultaneously executing commands and running programs. You can also run programs in the background while working on another job. All UNIX systems are multi-tasking operating systems.

multi-threaded

Codes or algorithms are multi-threaded when different processors can independently and concurrently execute the same code sequence while preserving the integrity of shared data.

offline

Taking a processor offline notifies the kernel that the processor is not available for use. The processor is not considered offline until any currently executing process has been moved and the processor is in an idle state. See `offline`(1M).

online

An online processor is in a state of availability and can execute applications. See `online`(1M) and `p_online`(2).

process

A process is a normal UNIX System V operating system execution entity. Each process has its own address space and context and shares the system with other processes in the system.

process binding

A process can be bound so that it will execute only on a specified processor. Use process binding for load balancing purposes or when processes require computer or input-output resources local to that processor. A processor with processes exclusively bound to it is not free to run other processes that are not exclusively bound to it, while a non-exclusively bound processor is free to run other, non-bound processes.

Note that some uniprocessor applications may have built-in uniprocessor timing assumptions that are no longer valid in the multi-processing environment. You should change these applications by removing the timing assumptions. Alternately, you can bind all processes belonging to the application to one processor to preserve these timing assumptions.

The `pbind`(1M) command or the `processor_bind`(2) system call can be used to bind a process to a processor. Processes can be exclusively bound to a processor with the `pexbind`(1M) command.

processor

The hardware component performing the actual program execution. By default, the processor on which the Release 4 Multi-Processor boots is assigned a processor ID of 0.

processor ID

The unique identification for each processor in the system.

symmetric and asymmetric multi-processor

If a multi-processing platform is designed in such a way that all I/O devices are directly accessible to all processors, then the multi-processing platform is said to support symmetric I/O. If there are I/O devices that are visible to some, but not to all processors, then the multi-processing platform is said to be asymmetric. To access an I/O device in an asymmetric multi-processing platform, the operating system has to switch the program out of the processor it is currently executing on and re-dispatch it to a processor that has access to the I/O device.

Setting Up Multi-Processing

UNIX System V Release 4 Multi-Processing boots initially on one processor. By default, this boot processor is assigned a processor ID of 0. During boot, the system uses a platform-dependent mechanism to scan the hardware for additional processors that are configured on the system. This information is reported on the system console.

After the kernel has finished booting, you should follow the set up procedure described in the "System Setup" chapter of this guide to set up your system requirements. In addition, you may need to bring the other processors in the system online. By default, the other processors in your multi-processing platform are brought online at system boot time. Optionally, the system may be configured to boot only specified processors. For more information on configuring certain processors to start at boot time, see "Processor Configuration" in this chapter.

Managing Processors

The basic multi-processing commands are divided into two groups. As described in this section, the first group manages the state of the processors configured in your system. The online(1M) and offline(1M) commands allow you to bring the processors online or offline, and the pinfo(1M) command allows you to query the system about the state of the processors. The second group, as described in the section "Process Binding," manages the relationship between processes and processors. For example, the pbind(1M) command allows you to specify the processor on which a process will execute.

Several of the multi-processing functions can be executed either through the command interface or the programming (system call) interface. The command interface and the system calls are described in this chapter.

NOTE Super-user privileges are necessary to run most of these commands.

Taking Processors Online and Offline

Processors are identified with a processor ID number that gives them a unique tag within the system. You can examine the state of the processors in your system by using the pinfo(1M) command or the processor_info(2) system call. pinfo(1M) reports whether the processor is online or offline. See pinfo(1M) and processor_info(2).

When a processor is online, it is free to dispatch processes and perform normal operating system functions. You can bring a processor online with the online(1M) command or the p_online(2) system call. Taking a processor offline removes the processor from operational status. The processor retains its place in the system but does not perform any useful operations while it remains idle.

Only processors with no bound processes can be taken offline. Some hardware platforms may require some processors to be online at all times. A processor can be taken offline with the offline(1M) command or the p_online(2) system call. Once a processor has been taken offline, it remains idle until it is brought back online using the online(1M) command or p_online(2) system call. See online(1M) and p_online(2).

Basic Procedure

The online command brings a processor online.

```
$ online
```

online with no arguments brings all processors in the system online. The command is silent by default. If the -v option is specified, the online command will report the original and the new operating status of each processor. For example:

```
$ online -v

  processor 0: on -> on
  processor 1: off -> on
  processor 2: off -> on
  processor 3: off -> on
```

If you want to enable the processors individually, you must specify the processor IDs as arguments on the command line.

```
$ online 1 3
```

online 1 3 brings only processors 1 and 3 online. The operating status of processor 2 remains unchanged.

The offline(1M) command is used to take a processor offline. As with the online command, it takes all processors offline if no argument is specified. Your multi-processing platform may require some processors to be online all the time. If that is the case, the offline command will report the processor as being busy and skip that processor. For example:

```
$ offline -v

  processor 0: busy
  processor 1: on -> off
```

You can query the status of the processors in your system by using the pinfo(1M) command. The pinfo command will reveal which processors have been brought online and which are idle or offline. For example:

```
$ pinfo

  processor 0: online
  processor 1: offline
```

where processors 0 and 1 are shown as online in the system. See online(1M), offline(1), and pinfo(1M).

Processor Configuration

When the system boots, processor 0 (the boot processor) performs kernel initialization. Once the kernel is initialized, the boot processor automatically starts any other processors that the system administrator has designated to start on boot. You may designate which processors are to boot by modifying the mp master file in the /etc/master.d directory.

The mp master file contains an array of integers associated with the variable bootstart. The ordinal position in the array (beginning with 0) identifies the CPU to which the value applies. A value of 1 indicates that the processor is to be started on boot. A value of 0 indicates that the processor should not be started. Processor 0 always starts regardless of its bootstart value.

To change which processors start on boot, edit the mp master file and follow the procedures for configuring the UNIX Operating System found in the "Performance Management" chapter of this guide.

Process Binding

Binding Processes to Processors

By default, a process can execute on any processor in the system. You can use the pbind(1M) command or the processor_bind(2) system call to make a process execute only on a specified processor. The same processor must, by default, also execute all child processes or timeout routines generated by the bound process. The -P option of ps(1) can be used to display the processor binding status of all processes in the system. The pbind(1M) command can also be used to unbind a process from a processor. Once unbound, the process can execute on any processor in the system. See pbind(1M), ps(1), and processor_bind(2).

Placing Processors in an Exclusive Binding Mode

You can use the pexbind(1M) command to place a processor in an exclusive binding mode so that it will execute only processes bound to it. Even when it is idle, the processor will not execute any other process unless an outside process requires a resource available only in the exclusively-bound processor. For example, if a device driver is exclusively bound to the processor, and a process requires the service of this driver, the operating system will bind the process to the processor just for the duration of the driver routine. Once the driver operation is complete, the operating system will unbind the outside process from the processor and restrict itself again only to processes that are bound to the processor.

In addition, processes that are exclusively bound to a processor are restricted to run only on that processor. See pexbind(1M) and ps(1).

System Activity Commands

This section outlines the uses of four system activity commands: sadc(1M), sar(1M), profiler(1M), and ps(1). The system activity reporting commands, sadc(1M) and sar(1M), sample, save, process and report system activity data. You can use them interactively during periods of poor system response time to check the activity of the system. sadc(1M) and sar(1M) specify command line options, and report global data such as the page fault rate. You can collect additional data, such as the %wio, (idle with some process waiting for block I/O), on a per processor basis. The profiler allows you to examine a running UNIX system, and the ps command prints information about active processes.

All four of these commands support a -P option, which allows you to request information about specific processors.

sar

By controlling the command options, you can generate a variety of reports with sar(1M), which provides a simplified command interface. By default, the reports display global system activity data only. When you specify the -P option, sar will ignore all command line options that request information not specific to that processor.

sadc

sadc(1M) provides a package of administrative utilities that you can use to automate the process of retrieving data. The sadc utility extracts and collects system activity data from the kernel and then writes records to the standard output or to a file specified in the command line. You can add the shell script sa1, a variant of sadc, to the crontab file to collect and store system activity records in binary format. The shell script sa2, a variant of sar, will generate a system activity report which, by default, will include only global data. You may use the -P option to request a report for a specific processor, in which case sa2 will ignore all command line options that request information not specific to that processor.

profiler

The profiler(1M) provides a system of programs to examine a running UNIX operating system. Use these programs to load symbol information into a running kernel, enable or disable the sampling mechanism, perform data collection functions, and format and print the data collected. The reports generated list the entry for all kernel procedures accessed during the profiling session and the percentage of system time the kernel spent in each procedure. By default, system-wide totals are reported. You can add the command option -P to request profiling information per processor.

ps

The ps(1) command prints information about active processes. If you add the -P option, ps will print the processor ID of the processor on which each process is running.

UNIX System Parameters

UNIX System V Release 4 Multi-Processing uses a number of tunable parameters to control the sizes of various data structures inside the kernel. These parameters are initialized to default values when UNIX System V is first installed into your system and may not necessarily be ideal for your application and environment.

Before changing any parameters, use the sadc command to analyze the performance of your current environment. The parameters have been chosen for a representative mix of activities that may not necessarily match your applications. You can change the kernel parameters to values that better match your application and memory size, and realize a gain of up to 25% for non-processor bound jobs with no noticeable degradation in other areas. A detailed discussion of all tunable parameters can be found in the section on tunable parameters in the "Performance Management" chapter in this guide.

This section describes the following parameters, including why they are important, how you can change their values for better performance, and what values you should use.

- NPROC
- BUFHWM
- NINODE
- UFSNINODE
- NHBUF
- NPBUF

NOTE Be especially careful when changing these parameters. You should consider the workload of your system to determine to what extent, if any, these changes should be made.

NPROC

NPROC defines the size of the directory name lookup cache (DNLC). The value of NPROC can be raised as long as it is about 200 less than NINODE or UFSNINODE. If NPROC is tuned high, so is the DNLC size. A higher DNLC size, for a given number of incore inodes (NINODE or UFSNINODE), results in fewer incore inodes on the free list. The consequence of a smaller free list is that purging the DNLC causes allocation of inodes. A DNLC purge is, in general terms, bad for performance because it causes more overhead in pathname lookups.

BUFHWM

BUFHWM is the amount of memory that can be used by block I/O buffers. It forms a buffer cache that allows the I/O subsystem to make large, organized transfers to minimize disk seek time and to avoid reading or writing to disk if the data required has already been loaded into the buffer.

Setting BUFHWM too low will cause excess disk I/O to flush the buffer before it can be re-used. Setting BUFHWM too high will reduce the page pool size and can increase paging. In general terms, BUFHWM should grow proportionately with memory size.

If the system being used is heavily processor-bound, set BUFHWM to no less than 5% of available memory (total memory minus kernel executable size). For general systems, setting BUFHWM to 12.5% of available memory is reasonable. If the system performs excessive file system activity, try bumping this to 25% of available memory. If the X Windows System is being used, memory is at a premium and BUFHWM should be reduced to an acceptable minimum (based on system use). For example, in a system with 16 megabytes and no windowing, a reasonable value of BUFHWM is 2048.

NINODE

This parameter is important when an S5 file system type is configured into your system. NINODE specifies the number of inode entries in the memory-resident S5 inode table. An S5 inode is a data structure that typically describes a file, directory, link and named pipe in an S5 file system type. Too few incore inodes would cause

- a poor bcache read/hit ratio
- a higher ipf (percent of S5 inode page-free activity)
- a poor page attach rate

These values represent the number of recycled inodes with one more valid page associated with inodes, and/or the poorer page attach rate. The percentage of recycled S5 inodes with one or more valid pages associated with the inode is represented by ipf. All three of these system statistics are reported by the sar(1M) command. NINODE should be incremented based on memory size.

UFSNINODE

UFSNINODE performs a function similar to NINODE except that it specifies the number of inode entries in the UFS inode table. When a UFS file is configured in your system, increment UFSNINODE based on memory size.

NHBUF

NHBUF specifies the size of the hash table used to locate a buffer, given a device number and a block number. The default value is 64 and the value must always be a power of 2. This value should be about one quarter of the total buffers available. A value between 1/8 and 1/4 of BUFHWM is typically sufficient.

NPBUF

NPBUF specifies how many physical I/O buffers to allocate; one is needed for each active physical read or write. The default value is 20. There is no hard and fast rule for adjusting NPBUF. However, if you anticipate a lot of I/O and file system activity, try raising NPBUF. In test cases, raising NPBUF to 40 improved system performance.

Parameters Specific to UNIX System V Release 4 Multi-Processing

The following table lists typical tuning parameter values for different memory sizes. You should investigate the workload of your system carefully before changing any parameters.

Table 8-1: Multi-Processing Parameters

Parameters:	4MB	8MB	12MB	16MB	32MB	64MB
NPROC	200	250	250	250	400	400
BUFHWM	100	400	1024	1024	2048	6144
NINODE	400	500	600	600	800	1000
NHBUF	100	400	400	800	2048	4096

If you are using a UFS file system, change UFSNINODE to the NINODE value specified above.

Modifying the Kernel

Modifying kernel parameters is a fairly simple task, though you should always insure that there is a functioning kernel that can be used to boot the system. For details, see "Tunable Parameters" in this guide.

9 Network Services

Introduction

This chapter describes the administrative command-level interface for Network Selection and the Basic Networking Utilities (BNU). Distributed File Systems (RFS and NFS) are documented in the *Network User's and Administrator's Guide.*

Network Selection and the Service Access Facility (SAF) grew out of the need for UNIX Systems to communicate both with other UNIX systems and with non-systems. Previous to UNIX System V/68 or V/88 Release 4, it was difficult for network applications to find out what transport providers were available on a given machine and to find the addresses of services. The Network Selection mechanism and the Name-to-Address Mapping facility described in this section provide both a consistent way to determine what transport providers are installed on a machine and a standardized mechanism for finding service addresses.

The new Network Selection feature described in this chapter generalizes the procedure by which an application chooses the network it connects to. This procedure is implemented as a set of library routines for inclusion in application programs. The system administrator is responsible for maintaining the network configuration database file (`/etc/netconfig`) used by these routines. He/she is also responsible for creating and maintaining the "host" and "service" files required for each of the Name-to-Address Mapping libraries.

The "Network Selection" section of this chapter describes the `netconfig` file and the NETPATH environment variable, which may be used by both users and the system administrator to customize the default list of networks an application tries to connect to. The "Name-to-Address Mapping" section describes the Name-to-Address Mapping files that must be in place before applications can use the Name-to-Address Mapping libraries. The library routines available to application programmers are described in the *Programmer's Guide: Networking Interfaces* on the `getnetconfig`(3N) manual page and on the `netdir`(3N) manual page.

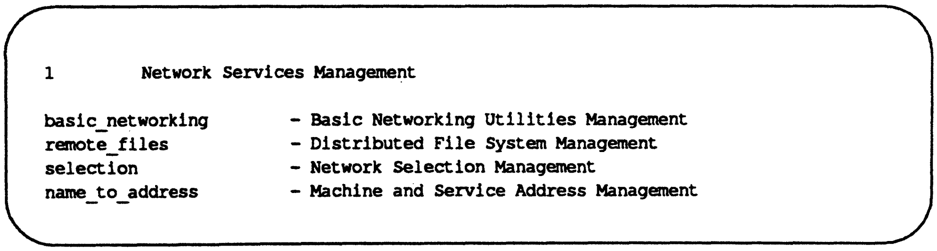
The remaining sections of this chapter describe the Basic Networking Utilities (BNU).

You can do any of the functions associated with Networking Services administration by selecting the appropriate "task" from a series of menus provided for administration. To access the "system administration" menu for using Networking Services, type:

```
        sysadm network_services
```

The following menu will appear on your screen:

Figure 9-1: Network Services Management Menu



```
1        Network Services Management

basic_networking       - Basic Networking Utilities Management
remote_files           - Distributed File System Management
selection              - Network Selection Management
name_to_address        - Machine and Service Address Management
```

NOTE

The `Distributed File System Management` option is described in the *Network User's and Administrator's Guide*

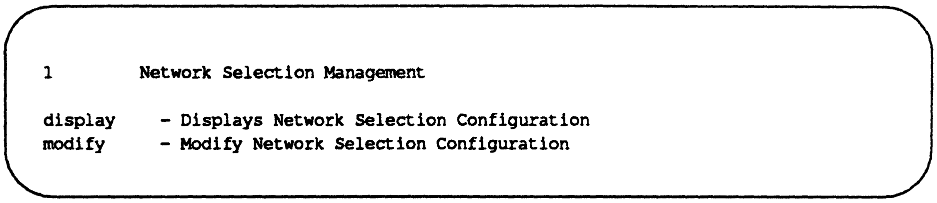
When you have selected the option you want, the submenus and instructions displayed on the screen are self-explanatory and lead you through the appropriate procedures.

Network Selection

In order for network applications to be portable to different environments, the application process must have a standard interface into the various networks available in any current environment. Network Selection provides a simple and consistent interface that allows user applications to select networks (at the transport level), enabling applications to be protocol- and media-independent. System V Networking Services applications that allow the user to influence the choice of networks use the standard interface outlined here.

Tasks associated with Network Selection administration may be performed using either the menu system or shell commands entered on the command line. The screen below is the top-level menu for Network Selection. It can be brought up on the screen by typing `sysadm network_services/selection.`

Figure 9-2: Network Selection Management Menu



```
1         Network Selection Management

display    - Displays Network Selection Configuration
modify     - Modify Network Selection Configuration
```

When you select an option, self-explanatory submenus and instructions lead you through the appropriate procedures.

You can also bypass the menu system by issuing commands directly to the shell. Where these commands involve editing sensitive system files, be sure to keep a backup copy of the file you are editing. When you have finished editing the file, use `diff(1)` on the edited file and the backup copy to verify that only the changes you want have been made.

Table 9-1: Command Alternatives to the Network Selection Management Menu

Task Description	Menu Item	Shell Command
Add networks to the network configuration database file `/etc/netconfig`	add	`vi /etc/netconfig`

Display the contents of the `netconfig` file; display the entry for network *netid*	display	`cat /etc/netconfig`
		`grep` *netid* `/etc/netconfig`

Change a `netconfig` entry	modify	`vi /etc/netconfig`
Delete a `netconfig` entry	remove	`vi /etc/netconfig`

Network Selection Overview

The UNIX System V/68 or V/88 Release 4 Network Selection component is built around:

- a network configuration database (the `/etc/netconfig` file) that contains entries for each network available to the system, and
- an optional `NETPATH` environment variable, set by a user or the system administrator, containing an ordered list of network identifiers. These network identifiers match the `netconfig` *network ID* field and are used as links to the records in the `netconfig` file.

The Network Selection application programming interface consists of a set of network configuration database access routines. One group of these library routines accesses only the `netconfig` entries identified by the `NETPATH` environment variable; another group of routines accesses `netconfig` directly. The routines are described in the *Programmer's Guide: Networking Interfaces*. The first group is also described in detail in the manual page getnetpath(3N). The second group is described in getnetconfig(3N).

Applications should use the routines that access NETPATH. They allow users to influence the selection of transports used by the application. If an application does not want the user to influence its decision, then the routines that access the netconfig database directly should be used.

The netconfig file, on which the Network Selection library routines depend, is maintained by the system administrator. The NETPATH environment variable is typically set or modified by application programmers and users, depending on the needs of their applications, but it may also be set by the system administrator in response to the needs of administrative applications.

The netconfig File

The system administrator is responsible for maintaining the network configuration database file /etc/netconfig. Entries in the netconfig file contain the following fields, in the order shown:

Table 9-2: Fields in netconfig Entries

network ID	*semantics*	*flag*	*protocol family*	*protocol name*	*network device*	*directory lookup libraries*

The fields correspond to elements of the struct netconfig structure. Pointers returned by Network Selection library routines are pointers to netconfig entries in struct netconfig format. The netconfig file is described in the manual page netconfig(4). The netconfig manual page also describes the elements of the struct netconfig structure. All symbolic names, structure definitions, and constant values for the Network Selection feature are defined in the header file /usr/include/netconfig.h.

`netconfig` fields are defined as follows:

network ID

The *network ID* field is a string used to identify a network. *network ID* consists of non-NULL characters, and has a length of at least 1. No maximum length is specified. This namespace is locally-significant and the local system administrator is the naming authority responsible for ensuring that all *network IDs* on a system are unique.

semantics

The *semantics* field is a string that identifies the "semantics" of the network, that is, the set of services it supports, by identifying the service interface it provides. This is closely related to, but not identical with, the API (Application Programming Interface) with which applications are "supposed" to access the network. Typically, an application will specify its API by pushing an appropriate STREAMS module (such as `timod`) and using an appropriate user-level library (such as the TLI library). The *semantics* field is mandatory. The following semantics are recognized.

`tpi_clts`	Transport Provider Interface, connectionless
`tpi_cots`	Transport Provider Interface, connection-oriented
`tpi_cots_ord`	Transport Provider Interface, connection-oriented and supports orderly release
`tpi_raw`	Transport Provider Interface, raw usage

flag

The *flag* field records certain two-valued ("true" and "false") attributes of networks. *flag* is a string composed of a combination of characters, each of which specifies the value of the corresponding attribute. If the character is present, the attribute is "true." If the character is absent, the attribute is "false." A hyphen (`-`) specifies that none of the attributes is present. Only one character is currently recognized:

`v`	Visible ("default") network. Used when the environment variable `NETPATH` is *unset*.

protocol family The *protocol family* and *protocol name* fields are provided for protocol-specific applications. The *protocol family* field contains a string that identifies a protocol family. The *protocol family* identifier follows the rules for *network IDs*, which are:

- The string consists of non-NULL characters.
- It has a length of at least `1`.
- There is no maximum length specified.

A hyphen (-) in the *protocol family* field indicates that none of the available protocol family identifiers applies, that is, the network is experimental. An application that wants to have family characteristics can match on the *protocol family* field when selecting a network (for example, an application can search for an "osi" family). In this case, the application is not protocol independent, since it has searched only for OSI entries. The following are examples of protocol family identifiers:

`loopback`	Loopback (local to host)
`inet`	Internetwork: UDP, TCP, etc.
`implink`	ARPANET imp addresses
`pup`	PUP protocols: for example, BSP
`chaos`	MIT CHAOS protocols
`ns`	XEROX NS protocols
`nbs`	NBS protocols
`ecma`	European Computer Manufacturers Association
`datakit`	DATAKIT protocols
`ccitt`	CCITT protocols, X.25, etc.
`sna`	IBM SNA
`decnet`	DECNET
`dli`	Direct data link interface

`lat`	LAT
`hylink`	NSC Hyperchannel
`appletalk`	Apple Talk
`nit`	Network Interface Tap
`ieee802`	IEEE 802.2; also ISO 8802
`osi`	Umbrella for all families used by OSI (for example, `protosw` lookup)
`x25`	CCITT X.25 in particular
`osinet`	AFI = 47, IDI = 4
`gosip`	U.S. Government OSI

protocol name

The *protocol name* field contains a string that identifies a protocol. This field is currently only used for the `inet` family. For any other family, the protocol name field contains a hyphen (-). The *protocol name* identifier follows the same rules as *network IDs*:

- The string consists of non-NULL characters.
- It has a length of at least `1`.
- There is no maximum length specified.

The *protocol name* field may contain:

`icmp`	Internet Control Message Protocol
`tcp`	Transmission Control Protocol
`udp`	User Datagram Protocol

network device

The *network device* is the full pathname of the device used to connect to the transport provider. Typically, this device will be in the `/dev` directory. The *network device* must be specified.

directory lookup libraries

The *directory lookup libraries* support a "directory service" (that is, a Name-to-Address Mapping service) for the network. This service is implemented by the UNIX System V Name-to-Address Mapping feature. If a network is not provided with such a library, the Name-to-Address Mapping feature will not work. A hyphen (-) in this field shows that lookup libraries, and therefore Name-to-Address Mapping, are unavailable.

The *directory lookup library* field consists of a comma-separated list of full pathnames to dynamically linked libraries. Literal commas may be embedded as "\,"; backslashes as "\\". Lines in /etc/netconfig that begin with a pound sign (#) in column 1 are comments.

The system administrator determines the *order* of the entries in the netconfig database. Since the Network Selection library routines that access netconfig directly return entries in order, beginning at the top of the /etc/netconfig file, the order in which networks are entered in the file by the system administrator becomes the default search path for applications choosing networks to connect to.

Figure 9-3: Sample netconfig File

```
starlan     tpi_cots      v  osinet     -    /dev/starlan    /usr/lib/straddr.so
starlandg   tpi_clts      v  osinet     -    /dev/starlandg  /usr/lib/straddr.so
npack       tpi_cots      v  localnet   -    /dev/npack      /usr/lib/npack.so
tcp         tpi_cots_ord  v  inet       tcp  /dev/tcp        /usr/lib/tcpip.so
udp         tpi_clts      v  inet       udp  /dev/udp        /usr/lib/tcpip.so
ticlts      tpi_clts      v  loopback   -    /dev/ticlts     /usr/lib/straddr.so
ticots      tpi_cots      v  loopback   -    /dev/ticots     /usr/lib/straddr.so
ticotsord   tpi_cots_ord  v  loopback   -    /dev/ticotsord
```

The NETPATH Environment Variable

Network Selection is unobtrusive. In most cases the user is not interested in which network handles a network operation, and the default network search path established by the system administrator (the netconfig file) is used to locate a network available for connection. However, if a user or the system administrator wants to influence the choices made by applications, the search path can be modified using a new standard shell variable, NETPATH. NETPATH is similar to the

PATH variable.

NETPATH consists of a colon-separated list of network IDs. Each network ID corresponds to the *network ID* field of a record in the netconfig database. A literal colon can be embedded as "\:", and a literal backslash as "\\". An empty component in NETPATH, signified by either a beginning colon, an ending colon, or two successive colons, is not a valid entry, since the empty string is not a valid network ID. NETPATH is described in the environ(5) manual page in the *User's Reference Manual*.

The NETPATH environment variable is not set in /etc/profile. It can, however, be set in a user's $HOME/.profile.

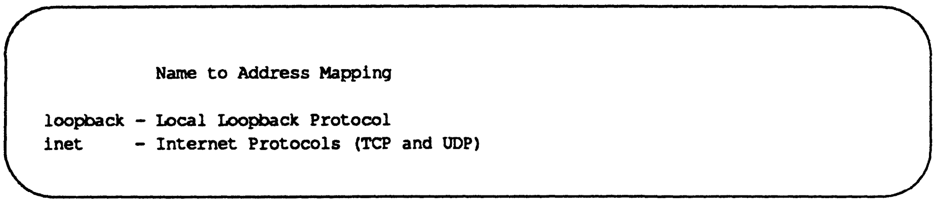
Users and system administrators alike should be aware that the set of "default" networks is different for the routines that access netconfig directly and the routines that access netconfig via the NETPATH environment variable. For the routines that access netconfig directly, the set of "default" networks is the entire netconfig file; the set of "default" networks for the routines that access netconfig via NETPATH is the "visible" networks in the netconfig file. A network is "visible" if the system administrator has included a v flag in the flag field. If NETPATH is unset, these "visible" networks are the default search path for this second group of access routines.

Name-to-Address Mapping

The Name-to-Address Mapping feature allows an application to obtain the address of a service on a specified machine in a transport-independent manner.

Tasks associated with Name-to-Address Mapping administration may be performed using the menu system or shell commands entered on the command line. The screen below is the top-level menu for Name-to-Address Mapping. It can be brought up on the screen by typing `sysadm name_to_address`.

Figure 9-4: Machine and Service Address Management Menu



```
           Name to Address Mapping

loopback - Local Loopback Protocol
inet     - Internet Protocols (TCP and UDP)
```

When you have selected the option you want, self-explanatory submenus and instructions will be displayed on the screen to lead you through the appropriate procedures.

If you want to bypass the menu system, you may issue commands directly to the shell, as shown in Table 9-3.

Table 9-3: Shell Commands for Name-to-Address Mapping

Task Description	Menu Item	Shell Command

Local Loopback Protocol	`loopback`	`vi /etc/net/`*transport*`/hosts`
		`vi /etc/net/`*transport*`/services`

Internet Protocols (TCP and UDP)	`inet`	`vi /etc/hosts`
		`vi /etc/services`

Name-to-Address Mapping consists of routines for use by application programs. These routines (which are described on the `netdir`(3N) manual page) are used to obtain addresses of services on given hosts. All routines are combined into a library, one for each transport provider. The library to use for a specific transport provider is named in the `/etc/netconfig` file in the entry for each network. The netdir_getbyname() routine dynamically links the library named in the *directory lookup libraries* field of the /etc/netconfig file.

The routines are:

```
netdir_getbyname
netdir_getbyaddr
netdir_free
netdir_mergeaddr
taddr2uaddr
uaddr2taddr
netdir_options
```

Each function takes a pointer to a netconfig structure and returns a list of addresses of the service and hostnames over a given transport provider. The library tcpip.so provides all the preceding Name-to-Address Mapping routines for the TCP/IP protocol suite.

Setting Up the Name-to-Address Mapping Libraries

Files for each of the libraries must be created and maintained by the system administrator. Each entry in the /etc/hosts file must contain at least two fields: the machine's IP address and the machine's official hostname. A list of aliases may follow the official hostname. For example:

```
192.11.108.01       bilbo
192.11.108.16       elvis.mot.com elvis hound_dog
```

The /etc/services file contains two fields, a service name and a port number with one of two protocol specifications, either tcp or udp. For example:

```
rpcbind         111/udp
rpcbind         111/tcp
login           513/tcp
listen          1025/tcp
```

For an application to use this library to request the address of a service on a particular host, the hostname must appear in the /etc/hosts file and the service name must appear in the /etc/services file. If one or the other does not appear, an error will be returned by the name to address mapping routines.

straddr.so

The routines in the straddr.so dynamic library create addresses from files that have the same format as the tcpip.so file described above. The straddr.so files are /etc/net/*transport*/hosts and /etc/net/*transport*/services. *transport* is the local name of the transport provider that accepts string addresses (specified in the *network ID* field of the /etc/netconfig file). For example, the host file for ticlts would be /etc/net/ticlts/hosts, and the service file for ticlts would be /etc/net/ticlts/services. For ticots, the files would be /etc/net/ticots/hosts and /etc/net/ticots/services.

Even though most string addresses do not distinguish between "host" and "service," separating the string into a host part and a service part provides consistency with other transport providers. The /etc/net/*transport*/hosts file will therefore contain a string that is considered to be the machine address, followed by the machine name. For example:

```
bilboaddr       bilbo
elvisaddr       elvis
frodoaddr       frodo
```

The /etc/net/*transport*/services file contains a service name followed by a string identifying the service port. For example:

```
rpcbind        rpc
listen         serve
```

The routines create the full string address by combining the "host address" and the "service port," separating the two with a dot (.). For example, the address of the "listen" service on bilbo would be bilboaddr.serve and the address of the "rpcbind" service on bilbo would be bilboaddr.rpc.

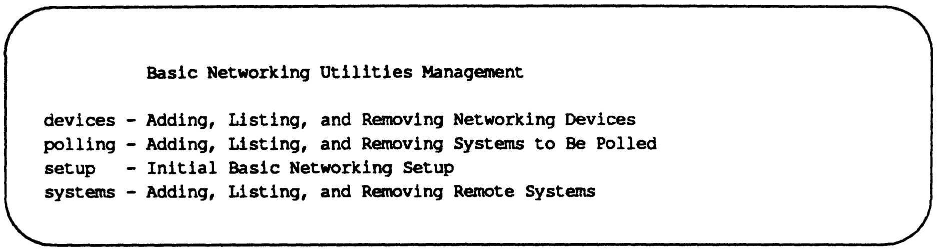
When an application requests the address of a service on a particular host on a transport provider that uses this library, the hostname must appear in /etc/net/*transport*/hosts and the service name must appear in /etc/net/*transport*/services. If one or the other does not appear, the Name-to-Address Mapping routines return an error.

Basic Networking Utilities

The Basic Networking Utilities Package (BNU) lets computers using the UNIX operating system communicate with each other and with remote terminals. These utilities range from those used to copy files between computers (uucp and uuto) to those used for remote login and command execution (cu, ct, and uux).

Tasks associated with Basic Networking Utilities administration may be performed using either the menu system or shell commands entered on the command line. The screen below is the top-level menu for BNU selection. It can be brought up on the screen by typing sysadm basic_networking.

Figure 9-5: The Basic Networking Utilities Management Menu



```
              Basic Networking Utilities Management

devices - Adding, Listing, and Removing Networking Devices
polling - Adding, Listing, and Removing Systems to Be Polled
setup   - Initial Basic Networking Setup
systems - Adding, Listing, and Removing Remote Systems
```

When you have selected the option you want, the self-explanatory submenus and instructions will be displayed on the screen to lead you through the appropriate procedures.

If you want to bypass the menu system, you may issue equivalent commands directly to the shell as shown in Table 9-4:

Table 9-4: Shell Commands for Basic Networking Utilities

Task Description	Menu Item	Shell Command
Add, list, and remove networking devices	`devices`	`vi /etc/uucp/Devices`
Add, list, and remove systems to be polled	`polling`	`vi /etc/uucp/Poll`
Set up basic networking	`setup`	*use menu item*
Add, list, and remove remote systems	`systems`	`vi /etc/uucp/Systems`

In this chapter, you will first get a high level introduction to the process of Basic Networking, as well as an introduction to the components involved in the process (network hardware, commands, daemons, and support files). Keep this high level view in mind as you begin to work with BNU; it will help you keep track of where you are in the overall process. Next, we will describe how to install, setup, maintain, debug, and remove your BNU package. The `sysadm` interface is provided to help you through these procedures. Sections detailing the structure and format of the various BNU data base support, administrative, and `uucp` log files are then described. As you gain more experience with BNU administration, these sections can be used as reference material for fine tuning your computer's configuration. Lastly, we discuss some points on how to configure your BNU package to work with direct links, telephone lines, and local area networks.

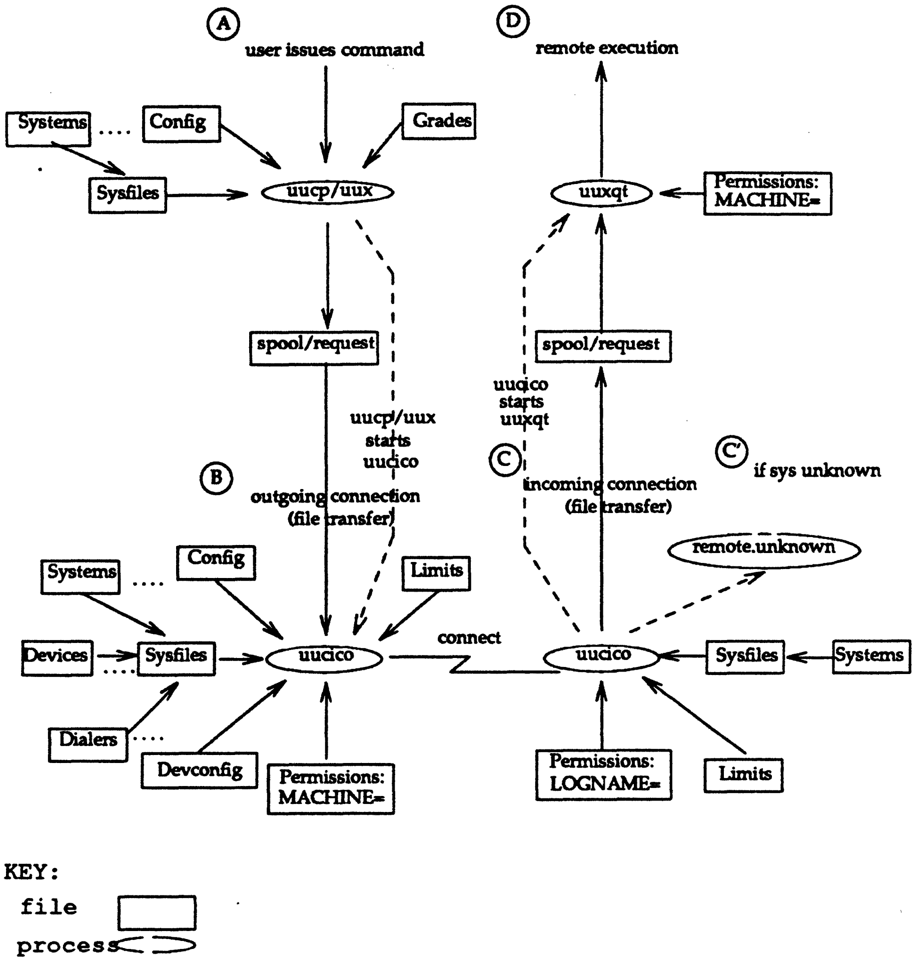
The Basic Networking Process

After you have properly installed and configured your BNU package, your computer will be ready to communicate with other computers that use the UNIX operating system. Here is a general view of how BNU accomplishes this communication process:

1. A user on your local machine issues a command requesting file transfer or remote execution communication with a remote computer (phase A). Several BNU data base support files are read to determine if the remote computer is accessible by your local computer and the priority of the user's request, as compared to other users' requests. This phase ends by queuing the user's request in a spool area on your local machine and triggering the next phase.
2. The `uucico` routine is triggered automatically (phase B). It reads several BNU data base support files to determine when the remote computer can be reached, how to establish the link to the remote computer, how to handle data flow between the local and remote computers, and if the maximum number of simultaneous requests for communication to the remote computer has been reached.
3. The `uucico` routine on the remote computer is triggered automatically when a call for communication is received from the local computer (phase C). In this phase, the remote `uucico` routine reads BNU data base support files on its computer to determine if the calling computer is allowed access, and what action to take if the calling computer is not allowed access (phase C'). For calling computers that are allowed access to the remote computer, the level of access is determined in this phase.
4. Requests initiated on the local computer may contain commands to be executed on the remote computer (phase D). When these commands arrive on the remote computer, they are stored in a spool area. The `uuxqt` routine on the remote computer runs these commands on the remote computer during this phase.

The following is a picture that describes the Basic Networking process:

Figure 9-6: Basic Networking Process



Networking Hardware

Before your computer can communicate with other computers, you must set up the hardware to complete the communications link. The cables and other hardware you need will depend on how you want to connect the computers: direct links, telephone lines, or local area networks.

Networking Programs

There are two categories of Basic Networking programs: user programs and administrative programs.

User Programs

You will find the user programs for Basic Networking in `/usr/bin`. No special permission is needed to use these programs; they are all fully described in the *User's Reference Manual*: `cu`, `ct`, `uucp`, `uuto`, `uupick`, `uux`, `uustat`, `uulog`, `uuglist`, `uuname`, `uudecode`, `uuencode`.

Administrative Programs

You will find most of the administrative programs in `/usr/lib/uucp`, along with Basic Networking shell scripts. The only exception is `uulog`, which is in `/usr/bin`. All of these commands are fully described in the *System Administrator's Reference Manual*: `uucleanup`, `Uutry`, `uucheck`.

You should use the `uucp` login ID only when you administer BNU because it owns the Basic Networking and spooled data files. The home directory of the `uucp` login ID is `/usr/lib/uucp`. (The other Basic Networking login ID is `nuucp`, used by remote computers to access your computer. Calls from `nuucp` are answered by `uucico`.)

Networking Daemons

There are three daemons in BNU. A daemon is a routine that runs as a background process and performs a system-wide public function. These daemons handle file transfers and command executions. They can also be run manually from the shell.

`uucico` Selects the device used for the link, establishes the link to the remote computer, performs the required login sequence and permission checks, transfers files (if requested), logs results, and notifies the user by `mail` of transfer completions. When the local `uucico` daemon calls a remote computer, it "talks" to the uucico daemon on the remote computer during the session.

uucico is executed by the uucp, uuto, and uux programs, after all the required files have been created, to contact the remote computer. It is also executed by uusched and Uutry.

uuxqt

Executes remote execution requests. It searches the spool directory for execute files (always named X.*file*) that have been sent from a remote computer. When an execute file is found, uuxqt opens it to get the list of data files that are required for the execution. It then checks to see if the required data files are available and accessible. If the files are present and can be accessed, uuxqt checks the Permissions file to verify that it has permission to execute the requested command. uuxqt is executed by the uudemon.hour shell script, which is started by cron.

uusched

Schedules the queued work in the spool directory. Before starting uucico, uusched randomizes the order in which remote computers will be called. uusched is executed by a shell script called uudemon.hour, which is started by cron.

Networking Support Files

There are three types of BNU support files:

Data Base

These files are responsible for much of the actual networking activity associated with your BNU package. They are located in /etc/uucp. In general, they determine: what computers your computer will communicate with, the devices over which the communication will take place, and the protocols for communicating with remote computers. See the section entitled "Data Base Support Files."

Administrative

Administrative files are created by network processes in spool directories to lock devices, hold temporary data, or store information about remote transfers or command executions. See the section entitled "Administrative Support Files."

Log

Log files keep track of overall statistics of your computer network, particularly in the areas of security and accounting. See the section entitled "Logs."

Basic Procedures

The following steps are performed to administer BNU.

Installation

Installing the BNU software package (placing it on your computer's hard disk) is first. To install your BNU package, you should follow the guidelines described under "Installing Software Packages" in the "Software Management" chapter of this guide.

Setup

Setting up the basic networking facility and configuring basic networking files. See the following section, "Setup Basic Networking Files."

Maintenance

Automatically and manually maintaining your basic networking files and transactions so that your operations continue to run smoothly. See "Basic Networking Maintenance" in this section.

Debugging

Identifying and correcting common problems in basic networking operations and administration. See "Basic Networking Debugging" in this section.

Removal

Removing the BNU software package (deleting it from your computer's hard disk) is last. To remove your BNU package, you should follow the guidelines described under "Removing Packages" in the "Software Management" chapter of this guide.

NOTE

See "BNU Software Removal Considerations" in this section for information to consider before you remove your BNU software package.

Setup Basic Networking Files

The BNU *Setup* procedure provides the steps for initializing your BNU data base. This information allows your local and remote systems to communicate with each other. This procedure is done by using the `sysadm` subcommands and your favorite text editor.

Begin by typing `sysadm basic_networking` and selecting `setup`. This will give you the starting menu. From this point, you can begin setting up (initializing) the various data base files. Continue making interactive menu selections until you complete the task(s). To get help along the way, press the HELP function key. This will give you a detailed description of a menu selection. Pressing the CANCEL function key exits help mode.

Setting up the `Permissions`, `Devconfig`, `Sysfiles` and `Limits` files, and adding `uucp` logins are principal functions in the initial Basic Networking setup process. Here is more detailed information in these areas.

Set Up Permissions File

The default `/etc/uucp/Permissions` file provides the maximum amount of security for your computer. The file, as delivered, contains the following entry:

```
LOGNAME=nuucp
```

You can set additional parameters for each machine you communicate with:

- the ways it can receive files from your machine
- the directories it can read and write
- the commands it can execute remotely

See the section "Data Base Support Files" for information on how to set up this file. You can modify this file to include the entries you desire.

Set Up Devconfig File

The /etc/uucp/Devconfig file is only used if you are using BNU over a Streams-based provider. If you are using a TCP/IP network, the two entries shown in the Devconfig file are all you need in this file.

```
service=cu      device=tcp  push=ttcompat:tirdwr
service=uucico  device=tcp  push=ttcompat:tirdwr
```

You must also create an entry for TCP in your Devices file. Descriptions in the Devices file tell how to define Transport Interface devices.

Devconfig entries define the STREAMS modules that are used for a particular device. (The push= variable shows the modules and the order they are pushed on to a stream.) Different modules and devices can be defined for cu and uucico services. You can modify this file to include the entries you desire.

Set Up Sysfiles File

/etc/uucp/Sysfiles lets you assign different files to be used by uucp and cu as Systems, Devices, and Dialers files. Here are some cases where this optional file may be useful:

- You may want different Systems files so requests for cu login services can be made to addresses other than uucp services.
- You may want different Dialers files to use different chat scripts for cu and uucp.
- You may want to have multiple Systems, Dialers, and Devices files. The Systems file in particular may become large, making it convenient to split it into several smaller files.

The format of the Sysfiles file is described in the section entitled "Data Base Support Files." The following is an example of the file.

```
service=uucico  systems=Systems.cico:Systems\
                dialers=Dialers.cico:Dialers\
                devices=Devices.cico:Devices
service=cu      systems=Systems.cu:Systems\
                dialers=Dialers.cu:Dialers\
                devices=Devices.cu:Devices
```

You can modify this file to include the entries you desire.

Set Up `Limits` File

The `/etc/uucp/Limits` file is used to limit the maximum number of simultaneous `uucicos`, `uuxqts`, and `uuscheds` that are running on your machine.

See the section entitled "Data Base Support Files" for a format description of the `Limits` file. The following is an example of the file.

```
service=uucico max=5
service=uuxqt max=5
service=uusched max=2
```

You can modify this file to include the entries you desire.

Add `uucp` logins

You may add one or more administrative logins to your system so incoming `uucp` (`uucico`) requests from different remote machines can be handled differently. Each remote machine should have an entry in its `Systems` file for your machine that contains the login ID and password that you add to your `/etc/passwd` file.

The default in the `/etc/passwd` file is shown below.

```
uucp:x:5:5:0000-uucp(0000):/usr/lib/uucp
nuucp:x:10:10:0000-uucp(0000):/var/spool/uucppublic:/usr/lib/uucp/uucico
```

This entry shows that a login request by nuucp is answered by /usr/lib/uucp/uucico. The home directory is /var/spool/uucppublic. The *x* indicates that the encrypted password is stored in /etc/shadow.

Basic Networking Maintenance

BNU maintenance involves automatically and manually maintaining your basic networking files so that they do not become large and consume too much disk space. It also means keeping BNU running smoothly.

BNU comes with four shell scripts:

```
uudemon.poll
uudemon.hour
uudemon.admin
uudemon.cleanup
```

These scripts will poll remote machines, reschedule transmissions, and clean up old log files and unsuccessful transmissions. They should be used regularly to keep your basic networking running smoothly. Normally, they are run automatically with cron(1M) although they can also be run manually.

Automated Networking Maintenance (cron)

BNU is delivered with entries for shell scripts in the /var/spool/cron/crontabs/root file. These entries will automatically handle some BNU administrative tasks for you. The shell scripts are in /usr/lib/uucp.

In multi-user mode, tasks scheduled by the crontab command are automatically performed by cron. The crontab file for root is generally used to schedule regular BNU maintenance using the following shell scripts:

uudemon.poll

The uudemon.poll shell script, as delivered, does the following:

- The shell script reads the Poll file (/etc/uucp/Poll).
- If any of the machines in the Poll file are scheduled to be polled, a work file (C.sysnxxxx) is placed in the /var/spool/uucp/*nodename* directory, where *nodename* is replaced by the name of the machine being polled.

By default, the shell script is scheduled to run twice an hour, just before uudemon.hour, so that the work files will be there when uudemon.hour is called. The default root crontab entry for uudemon.poll is as follows:

```
        1,30 * * * * /usr/lib/uucp/uudemon.poll > /dev/null
```

uudemon.hour

The uudemon.hour shell script you receive with your machine does the following:

- calls the uusched program to search the spool directories for work files (C.) that have not been processed and schedules these files for transfer to a remote machine
- calls the uuxqt daemon to search the spool directories for execute files (X.) that have been transferred to your computer and were not processed at the time they were transferred

The default root crontab entry for uudemon.hour is as follows:

```
        41,11 * * * * /usr/lib/uucp/uudemon.hour > /dev/null
```

As delivered, this is run twice an hour. You may want it to run more often if you expect high failure rates.

uudemon.admin

The uudemon.admin shell script, as delivered, does the following:

- runs the uustat command with -p and -q options. The -q reports on the status of work files (C.), data files (D.), and execute files (X.) that are queued. The -p prints process information for networking processes listed in the lock files (/var/spool/locks)
- sends resulting status information to the uucp administrative login via mail(1)

There is no default crontab entry for uudemon.admin. The following is recommended:

```
48 8,12,16 * * * /bin/su uucp -c "/etc/uucp/uudemon.admin" > /dev/null
```

uudemon.cleanup

The delivered uudemon.cleanup shell script does the following:

- takes log files for individual machines from the /var/spool/uucp/.Log directory, merges them, and places them in the /var/spool/uucp/.Old directory with other old log information (if log files get large, the ulimit parameter may need to be increased)
- removes work files (C.) 7-days old or older, data files (D.) 7 days old or older, and execute files (X.) two days old or older from the spool directory
- returns mail that cannot be delivered to the sender
- mails a summary of the status information gathered during the current day to the uucp administrative login

There is no default crontab entry for uudemon.cleanup. The following is recommended:

```
45 23 * * * ulimit 5000; /usr/bin/su uucp -c \
"/usr/lib/uucp/uudemon.cleanup" \
 > /dev/null 2>&1
```

uudemon.cleanup is described in detail below, under "Cleaning up Log Files."

Manual Maintenance

Some files may grow indirectly from uucp and other Basic Networking activities. Here are two files you should check and delete if they have become too large.

/usr/adm/sulog

This file keeps a history of all super-user commands. Since the uudemon entries in the /usr/cron/root file use the su command, the sulog will grow over time. After examining it for tampering, you should delete this file if it becomes too large.

/usr/lib/cron/log — This file is a log of cron activities. While it grows with use, it is automatically truncated when the system goes to the multi-user state.

Cleaning up Log Files

uudemon.cleanup is a shell script that should be invoked daily by cron to clean up uucp's spool directory and to consolidate and dispose of the log files that currently exist for uucp. Currently, uudemon.cleanup uses two techniques to clean up the log files:

1. The *Multi-day* (multi) technique. Three days of logs are kept in files called Old-1, Old-2, and Old-3. Whenever uudemon.cleanup is run, Old-2 is renamed Old-3, Old-1 is renamed Old-2, and the current log is renamed Old-1 and stored in /var/spool/uucp/.Old/Old-1.
2. The *Single-day* (single) technique. The current log is moved to /var/spool/uucp/.Old. This means that the log is preserved for one day only (assuming that uudemon.cleanup is run daily).

Although uudemon.cleanup does take care of removing old log entries, as an administrator, you should monitor the size of the log files. The exact procedure uudemon.cleanup uses to remove old log entries varies according to the type of log file. Here's a description of how the uucp log files will be cleaned up.

Table 9-5: Summary of BNU Log Files

File Use	File Name (beginning with /var/uucp)	Cleanup Technique
Command	.../.Admin/command	single
History	.../.Log/[uucp|uucico|uux|uuxqt]/system	multi
Foreign	.../.Admin/Foreign	single
Error	.../.Admin/errors	single
Transfer	.../.Admin/xferstats	single
Accounting	.../.Admin/account	multi
Security	.../.Admin/security	multi
Performance	.../.Admin/perflog	single

Basic Networking Debugging

The *Debug* procedures are intended to help you identify and correct common problems in basic networking operations and administration. Here is a list of the available monitoring tools you can use to detect and solve basic networking problems:

```
uustat(1)
cu(1)
Uutry(1)
uuname(1M)
uulog(1)
uucheck(1)
```

Check Basic Information

There are several commands you can use to check for Basic Networking information.

uuname — Use this command to list those machines your machine can contact.

uulog — Use this command to display the contents of the log directories for particular hosts.

`uucheck -v` Run this command to check for the presence of files and directories needed by `uucp`. This command also checks the `Permissions` file and provides information on the permissions you have set up.

Check for Faulty ACU/Modem

You can check if the automatic call units or modems are not working properly in several ways.

- Run `uustat -q`. This will give counts and reasons for contact failure.
- Run `cu -d -l`*line*. This will let you call over a particular communications line and print debugging information on the attempt. If the communications line, *line*, is connected to an autodialer, you must add a telephone number at the end of the command line you execute. Otherwise, *line* must be defined as `direct` in the `Devices` file.

Check Systems File

Check that you have up-to-date information in your systems file if you are having trouble contacting a particular machine. Some things that may be out of date for a machine are its:

- phone number
- login
- password

Debug Transmissions

If you are unable to contact a particular machine, you can check out communications to that machine with `Uutry` and `uucp`.

Step 1: To try to make contact, run:

```
$ /usr/lib/uucp/Uutry -r machine
```

where *machine* is replaced with the node name of the machine you are having problems contacting. This command will:

1. Start the transfer daemon (`uucico`) with debugging. You will get more debugging information if you are `root`.

2. Direct the debugging output to /tmp/*machine*.
3. Print the debugging output to your terminal, using `tail -f`. Hit BREAK to end output.

You can copy the output from /tmp/*machine* if you want to save it.

Step 2: If `Uutry` does not isolate the problem, try to queue a job by running:

```
$ uucp -r file machine!/dir/file
```

where *file* is replaced by the file you want to transfer, *machine* is replaced by the machine you want to copy to, and *dir/file* is where the file will be placed on the other machine. The `-r` option will queue a job but not start the transfer.

Now use `Uutry` again. If you still cannot solve the problem, you may need to call support personnel. Save the debugging output; it will help diagnose the problem.

BNU Software Removal Considerations

Since removing BNU leaves you without a means of communicating with other computers, this should be a last resort for freeing disk space. Before you remove the BNU software, you may want to save the information in /etc/uucp.

Data Base Support Files

The BNU support files are in the /etc/uucp directory. Most changes to these files can be made using the System Administration Menu commands described in the *Setup* procedure. The descriptions below, however, provide details on the structure of these files so you can edit them manually.

`Config` This file contains a list of variable parameters within BNU. The administrator can set these parameters to configure the network manually.

`Devconfig` This file is used to configure your network connections on TCP/IP or some other network provider.

`Devices` This file contains information concerning the location and line speed of automatic call units, direct links, and network devices.

`Dialcodes` This file contains dial-code abbreviations that may be used in the *telephone number* field of `Systems` file entries.

`Dialers` This file contains character strings required to communicate with network devices, automatic calling units, and direct links.

`Grades` This file is used to define the job grades, and the permissions associated with each job grade, that users may specify to queue jobs to a remote computer.

`Limits` This file defines the maximum number of simultaneous `uucicos`, `uuxqts`, and `uuscheds` permitted on your machine.

`Permissions` This file defines the level of access that is granted to computers when they attempt to transfer files or execute remote commands on your computer.

`Poll` This file defines computers that are to be polled by your system and when they are polled.

`Sysfiles` This file is used to assign different or multiple files to be used by `uucico` and `cu` as `Systems`, `Devices`, and `Dialers` files.

`Systems` This file contains information needed by the `uucico` daemon and the `cu` program to establish a link to a remote computer, such as the name of the remote computer, the name of the connecting device associated with the remote computer, when the computer can be reached, telephone number or network address, login ID, and password.

There is one file that may be considered part of the supporting data base, but is not directly related to the process of establishing a link and transferring files. This file, `remote.unknown`, is described briefly in the section entitled "Other Networking Files."

Config File

The /etc/uucp/Config file allows the administrator to override certain parameters within BNU manually. Each entry in the Config file has the following format:

parameter=value

Where *parameter* is one of the configurable parameters and *value* is the value to be assigned to that parameter. See the Config file provided with your system for a complete list of configurable parameter names.

The following Config file entry sets the default protocol ordering to "Gge" and changes the "G" protocol defaults to 7 windows and 512 byte packets.

```
Protocol=G(7,512)ge
```

Devices File

The Devices file (/etc/uucp/Devices) contains information for all the devices that may be used to establish a link to a remote computer. Provisions are made for several types of devices, such as automatic call units, direct links, and network connections.

NOTE This file works closely with the Dialers, Systems, and Dialcodes files. Before you make changes in any of these files, you should be familiar with them all. A change to an entry in one file may require a change to a related entry in another file.

Each entry in the Devices file has the following format:

Type Line Line2 Class Dialer-Token-Pairs

These fields are defined as:

Type This field may contain one of two keywords (Direct or ACU), the name of a Local Area Network, switch, or system.

Direct This keyword indicates a Direct Link to another computer or a switch.

ACU This keyword specifies that the link to a remote computer is made through an automatic call unit (Automatic Dial Modem). This modem may be connected either directly to your computer or indirectly through a Local Area

Network (LAN) switch.

LAN_Switch This is the name of the LAN or switch. For instance, Develcon could be the name for a Develcon switch connection.

Sys-Name This value specifies a direct link to a particular computer. (*Sys-Name* is replaced by the name of the computer.) This naming scheme is used to convey the fact that the line associated with this Devices entry is for a particular computer in the Systems file.

The keyword used in the *Type* field is matched against the third field of Systems file entries as shown below:

```
Devices: ACU term/11 - 1200 penril

Systems: eagle Any ACU 1200 3251 ogin: nuucp \
               ssword: Oakgrass
```

You can designate a protocol to use for a device within this field. See the "Protocols" section at the end of this file.

Line This field contains the device name of the line (port) associated with the Devices entry. For instance, if the Automatic Dial Modem for a particular entry was attached to the /dev/term/11 line, the name entered in this field would be term/11. There is an optional modem control flag, *M*, that can be used in the *Line* field to indicate that the device should be opened without waiting for a carrier. For example:

```
term/11,M
```

Line2 If the keyword ACU was used in the *Type* field and the ACU is an 801 type dialer, *Line2* would contain the device name of the 801 dialer. (801 type ACUs do not contain a modem. Therefore, a separate modem is required and would be connected to a different line, defined in the *Line* field.) This means that one line would be allocated to the modem and another to the dialer. Since non-801 dialers will not normally use this configuration, the *Line2* field will be ignored by them, but it must still contain a hyphen (-) as a placeholder.

Class If the keyword ACU or Direct is used in the *Type* field, *Class* may be just the speed of the device. However, it may contain a letter and a speed (for example, C1200, D1200) to differentiate between classes of dialers (Centrex or Dimension PBX). This is necessary because many larger offices may have more than one type of telephone network: one network may be dedicated to serving only internal office communications while another handles the external communications. In such a case, it becomes necessary to distinguish which line(s) should be used for internal communications and which should be used for external communications. The keyword used in the *Class* field of the Devices file is matched against the fourth field of Systems file entries as shown below:

```
Devices: ACU tty11 - D1200 penril

Systems: eagle Any ACU D1200 3251 ogin: nuucp \
          ssword: Oakgrass
```

Some devices can be used at any speed, so the keyword Any may be used in the *Class* field. If Any is used, the line will match any speed requested in a Systems file entry. If this field is Any and the Systems file *Class* field is Any, the speed defaults to 1200 bps.

Dialer-Token-Pairs:

This field contains pairs of dialers and tokens. The *dialer* portion may be the name of an automatic dial modem, a LAN switch, or it may be direct or uudirect for a Direct Link device. You can have any number of Dialer-Token-Pairs. The *token* portion may be supplied immediately following the *dialer* portion, or, if not present, it will be taken from a related entry in the Systems file.

This field has the format:

dialer token [*dialer token*]

where the last pair may or may not be present, depending on the associated device (dialer). In most cases, the last pair contains only a *dialer* portion and the *token* portion is retrieved from the *Phone* field of the Systems file entry.

A valid entry in the *dialer* portion may be defined in the Dialers file or may be one of several special dialer types. These special dialer types are compiled into the software and are therefore available without having entries in the Dialers file.

801	Bell 801 auto dialer
TLI	Transport Level Interface Network (without STREAMS)
TLIS	Transport Level Interface Network (with STREAMS)

The *Dialer-Token-Pairs* (*DTP*) field may be structured differently, depending on the device associated with the entry:

- If an automatic dialing modem is connected directly to a port on your computer, the *DTP* field of the associated `Devices` file entry will only have one pair. This pair would normally be the name of the modem. This name is used to match the particular `Devices` file entry with an entry in the `Dialers` file. Therefore, the *dialer* field must match the first field of a `Dialers` file entry as shown below:

  ```
  Devices: ACU term/11 - 1200 att2212c
  Dialers:  att2212c =+-, "" atzod,o12=y,o4=n\r\c \
          \006 atT\T\r\c ed
  ```

  Notice that only the *dialer* portion (`att2212c`) is present in the *DTP* field of the `Devices` file entry. This means that the *token* to be passed on to the dialer (in this case the phone number) is taken from the *Phone* field of a `Systems` file entry. (`\T` is implied; backslash sequences are described below.)

- If a direct link is established to a particular computer, the *DTP* field of the associated entry would contain the keyword `direct` or `uudirect`. This is true for both types of direct link entries, `Direct` and *System-Name* (refer to discussion on the *Type* field).

- If you wish to communicate with a computer that is on the same local network switch as your computer, your computer must first access the switch. Then the switch can make the connection to the other computer. In this type of entry, there is only one pair. The *dialer* portion is used to match a `Dialers` file entry as shown below:

  ```
  Devices: develcon term/13 - 1200 develcon
  Dialers: develcon "" "" \pr\ps\c est:\007 \E\D\e \007
  ```

  As shown, the *token* portion is left blank. This indicates that it is retrieved from the `Systems` file. The `Systems` file entry for this particular computer will contain the token in the *Phone* field, which is normally reserved for the telephone number of the computer (refer to

Systems file, *Phone* field). This type of *DTP* contains an escape character (\D), which ensures that the contents of the *Phone* field will not be interpreted as a valid entry in the Dialcodes file.

- If an automatic dialing modem is connected to a switch, your computer must first access the switch, and then the switch will make the connection to the automatic dialing modem. This type of entry requires two *dialer-token-pairs*. The *dialer* portion of each pair (fifth and seventh fields of entry) will be used to match entries in the Dialers file as shown below:

```
Devices:  ACU term/14 - 1200 develcon dial att2212c

Dialers:  develcon "" "" \pr\ps\c est:\007 \E\D\e \007
Dialers:   att2212c =+-, "" atzod,o12=y,o4=n\r\c \006 atT\T\r\c ed
```

In the first pair, develcon is the dialer and dial is the token that is passed to the Develcon switch to tell it which device (auto-dial modem) to connect to your computer. This token would be unique for each LAN switch since each switch may be set up differently. Once the modem has been connected, the second pair is accessed, where att2212c is the dialer and the token is retrieved from the Systems file.

There are two escape characters that may appear in a *DTP* field:

\T Specifies that the *Phone* (*token*) field should be translated using the Dialcodes file. This escape character is normally placed in the Dialers file for each caller script associated with an automatic dial modem. Therefore, the translation will not take place until the caller script is accessed.

\D Indicates that the *Phone* (*token*) field should not be translated using the Dialcodes file. If the dialer is an internal dialer, \T is the default. Otherwise, \D is the default. A \D is also used in the Dialers file with entries associated with network switches (develcon and micom).

Protocols

You can choose the protocol to use with each device. Usually, it is not needed since you can use the default. If you do specify the protocol, you must do so in the form *Type,Protocol[(parameters)]* (for example, `tcp,eg`). Available protocols are:

- g This is a generic packet protocol. It provides error detection and retransmission intended for use over potentially noisy lines. By its nature, it is relatively slow. Two parameters characterize the `g` protocol, *windows* and *packetsize*. *windows* indicates the number of packets which may be transmitted without waiting for an acknowledgement from the remote host. *packetsize* indicates the number of data bytes in each packet. *windows* value is set at 7, and *packetsize* is set at 64 bytes.
- G This protocol is identical to the `g` protocol in that it provides the same error detection and retransmission. However, in addition, the `G` protocol allows the number of windows and the packet size to be varied to match the characteristics of the transmission medium. When properly configured, performance can be significantly better than the `g` protocol. *windows* may range from 1 to 7, and *packetsize* may range from 32 to 4096 bytes, in powers of 2 (that is, 32, 64, 128, 256, 512, 1024, 2048, 4096).
- e This protocol assumes error free transmission and performs no error checking or retransmission. Therefore it is the fastest of these protocols. It should be used for reliable local area networks. There are no parameters to be tuned within the `e` protocol.

The following is an example that uses the `e` protocol over a tcp local area network. If the `e` protocol is not available, `g` will be used.

```
TCP,eg tcp - - TLIS \D
```

The following is an example that uses the `G` protocol on a high speed modem. The number of windows is set to 7, and the packetsize is 512 bytes. If the `G` protocol is unavailable, the standard `g` protocol will be used.

```
ACU,G(7,512)g term/11 - 9600 att2296a
```

Presumably, seven windows with a packet size of 512 bytes will provide optimum throughput for the specified device.

For incoming connections, the preferred protocol priority and parameters may be specified in the Config file using the *Protocol* parameter.

Dialers File

The Dialers file (/etc/uucp/Dialers) specifies the initial conversation that must take place on a line before it can be made available for transferring data. This conversation is usually a sequence of character strings that is transmitted and expected, and it is often used to dial a telephone number using an Automatic Call Unit.

As shown in the above examples, the fifth and subsequent odd numbered fields in the Devices file is an index into the Dialers file or internal list of special dialer types (801, TLI, or TLIS). If the match succeeds, the Dialers entry is interpreted to perform the dialer conversation. Each entry in the Dialers file has the following format:

dialer substitutions expect-send ...

The *dialer* field matches the fifth and additional odd numbered fields in the Devices file. The *substitutions* field is a translate string: the first of each pair of characters is mapped to the second character in the pair. This is usually used to translate the equal sign (=) and the dash (-) into whatever the dialer requires for "wait for dialtone" and "pause."

The remaining *expect-send* fields are character strings. Figure 9-7 shows some character strings distributed with BNU in the Dialers file.

Figure 9-7: Sample Character Strings in `Dialers` File

```
penril =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-)
                y\c : \E\TP > 9\c OK
ventel =&-% "" \r\p\r\c $ <K\T%%\r>\c ONLINE!
vadic =K-K "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE
develcon "" "" \pr\ps\c est:\007 \E\D\e \n\007
micom "" "" \s\c NAME? \D\r\c GO
direct
uudirect "" "" \r\d in:--in:
rixon =&-% "" \r\r\d $ s9\c )-W\r\ds9\c-) s\c : \T\r\c $ 9\c LINE
hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
att4000 =,-, "" ATZ\r\p\p  OK\r ATZ\r OK\r\c \EATDT\T\r\c CONNECT
att4024 =+-, "" atzod,o12=y,o4=n\r\c \006 atT\T\r\c ed
att2212c =+-, "" atzod,o12=y,o4=n\r\c \006 atT\T\r\c ed
att2224b =+-, "" atT\T\r\c ed
att2224ceo =+-, "" atzod,o12=y,o4=n,\\n3\\c1\\j0\\q0\\g0\r\c \006
           atT\T\r\c Connected
att2224g =+-, "" atzod,o12=y,o4=n,o1=n\r\c \006
           atz\\n3\\c1\\j0\\q0\\g0\r\c "" \datT\T\r\c Connected
att2224 =+-,"" \r\c :--: T\T\r\c red
att2248a =+-, "" atzod,o12=y\r\c \006 atT\T\r\c Connected
att2296a =+-, "" atzod,o12=y,o50=y,o51=n,o55=n,o69=n\r\c \006
           atz\\n3\\c1\\j0\\q0\\g0\r\c "" \datT\T\r\c Connected
nls "" "" NLPS:000:001:1\N\c
```

The meaning of some of the escape characters (those beginning with "\") used in the `Dialers` file are listed below:

`\p`	Pause (approximately ¼ to ½ second)
`\d`	Delay (approximately 2 seconds)
`\D`	Telephone number or token without `Dialcodes` translation
`\T`	Telephone number or token with `Dialcodes` translation
`\K`	Insert a BREAK
`\E`	Enable echo checking (for slow devices)
`\e`	Disable echo checking

\r	Carriage return
\c	No new-line or carriage return
\M	Turn on CLOCAL flag
\m	Turn off CLOCAL flag
\n	Send new-line
\nnn	Send character represented by octal number *nnn*

Additional escape characters that may be used are listed in the section "Systems File."

The att2212c entry in the Dialers file is executed as follows. First, the telephone number argument is translated, replacing any equal sign (=) with a W (wait for dialtone) and replacing any dash (-) with a P (pause). The handshake given by the remainder of the line works as follows:

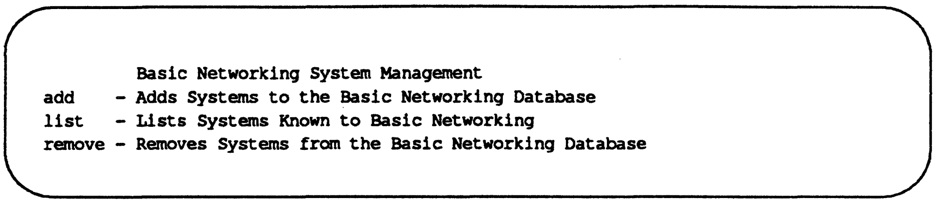
""	wait for nothing (proceed to the *expect-send* string)
=+-	secondary dial tone and pause
atzod	enter command mode, reset modem, set options to default
o12=y	set option 12 to 'y' (transparent data mode)
o4=n\r\c	set option 4 to 'n' (do not disconnect on received spaces) and terminate with a carriage return but no newline
\006	wait for acknowledge signal (ACK)
atT\T\r\c	enter command mode, use tone dialing, translate the phone number and terminate with a carriage return, but no newline
ed	expect "ed" (answered)

Systems File

The Systems file (/etc/uucp/Systems) contains the information needed by the uucico daemon to establish a communication link to a remote computer. Each entry in the file represents a computer that can be called by your computer. In addition, the Basic Networking software is configured to prevent any computer that does not appear in this file from logging in on your computer (refer to the section "Data Base Support Files" in this chapter for a description of the remote.unknown file). More than one entry may be present for a particular computer. The additional entries represent alternative communication paths that will be tried in sequential order.

The screen below allows you to do BNU systems management. It can be brought up on the screen by typing sysadm systems.

Figure 9-8: Basic Networking System Management Menu



```
           Basic Networking System Management
add    - Adds Systems to the Basic Networking Database
list   - Lists Systems Known to Basic Networking
remove - Removes Systems from the Basic Networking Database
```

If you want to bypass the menu system, you may issue equivalent commands directly to the shell as shown in Table 9-6:

Table 9-6: Shell Commands for Basic Networking System Management

Task Description	Menu Item	Shell Command
Add systems to the Basic Networking Database	`add`	`vi /etc/uucp/Systems`
List systems in the Basic Networking Database	`list`	`cat /etc/uucp/Systems`
Remove systems from the Basic Networking Database	`remove`	`vi /etc/uucp/Systems`

Using the `Sysfiles` file, you can define several files to be used as "Systems" files. See the description of the `Sysfiles` file for details. Each entry in the `Systems` file has the following format:

System-Name Time Type Class Phone Login

These fields are defined as:

System-name
This field contains the node name of the remote computer.

Time This field is a string that specifies the day-of-week and time-of-day when the remote computer can be called. The format of the *Time* field is:

daytime[*;retry*]

The day portion may be a list containing some of the following:

`Su Mo Tu We Th Fr Sa`
for individual days

`Wk` for any weekday (Mo Tu We Th Fr)

`Any` for any day

`Never` for a passive arrangement with the remote computer. If the *Time* field is `Never`, your computer will never initiate a call to the remote computer. The call must be initiated by the remote computer. In other words, your computer is in a passive mode with respect to the remote computer (see the section "`Permissions` File").

The *time* portion should be a range of times specified in 24-hour notation, for example 0800-1230 for "8:30 a.m. to 12:30 p.m." If no *time* portion is specified, any time of day is assumed to be allowed for the call. A time range that spans 0000 is permitted. For example, `0800-0600` means all times are allowed other than times between 6 a.m. and 8 a.m. An optional subfield, *retry*, is available to specify the minimum time (in minutes) before a retry, following a failed attempt. The default wait is 60 minutes. The subfield separator is a semicolon (;). For example, `Any;9` is interpreted as call any time, but wait at least 9 minutes before retrying after a failure occurs. The following is an example:

```
Wk 1700-0800,Sa,Su
```

This example allows calls from 5:00 p.m. to 8:00 a.m., Monday through Friday, and calls any time Saturday and Sunday. The example would be an effective way to call only when telephone rates are low, if immediate transfer is not critical.

Type This field contains the device type that should be used to establish the communication link to the remote computer. The keyword used in this field is matched against the first field of `Devices` file entries as shown below:

```
Systems:  eagle Any ACU,g D1200 3251 ogin: nuucp \
           ssword: Oakgrass

Devices:  ACU tty11 - D1200 penril
```

You can define the protocol used to contact the system by adding it on to the *Type* field. The example above shows how to attach the protocol g to the device type ACU. See the information in the section "Protocols" under "Devices File" for details.

Class This field specifies the transfer speed of the device used in establishing the communication link. It may contain a letter and speed (for example, C1200, D1200) to differentiate between classes of dialers (refer to the discussion of the *Class* field under "Devices File"). Some devices can be used at any speed, so the keyword Any may be used. This field must match the *Class* field in the associated Devices file entry as shown below:

```
Systems:  eagle Any ACU D1200 NY3251 ogin: nuucp \
            ssword: Oakgrass

Devices:  ACU tty11 - D1200 penril
```

If information is not required for this field, use a dash (-) as a place holder for the field.

Phone This field allows you to specify the telephone number (token) of the remote computer for automatic dialers (LAN switches). The telephone number is made up of an optional alphabetic abbreviation and a numeric part. If an abbreviation is used, it must be one that is listed in the Dialcodes file. For example:

```
Systems:  eagle Any ACU D1200 NY3251 ogin: nuucp \
            ssword: Oakgrass

Dialcodes:  NY 9=1212555
```

In this string, an equal sign (=) tells the ACU to wait for a secondary dial tone before dialing the remaining digits. A dash (-) in the string instructs the ACU to pause 4 seconds before dialing the next digit.

If your computer is connected to a LAN switch, you may access other computers that are connected to that switch. The Systems file entries for these computers will not have a telephone number in the *Phone* field. Instead, this field will contain the token that must be passed on to the switch so it will know which computer your computer wishes to communicate with (this is usually just the system name). The associated Devices file entry should have a \D at the end of the entry to ensure that this field is not translated using the Dialcodes file.

Login This field contains login information given as a series of fields and subfields of the format:

expect send

where *expect* is the string that is received and *send* is the string that is sent when the *expect* string is received.

The *expect* field may be made up of subfields of the form:

expect[*-send-expect*]...

where the *send* is sent if the prior *expect* is not successfully read and the *expect* following the *send* is the next expected string. For example, with `login--login`, uucp will expect `login`. If uucp gets `login`, it will go on to the next field. If it does not get `login`, it will send nothing followed by a new line, then look for `login` again. If no characters are initially expected from the remote computer, the characters `""` (null string) should be used in the first *expect* field. Note that all *send* fields will be sent followed by a new-line unless the *send* string is terminated with a `\c`.

Here is an example of a `Systems` file entry that uses an expect-send string:

```
owl Any ACU 1200 Chicago6013 "" \r ogin:-BREAK-ogin: \
uucpx word: xyzzy
```

This example says do not wait, just send a carriage return and wait for `ogin:` (for `Login:`). If you do not get `ogin`, send a `BREAK`. When you do get `ogin:` send the login name `uucpx`, then when you get `word:` (for `Password:`), send the password `xyzzy`.

There are several escape characters that cause specific actions when they are a part of a string sent during the login sequence. The following escape characters are useful when using BNU communications:

`\N`	Send or expect a null character (ASCII NULL).
`\b`	Send or expect a backspace character.
`\c`	If at the end of a string, suppress the new-line that is normally sent. Ignored otherwise.

`\d` Delay two seconds before sending or reading more characters.

`\p` Pause for approximately ¼ to ½ second.

`\E` Start echo checking. (From this point on, whenever a character is transmitted, it will wait for the character to be received before doing anything else.)

`\e` Echo check off.

`\M` Turn on `CLOCAL` flag.

`\m` Turn off `CLOCAL` flag.

`\n` Send a new-line character.

`\r` Send or expect a carriage-return.

`\s` Send or expect a space character.

`\t` Send or expect a tab character.

`\\` Send or expect a \ character.

`EOT` Send or expect EOT new-line twice.

`BREAK` Send or expect a break character.

`\K` Same as BREAK.

`\ddd` Collapse the octal digits (ddd) into a single character.

`Dialcodes` File

The `Dialcodes` file (`/etc/uucp/Dialcodes`) contains the dial-code abbreviations that can be used in the *Phone* field of the `Systems` file. Each entry has the format:

abb dial-seq

where *abb* is the abbreviation used in the `Systems` file *Phone* field and *dial-seq* is the dial sequence that is passed to the dialer when that particular `Systems` file entry is accessed.

The entry

```
jt 9=555-
```

would be set up to work with a *Phone* field in the `Systems` file such as `jt7867`. When the entry containing `jt7867` is encountered, the sequence 9=555-7867 would be sent to the dialer if the token in the dialer-token-pair is `\T`.

`Permissions` File

The `Permissions` file (`/etc/uucp/Permissions`) specifies the permissions that remote computers have with respect to login, file access, and command execution. There are options that restrict the remote computer's ability to request files and its ability to receive files queued by the local site. Another option is available that specifies the commands that a remote site can execute on the local computer.

The following items should be considered when using the `Permissions` file to restrict the level of access granted to remote computers:

- All login IDs used by remote computers to login for `uucp` communications must appear in one and only one `LOGNAME` entry.
- Any site that is called whose name does not appear in a `MACHINE` entry, will have the following default permissions/restrictions:
    1. Local send and receive requests will be executed.
    2. The remote computer can send files to your computer's `/var/spool/uucppublic` directory.
    3. The commands sent by the remote computer for execution on your computer must be one of the default commands; usually `rmail`.
- When a remote machine calls you, unless you have a unique login and password for that machine, you do not know if the machine's name is authentic.

How Entries Are Structured

Each entry is a logical line with physical lines terminated by a backslash (\) to specify that the entry continues on the next line. Entries are made up of options delimited by white space. Each option is a name/value pair in the following format:

name=value

Note that no white space is allowed within an option assignment.

Comment lines begin with a pound sign (#) and they occupy the entire line up to a newline character. Blank lines are ignored (even within multi-line entries).

There are two types of `Permissions` file entries:

`LOGNAME` Specifies the permissions that take effect when a remote computer logs in on (calls) your computer.

`MACHINE` Specifies permissions that take effect when your computer logs in on (calls) a remote computer.

Options

This section describes each option, specifies how they are used, and lists their default values.

`REQUEST` When a remote computer calls your computer and requests the transfer of a file, this request is granted or denied based on the value of the `REQUEST` option. The string

```
REQUEST=yes
```

specifies that the remote computer can request the transfer of files from your computer. The string

```
REQUEST=no
```

specifies that the remote computer cannot request file transfers from your computer. This is the default value. It will be used if the `REQUEST` option is not specified. The `REQUEST` option can appear in either a `LOGNAME` (remote calls you) entry or a `MACHINE` (you call remote) entry.

`SENDFILES` When a remote computer calls your computer and completes its work, it may attempt to take work your computer has queued for it. The `SENDFILES` option specifies whether your computer can send the work queued for the remote computer.

The string

```
SENDFILES=yes
```

specifies that your computer may send the work that is queued for the remote computer as long as it logged in as one of the names in the `LOGNAME` option. This string is mandatory if your computer is in a "passive mode" with respect to the remote computer.

The string

```
SENDFILES=call
```

specifies that files queued in your computer will be sent only when your computer calls the remote computer. The call value is the default for the SENDFILES option. This option is only significant in LOGNAME entries, since MACHINE entries apply when calls are made out to remote computers. If the option is used with a MACHINE entry, it will be ignored.

READ and WRITE

These options specify the various parts of the file system that uucico can read from or write to. The READ and WRITE options can be used with either MACHINE or LOGNAME entries.

The default for both the READ and WRITE options is the uucppublic directory as shown in the following strings:

```
READ=/var/spool/uucppublic
WRITE=/var/spool/uucppublic
```

The strings

```
READ=/ WRITE=/
```

specify permission to access any file that can be accessed by a local user whose access permissions are set to "other."

The value of these entries is a colon separated list of pathnames. The READ option is for requesting files, and the WRITE option for depositing files. One of the values must be a component of any full pathname of a file coming in or going out. To grant permission to deposit files in /usr/news as well as the public directory, the following values would be used with the WRITE option:

```
WRITE=/var/spool/uucppublic:/usr/news
```

It should be pointed out that if the READ and WRITE options are used, all pathnames must be specified because the pathnames are not added to the default list. For instance, if the /usr/news pathname was the only one specified in a WRITE option, permission to deposit files in the public directory would be denied.

You should be careful what directories you make accessible for reading and writing by remote systems. For example, you probably would not want remote computers to be able to write over your /etc/passwd file, so /etc should not be open to writes.

NOREAD and NOWRITE

The NOREAD and NOWRITE options specify exceptions to the READ and WRITE options or defaults. The strings

```
NOREAD=/etc    WRITE=/var/spool/uucppublic
   READ=/
```

would permit reading any file except those in the /etc directory (and its subdirectories—remember, these are prefixes) and writing only to the default /var/spool/uucppublic directory. NOWRITE works in the same manner as the NOREAD option. The NOREAD and NOWRITE can be used in both LOGNAME and MACHINE entries.

CALLBACK

The CALLBACK option is used in LOGNAME entries to specify that no transaction will take place until the calling system is called back. There are two examples of when you would use CALLBACK. From a security standpoint, if you call back a machine, you can be fairly certain it is the machine it says it is. If you are doing long data transmissions, you can choose the machine that will be billed for the longer call.

The string

```
CALLBACK=yes
```

specifies that your computer must call the remote computer back before any file transfers will take place.

The default for the COMMAND option is

```
CALLBACK=no
```

The CALLBACK option is very rarely used. Note that if two sites have this option set for each other, a conversation will never get started.

COMMANDS

CAUTION **The COMMANDS option can compromise the security of your system. Use it with extreme care.**

The uux program will generate remote execution requests and queue them to be transferred to the remote computer. Files and a command are sent to the target computer for remote execution. The COMMANDS option can be used in MACHINE entries to specify the commands that a remote computer can execute on your computer. Note that COMMANDS is not relevant in a LOGNAME entry; COMMANDS in MACHINE entries define command permissions whether we call the remote system or it calls us.

The string

```
COMMANDS=rmail
```

specifies the default commands that a remote computer can execute on your computer. If a command string is used in a MACHINE entry, the default commands are overridden. For instance, the entry

```
MACHINE=owl:raven:hawk:dove \
COMMANDS=rmail:rnews:lp
```

overrides the COMMANDS default so that the computers owl, raven, hawk, and dove can now execute rmail, rnews, and lp on your computer.

NOTE **See the section "Permissions File" in this chapter.**

In addition to the names as specified above, there can be full pathnames of commands. For example,

```
    COMMANDS=rmail:/usr/lbin/rnews:/usr/local/lp
```

specifies that the command `rmail` uses the default path. The default path for remote execution is `/usr/bin`. When the remote computer specifies `rnews` or `/usr/lbin/rnews` for the command to be executed, `/usr/lbin/rnews` will be executed regardless of the default path. Likewise, `/usr/local/lp` is the `lp` command that will be executed.

CAUTION Including the `ALL` value in the list means that any command from the remote computer(s) specified in the entry will be executed. If you use this value, you give the remote computer full access to your computer. Be careful. This allows far more access than normal users have.

The string

```
COMMANDS=/usr/lbin/rnews:ALL:/usr/local/lp
```

illustrates two points:

- The `ALL` value can appear anywhere in the string.
- The pathnames specified for `rnews` and `lp` will be used (instead of the default) if the requested command does not contain the full pathnames for `rnews` or `lp`.

The `VALIDATE` option should be used with the `COMMANDS` option whenever potentially dangerous commands like `cat` and `uucp` are specified with the `COMMANDS` option. Any command that reads or writes files is potentially dangerous to local security when executed by the `uucp` remote execution daemon (`uuxqt`).

`VALIDATE` The `VALIDATE` option is used in conjunction with the `COMMANDS` option when specifying commands that are potentially dangerous to your computer's security. It is used to provide a certain degree of verification of the caller's identity. The use of the `VALIDATE` option requires that privileged computers have a unique login/password for `uucp` transactions. An important aspect of this validation is that the login/password associated with this entry be protected. If an outsider gets that information, that particular VALIDATE option can no longer be considered secure. VALIDATE is merely an added level of security on top of the COMMANDS option (though it is a more secure way to open command access than ALL).

Careful consideration should be given to providing a remote computer with a privileged login and password for uucp transactions. Giving a remote computer a special login and password with file access and remote execution capability is like giving anyone on that computer a normal login and password on your computer. Therefore, if you cannot trust users on the remote computer, do not provide that computer with a privileged login and password.

The LOGNAME entry

```
        LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

specifies that if one of the remote computers that claims to be eagle, owl, or hawk logs in on your computer, it must have used the login uucpfriend. If an outsider gets the uucpfriend login/password, masquerading is trivial.

But what does this have to do with the COMMANDS option, which only appears in MACHINE entries? It links the MACHINE entry (and COMMANDS option) with a LOGNAME entry associated with a privileged login. This link is needed because the execution daemon is not running while the remote computer is logged in. In fact, it is an asynchronous process with no knowledge of what computer sent the execution request. Therefore, the real question is: How does your computer know where the execution files came from?

Each remote computer has its own "spool" directory on your computer. These spool directories have write permission given only to the UUCP family of programs. The execution files from the remote computer are put in its spool directory after being transferred to your computer. When the uuxqt daemon runs, it can use the spool directory name to find the MACHINE entry in the Permissions file and get the COMMANDS list, or if the computer name does not appear in the Permissions file, the default list will be used.

The following example shows the relationship between the MACHINE and LOGNAME entries:

```
    MACHINE=eagle:owl:hawk REQUEST=yes \
    COMMANDS=rmail:/usr/lbin/rnews \
    READ=/  WRITE=/

    LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
    REQUEST=yes SENDFILES=yes \
    READ=/  WRITE=/
```

The value in the COMMANDS option means that remote mail and /usr/lbin/rnews can be executed by remote users.

In the first entry, you must make the assumption that when you want to call one of the computers listed, you are really calling either eagle, owl, or hawk. Therefore, any file put into one of the eagle, owl, or hawk spool directories is put there by one of those computers. If a remote computer logs in and says that it is one of these three computers, its execution files will also be put in the privileged spool directory. You therefore have to validate that the computer has the privileged login uucpz.

MACHINE Entry for "Other" Systems

You may want to specify different option values for the computers your computer calls that are not mentioned in specific MACHINE entries. This may occur when there are many computers calling in and the command set changes from time to time. The name OTHER for the computer name is used for this entry as shown below:

```
MACHINE=OTHER \
COMMANDS=rmail:rnews:/usr/lbin/Photo:/usr/lbin/xp
```

All other options available for the MACHINE entry may also be set for the computers that are not mentioned in other MACHINE entries.

Combining MACHINE and LOGNAME Entries

It is possible to combine MACHINE and LOGNAME entries into a single entry where the common options are the same. For example, the two entries

```
MACHINE=eagle:owl:hawk REQUEST=yes \
  READ=/  WRITE=/

LOGNAME=uucpz REQUEST=yes SENDFILES=yes \
  READ=/  WRITE=/
```

share the same REQUEST, READ, and WRITE options. These two entries can be merged as shown below:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
LOGNAME=uucpz SENDFILES=yes \
  READ=/  WRITE=/
```

Poll **File**

The Poll file (/etc/uucp/Poll) contains information for polling remote computers. Each entry in the Poll file contains the name of a remote computer to call, followed by a TAB character (a space will not work), and finally the hours the computer should be called. The format of entries in the Poll file are:

*sys-name*TAB*hour* ...

For example the entry:

```
eagle   0       4       8       12      16      20
```

provides polling of the computer eagle every four hours.

The uudemon.poll script does not actually perform the poll. It merely sets up a polling work file (always named C.*file*) in the spool directory. uudemon.hour starts the scheduler, and the scheduler examines all work files in the spool directory.

Devconfig **File**

The /etc/uucp/Devconfig file is used when your computer communicates over a TCP/IP network or some other network.

Devconfig entries define the STREAMS modules that are used for a particular device. Entries in the Devconfig file have the format:

```
service=x device=y push=z[:z . . .]
```

where *x* can be cu, uucico, or both separated by a colon; *y* is the name of a network and must match an entry in the Devices file; and z is replaced by the names of STREAMS modules in the order that they are to be pushed onto the Stream. Different modules and devices can be defined for cu and uucp services.

The following entries would most commonly be used in the file:

```
service=cu       device=tcp  push=ntty:tirdwr
service=uucico   device=tcp  push=ntty:tirdwr
```

This example pushes ntty, then tirdwr. The Devconfig file cannot be modified with the sysadm menu interface. If you want to change the contents of this file, you can use a text editor.

Sysfiles File

The /etc/uucp/Sysfiles file lets you assign different files to be used by uucp and cu as Systems, Devices, and Dialers files. The following are some cases where this optional file may be useful.

- You may want different Systems files so requests for login services can be made to different addresses from requests for uucp services.
- You may want different Dialers files to use different handshaking for cu and uucp.
- You may want to have multiple Systems, Dialers, and Devices files. The Systems file in particular may become large, making it more convenient to split it into several smaller files.

The format of the Sysfiles file is

```
service=w  systems=x[:x...] dialers=y[:y...] devices=z[:z...]
```

where *w* is replaced by uucico, cu, or both, separated by a colon; *x* is one or more files to be used as the Systems file, with each filename separated by a colon and read in the order presented; *y* is one or more files to be used as the Dialers file; and z is one or more files to be used as the Devices file. Each file is assumed to be relative to the /etc/uucp directory, unless a full path is given. A backslash (\) can be used to continue an entry on to the next line.

The following is an example of using a local Systems file in addition to the usual Systems file:

```
        service=uucico:cu       systems=Systems:Local_Systems
```

If this is in /etc/uucp/Sysfiles, then both uucico and cu will first look in /etc/uucp/Systems. If the system they are trying to call does not have an entry in that file, or if the entries in the file fail, then they will look in /etc/uucp/Local_Systems.

When different Systems files are defined for uucico and cu services, your machine will store two different lists of Systems. You can print the uucico list using the uuname command or the cu list using the uuname -c command.

Limits File

The /etc/uucp/Limits file is used to control the maximum number of simultaneous uucicos, uuxqts, and uuscheds that are running in the uucp networking.

The format of the Limits file is

```
        service=x  max=y
```

where x can be uucico, uuxqt or uusched; and y is the limit that is permitted for that service.

The fields are order insensitive and lower case.

The following entries should most commonly be used in the file:

```
        service=uucico max=5
        service=uuxqt max=5
        service=uusched max=2
```

The example allows five uucicos, five uuxqts, and two uuscheds running on your machine. The Limits file cannot be modified with the System Administration Menus command sysadm. If you want to change the contents of this file, you must use one of the UNIX system text editors.

`Grades` File

The `Grades` file (`/etc/uucp/Grades`) contains the definitions for the job grades that may be used to queue jobs to a remote computer. It also contains the permissions for each job grade. Each entry in this file represents a definition of an administrator defined job grade that allows users to queue jobs.

Each entry in the `Grades` file has the following format:

User-job-grade System-job-grade Job-size Permit-type ID-list

Each entry in this file contains fields that are separated by white space. The last field in the entry is made up of sub-fields that are also white space separated. If an entry takes up more than one physical line, then a backslash (\) is used to continue the entry onto the following line. Comment lines begin with a pound sign (#) and occupy the entire line. Blank lines are always ignored. Here is a description of each field:

User-job-grade This field contains an administrative defined user job grade name of up to 64 characters.

System-job-grade This field contains a one character job grade to which *User-job-grade* will be mapped. The valid list of characters is A-Z, a-z, with A having the highest priority and z the lowest.

Job-size This field specifies the maximum job size that can be entered in the queue. *Job-size* is measured in bytes and may be a list of the following:

nnnn where *nnnn* is an integer that specifies the maximum job size for this job grade
n`K` where *n* is a decimal number that represents the number of kilobytes and `K` is an abbreviation for kilobyte
n`M` where *n* is a decimal number that represents the number of megabytes and `M` is an abbreviation for megabyte
`Any` a keyword to specify that there is no maximum job size

The following are some examples:

`5000` represents 5000 bytes
`10K` represents 10 kilobytes
`2M` represents 2 megabytes

Permit-type This field contains a keyword that denotes how to interpret the ID list. The following list contains the keywords and their meanings:

- `User` This ID list contains the login names of users permitted to use this job grade.
- `Non-user` This ID list contains the login names of users not permitted to use this job grade.
- `Group` This ID list contains the group names whose members are permitted to use this group.
- `Non-group` This ID list contains the group names whose members are not permitted to use this job grade.

Id-list This contains a list of login names or group names that are to be permitted or denied queuing to this job grade. The names in the list are separated by white space and terminated by a newline character. The keyword `Any` is used to denote that anyone is permitted to queue to this job grade.

The user job grade may be bound to more than one system job grade. It is important to note that the `Grades` file will be searched sequentially for occurrences of a user job grade. Therefore, any multiple occurrences of a system job grade should be listed according to the restriction on the maximum job size.

While there is no maximum number for the user job grades, the maximum number of system job grades allowed is 52. The reason is that more than one *User-job-grade* can be mapped to a *System-job-grade,* but each *User-job-grade* must be on a separate line in the `Grades` file. The following is an example:

```
mail       N     Any      User      Any
netnews    N     Any      User      Any
```

Given this configuration in a `Grades` file, these two *User-job-grade* grades will share the same *System-job-grade.* Since the permissions for a *Job-grade* are associated with a *User-job-grade* and not a *System-job-grade,* it is even possible for two *User-job-grades* to share the same *System-job-grades* and have two different sets of permissions for each one.

Default Grade

The binding of a default *User-job-grade* to a system job grade can be defined by the administrator. The administrator must use the keyword `default` as user job grade in the *User-job-grade* field of the `Grades` file and the system job grade that it is bound to. The restrictions and ID fields should be defined as `Any` so that any user and any size job can be queued to this grade. The following is an example:

```
        default    a       Any       User       Any
```

If the default user job grade is not defined by the administrator, then the built-in default grade, `Z`, will be used. Because it is assumed that the restriction field is `Any`, multiple occurrences of the default grade is not checked.

`remote.unknown` File

There is one other networking file that affects the use of Basic Networking facilities, the `remote.unknown` file. This file is a binary program that executes when a machine not found in any of the `Systems` files starts a conversation. It will log the conversation attempt and drop the connection.

CAUTION If you change the permissions of the `remote.unknown` file so it cannot execute, your system will accept connections from any system.

Administrative Support Files

The Basic Networking administrative files are described below. These files are created in spool directories to lock devices, hold temporary data, or store information about remote transfers or executions.

`TM` (temporary data file)

These data files are created by Basic Networking processes under the spool directory (that is, `/var/spool/uucp/`*X*) when a file is received from another computer. The directory *X* has the same name as the remote computer that is sending the file. The names of the temporary data files have the format:

`TM.`*pid.ddd*

where *pid* is a process-ID and *ddd* is a sequential three digit number starting at 0.

When the entire file is received, the `TM.`*pid.ddd* file is moved or copied to the pathname specified in the `C.`*sysnxxxx* file (discussed below) that caused the transmission. If processing is abnormally terminated, the `TM.`*pid.ddd* file may remain in the *X* directory. These files should be automatically removed by `uucleanup`.

`LCK` (lock file) These lock files prevent duplicate conversations and transfers. The names have the form:

`LCK.`*system.grade*

where *system* is the name of the remote system and *grade* is the *System-Job-grade* being processed. The lock file contains the process ID of the process holding the lock. This process ID remains valid as long as the process is active.

`LK` (lock file) These lock files prevent duplicate use of calling devices. The names have the form:

`LK.`*MAJ.maj.min*

where *MAJ* is the major number of the device containing the directory entry. *maj* and *min* are the major and minor numbers, respectively, of the device itself. The lock file contains the process ID of the process holding the lock. This process ID remains valid as long as the process is active.

`C.` (work file) Work files are created in a spool directory when work (file transfers or remote command executions) has been queued for a remote computer. The names of work files have the format:

`C.`*sysnxxxx*

where *sys* is the name of the remote computer, *n* is the ASCII character representing the grade (priority) of the work, and *xxxx* is the four digit job sequence number assigned by `uucp`. Work files contain the following information:

- Type of request, S (send) or R (receive)
- Pathname of the file to be sent or received
- Pathname of the destination or user filename
- User login name
- List of options
- Name of associated data file in the spool directory (if the uucp -c or uuto -p option was specified, a dummy name may be used)
- Mode bits of the source file
- Remote user's login name to be notified upon completion of the transfer

D. (Data file) Data files are created when it is specified in the command line to copy the source file to the spool directory. The names of data files have the following format:

D.*systmxxxxyyy*

where *systm* is the first five characters in the name of the remote computer and *xxxx* is a four-digit job sequence number assigned by uucp. The four digit job sequence number may be followed by a sub-sequence number, *yyy*, which is used when there are several D. files created for a work (C.) file.

P. (Checkpoint file)

A checkpoint file is created when processing terminates abnormally. Unlike the TM.*pid.ddd* file, it is not removed. Therefore, when the transfer session is re-established, the length of the *checkpoint* file serves as an appropriate place to restart the transfer of the file, instead of restarting from the beginning. When checkpointing is specified for a file being transferred from another computer, the checkpoint file is used instead of a TM file. *checkpoint* files are created in the spool directory. Checkpointing occurs only between two systems that have the SVR4.0 BNU enhancements. The names of the checkpoint files have the following format:

P.*systmxxxxyyy*

where *systm* is the first five characters in the name of the remote computer, *xxxx* is a four-digit job sequence number assigned by uucp. The four digit job sequence number may be followed by a sub-sequence number, *yyy*, which is used when there are several P. files created for a work (C.) file. When the entire file is received, the P.*systmxxxxyyy* file is moved to the pathname specified in the C.*sysnxxxx* file (discussed above) that caused the transmission.

X. (Execute file) Execute files are created in the spool directory prior to remote command executions. The names of Execute files have the following format:

X.*sysnxxxx*

where *sys* is the name of the remote computer, *n* is the character representing the grade (priority) of the work, and *xxxx* is a four digit number assigned by uucp. Execute files contain the following information:

- requester's login and computer name
- name of file(s) required for execution
- input filename to be used as the standard input to the command string
- computer and filename to receive standard output and stderr from the command execution
- command string
- option lines for return status requests

Logs

The BNU commands provide eight logs, some of which are optional. They are described below.

Command Log

The command log contains the commands issued by the user, the administrator, and the operator. It can help the system administrator in trouble-shooting. The full pathname of the command log is `/var/spool/uucp/.Admin/command`. The format of each entry is as follows:

```
 user1  (5/16-19:00:31)  uucp test.c mach1!~/user2
|______||_______________||_________________________|
    a            b                     c
```

KEY:

a user login name
b date and time the command was issued
c command line

System History Log

The system history log contains a record of each action that alters the state of the system and queue. The system history log can be generated by `uucp`, `uucico`, `uux`, or `uuxqt` programs and put into the `uucp`, `uucico`, `uux`, `uuxqt` subdirectories of the `/var/spool/uucp/.Log` directory. Below is a sample entry that `uucico` writes into `/var/spool/uucp/.Log/uucico/mach1`.

```
 uucp  mach1    (2/12-8:18:42,  2427,  0)  CONN FAILED
|_____||______||_||______________||_____||__||____________|
   a      b     c         d          e    f        g
 (NO DEVICES AVAILABLE)
└──────────────────────┘
           h
```

KEY:

a	user who submitted the job
b	name of the remote machine
c	job ID if a job is currently being processed; otherwise null, as in this line
d	date and time that the entry was written to the file
e	process ID of `uucico` in this example
f	file transfer sequence number within the current invocation of `uucico`
g	status message
h	status message elaboration

Error Log

The error log contains the error messages in the network. Error messages appear in the `/var/spool/uucp/.Admin/errors` file. When the errors occur, the program aborts. In most cases, this results from file-system problems. A typical entry is as follows:

```
 ASSERT ERROR  (uucico)  pid: 1513  (5/15-9:03:45)  can't open
└────────────┘└────────┘└─────────┘└──────────────┘└──────────┘
      a            b         c             d             e

 system  3  [FILE:pkl.c,  LINE:356]
└──────┘└─┘└────────────┘└─────────┘
    f    g        h            i
```

KEY:

a	error type
b	program name
c	process ID
d	date and time that this entry was written to the file
e	error message part 1
f	error message part 2

g error number
h name of calling module
i line of calling module where the error occurred

Transfer Log

The transfer log contains information pertaining to file transfer. For example, it shows the number of bytes transferred and how long the transfer took. After both `uucicos` (master and slave) agree on the protocol, the transfer information of each file is written into `/var/spool/uucp/.Admin/xferstats`. A typical entry is as follows:

```
ihx!  user  M  (5/20-6:10:10)  (C,  6559,  1)  [DKH]
 a     b    c        d          e     f    g     h
```

```
→ 1024/0.13 secs,  7877 bytes/sec  ""
         i                j         k
```

KEY:

a name of remote system
b user login
c M=master, S=slave
d date and time that entry was written to log
e C=uucico, U=uucp, X=uux, Q=uuxqt
f process ID of `uucico`, `uucp`, `uux`, or `uuxqt`
g sequential number for each file transferred in a session
h device name of media
i direction and data bytes / clock time to transfer
j transfer rate (bytes/sec)
k `"PARTIAL FILE"` if the file was not completed because of transmission error; `""` if the file was transmitted completely (as in this record)

Report Statistics of File Transfer

By specifying the -s*file* option with `uucp`, you can have the statistics of your file transfer reported to *file*. For each file transfer request, *file* will contain three lines. For multiple file transfer requests, the size of *file* increases accordingly. A typical entry is:

First line:

```
uucp job:  mach1N1131  (6/12-13:34:58)  (0:0:19)
    a          b              c             d
```

KEY:

- a message header (always the same)
- b job ID assigned by uucp
- c date and time the file transfer completed
- d *hh:mm:ss* of elapsed time between time of creation of queue file and completion of the file transfer

Second Line:

```
ihnp1!  /n15/user1/liblog  →  mach1!  ~/userid  (user1)
  a             b          c    d        e         f
```

KEY:

- a sender node name
- b source filename
- c direction (always the same)
- d receiver node name
- e destination directory/filename
- f requester login ID

Third line (status):

```
copy succeeded
      a
```

KEY:

- a a message

Accounting Log

The accounting log contains information needed for network charging. Upon completing a successful job transaction, accounting information is written to /var/spool/uucp/.Admin/account. If the job transaction is a file transfer, then accounting information is written to /var/spool/uucp/.Admin/account on the requesting site. If the job transaction is a remote execution, then accounting information is written to /var/spool/uucp/.Admin/account on the executing (target) site.

Accounting information is only collected if an account file exists and is writable by uucp; the file is not created automatically. As an administrator, you can create or remove the accounting log on your machine. A typical entry is as follows:

```
 5432  mach1N11152  4514  C  S  A  ihnp1  user1  (5/20-3:10:12)
   a        b         c   d  e  f    g      h          i

 mach1  user2  DKH  ""  xfer  ""
   j      k     l    m    n    o
```

KEY:

a user ID
b job ID assigned by uucp
c size of job in bytes if the job transaction is a file transfer; time of job in seconds if the job transaction is a remote execution
d C = job completed, P = partial job completed
e service class of the job, namely: premium, standard, or economy (at present, only "standard" service class is supported)
f job grade identification
g originating system's name
h originator's login name
i date and time the job originated
j destination system's name
k destination user's login name
l device name of media
m ID of physical network (always "")

n type of transaction: xfer = file transfer, rexe = remote execution
o the command if the job transaction is a remote execution

Security Log

The security log contains the job transactions that attempt to violate system and user security measures. It is used to aid in detecting attacks on the systems. An attempted security violation is detected when the requester fails to pass the security checks specified in the `/etc/uucp/Permissions` file or tries to access a protected source or destination file. The occurrence is logged for further analysis in the `/var/spool/uucp/.Admin/security` file. Two different entries can appear in the security log.

`xfer`	file transfer
`rexe`	remote execution

Their formats are as follows:

```
 xfer   ihnp1   user1   mach1   user2   uucp.c   ihnp1   user1   uucp.c
|_____||______||______||______||______||_______||______||______||_______|
   a      b       c       d       e        f        g       h        i

 34567   (5/19-16:10)   (5/20-11:10:29)   (5/20-11:18:20)
|______||_____________||________________||________________|
    j          k                l                  m
```

KEY:

a record type (always `xfer`)
b requester node name
c requester user login
d destination node name
e destination user login
f destination filename
g source node name
h source file owner login
i source filename
j source file size in bytes

k modification date and time of source file
l date and time that transfer started
m date and time that transfer completed

```
 rexe  ihnpl  user1  user2  (5/20-15:28:32)  (pwd)
|____||_____||_____||_____||_______________||_____|
   a     b      c      d           e           f
```

KEY:

a record type (always rexe)
b client (requesting) node name
c client (requesting) user login
d server (destination) user login
e date and time that command was executed by server
f command name and options

Performance Log

The performance log contains statistics about the operation of `uucico`. `uucico` writes the log entries to `/var/spool/uucp/.Admin/perflog`. Statistics are only collected if `perflog` exists when `uucico` starts; the file is not created automatically. As an administrator, you can turn on or turn off performance logging at your machine. Two types of records will be written to the file; each is identified by a mnemonic type at the beginning of the record. The record types are:

NOTE Fields that are not applicable or unknown are marked by double quotes ("").

`conn` contains statistics about the successful establishment of a connection

`xfer` contains statistics about a file transfer

Their formats are as follows:

```
conn  860516175047  23517  machl  M  ihnpl  DKH  d  ""
 a         b          c      d    e    f     g   h  i

20.85  3.15  0.94
  j     k     l
```

KEY:

- a record type (always conn)
- b time stamp *YYMMDDhhmmss* for sorting
- c `uucico`'s process ID
- d the name of the machine where the record was written
- e M = master, S = slave
- f name of remote system
- g device name of media
- h the protocol that was used for communications
- i physical network ID (always `""`)
- j real time to connect
- k user time to connect
- l system (kernel) time to connect

```
xfer  N  860516175051  23517  machl  M  ihnpl  DKH  d
 a    b        c         d      e    f    g     h   i

""  ihnplN2c76  118.00  121.00  1000  -dc  0.08  0.00
j       k          l       m      n    o     p     q

0.04  0.91  0.06  0.13  0.22  0.02  0.06  ""
 r     s     t     u     v     w     x    y
```

KEY:

- a record type (always `xfer`)

b job grade ID
c time stamp (*YYMMDDhhmmss*) for sorting
d `uucico`'s process ID
e the name of the machine where the record was written
f M = master, S = slave
g name of remote system
h device name of media
i the protocol that was used for communications
j physical network ID (always "")
k job ID if master, "" if slave
l time in seconds that job was in queue if master, "" if slave
m turn around time in seconds if master, "" if slave
n size of the file that was actually transferred successfully or partially because of transmission error
o command line options if master, "" if slave
p real time to start up transfer
q user time to start up transfer
r system (kernel) time to start up transfer
s real time to transfer file
t user time to transfer file
u system (kernel) time to transfer file
v real time to terminate the transfer
w user time to terminate the transfer
x system (kernel) time to terminate the transfer
y `"PARTIAL FILE"` if the file was not completed because of transmission error; "" if the file was transmitted completely (as in this record)

NOTE Fields that are not applicable or unknown are marked by double quotes ("").

Start-up time includes the time for the master to search the queues for the next job, for the master and slave to exchange work vectors, and the time to open files.

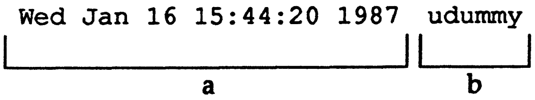
Transfer time is the time it takes to transfer the data, close the file, and exchange confirmation messages.

Termination time is the time it takes to send mail notifications and write status files.

Turnaround time is the difference between the time that the job was queued and the time that the final notification was sent.

Foreign Log

The foreign log is a list of unknown systems that attempted to connect to the current machine. The list appears in `/var/spool/uucp/.Admin/Foreign`. The format produced by `remote.unknown` is as follows:



```
Wed Jan 16 15:44:20 1987  udummy
```

KEY:

- a date and time that the entry was written to the file
- b the message logged by `remote.unknown`

10 Performance Management

An Overview of Performance Management

Performance in relation to a computer system is the way in which the system executes its tasks; its timeliness or responsiveness, including down time.

This chapter describes ways to monitor and enhance the performance of your UNIX system by telling you how to find and fix performance problems, by providing examples of how to improve performance, and by listing tools that monitor system performance. All these activities make up the task of performance management.

Performance management is a task, like all administrative tasks, that is a continual process. Usually it can be done routinely, on a regular schedule, but there are times when it will require your immediate attention.

In fact, performance management may need to be performed as soon as you set up your UNIX system. When you set up your system for the first time, it is automatically set to a basic configuration that is usually satisfactory for most applications. This default configuration controls the chief characteristics of your operating system, and this configuration may not take into consideration the traffic on your system nor the behavior of certain applications on your system. For this reason, you may need to reconfigure your system immediately in order to provide the service required by your users and their applications. For a full explanation of your system's default configuration, refer to your computer installation guide. Reconfiguration of your system will be described later in the chapter.

However, you may not know what your system's configuration should be at first. So, for argument's sake, we'll assume that you've accepted the default configuration.

Just as a car performs best when properly tuned, your system will perform best when it is properly tuned. For example, you may notice that the response time at the console is slow and something needs to be adjusted. Tuning is not just to correct performance problems. It is to maximize customer satisfaction.

At this point, you may want to utilize some of the tools which monitor performance to pinpoint the performance problem. These tools help you determine whether the problem is user-related or application-related, and are listed in the section "Monitoring System Performance."

Once you identify the problem, you will need to take some corrective action. Suggestions for these actions will be provided in the next section.

This is the extent of performance management. It is very possible that you may never need to do any special fine tuning to your system, and that your only experience with reconfiguration will be when you add new memory and peripherals. However, the rest of this chapter will be a handy reference when you need help recognizing performance problems and want advice on eliminating those problems.

Improving and Controlling System Performance

There are many items, both user-related and application-related, that can affect your system's performance. Possible problems and their solutions will be covered in this section. Keep in mind that any system tuning should be done during nonprime time.

Modifying the Tunable Configuration Parameters

Parameters exist which define your system's configuration; these parameters can be altered. This procedure is usually referred to as tuning the kernel, as you are adjusting the essential control structures at the heart of the system (the kernel). Use the `/usr/sbin/sysdef` shell command to see what the current parameter values are in the present configuration of your system. These parameters and their values are described in detail in your computer installation manual. How to tune these parameters is also explained in your computer installation manual. However, an example of this procedure appears at the end of this chapter.

Improving and Controlling File System Usage

Making files is easy under the UNIX operating system. Therefore, users tend to create numerous files using large amounts of space. The file systems containing the following directories should maintain, at the very least, the following start-of-day counts.

`/tmp`	2000 to 4000 512-byte blocks
`/usr`	500 to 1000 512-byte blocks
`/home`	3000 to 6000 512-byte blocks
`/var`	4000 to 8000 512-byte blocks

Other file systems should have 6 to 10 percent of their capacity available.

The default system configuration is set up so that the file blocks are allocated in an optimum manner. Refer to the "File System Administration" chapter for more information on file system allocation.

Balancing File System Space

You can also control file system space by balancing the load between file systems. To do this, user directories often need to be moved. Users should be notified of moves well enough in advance so that they can program around the expected change.

In order to move directories and manipulate the file system tree, you must use the find and cpio commands. The following command sequence shows how to do this. This example moves directory trees userx and usery from file system fs1 to fs2 where, presumably, there is more space available.

```
cd /fs1
find userx usery -print -depth | cpio -pdm /fs2
rm -rf /fs1/userx /fs1/usery
```

Once this sequence is entered, verify that the copy was made. Then change the userx and usery default login directories with the usermod shell command. You must also notify userx and usery, preferably via mail, that they have been moved and that their pathname dependencies may need to be changed.

Whenever moving users in this way, make sure that users with common interests are in the same file system. Furthermore, move groups of users with a single cpio command, as shown in the example above, otherwise linked files will be unlinked and duplicated.

Selecting a File System Type

The default file system type is the S5 file system type with a logical block size of 2K (2048 bytes). For most applications this should be best. Depending on the average size of the file, however, you may want to change either the logical block size or even the file system type of the file system. There are three logical block sizes of S5 file systems: 512 byte, 1K (1024 bytes), and 2K (the default). The UFS file system has two logical block sizes: 4K (4096 bytes) and 8K (8192 bytes). Each has its advantages and disadvantages in terms of performance.

The UNIX kernel uses the logical block size when reading and writing files. If the logical block size of the file system is 2K, whenever I/O is done between a file and memory, 2K chunks of the file are read into or out of memory.

Large logical block size improves disk I/O performance by reducing seek time and also decreases CPU I/O overhead. On the other hand, if the logical block size is too large then disk space is wasted. The extra space is lost because even if only a portion of a block is needed the entire block is allocated. For example, if files are stored in 1K (1024 bytes) logical blocks, then a 24-byte file wastes 1000 bytes. If the same 24-byte file is stored on a file system with a 2K (2048 bytes) logical block size, then 2024 bytes are wasted. However, if most files on the file system are very large this waste is reduced considerably.

For a file system with mostly small files, small logical block sizes (512 byte and 1K) have the advantage of less wasted space on disk. However, CPU overhead may be increased for files larger than the block size.

The `sar -u` command, described later in the chapter, can help determine if large I/O transfers are slowing the system down.

Controlling Directory Size

Very large directories are very inefficient and can affect performance. Two directories in particular, `/var/mail` and `/var/spool/uucp`, tend to get very large and should be monitored periodically. If a directory becomes bigger than 10 logical blocks (forty, 512-byte blocks or 1280 entries for a 2K logical block size) or 2560 entries, whichever is smaller, then directory searches are likely causing performance problems. The `find` command, as shown below, can ferret out such problem directories.

```
find / -type d -size +40 -print
```

NOTE

`find` thinks in terms of 512-byte blocks.

Another important thing to remember is that removing files from directories *does not* make those directories any smaller. When a file is removed from a directory, the inode (file header) number is nulled out. This leaves an unused slot for that inode; over time the number of empty slots may become quite large. For example, if you have a directory with 100 files in it and you remove the first 99 files, the directory still contains the 99 empty slots, at 16 bytes per slot, preceding the active slot. In effect, unless a directory is reorganized on the disk, it will retain the largest size it has ever achieved. Use the `/usr/sbin/dcopy` shell command to compress the file system. This procedure copies the file system temporarily to the cartridge tape and then back to its original location.

During reorganization, the system can be up but the file system being compressed must be unmounted. Root reorganization should be done once a week (requires a system reboot) and user file systems should be reorganized once a month in order to maintain maximum system performance.

An example of the dcopy command is:

```
/usr/sbin/dcopy fs1 fs2
```

where *fs1* and *fs2* can be the same name, however, the original *fs1* will be written over.

NOTE

dcopy normally catches interrupts and quit signals and reports its progress. To kill dcopy, send it a quit signal followed by an interrupt (see dcopy(1M) for more information).

If you want to compress a single directory, you must perform a series of commands in order to perform this procedure. These commands are as follows:

```
mkdir /var/omail
mv /var/mail /var/omail
chmod 777 /var/omail
cd /var/omail
find . -print | cpio -plm ../mail
cd ..
rm -rf omail
```

You can also reduce directory size by locating inactive files, backing them up, and then deleting them. The find command can be used to locate inactive files. For example:

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

where *filename* will contain the name of the files neither written to nor accessed within a specified time period. In this example, we used 90 days (+90). If these inactive files are causing problems, it is wise to first contact the user to see if these files can be deleted.

Controlling System Work Loads

Another step that can be performed to improve system performance is to check whether prime-time load can be reduced. You should control:

- less important jobs interfering with more important jobs
- scheduling of large jobs when the system is busy
- the efficiency of user-defined features, such as PATH variables

Controlling User PATH Variables

User PATH variables are the most difficult items to control. Regular mail should be sent to users on this subject informing them of how these items can cause system problems.

$PATH is a command line in the user's .profile file that is searched upon each command execution. Before outputting the not found error message, the system must search every directory in $PATH. These searches require both processor and disk time. If there is a disk or processor slowdown, changes here can help performance.

Some things that should checked for in user PATH variables are:

- $PATH is read left to right, so the most likely places to find the command should be first in the path. Make sure that a directory is not searched more than once for a command.
- Users may prefer to have the current directory listed first in the path (:/usr/bin).
- In general, $PATH should have the least number of required entries.
- Searches of large directories should be avoided if possible. Put any large directories at the end of $PATH.
- Directories that are actually a symbolic link to another directory should not appear in $PATH, for example, /usr/bin should not be in $PATH, but rather /usr/bin should.

Controlling Runaway Processes

The ps command is used to obtain information about active processes. This command gives a "snapshot" of what is going on, which is useful when you are trying to identify what processes are loading the system. The entries you should be interested in are `TIME` (minutes and seconds of CPU time used by processes) and `STIME` (time when the process first started).

When you spot a runaway process (one that uses progressively more system resources over a period of time while you are monitoring it), you should check with the owner. It is possible that such a process should be terminated immediately via the `kill -9` shell command. When you have a real runaway, it continues to eat up system resources until everything grinds to a halt.

When you spot processes that take a very long time to execute you should consider using the `cron` or `at` command to execute the job during off-hours.

Monitoring System Performance

The need to improve and control system usage may not be evident unless you monitor your system regularly. For example, it may not be obvious that a system is degraded. Just as a driver might not notice the difference between 48 and 50 miles per hour without the aid of a speedometer, you might not notice a 4 percent degradation without performance monitoring. This section will show you many ways to monitor your system's performance.

df and du Usage Reports

You can monitor your file system usage by executing the df command regularly during the day. The df command prints out the number of free file blocks and inodes.

The du command can be executed daily after hours. The du command summarizes file system usage with a total for each directory being printed out. Note that if there are links between files in different directories, where the directories are on separate branches of the file system hierarchy, du will count the excess files more than once.

The output from these commands can be kept for later comparison. In this way, directories which rapidly increase their space usage can be spotted. However, these reports may not provide the amount of detail that you need to pinpoint exact performance problems. The following section describes monitoring tools that do provide a high amount of detail.

System Performance Analysis Utilities (SPAU) Tools

The tools within SPAU provide you with commands for collecting and examining system usage data. These reports can be used to analyze the current performance of the computer and determine load-balancing and system-tuning strategies that will improve performance.

The sections, "Kernel Profiling" and "System Activity Reporting," "System Activity Reporting," describe each SPAU command, its purpose, the use of its options, and examples of command usage. The examples provided are from a M88000 family of processors system. The values you receive, whether on a M68000 or M88000 family of processors system, may be different. However, the output you receive should help you determine traffic problems and whether they are user- or application-related.

Installing SPAU

Before you install the appropriate SPAU package, you must remove any release of performance management utilities that may have been previously installed. Refer to your computer installation manual for the procedure to install and remove software.

Check the amount of free space in the `root` and `/usr` directories before installation. This package requires 1 block in `root` and 618 blocks in `/usr`.

Summary of SPAU commands

SPAU contains ten commands and two shell scripts that are listed below:

`prfdc` Performs the data collection function of the profiler by copying the current value of all the text address counters to a file where the data can be analyzed.

`prfld` Used to initialize the recording mechanism in the system.

`prfpr` Formats the data collected by `prfdc` or `prfsnap`.

`prfsnap` Collects data (like `prfdc`) at the time of invocation only.

`prfstat` Used to enable, disable, or check the status of the sampling mechanism.

`sadc` Used to sample and save the system activity data.

`sadp` Reports disk access location and seek distance in tabular form.

`sag` Graphically displays the information collected by `sar`.

`sar` Calls `sadc` or uses files created by `sadc` to sample cumulative activity counters internal to the UNIX system and provides reports on various system-wide activities. Results are saved in binary format.

`sa1` Shell script used to collect and store data in binary file `/var/adm/sa/sa`*dd* where *dd* is the current day.

`sa2` Shell script that writes a daily report in file `/var/adm/sa/sar`*dd* where *dd* is the current day.

`timex` When `timex` is used to execute another command, the elapsed time, user time, and system time spent in execution of the command are reported in seconds.

To use any of the commands beginning with `prf`, you must be logged in as `root`. The same is true for both `sa1` and `sa2`.

The next two sections describe the SPAU tools in detail.

Kernel Profiling

Kernel profiling is a mechanism that allows you to determine where the operating system is spending its time during operation. It consists of commands that control the profiling process and generate reports (see `profiler`(1M) for a description of these commands). The system profiler samples the program counter on every clock interrupt and increments the counter corresponding to the function shown by that value of the program counter.

The system profiler initializes the sampling mechanism. It will then generate a table containing the starting address of each system subroutine as extracted from the UNIX system kernel. To operate the system profiler, you must perform the following steps:

1. Use the `prfld` command to initialize or load the profiler.
2. Use the `prfstat` command to turn on the sampling mechanism.
3. Use the `prfdc` or `prfsnap` command to collect and enter the data into a file.
4. Use the `prfpr` command to print the contents of the data, collected by either `prfdc` or `prfsnap`.
5. Use the `prfstat` command to turn off the sampling mechanism.

The system profiler must be loaded and turned on after every boot. If you want the profiler to begin automatically when you boot the system to multi-user mode, you can add the following lines to the `/etc/init.d/perf` file:

```
/usr/sbin/prfld
/usr/sbin/prfstat on
```

The `prf` shell script will be executed during system initialization and the following messages will be displayed:

```
profiling enabled

xxx kernel text addresses
```

where *xxx* states how many kernel text addresses are in the current UNIX system kernel.

The following sections describe kernel profiling commands in detail.

Loading the System Profiler

The `prfld` command initializes, or loads, the system profiler mechanism. The command has the following format:

```
        /usr/sbin/prfld [namelist]
```

This command generates a table, in memory, containing the starting address of each subroutine as extracted from *namelist*. The default of *namelist* is `/stand/unix`. If *namelist* is not indicated, the starting address of each subroutine is recorded. If the number of kernel text addresses is greater than PRFMAX defined in `/etc/master.d/prf`, then PRFMAX should be increased and the system rebuilt (see the section, "Reconfiguring the System Through a Reboot").

Enabling/Disabling the Sampling Mechanism

The `prfstat` command enables or disables the sampling mechanism of the system profiler initialized by `prfld`. The `prfstat` command has the following format:

```
        /usr/sbin/prfstat [on | off]
```

Profiler overhead is less than 3 percent as calculated for 2000 text addresses. If neither of the optional parameters is entered, the status of the profiler is displayed. If the `on` parameter is supplied, the sampling mechanism is turned on. The opposite happens if `off` is indicated.

Collecting Profiling Data

The `prfdc` command performs the data collection function of the profiler by copying the current value of all the text address counters to a file where the data can be analyzed. The `prfdc` command has the following format:

```
        /usr/sbin/prfdc file [period [off_hour]]
```

This command stores the contents of the counters in a *file* every *period* minutes and turns off at *off_hour*. Valid values for *off_hour* are 0 through 24.

For example, the following copies the current value of all the text address counters into a file called `temp` every five minutes and turns off at 4:00 P.M.:

```
/usr/sbin/prfdc temp 5 16
```

The `prfsnap` command also performs data collection, but takes a snapshot of the system at the time it is called. The format of this command is as follows:

`/usr/sbin/prfsnap [`*file*`]`

where the command appends the counter values to *file.*

Formatting the Collected Data

The `prfpr` command formats the contents of *file* (data that was collected by `prfdc` or `prfsnap`). The `prfpr` command has the following format:

`/usr/sbin/prfpr` *file* `[`*cutoff* `[`*namelist*`] ]`

Each text address is converted to a system function name and the percentage of time used by that function is printed if the the activity percentage is greater than the *cutoff* number that you specify. The range of *cutoff* is 0 percent to 99 percent where 0 prints all contents. The default *cutoff* is 1 percent. The default *namelist* is `/stand/unix`.

The following screen display illustrates the output of the `prfpr` command.

```
# /usr/sbin/prfpr temp 1

07/24/90 13:59
07/24/90 14:04

excep 1.0
bcopy 7.5
struct_zero 5.2
idleloop 6.5
systrap 1.3
hat_pteload 1.2
hat_pteunload 1.5
user     48.4
```

These are the function calls in the kernel. For detailed information on function calls, refer to your computer installation manual, source code, or experienced user.

System Activity Reporting

Another SPAU tool is system activity reporting. As various system actions occur, counters in the operating system are incremented to keep track of these activities. System activities that are tracked are:

- central processing unit (CPU) utilization
- buffer usage
- disk and tape input/output activity
- terminal device activity
- system call activity
- switching
- file access
- queue activity
- kernel tables
- interprocess communication
- paging
- free memory and swap space
- Kernel Memory Allocation (KMA)
- Remote File Sharing (RFS)

System activity data can be accessed on a special request basis using the `sar` command or it can be saved automatically on a routine basis using the `sadc` command and the shell scripts `sa1` and `sa2` (see `sar`(1M)). Generally, the demand system activity reports are used to pinpoint specific performance problems, and the automatic reports are generated as a measure to monitor system performance.

The following sections describe both methods of activity reporting in detail.

Automatically Collecting System Activity Data

The sadc command can automatically sample system data. The format of this command is as follows:

```
        /usr/lib/sa/sadc [t n] [ofile]
```

The command samples *n* times with an interval of *t* seconds (*t* should be greater than 5 seconds) between samples. It then writes, in binary format, to the file *ofile*, or to standard output. If *t* and *n* are omitted, a default interval is used.

When the performance package is installed, a number of files should be automatically created and/or appended that will cause system activity commands to be run automatically.

The file /etc/init.d/perf, which is linked to /etc/rc2.d/S21perf, causes the sadc command to be invoked to mark usage from when the counters are reset to zero. The output of sadc is put in the file sa*dd* which acts as the daily system activity record. The command entry in the /etc/init.d/perf file that does this is as follows:

```
        su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
```

Once the performance package is installed, the cron file /var/spool/cron/crontabs/sys contains commands to cause the automatic collection of system activity data. The commands in the cron file are sa1 and sa2. The shell script sa1 has the following format:

```
        /usr/lib/sa/sa1 [t n]
```

The arguments *t* and *n* cause records to be written *n* times at an interval of *t* seconds. If these arguments are omitted, the records are written only one time. The records are written to the binary file /var/adm/sa/sa*dd*, where *dd* is the current date. The sa1 command is performed automatically by cron using the following two entries found in /var/spool/cron/crontab/sys:

```
                0 * * * 0-6 /usr/lib/sa/sa1
                20,40 8-17 * * 1-5 /usr/lib/sa/sa1
```

The first causes a record to be written to /var/adm/sa/sa*dd* on the hour, every hour, seven days a week. The second entry causes a record to be written to /var/adm/sa/sa*dd* 20 minutes and 40 minutes after each hour from 8:00 a.m. to 5:00 p.m., Monday through Friday, typically considered to be peak working hours. Thus, these two crontab entries cause a record to be written to

/var/adm/sa/sa*dd* every 20 minutes from 8:00 a.m. to 5:00 p.m., Monday through Friday, and every hour on the hour otherwise. These defaults can easily be changed to meet your daily needs.

The shell script sa2 has the following format:

```
/usr/lib/sa/sa2 [-abcdgkmpqruvwxyADSC] [-s time] [-e time] [-i sec]
```

The sa2 command invokes the sar command with the arguments given and writes the ASCII output to the file /var/adm/sa/sar*dd* where *dd* is the current date. The report starts at -s *time*, ends at -e *time*, and is taken as close to -i *sec* intervals as possible. See the sar command, later in the chapter, for an explanation of the remaining options.

When installed, the performance package includes the following entry in the /var/spool/cron/crontabs/sys file:

```
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

This causes a sar -A report to be generated from /var/adm/sa/sa*dd*. The report covers twenty-minute intervals in the time period from 8:00 a.m. to 6:01 p.m., Monday through Friday. Note that since /var/adm/sa/sa*dd* does not have data for 5:20 and 5:40 if the above sa1 cron entries are used, that the sar report will not have data for those times either.

Collecting System Activity Data on Demand

The sar command can either be used to gather system activity data itself or to extract what has been collected in the daily activity files created by sa1 and sa2.

The sar command has the following formats:

```
sar [-abcdgkmpqruvwxyADSC] [-o file] t [n]
sar [-abcdgkmpqruvwxyADSC] [-s time] [-e time] \
    [-i sec] [-f file]
```

In the first format, sar samples cumulative activity counters in the operating system at intervals specified by *n* for a time (in seconds) specified by *t* (*t* should be 5 seconds or greater). The default value of *n* is 1. If the -o option is specified, samples are saved in *file* in binary format.

In the second format, with no sampling interval specified, `sar` extracts data from a previously recorded *file*, either the one specified by the `-f` option or, by default, the standard daily activity file, `/var/adm/sa/sar`*dd*. The `-s` and `-e` options define the starting and ending times for the report. Starting and ending times are of the form *hh[:mm[:ss]]*. The `-i` option specifies, in seconds, the intervals to select records. If the `-i` option is not included, all intervals found in the daily activity file are reported.

The following summarizes `sar` options and their results:

`-a`	checks file access operations
`-b`	checks buffer activity
`-c`	checks system calls
`-d`	checks disk activity
`-g`	checks page-out and memory freeing
`-k`	checks kernel memory allocation
`-m`	checks interprocess communication
`-p`	checks page-in and fault activity
`-q`	checks queue activity
`-r`	checks unused memory
`-u`	checks CPU utilization
`-v`	checks system table status
`-w`	checks swapping and switching volume
`-y`	checks terminal activity
`-A`	reports overall system performance; same as entering all options

The options `-x, -D, -S,` and `-C` are remote file sharing options, and are described in the "Network Services" chapter.

Checking File Access with `sar -a`

The `sar -a` option reports on the use of file access operations. The UNIX operating system routines reported are as follows:

`iget/s` Number of S5 and UFS files located by inode entry per second.

`namei/s` Number of file system path searches per second. If `namei` does not find a directory name in the directory name logic cache, it will call `iget` to get the directory. Hence, most `iget`s are the result of directory name logic cache misses.

`dirbk/s` Number of S5 directory block reads issued per second.

The following is an example of `sar -a` output. It illustrates a one-minute sampling interval.

```
UNIX_System_V colorado 4.0 UE4.0 m88k     07/24/90

14:28:12  iget/s namei/s dirbk/s
14:29:12       0       2       1
14:30:12       0       4       1
14:31:12       0       3       1

Average        0       3       1
```

The larger the values reported, the more time the UNIX kernel is spending to access user files. The amount of time reflects how heavily programs and applications are using the file system(s). The `-a` option is helpful for understanding how disk- dependent an application system is; it is not used for any specific tuning step.

Checking Buffer Activity with `sar -b`

The `-b` option reports on the following buffer activities.

`bread/s` Average number of physical block (512 bytes each) reads into the system from the disk per second.

`lread/s` Average number of logical reads from system buffers per second.

`%rcache` Fraction of logical reads found in the system buffers (100% minus the ratio of `breads` to `lreads`).

`bwrit/s` Average number of physical writes from the system buffers to disk per second.

`lwrit/s` Average number of logical writes to system buffers per second.

`%wcache` Fraction of logical writes found in the system buffers (100% minus the ratio of `bwrit/s` to `lwrit/s`).

`pread/s` Average number of physical read requests per second.

`pwrit/s` Average number of physical write requests per second.

The most important entries are the cache hit ratios `%rcache` and `%wcache`, which measure the effectiveness of system buffering. If `%rcache` falls below 90, or if `%wcache` falls below 65, it may be possible to improve performance by increasing the buffer space by adjusting the tunable `BUFHWM` in `/etc/master.d/kernel`.

The following is an example of `sar -b` output:

```
UNIX_System_V colorado 4.0 UE4.0 m88k     07/24/90

14:28:12 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
14:29:12       0      14     100       6      17      67       0       0
14:30:12       0      12      99       6      16      65       0       0
14:31:12       0      12     100       6      16      65       0       0

Average        0      12     100       6      16      66       0       0
```

This example shows that the buffers are not causing any slowdowns, because all the data are within acceptable limits.

A parallel option is available for systems that have Remote File Sharing installed. See the description of `sar -Db` in the "Network Services" chapter.

Checking System Calls with `sar -c`

The `-c` option reports on system calls in the following categories:

`scall/s` All types of system calls per second (generally about 30 per second on a busy four- to six-user system).

`sread/s` `read` system calls per second.

`swrit/s` `write` system calls per second.

`fork/s` `fork` system calls per second, about 0.5 per second on a four- to six-user system. This number will increase if shell scripts are running.

`exec/s` `exec` system calls per second. (If (`exec/s`) / (`fork/s`) is greater than 3, look for inefficient `PATH` variables.)

`rchar/s` Characters (bytes) transferred by `read` system calls per second.

`wchar/s` Characters (bytes) transferred by `write` system calls per second.

Typically, `reads` plus `writes` account for about half of the total system calls, although the percentage varies greatly with the activities that are being performed by the system.

This is an example of `sar -c` output:

```
UNIX_System_V colorado 4.0 UE4.0 m88k     07/24/90

14:28:12 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
14:29:12      17       2       2    0.28    0.28    2527    1542
14:30:12      25       2       1    0.50    0.47    1624     295
14:31:12      21       2       2    0.35    0.35    1812     703

Average       21       2       2    0.38    0.37    1987     847
```

A parallel option is available for systems that have Remote File Sharing installed. See the description of `sar -Dc` in the "Network Services" chapter.

Checking Disk Activity with `sar -d`

The `sar -d` option reports the activities of disk devices.

`device` Name of the disk device(s) monitored.

`%busy` Percentage of time the device spent servicing a transfer request.

`avque` The average number of requests outstanding during the monitored period (measured only when the queue was occupied).

`r+w/s` Number of read and write transfers to the device per second.

`blks/s` Number of 512 byte blocks transferred to the device per second.

`avwait` Average time in milliseconds that transfer requests wait idly in the queue (measured only when the queue is occupied).

`avserv` Average time in milliseconds for a transfer request to be completed by the device (for disks this includes seek, rotational latency, and data transfer times).

The following example illustrates the `sar -d` output. ..

```
UNIX_System_V colorado 4.0 UE4.0 m88k     07/24/90
15:32:20  device %busy  avque r+w/s blks/s  avwait  avserv
15:33:20  m328-0   15   1.5     6    184     13.0    25.6
          m328-1    6   3.5     3    102     43.5    17.6
15:34:20  m328-0   15   3.1     6    202     51.4    24.4
          m328-1    6   6.7     3    107    100.5    17.5
15:35:20  m328-0   15   1.6     6    189     16.4    25.7
          m328-1    6   3.7     3     97     47.6    17.4
Average   m328-0   15   2.1     6    192     27.5    25.2
          m328-1    6   4.7     3    102     64.6    17.5
```

Note that queue lengths and wait times are measured when there is something in the queue. If `%busy` is small, large queues and service times probably represent the periodic `sync` efforts by the system to ensure that altered blocks are written to the disk in a timely fashion.

Checking Page-Out and Memory Freeing Activity with `sar -g`

The `sar -g` option reports page-out and memory freeing activities as follows:

`pgout/s` The number of times per second file that systems receive page-out requests.

`ppgout/s` The number of pages that are paged-out per second. (A single page-out request may involve paging-out multiple pages.)

`pgfree/s` The number of pages per second that are placed on the freelist by the page-stealing daemon. If this value is greater than 5, it may be an indication that more memory is needed. (This is the same as `rclm/s` previously reported by option `-p`.)

`pgscan/s` The number of pages per second scanned by the page-stealing daemon. If this value is greater than 5, the page-out daemon is spending a lot of time checking for free memory. This implies that more memory may be needed.

`%s5ipf` The percentage of S5 inodes taken off the freelist by `iget` which had reusable pages associated with them. These pages are flushed and cannot be reclaimed by processes. Thus, this is the percentage of `igets` with page flushes. If this value is greater than 10 percent, then the freelist of inodes is considered to be page-bound and the number of S5 inodes should be increased.

The following is an example of `sar -g` output:

```
UNIX_System_V colorado 4.0 UE4.0 m88k    07/24/90

14:28:12  pgout/s ppgout/s pgfree/s pgscan/s %s5ipf
14:29:12     0.00     0.00     0.35     8.18   0.00
14:30:12     0.00     0.00     0.00     0.00   0.00
14:31:12     0.00     0.00     0.00     0.00   0.00

Average      0.00     0.00     0.12     2.72   0.00
```

`sar -g` is a good indicator of whether more memory may be needed. The number of cycles used by the page-stealing daemon can be found using `ps -elf`. If it has used many cycles then coupled with high `pgfree/s` and `pgscan/s` values it is a good indicator of a memory shortage. `sar -p, sar -u, sar -r,` and `sar -w` are also good memory shortage indicators.

`sar -g` also shows whether inodes are being recycled too quickly, causing a loss of reusable pages.

Checking Kernel Memory Allocation Activity with `sar -k`

The `-k` option reports on the following activities of the Kernel Memory Allocator (KMA).

`sml_mem` The amount of memory in bytes the KMA has available in the small memory request pool (a small request is less than 256 bytes).

`alloc` The amount of memory in bytes the KMA has allocated from its small memory request pool to small memory requests.

`fail` The number of requests for small amounts of memory that failed.

`lg_mem` The amount of memory in bytes the KMA has available in the large memory request pool (a large request is from 512 bytes to 4K bytes).

`alloc` The amount of memory in bytes the KMA has allocated from its large memory request pool to large memory requests.

`fail` The number of requests for large amounts of memory that failed.

`ovsz_alloc` The amount of memory allocated for oversized requests (those greater than 4K). These requests are satisfied by the page allocator; thus, there is no pool.

`fail` The number of requests for oversized amounts of memory that failed.

The KMA allows a kernel subsystem to allocate and free memory as needed. Rather than statically allocating the maximum amount of memory it is expected to require under peak load, the KMA divides requests for memory into three categories: small (less than 256 bytes), large (512 - 4K bytes), and oversized (greater than 4K bytes). It keeps two pools of memory to satisfy small and large requests. The oversized requests are satisfied by allocating memory from the system page allocator.

If your system is being used to write drivers or STREAMS that use KMA resources then `sar -k` will likely prove useful. Otherwise, you will probably not need the information it provides. Any driver or module that uses KMA resources but does not specifically return the resources before it exits can create a memory leak. A memory leak will cause the amount of memory allocated by KMA to increase over time. Thus, if the `alloc` fields of `sar -k` increase steadily over time, then there may be a memory leak. Another indication of a memory leak is failed requests. If this occurs then it is likely that a memory leak has caused KMA to be unable to reserve and allocate memory.

If it appears that a memory leak has occurred, you should check any locally written drivers or STREAMS that may have requested memory from KMA and not returned it.

The following is an example of `sar -k` output:

```
UNIX_System_V colorado 4.0 UE4.0 m88k    07/24/90

14:28:12 sml_mem   alloc  fail  lg_mem   alloc  fail  ovsz_alloc  fail
14:29:12   95232   73472     0  311296  198656     0      180224     0
14:30:12   95232   75120     0  311296  198656     0      180224     0
14:31:12   95232   73600     0  311296  197632     0      180224     0

Average    95232   74064     0  311296  198314     0      180224     0
```

Checking Interprocess Communication with `sar -m`

The `sar -m` option reports interprocess communication activities. Message and semaphore calls are reported as follows:

`msg/s`	Number of message operations (sends and receives) per second.
`sema/s`	Number of semaphore operations per second.

An example of `sar -m` output follows:

```
UNIX_System_V colorado 4.0 UE4.0 m88k     07/24/90

14:28:12   msg/s  sema/s
14:29:12    0.00    0.00
14:30:12    0.00    0.00
14:31:12    0.00    0.00

Average     0.00    0.00
```

These figures will usually be zero (0.00) unless you are running applications that use messages or semaphores.

Checking Page-In Activity with `sar -p`

The `sar -p` option reports paging-in activity which includes protection and validity faults. (Note: This option has changed significantly from past releases due to the adoption of Virtual Memory.)

`atch/s` The number of page faults per second that are satisfied by reclaiming a page currently in memory (attaches per second). Instances of this include reclaiming an invalid page from the free list and sharing a page of text currently being used by another process (for example, two or more processes accessing the code for data.)

`pgin/s` The number of times per second file systems receive page-in requests. (This encompasses and replaces the old `sar -p` report of `pgfil/s`, which previously reported the number of validity faults per second satisfied by a page-in from the file system.)

`ppgin/s` This new field reports the number of pages paged in per second. (A single page-in request, such as a softlock request as described below, or a large block size, may involve paging-in multiple pages.)

pflt/s The number of page faults from protection errors per second. Instances of protection faults are illegal access to a page and "copy-on-writes." Generally, this number consists primarily of "copy-on-writes." (This field is carried over from the old -p option.)

vflt/s The number of address translation page faults per second. These are known as validity faults and occur when a valid page is not present in memory. (This field is carried over from the old -p option.)

slock/s This new field reports the number of faults per second caused by software lock requests requiring physical I/O. An example of the occurrence of a softlock request is the transfer of data from a disk to memory. To ensure that the page which is to receive the data is not claimed and used by another process, it is locked by the system hardware.

The following is an example of sar -p output:

```
UNIX_System_V colorado 4.0 UE4.0 m88k    07/24/90

14:28:12  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
14:29:12    1.17   12.87   12.87    5.67   11.28    1.15
14:30:12    1.67    7.08    7.08    9.12    6.33    0.67
14:31:12    1.37   12.48   12.48    6.83   10.78    1.03

Average     1.40   10.81   10.81    7.21    9.46    0.95
```

If vflt/s becomes much higher than 15, then sar -g should be looked at to determine if there is a memory shortage or if the S5 inode freelist is page bound. (See sar -g for more details). In addition, sar -u, sar -w, and sar -r can help verify whether memory is a bottleneck.

Checking Queue Activity with `sar -q`

The `sar -q` option reports the average queue length while the queue is occupied and the percentage of time that the queue is occupied.

`runq-sz` The number of processes waiting, in memory, to run. Typically, this should be less than 2. Consistently higher values mean you are CPU-bound.

`%runocc` The percentage of time the run queue is occupied. The larger this value, the better.

`swpq-sz,` Values for these headings are no longer reported due to the removal of swap queues.

An example of `sar -q` output follows:

```
UNIX_System_V colorado 4.0 UE4.0 m88k     07/24/90

14:28:12 runq-sz %runocc swpq-sz %swpocc
14:29:12     1.2      53
14:30:12     1.3      38
14:31:12     1.1      37

Average      1.2      43
```

If `%runocc` is greater than 90 percent and `runq-sz` is greater than 2, the CPU is heavily loaded and response is degraded. In this case, additional CPU capacity may be required to obtain acceptable system response. If `sar -p` shows a large number of validity faults and `sar -g` shows high page-out activity, then more memory may be required.

Checking Unused Memory with `sar -r`

The `-r` option records the number of memory pages and swap file disk blocks that are currently unused.

`freemem` Average number of 2K pages of memory available to user processes over the intervals sampled by the command.

`freeswap` Number of 512-byte disk blocks available for page swapping.

An example of `sar -r` output follows:

```
UNIX_System_V colorado 4.0 UE4.0 m88k     07/24/90

14:28:12 freemem freeswp
14:29:12     268    3034
14:30:12     351    3009
14:31:12     297    3033

Average      306   3025
```

Checking CPU Utilization with `sar -u`

The CPU utilization is listed by `sar -u`. At any given moment the processor is either busy or idle. When busy, the processor is in either user or system mode. When idle, the processor is either waiting for input/output completion or "sitting still" with no work to do. `sar -u` lists the percentage of time that the processor is in system mode (`%sys`), in user mode (`%user`), waiting for input/output completion (`%wio`), and idle (`%idle`).

In typical timesharing use, `%sys` and `%usr` are about the same value. In special applications, either of these may be larger than the other without anything being abnormal. A high `%wio` generally means a disk slowdown has occurred. A high `%idle`, with degraded response time, may mean memory constraints are present; time spent waiting for memory is attributed to `%idle`.

The following is an example of `sar -u` output:

```
UNIX_System_V colorado 4.0 UE4.0 m88k     07/24/90

14:28:12    %usr    %sys    %wio   %idle
14:29:12      22      27      18      32
14:30:12       6      24      13      57
14:31:12       8      28      19      45

Average       12      27      17      45
```

A parallel option is available for systems on which Remote File Sharing is installed. See the description of `sar -Du` in the "Network Services" chapter.

Checking System Table Status with `sar -v`

The `-v` option reports the status of the process, inode, file, and shared memory record table. >From this report you know when the system tables need to be modified.

`proc-sz`	Number of process table entries currently being used/allocated in the kernel.
`inod-sz`	Number of inode table entries currently being used/allocated in the kernel.
`file-sz`	Number of file table entries currently being used in the kernel. The `sz` is given as 0 since space is allocated dynamically for the file table.
`ov`	Number of times a table has overflowed (reported for the three tables listed above).
`lock-sz`	Number of shared memory record table entries currently being used/allocated in the kernel. The `sz` is given as 0 because space is allocated dynamically for the shared memory record table.

An example of sar -v output follows:

```
UNIX_System_V colorado 4.0 UE4.0 m88k    07/24/90

14:28:12 proc-sz ov inod-sz ov file-sz ov lock-sz
14:29:12  28/200  0 297/300  0  63/  0  0   6/  0
14:30:12  30/200  0 297/300  0  65/  0  0   6/  0
14:31:12  28/200  0 296/300  0  63/  0  0   6/  0
```

This example shows that all tables are large enough to have no overflows. If the values in these tables never exceed those shown here, you could reduce the sizes of the tables as a way of saving space in main memory.

Checking Swapping and Switching Volume with sar -w

The -w option reports swapping and switching activity. The following are some target values and observations.

swpin/s Number of transfers into memory per second.

bswin/s Number of 512-byte blocks transferred for swap-ins (including initial loading of some programs) per second.

swpot/s Number of transfers from memory to the disk swap area per second. If greater than 1, you may need to increase memory or decrease the amount of buffer space.

bswot/s Number of blocks transferred for swap-outs per second.

pswch/s Process switches per second. This should be 30 to 50 on a busy 4- to 6-user system.

An example of sar -w output follows:

```
UNIX_System_V colorado 4.0 UE4.0 m88k    07/24/90

14:28:12 swpin/s pswin/s swpot/s pswot/s pswch/s
14:29:12    0.00     0.0    0.00     0.0      22
14:30:12    0.00     0.0    0.00     0.0      12
14:31:12    0.00     0.0    0.00     0.0      18

Average     0.00     0.0    0.00     0.0      18
```

This example shows that because no swapping is occurring, there is sufficient memory for the currently active users.

Checking Terminal Activity with `sar -y`

The `-y` option monitors terminal device activities. If you have a lot of terminal I/O, you can use this report to determine if there are any bad lines. The activities recorded are defined as follows:

`rawch/s` input characters (raw queue) per second

`canch/s` input characters processed by canon (canonical queue) per second

`outch/s` output characters (output queue) per second

`rcvin/s` receiver hardware interrupts per second

`xmtin/s` transmitter hardware interrupts per second

`mdmin/s` modem interrupts per second

The number of modem interrupts per second (`mdmin/s`) should be close to 0, and the receive and transmit interrupts per second (`xmtin/s` and `rcvin/s`) should be less than or equal to the number of incoming or outgoing characters, respectively. If this is not the case, check for bad lines.

An example of `sar -y` output follows:

```
UNIX_System_V colorado 4.0 UE4.0 m88k    07/24/90

14:28:12 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
14:29:12       0       1     157       1       3       0
14:30:12       0       2      34       2       2       0
14:31:12       0       1      11       1       2       0

Average        0       1      67       1       2       0
```

Checking Overall System Performance with `sar -A`

The `-A` option provides a view of overall system performance. Use it to get a more global perspective. If data from more than one time slice is shown, the report includes averages.

An example of `sar -A` output follows:

```
UNIX_System_V colorado 4.0 UE4.0 m88k    07/24/90

14:28:12    %usr    %sys    %sys    %wio   %idle
                   local  remote
14:29:12      22      27       0      18      32
14:30:12       6      24       0      13      57

Average       14      26       0      16      44



14:28:12  device   %busy   avque   r+w/s  blks/s  avwait  avserv

14:29:12 hdsk-0       34    10.8      20      39   170.2    17.4

14:30:12 hdsk-0       24    13.6      13      26   236.4    18.8

Average  hdsk-0       29    12.0      16      32   196.6    17.9

14:28:12 runq-sz %runocc swpq-sz %swpocc
14:29:12     1.2      53
14:30:12     1.3      38
```

(continued on next page)

```
Average      1.2      11

14:28:12 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
14:29:12
   local      0     14     100       6      17      67       0       0
   remote     0      0       0       0       0       0
14:30:12
   local      0     12      99       6      16      65       0       0
   remote     0      0       0       0       0       0

Average
   local      0     13     100       6      17      66       0       0
   remote     0      0       0       0       0       0

14:28:12 swpin/s pswin/s swpot/s pswot/s pswch/s
14:29:12    0.00     0.0    0.00     0.0      22
14:30:12    0.00     0.0    0.00     0.0      12

Average     0.00     0.0    0.00     0.0      17

14:28:12 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
14:29:12
   in          0       0       0            0.00       0       0
   out         0       0       0            0.00       0       0
   local      17       2       2    0.28    0.28    2527    1542
14:30:12
   in          0       0       0            0.00       0       0
   out         0       0       0            0.00       0       0
   local      25       2       1    0.50    0.47    1624     295

Average
   in          0       0       0            0.00       0       0
   out         0       0       0            0.00       0       0
   local      21       2       2    0.39    0.38    2075     918

14:28:12  iget/s namei/s dirbk/s
14:29:12       0       2       1
14:30:12       0       4       1

Average        0       3       1

14:28:12 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
14:29:12       0       1     157       1       3       0
14:30:12       0       2      34       2       2       0
```

(continued on next page)

```
Average        0       1      95       1       3       0

14:28:12 proc-sz ov inod-sz ov file-sz ov lock-sz
14:29:12  28/200  0 297/300  0  63/  0  0    6/100
14:30:12  30/200  0 297/300  0  65/  0  0    6/100


14:28:12   msg/s  sema/s
14:29:12    0.00    0.00
14:30:12    0.00    0.00

Average     0.00    0.00

14:28:12  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
14:29:12    1.17   12.87   12.87    5.67   11.28    1.15
14:30:12    1.67    7.08    7.08    9.12    6.33    0.67

Average     1.42    9.97    9.97    7.40    8.81    0.91

14:28:12  pgout/s ppgout/s pgfree/s pgscan/s %s5ipf
14:29:12     0.00     0.00     0.35     8.18   0.00
14:30:12     0.00     0.00     0.00     0.00   0.00

Average      0.00     0.00     0.18     4.09   0.00

14:28:12 freemem freeswp
14:29:12     268    3034
14:30:12     351    3009

Average      310   3022

14:28:12 sml_mem   alloc  fail  lg_mem   alloc  fail  ovsz_alloc  fail
14:29:12   95232   73472     0  311296  198656     0      180224     0
14:30:12   95232   75120     0  311296  198656     0      180224     0

Average    95232   74296     0  311296  198656     0      180224     0

14:28:12 open/s create/s lookup/s readdir/s getpage/s putpage/s other/s
14:29:12
  in       0.00     0.00     0.00      0.00      0.00      0.00    0.08
  out      0.00     0.00     0.00      0.00      0.00      0.00    0.00
14:30:12
  in       0.00     0.00     0.00      0.00      0.00      0.00    0.08
  out      0.00     0.00     0.00      0.00      0.00      0.00    0.00

Average
```

(continued on next page)

```
  in       0.00     0.00     0.00      0.00      0.00      0.00    0.08
  out      0.00     0.00     0.00      0.00      0.00      0.00    0.00

14:28:12   snd-inv/s   snd-msg/s   rcv-inv/s   rcv-msg/s dis-bread/s   blk-inv/s
14:29:12         0.0         0.0         0.0         0.0         0.0         0.0
14:30:12         0.0         0.0         0.0         0.0         0.0         0.0

Average          0.0         0.0         0.0         0.0         0.0         0.0

14:28:12 serv/lo - hi    request   request    server    server
              3 -  6      %busy   avg lgth   %avail  avg avail
14:29:12        0          0.0         0        0.0          0
14:30:12        0          0.0         0        0.0          0

Average         0          0.0         0        0.0          0

14:28:12 evpoll/s evpost/s evtrap/s
14:29:12     0.00     0.00     0.00
14:30:12     0.00     0.00     0.00

Average      0.00     0.00     0.00
```

Reporting Application Turnaround with `timex`

The `timex` command records the amount of time taken by a command to execute, and reports the system activities that occurred during the time the command was executing. If no other programs are running, then `timex` can give you a good idea of which resources a specific command uses during its execution. A record of system consumption can be collected for each application program and then used for tuning the heavily loaded resources. In the following example, the `date` command is used.

```
$ timex -s date
Tue Aug 22 15:07:09 EDT 1990
real        0.17
user        0.00
sys         0.13


UNIX_System_V colorado 4.0 UE4.0 m88k    07/24/90

15:07:09    %usr    %sys    %sys    %wio   %idle
                   local  remote
15:07:09       8      90       0       2       0


15:07:09 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
15:07:09
   local     0       4     100       1       1       0       0       0
   remote    0       0       0       0       0       0

15:07:09  device   %busy   avque   r+w/s  blks/s  avwait  avserv

15:07:09 hdsk-0         2     1.0       1       2     0.0    20.0

15:07:09 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
15:07:09       0       0      31       0       1       0

15:07:09 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
15:07:09
   in           0       0       0            0.00       0       0
   out          0       0       0            0.00       0       0
   local      157      23       2    3.23    3.23   67918     775

15:07:09 swpin/s pswin/s swpot/s pswot/s pswch/s
15:07:09    0.00    0.0     0.00    0.0       14

15:07:09  iget/s namei/s dirbk/s
15:07:09       0      23       0

15:07:09 runq-sz %runocc swpq-sz %swpocc
15:07:09     1.0     100

15:07:09 proc-sz ov inod-sz ov file-sz ov lock-sz
15:07:09  28/200  0 300/300  0  61/  0  0   6/100

15:07:09   msg/s  sema/s
15:07:09    0.00    0.00
```

(continued on next page)

```
15:07:09  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
15:07:09   49.46    0.00    0.00   24.73   44.09    2.15

15:07:09  pgout/s ppgout/s pgfree/s pgscan/s %s5ipf
15:07:09     0.00     0.00     0.00     0.00   0.00

15:07:09 sml_mem   alloc  fail  lg_mem   alloc  fail  ovsz_alloc  fail
15:07:09   95232   73344     0  311296  200704     0      180224     0

15:07:09 freemem freeswp
15:07:09     428    3044

15:07:09 open/s create/s lookup/s readdir/s getpage/s putpage/s other/s
15:07:09
  in       0.00     0.00     0.00      0.00      0.00      0.00    0.00
  out      0.00     0.00     0.00      0.00      0.00      0.00    0.00

15:07:09   snd-inv/s   snd-msg/s   rcv-inv/s   rcv-msg/s dis-bread/s   blk-inv/s
15:07:09         0.0         0.0         0.0         0.0         0.0         0.0

15:07:09 serv/lo - hi   request   request    server    server
                3 -  6    %busy  avg lgth    %avail  avg avail
15:07:09          0        0.0         0       0.0         0
```

While date, for its simplicity, was used for the preceding demonstration, it is not representative of many commands because most commands use more system resources.

timex can be used in the following way:

timex -s *application_program*

Your application program will operate normally. When you finish running your application and exit, the timex result will be printed on your screen. This can be extremely interesting; you get a precise record of the system resources used while your program was executing.

If accounting is installed and running, then timex can present the information it collects as well. The following options can be used with the existence of accounting:

timex [-p[-fhkmrt]] [-o] *application_program*

See timex(1) for more information.

Reporting Location and Seek Distance with sadp

The sadp command has the following format:

sadp [-th] [-d *drive*] s [*n*]

sadp reports disk access locations and seek distance in tabular (-t) or histogram (-h) form. If neither option is designated, the two reports will be in tabular form. *drive* specifies the disk drive.

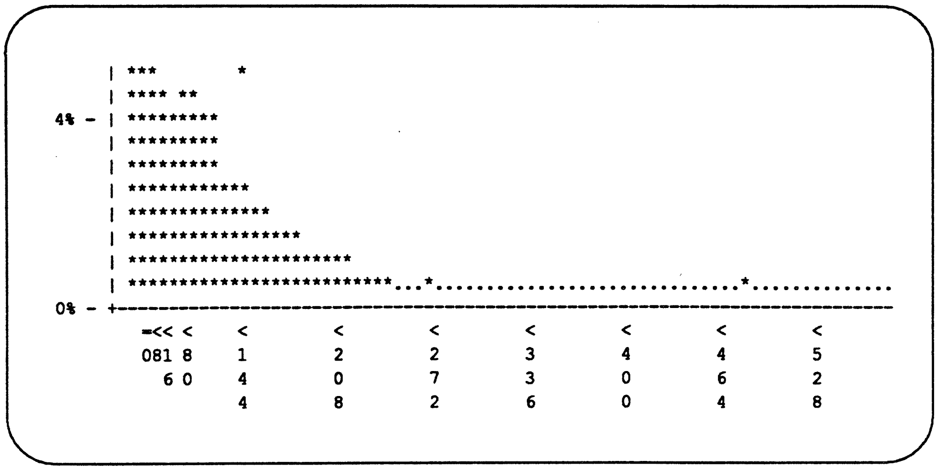
Disk activity is sampled once every second during a specified interval of length *n*. Slice usage and seek distance are recorded in units of sectors. The s option specifies the duration of the sampling interval in seconds. The sampling interval must be 10 seconds or greater. The *n* argument specifies the number of reports to be generated during the sampling interval. The default of *n* is 1.

Figure 10-1: Output from sadp: Seek Distance Histogram

```
SEEK DISTANCE HISTOGRAM
disk-0:
Total seeks = 30308
16% - | *
      | *
      | *
      | *
      | **
      | **
      | **
      | **
12% - | **
      | **
      | **
      | **
      | **
      | **
      | **
      | **
 8% - | **
      | **
      | **
      | **
      | ***
      | ***
```

(continued on next page)

Figure 10-1: Output from sadp: Seek Distance Histogram (continued)



```
     | ***        *
     | **** **
4% - | *********
     | *********
     | *********
     | *************
     | ***************
     | ******************
     | ***********************
     | ***************************...*.................................*...............
0% - +--------------------------------------------------------------------------------
        =<< <     <         <         <         <         <         <         <
        081 8     1         2         2         3         4         4         5
          6 0     4         0         7         3         0         6         2
                  4         8         2         6         0         4         8
```

Using the sadp output, you can identify the file systems with a large amount of I/O activity. In general, try to move files with high activity close together. This will reduce the number of seeks over large distances.

The graph above shows how finely you can tune your system. If, after a working period of weeks or months, you can identify which file systems are consistently the most active, you might consider repartitioning your disks to achieve the maximum from disk access activity (see the "File System Administration" and "Storage Device Management" chapters for more information).

Samples of Performance Management Procedures

This section describes typical approaches to performance management. First, it describes a general procedure for troubleshooting performance problems. Then, it provides a procedure for reconfiguring the system and shows a sample of a typical system reconfiguration. Finally, it provides a procedure for recovering after an unsuccessful attempt at reconfiguring the system.

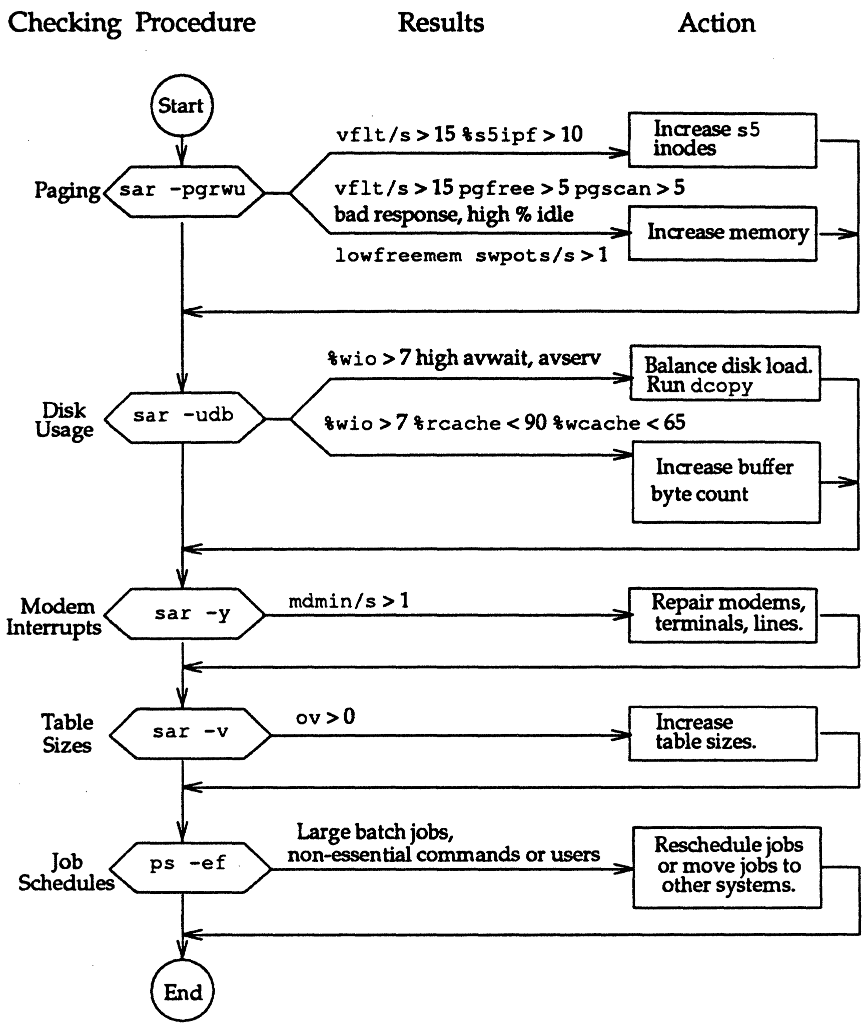
In this section, references are made to tunable configuration parameters. Refer to the the section "Tunable Parameters." for the default values of these parameters and complete instructions for altering them.

Investigating Performance Problems

Locating the source of the problem can require some careful detective work. Hence, the following is not a canned procedure, but a sample approach. It covers basic areas where problems usually surface, and suggests some actions that will alleviate the problem. The most common indication that a problem exists is consistently poor response time. If you have identified a familiar problem area and you need to make changes to your system parameters, see the section "Configuring the UNIX Operating System" in this chapter.

The following figure is an outline of the general approach to troubleshooting:

Figure 10-2: Outline of Typical Troubleshooting Procedure


Checking Procedure
Results
Action
Start
Paging
sar -pgrwu
vflt/s > 15 %s5ipf > 10
Increase s5 inodes
vflt/s > 15 pgfree > 5 pgscan > 5
bad response, high % idle
lowfreemem swpots/s > 1
Increase memory
Disk Usage
sar -udb
%wio > 7 high avwait, avserv
Balance disk load. Run dcopy
%wio > 7 %rcache < 90 %wcache < 65
Increase buffer byte count
Modem Interrupts
sar -y
mdmin/s > 1
Repair modems, terminals, lines.
Table Sizes
sar -v
ov > 0
Increase table sizes.
Job Schedules
ps -ef
Large batch jobs, non-essential commands or users
Reschedule jobs or move jobs to other systems.
End

Checking for Excess Swapping

The first thing to look at is the paging activity, since the paging-in and -out of pages is costly in both disk and CPU overhead. Get the `sar -pgrwu` report. By using this information you can reasonably determine whether more memory is needed or if the excess paging activity is due to too few inodes which indirectly causes reusable pages to be discarded.

If the `vflt/s` value shown by `sar -p` is greater than 15, then look at `sar -g`. High values (greater than 5) for `pgscan/s` and `pgfree/s` imply that the page-stealing daemon is working overtime to find free pages because of a memory shortage. This can be verified by looking at the `ps -elf` report which gives the number of cycles used by the page stealing daemon. The value of `%s5ipf` should also be considered. If it is greater than 10 percent then the freelist of S5 inodes is page-bound which causes reusable pages to be discarded whenever `iget` takes an inode off the freelist. This can be remedied by increasing the tunable `NINODES` found in `/etc/master.d/s5`.

Other indicators of a memory shortage are the `freemem` value of `sar -r` and the `swpot/s` value of `sar -w`. `swpot/s` greater than 1.0 is also an indicator of memory shortage.

If memory shortage occurs frequently then memory should be increased in some way. This can be done in two ways. Uninstalling optional kernel utilities that are not needed by your applications frees the memory used by the utilities so that it can be used by user applications. If this is not possible then extra memory probably needs to be added.

Checking for Disk Slowdowns

If the value of `%wio` (from the `sar -u` report above) is greater than 10 percent, or if the `%busy` for a disk drive (obtained by `sar -d`) is greater than 50 percent, then the system has a disk slowdown. Some ways to alleviate a disk slowdown are:

1. Increase the amount of buffer space.
2. Organize the file system to minimize disk activity. If you have two disks, distribute the file systems for a more balanced load.
3. Consider adding more memory if the situation persists. Additional memory reduces swapping/paging traffic and allows pages to remain in memory (reducing the number of user-level reads and writes that need to go out to disk).

4. Consider adding an additional disk and balancing the most active file systems across the two disks.
5. Consider increasing the logical block size of S5 file systems with lots of large files or even changing the file system type. See the section "Logical Block Size."

Checking for Modem Interrupts

Run `sar -y` to get a report describing activity on terminal devices. If the number of modem interrupts per second, `mdmin/s`, is much greater than 0, your system may have faulty communications hardware.

Checking for Table Overflows

To check for potential table overflows, get the `sar -v` report. This report will let you know if overflows have occurred in the process or inode tables. Overflows in these tables are avoided by increasing `NPROC` and `NINODES` in the `etc/master.d/kernel` and `/etc/master.d/s5` files.

Shifting the Workload to Off-Peak Hours

Examine the files in `/var/spool/cron/crontabs` to see if jobs are queued up for peak periods that might better be run at times when the system is idle. Use the `ps` command to determine what processes are heavily loading the system. Encourage users to run large, noninteractive commands (such as `nroff` or `troff`) at off-peak hours. You may also want to run such commands with a low priority by using the `nice` or `batch` shell commands.

Configuring the UNIX Operating System

During the boot procedure, the boot program reads from disk a program called unix, loads it into memory, and executes it. This file, unix (often referred to as the bootable operating system), defines the running UNIX system on your machine.

The absolute path name of this file is /stand/unix.

NOTE: /unix is symbolically linked to /stand/unix for compatibility with earlier releases.

For simplicity, we will refer to this file as the bootable operating system, or unix.

The unix that runs on your machine is specifically configured for the hardware and software currently on your machine.

All software changes that need to be incorporated into the bootable operating system are specified in the /stand/system file, or in one of the /etc/master.d files. For simplicity, we refer to /stand/system as the system file, and to files found in /etc/master.d as master files.

There are two reasons that the bootable operating system would need to be reconfigured:

- software changes were made to the system that need to be incorporated into the bootable operating system
- hardware resources were added to or removed from the system

The software changes that could be made to a system include the following:

- changing the definition of a driver in the /etc/master.d directory
- changing tunable parameters found in a master file
- adding or deleting a driver or module definition in a master file
- making changes to the system file
- adding or removing driver modules from /boot

While the system file is read directly by the configuration software, changes to master files require execution of the mkboot(1M) command before the configuration process is begun. First, you must change the appropriate file(s) in the directory /etc/master.d, and then execute a separate mkboot command for each changed master file. The mkboot command will read the master file and create a new object file in the /boot directory.

The /boot directory contains object files for use in the configuration process; these object files contain configuration information necessary for various hardware and software drivers on your system. The system file is used to specify which of the object files in /boot must be configured into unix, and which are to be excluded.

Some modules in /boot are included in the bootable operating system even if they are not explicitly included in the system file. If you want to explicitly exclude a driver in /boot, it is best to do so with an EXCLUDE statement in /stand/system (see system(4)). Do not move the driver to a new file name in /boot or some other directory to exclude it from the bootable operating system.

Hardware changes always require the configuration of a new unix. These changes include adding or removing a board, a tape drive, and so on.

One topic we have not considered is the installation of a new software package. Usually, the installation procedure for installing a new software package that requires a modification of the bootable operating system includes some modification of the system file and drivers in /boot; this is usually done directly by the installation software provided with the software package. The system is then rebooted at the end of the installation process, and a new bootable operating system is built, loaded, and executed.

Standard procedures for installing new hardware are given in the documentation that accompanied your computer.

Configuration Scenarios

The configuration of a new unix can occur in one of five ways:

- on powerup, the system detects that the system file is newer than the unix file
- a shutdown -i6 or init 6 is executed at the shell prompt and the system file is newer than unix

- the name of the system file (or another text file) is entered at the firmware prompt
- the cunix command is executed at the shell prompt
- a hardware change is made that makes the current unix inoperable

Of all the above methods, using the cunix command provides the most flexibility, and has the added advantages of allowing you to build a new bootable operating system without rebooting the machine, and without overwriting the currently running bootable operating system. The cunix command is particularly useful in development environments where the possibility exists of creating a bootable operating system that will not execute properly, or not at all, because of bad changes made to the system file, a bad change made to a master file, or symbol-referencing problems.

The remaining sections in this chapter describe configuring a new operating system through a reboot, recovering from an attempt to boot an unbootable operating system, and the use of cunix. A procedure for configuring a new mUNIX (a version of the UNIX operating system that runs during the configuration process) is also given.

Reconfiguring the System Through a Reboot

This section tells you how to configure a new bootable operating system through a system reboot. The procedure includes the modification of system tunable parameters as an example, though any change in the hardware or software configuration of your system could be substituted for that part of the procedure.

There are four things to remember when reconfiguring the operating system:

1. Always copy the existing bootable operating system (/stand/unix) to another file; it is recommended that you copy it to a file in the root directory and not to another file in the /stand directory. This way, in the event you create an unbootable unix, you can boot mUNIX from firmware and copy the existing bootable operating system from the root directory back to /stand/unix.

2. For each driver or module that you modified in /etc/master.d, execute a separate mkboot command.
3. When reconfiguring the operating system do not arbitrarily change the node name (NODE) of the computer. If basic networking has been established, a change in node name must be coordinated with all interfacing systems.
4. Take detailed notes of everything you do so that you can identify and correct any mistakes you may make. In the event that you need to contact a service representative for your computer, your notes will be an invaluable aid in resolving your problem.

There are two major steps in reconfiguring the operating system:

1. change the system configuration (in this example, modify tunable parameters)
2. rebuild the operating system

These steps are described in the procedures below.

Modifying the Tunable Parameters

Step 1: Log in as root.

Step 2: Copy the existing /stand/unix to /oldunix.

```
# cp /stand/unix /oldunix
```

Step 3: Change the present working directory to /etc/master.d.

```
# cd /etc/master.d
```

Step 4: Edit the applicable master files to modify the tunable parameters. Keep a record of all changes you make; this is important in case you make a mistake that results in the configuration of an unbootable operating system or the failure of the configuration process.

Configuring a New Bootable Operating System

Step 5: Change the present working directory to /boot.

```
# cd /boot
```

Step 6: Execute the mkboot command to create a bootable object file for each of the files modified in /etc/master.d. For example, if some of the tunable parameters in the /etc/master.d/kernel were modified, you would enter:

```
# /usr/sbin/mkboot -k KERNEL
```

If the tunables that you changed were in another file (for example, /etc/master.d/sem), the mkboot command does not take the -k option:

```
# /usr/sbin/mkboot SEM
```

Step 7: Execute the cunix command to make a new unix:

```
# cunix -o /newunix
```

If cunix works successfully, move newunix to /stand/unix:

```
# mv /newunix /stand/unix
```

and reboot, otherwise fix the problems and try again. (For more information in cunix, see cunix(1M).)

NOTE Using the touch command on the file stand/system and then rebooting will also remake your system, as in previous releases, but will not allow you to check for errors before you shut down.

See "Sample System Configuration" in the next section for an example of the prompts that appear during the reboot and configuration process.

Step 8: Reboot the system:

```
#shutdown -i6
```

If the system will not boot, you must boot `/stand/mUNIX` to bring up a system so that you can repair problems.

If you cannot determine and/or fix the problem, you must undo the changes made to the files in `/etc/master.d` in Step 4; then, repeat Steps 5 and 6. After completing Step 6, move `/oldunix` back to `/stand/unix` and reboot.

Sample System Reconfiguration

The following is an illustration of a typical scenario for reconfiguring supported DeltaSeries and DeltaSERVER platforms to add more memory.

Because of the additional memory, many tunable parameters should be increased. Most of them are in the `/etc/master.d/kernel` file. The command line entries and system responses in the illustration below show the reconfiguration and rebooting of the operating system to support these new parameters. The illustration also indicates that tunable parameters for semaphores are being modified. The editing of the `/etc/master.d/kernel` and `/etc/master.d/sem` files is not shown.

```
# cp /stand/unix /oldunix
# cd /etc/master.d
# ed kernel

        NOTE:  Editing of /etc/master.d/kernel is not shown.

q
# cd /boot
# mkboot -k KERNEL
# cd /etc/master.d
# ed sem

        NOTE:  Editing of /etc/master.d/sem is not shown.

q
# cd /boot
# mkboot SEM

        NOTE:  If parameters in other /etc/master.d files are changed,
        execute mkboot on the uppercase name for each changed file.
        Only the KERNEL file requires the -k option.
        See mkboot(1M).

# cd
# touch /stand/system
```

(continued on next page)

```
# shutdown -i6
```

NOTE: A series of messages are displayed ending with the following:

```
INIT: New Run level: 6
The system is coming down. Please wait.
System services are now being stopped.
A new unix is being built

/stand/unix is being created

CONFIGURATION SUMMARY
=====================
     ----driver---- #devices major
     LOG                1     50
     CLONE              1     63
     PRF                1     49
     SXT                1     48
     GENTTY             1     20
     PORTS              2      1,   2
     MEM                1     18
     IUART              1      0
     IDISK              1     17
     HDELOG             1     16

     ----module---
     INTP
     ELF
     COFF
     FIFOFS
     BFS
     .
     .
     .

     ----device info----
                   major              minor
     rootdev         17                  0
     swapdev         17                  1
```

NOTE: A series of messages is displayed ending with the following:

```
The system is down.

NOTICE:  System Reboot Requested (0)
```

Recovering from an Unbootable Operating System

If your attempt at configuring a new bootable operating system is unsuccessful, resulting in an unbootable operating system or the failure of the configuration process, you can get a viable version of the system running using the procedure outlined below. This procedure can also be used if you configure a new bootable operating system that performs poorly and you want to recover the previously used /stand/unix.

It is assumed that you previously saved a copy of /stand/unix as /oldunix.

Step 1: If the system is in the firmware state, skip to Step 2.

If you are in single- or multi-user mode, bring the system to the firmware state:

```
# shutdown -i5
```

Note that if you are in multi-user mode, you must be logged in as root.

If you are in the shell spawned during an error in the configuration process, use exit or CTRL-d to go to firmware mode.

If for some reason the system is not able to come up at all, see the hardware documentation for your computer for instructions on getting the system to firmware mode in this case (most computers are equipped with some form of hardware reset switch).

Step 2: When the firmware prints the COLD START message, enter any key. The firmware should respond with a Menu followed by:

```
Enter Menu #
```

Enter 3.

When prompted with:

```
181-DIAG>
188-BUG>
```

Enter:

```
bo 8 0 /stand/mUNIX
   6 0
```

The system boots mUNIX, a version of the operating system that always resides in /stand.

Step 3: After the system has rebooted and you have logged in as root, move /oldunix back to /stand/unix:

```
# mv /oldunix /stand/unix
```

This returns the previous working version of the operating system to /stand/unix.

Step 4: If you made any changes to files in /etc/master.d or to the /stand/system file before a configuration attempt that failed, undo all the changes made at this time (if you have not done so already) so that these system files match the current bootable operating system in /stand/unix. If you had saved copies of these files before changing them, simply move the old files back to /etc/master.d. Then, execute a separate mkboot for each module corresponding to the master files as shown in the section "Configuring a New Bootable Operating System."

Step 5: Reboot the system:

```
# shutdown -i6
```

Step 6: When the Console Login: prompt appears you can log in to your system.

Try to determine what you did to cause the configuration process to fail so that you can avoid the problem in the future. If you are unable to determine what went wrong, and repeated attempts at configuring a new operating system fail, retain all notes and other documentation that you kept during the attempt(s) (including changes made to master files and/or the /stand/system file), and contact your computer service representative.

User-Level Configuration of the UNIX System

Configuring unix at the user-level has several major advantages over configuring unix automatically through a reboot:

- no reboot of the system is necessary (i.e., system remains available to all users)
- the reconfigured operating system can be placed in a file other than /stand/unix, leaving the current bootable operating system intact
- an alternate system file and /boot directory can be specified

Typically, the cunix command will be used as in the following example:

```
cunix [-f system] [-o new_unix]
```

The -f option specifies the pathname of the system file to be used for the configuration of the new bootable operating system. By default, this is /stand/system, but can be any text file in system(4) format.

There are also other options to cunix that allow you to further customize your configuration environment. See the cunix(1M) manual page for more information.

Configuring a New mUNIX

As previously described, mUNIX is a version of the UNIX system that runs during the configuration process. It was configured originally using the /stand/mini_system file as the system file.

You may want to configure a new mUNIX to account for your particular operating environment. To do this, just make changes to the /stand/mini_system file as necessary and execute the cunix command, as in the following example:

```
cunix -f /stand/mini_system -o new_mUNIX
```

This command will build new_mUNIX (in the current directory) using the mini_system file as the system file. Do not make any changes to master files unless you also intend to reconfigure unix along with mUNIX.

Once the configuration process is complete, you should move the new mUNIX to /stand, optionally saving the old one before doing so, as in the following:

```
mv /stand/mUNIX old_mini_unix
mv new_mUNIX /stand/mini_unix
```

This causes your newly configured mUNIX to be the one used for future configurations of the operating system, and retains a copy of the old mUNIX in the current directory. Of course, you can keep an old copy of mUNIX in any directory, including the root directory.

Tunable Parameters

Tunable system parameters are used to set various table sizes and system thresholds to handle the expected system load. Caution should be used when changing these variables since such changes can directly affect system performance. For the most part, the initial tunable parameter values for a new computer are acceptable for most configurations and applications. If your application has special performance needs, you may have to experiment with different combinations of parameter values to find an optimal set.

Note that whenever a parameter's value is being changed from the default value to a much higher value, you should read the `master.d` file to determine the type of parameter data that affects its maximum value.

The tunables for the core package can be found in the following `/etc/master.d` files delivered with the core package:

`kernel`	Kernel Tunables
`hrt`	High Resolution Timers
`mvme332xt`	mvme332 Board
`log`	STREAMS Log Driver
`sad`	STREAMS Administrative Driver
`ts`	Time Sharing Scheduler
`s5`	System V File System Type

Figure 10-5 shows the default values for the tunable parameters found in these files for systems equipped with 4 megabytes of random access memory (RAM).

Also included in Figure 10-5 are the default values for the tunable parameters found in the in the `/etc/master.d` files of the following packages:

UFS Utilities

`/etc/master.d/ufs`	4.2BSD Fast File System

System Performance Analysis Utilities

`/etc/master.d/prf`	Kernel Profiler

Interprocess Communication Utilities

`/etc/master.d/msg`	Messages
`/etc/master.d/shm`	Shared Memory
`/etc/master.d/sem`	Semaphores

The following packages also have tunable parameters found in /etc/master.d files. The files associated with the package and where the respective tunables are found are also given.

Internet Utilities

/etc/master.d/arp	Address Resolution Protocol
/etc/master.d/ip	Internet Protocol
/etc/master.d/tcp	Transmission Control Protocol
/etc/master.d/udp	User Datagram Protocol
/etc/master.d/llcloop	Loopback Driver

Found in: *Network User's and Administrator's Guide*

Network File System Utilities

/etc/master.d/nfs	Network File System

Found in *Network User's and Administrator's Guide*

Remote File Sharing Utilities

/etc/master.d/rfs	Remote File Sharing

Found in: *Network User's and Administrator's Guide*

XENIX Compatibility Package

/etc/master.d/xnamfs	XENIX Semaphores

The following notes apply to Figure 10-5:

- The parameters are set to specific values, as defined in the appropriate /etc/master.d file. The default value and the size in bytes for each entry are shown in the figure.
- A dash (-) is used in the size information to indicate parameters that do not affect the size of the kernel when the values are changed. These parameters instead act as flags, limits, or provide a naming function.

Figure 10-3: Suggested Parameter Values

master.d File	Tunable Type	Parameter	Default Value	Size per Entry in Bytes
kernel	Gen. Kernel Tunables	NCALL	60	16
		ARG_MAX	5120	-
		FLCKREC	300	-
		IOPBMEM	91	4096
		MAPALLPHYS	1	-
		MAXUP	100	-
		NCLIST	0	-
		NPROC	200	212
		ROOTFSTYPE	"s5"	-
	System Information	SYS	"UNIX_Sys._V"	-
		NODE	"unix"	-
		REL	"4.0"	-
		VER	"2"	-
		SRPC_DOMAIN	""	-
	Hardware Information	ARCHITECTURE	"m68000" or "MC88100"	-
		HW_SERIAL	""	-
		HW_PROVIDER	"MOTOROLA"	-
	Buffer Cache	NBUF	100	88
		NHBUF	64	12
		NPBUF	20	52
		BUFHWM	200	-

Figure 10-3: **Suggested Parameter Values** (continued)

master.d File	Tunable Type	Parameter	Default Value	Size per Entry in Bytes
	Paging	FSFLUSHR	1	-
		NAUTOUP	60	-
		SPTMAP	0	
		KVSIZE	0	
		MAXPMEM	0	
		GPGSLO	25	-
		MINARMEM	25	-
		MINASMEM	25	-
	Parity Error	PARITYON	1	-
		IO_PE_RETRYS	3	-
	Per Process Limits	SHLBMAX	2	-
		SCPULIM	0x7fffffff	-
		HCPULIM	0x7fffffff	-
		SFSZLIM	0x100000	-
		HFSZLIM	0x100000	-
		SDATLIM	0x1000000	-
		HDATLIM	0x1000000	-
		SSTKLIM	0x1000000	-
		HSTKLIM	0x1000000	-
		SCORLIM	0x100000	-
		HCORLIM	0x1000000	-
		SFNOLIM	0x40	-
		HFNOLIM	0x400	-
		SVMMLIM	0x1000000	-
		HVMMLIM	0x1000000	-
	File Access Features	RSTCHOWN	1	-

Figure 10-3: Suggested Parameter Values (continued)

master.d File	Tunable Type	Parameter	Default Value	Size per Entry in Bytes
		NGROUPS_MAX	16	-
	STREAMS	NSTRPUSH	9	-
		STRCTLSZ	1024	-
		STRMSGSZ	0	-
		STRTHRESH	2097152	-
	Scheduler Information	MAXCLSYSPRI	99	-
		SYS_NAME	"SYS"	-
		INITCLASS	"TS"	-
	XENIX Shared Data	XSDSEGS	0	12
		XSDSLOTS	0	20*XSDSEGS
hrt	High Resolution Timers	HRTIME	50	60
		HRVTIME	50	60
mvme332xt	mvme332 board	DEFLD	0	4
		DEFVTIME	1	4
		DEFVMIN	64	4
log	STREAMS Logging	NLOG	16	12
sad	STREAMS Admin. Driver	NSTRPHASH	64	4
		NAUTOPUSH	32	44

Figure 10-3: Suggested Parameter Values (continued)

master.d File	Tunable Type	Parameter	Default Value	Size per Entry in Bytes
ts	Time Sharing	TSMAXUPRI	20	-
s5	S5 File Type	NINODE	800	132
ufs	Fast File System	UFSNINODE	600	268
		NDQUOT	200	60
prf	Profiler	PRFMAX	4096	1
msg	Message	MSGMAP	100	8
		MSGMAX	2048	-
		MSGMNB	4096	-
		MSGMNI	50	53
		MSGSSZ	8	1024
		MSGTQL	40	12
		MSGSEG	1024	8
sem	Semaphores	NBPW	4	-
		SEMMAP	10	8
		SEMMNI	10	84
		SEMMNS	60	12
		SEMMNU	30	8x(SEMUME+2)
		SEMMSL	25	-
		SEMOPM	10	8
		SEMUME	10	8x(SEMMNU)
		SEMVMX	32767	-
		SEMAEM	16384	-
shm	Shared Memory	SHMMAX	131072	-
		SHMMIN	1	-
		SHMMNI	100	112
		SHMSEG	6	see text below

Kernel Tunables

General Kernel Tunables

The following general kernel parameters are defined in the /etc/master.d/kernel file.

ARG_MAX
: This is the maximum number of characters (including NULL characters) allowed in the argument and environment strings passed to an exec system call. This can be increased to allow larger argument lists, but it should not be less than 5120. If it is increased, it should be no more than about an eighth of SSTKLIM (see "Per Process Limits" below) so that there is room for both the pointer arrays and the ordinary stack frames.

FLCKREC
: Specifies the number of records that can be locked by the system. The default is 300.

MAPALLPHYS
: Used to control the mapping of physical memory. If this parameter is non-zero, all physical memory is mapped 1-to-1, allowing the system to take advantage of some performance improvements. This tuneable is only available on the 88000.

MAXUP
: Specifies how many concurrent processes a non-superuser is allowed to run. The entry is normally in the range of 15 to 25. This value should not exceed the value of NPROC (NPROC should be at least 10 percent more than MAXUP). This value is per user identification number, not per terminal. For example, if twelve people are logged in with the same user identification, the default limit would be reached very quickly.

NCALL
: Specifies how many call-out table entries to allocate. Each entry represents a function to be invoked at a later time by the clock handler portion of the kernel. This value must be greater than 2 and is normally in the range of 10 to 70. The default value is 60. Each entry contains 16 bytes.

Software drivers may use call entries to check hardware device status. When the call-out table overflows, the system crashes and outputs the following message on the system console:

```
PANIC: Timeout table overflow
```

NCLIST

NCLIST is not used by supported Delta Series and DeltaSERVER platforms.

NPROC

Specifies how many process table entries to allocate. Each table entry represents an active process. The swapper is always the first entry and /sbin/init is always the second entry. The number of entries depends on the number of terminal lines available and the number of processes spawned by each user. The average number of processes per user is in the range of 2 to 5 (also see MAXUP, default value 25). When full, the fork(2) system call returns the error EAGAIN. The default value of NPROC is 200. It should probably be no less than 50.

PUTBUFSZ

The size, in bytes, of the circular buffer putbuf used to record various system messages, including PANIC messages. The messages in putbuf can be seen in a system dump by using the crash command to print the contents of putbuf as ASCII characters. There should be no reason to change this value unless you are doing operating system development.

ROOTFSTYPE

Specifies the file system type of the root file system. This is used to determine the format of the file system.

System Information

The following system information tunables are defined in the /etc/master.d/kernel file.

SYS

Specifies the system name. The default system name is UNIX_System_V (see the procedure "Changing the System Name and Node Name" in the "System Setup" chapter).

`NODE` Specifies the node name of the system. The default node name is `unix` (see the procedure “Changing the System Name and Node Name” in the “System Setup” chapter.

`REL` Specifies the UNIX system release.

`VER` Specifies the version. This value may be 1 or 2.

`SRPC_DOMAIN` The name of the “Secure RPC Domain,” the realm in which a uid space is unique. When using secure RPC, each user is assigned a “netname” which is of the form

```
Operating System.UserId@Realm
```

So, if the user id of each user is the same for machines A, B, and C, then `SRPC_DOMAIN` should be set to the same name on A, B, and C. Thus, if the `SRPC_DOMAIN` name for machines A, B, and C is “documentation,” then user 1701 from any one of those machines would have the netname

```
    unix.1701@documentation
```

See “RPC Administration” in the *Programmer's Guide: Networking Interfaces* for more information on secure domain.

Hardware Information

The following parameters are defined in the `/etc/master.d/kernel` file.

`ARCHITECTURE` The machine architecture information.

`HW_SERIAL` The serial number of your machine. Its value, of course, must be filled in by the administrator.

`HW_PROVIDER` The hardware provider's name.

Buffer Cache

The following parameters are defined in the `/etc/master.d/kernel` file.

`NBUF` Block I/O uses both buffers and buffer headers. Whenever a buffer header is needed, but no free ones are available, the system dynamically allocates more buffer headers in chunks of `NBUF` at a time. There is no limit to the total number of buffer headers in the system, however, the tunable BUFHWM limits the number of kilobytes that may be used by buffers. This effectively limits the number of buffer headers that will be allocated.

Once allocated, buffer header space cannot be freed for other uses. Thus, care should be taken when raising the value of NBUF. A higher value of NBUF will decrease the number of times the Kernel Memory Allocator must be called to allocate space for buffer headers, but this could also result in the allocation of headers that are not used.

NHBUF

Specifies how many "hash buckets" to allocate for 1K buffers. These are used to search for a buffer given a device number and block number, rather than a linear search through the entire list of buffers. This value must be a power of 2. Each entry contains 12 bytes. NHBUF must be specified in /etc/master.d/kernel.

NPBUF

Specifies how many physical I/O buffers to allocate. One I/O buffer is needed for each physical read or write active. Each entry contains 52 bytes. The default value is 20.

BUFHWM

BUFHWM limits the number of kilobytes of memory that can be used by block I/O buffers. If sar -b shows the buffer hit ratio to be low, then BUFHWM and/or NBUF should be increased.

Paging

A paging daemon, pageout, exists in the system. Its sole responsibility is to free memory as the need arises. It uses a "least recently used" algorithm to approximate process working sets and writes those pages that have not been touched during some period of time out to disk. The page size is 2048 bytes. When memory is exceptionally tight, the working sets of entire processes may be swapped out.

The first two tunables are for file system hardening. The remaining tunable parameters determine how often pageout runs and under what conditions. The default values in /etc/master.d/kernel should be adequate for most applications.

FSFLUSHR

This is the file system flush rate. It specifies the rate in seconds for checking the need to write the file system buffers, modified inodes, and mapped pages to disk. The default is one second. (This has replaced BDFLUSHR of previous releases.)

NAUTOUP

The NAUTOUP entry specifies the buffer age in seconds for automatic file system updates. System buffers and other cached file attributes (such as inodes) are written to the hard disk when they have been memory-resident for the interval specified by the NAUTOUP parameter. Specifying a smaller limit increases system reliability by writing the buffers to disk more frequently and decreases system performance. Specifying a larger limit increases system performance at the expense of reliability.

SPTMAP

The number of map entries for free space accounting for the dynamic portion of the kernel virtual address space. The dynamic kernel space is used by drivers and kmem_alloc for execution time allocation of kernel memory. It should never be made smaller. It should be made larger if the following warning message is seen:

```
rmfree map overflow SPTMAPADDR.
Lost N items at LOSTADDRESS.
```

where SPTMAPADDR is the hexadecimal address of the sptmap array and LOSTADDRESS is a kernel virtual address in the dynamic allocation address interval. (On the M68000 family of processors, this interval is 0x46000000 and the upper end is configurable. On the M88000 family of processors, this interval is 0x40000000 and the upper end is also configurable.)

MAXPMEM

Specifies the maximum amount of physical memory to use in pages. The default value of 0 specifies that all available physical memory be used.

GPGSLO

Specifies the low water mark of free memory in pages for pageout to start stealing pages from processes. The default is 25. Increase the value to make the daemon more active; decrease the value to make the daemon less active (must be an integer ≥ 0.)

MINARMEM Specifies the minimum number of memory pages reserved for the text and data segments of user processes.

MINASMEM Threshold value that specifies the number of memory and swap pages reserved for system purposes (unavailable for the text and data segments of user processes).

PAGES_UNLOCK Unused.

Per Process Limits

SHLBMAX Specifies the maximum number of shared libraries that can be attached to a process at one time. This applies only to COFF shared executables.

The following tunables are soft and hard limit pairs on process resource limits. These limits are given to process 0; thereafter child processes inherit the parent process's hard and soft limits. However, whenever a process execs a file whose set-user-id or set-group-id bit has been set, the resource limits of that process are reinitialized to the default system limits.

Processes can change their own values of these limits using setrlimit (see getrlimit(2)). Soft limits may be changed but must remain less than or equal to the hard limits. Only processes whose effective user ID is equal to 0 (root) may raise their hard limits. Any process may lower its hard limit.

A value equal to RLIMIT_INFINITY (0x7fffffff) indicates a resource without limitation.

See getrlimit(2) for more information on hard and soft limits.

SCPULIM The soft limit of the maximum combined user and system CPU time, in seconds, that a process is allowed. A SIGXCPU signal will be sent to processes whose CPU time exceeds this value.

HCPULIM The maximum value of SCPULIM.

SFSZLIM The soft limit specifying the largest offset, in bytes, of any single file that may be created by the process. A SIGXFSX signal will be sent to processes that attempt to write a file whose offset is greater than this value. In addition, the write will fail with an EFBIG error.

HFSZLIM The maximum value of SFSZLIM.

SDATLIM The soft limit specifying the maximum size, in bytes, of a process's heap. If a process attempts to extend its heap beyond this limit using brk(2), the attempt will fail and errno will be set to ENOMEM.

HDATLIM The maximum value of SDATLIM.

SSTKLIM The soft limit specifying the maximum size, in bytes, of the stack segment for a process. This defines the limit of automatic stack growth by the system. A SIGSEGV signal will be sent to processes that attempt to grow the stack beyond this value. Unless the process has arranged to catch this signal on a separate stack (see signalstack(2)) this will terminate the process.

HSTKLIM The maximum value of SSTKLIM.

SCORLIM The soft limit specifying the largest size, in bytes, of a core file that may be created. A soft limit of 0 will prevent the creation of core files.

HCORLIM The maximum value of SCORLIM.

SFNOLIM The soft limit specifying the maximum number of open files the process may have. When this limit is exceeded, attempts to open files will fail and errno will be set to EMFILE.

HFNOLIM The maximum value of SFNOLIM.

SVMMLIM The soft limit specifying the maximum address space that may be mapped to a process. Attempts to increase a process's address space beyond this value (i.e., brk(2), shmat(2), mmap(2)) will fail with a ENOMEM error.

HVMMLIM The maximum value of SVMMLIM.

File Access Features

RSTCHOWN — RSTCHOWN is the restricted file ownership changes flag. Only 0 and 1 are valid values for RSTCHOWN. A value of 0 is the System V Release 3 compatibility mode. As in Release 3, the owner of a file can change user ID and group ID of the file to any value, including nonexistent user IDs and group IDs. RSTCHOWN set to 1 designates the FIPS/BSD compatibility mode. This restricts the ability to change ownership of the file. Only the superuser or root processes (those whose UID is 0) are able to change the ownership of a file. The owner of the file may only change the group ID of the file to one of the groups in which the owner has membership (see getgroups(1)). Superuser and root processes may change the group ID of any file to any value. (RSTCHOWN set to 1 is FIPS/BSD compatibility mode.)

NGROUPS_MAX — Specifies the maximum number of groups in which a process can have membership (see getgroups(1)).

STREAMS

The following tunable parameters are associated with STREAMS processing. These parameters are defined in the /etc/master.d/kernel file.

NSTRPUSH — The maximum number of modules that may be pushed onto a Stream. This is used to prevent an errant user process from consuming all of the available queues on a single Stream. By default this value is 9, but in practice, existing applications have pushed, at most, four modules on a Stream.

STRMSGSZ — The maximum allowable size of the data portion of any STREAMS message. This should usually be set just large enough to accommodate the maximum packet size restrictions of the configured STREAMS modules. If it is larger than necessary, a single write or putmsg can consume an inordinate number of message blocks. A value of zero indicates no upper bound. A value of 4096 is sufficient for existing applications.

STRCTLSZ

The maximum allowable size of the control portion of any STREAMS message. The control portion of a putmsg message is not subject to the constraints of the minimum/maximum packet size, so the value entered here is the only way of providing a limit for the control part of a message. The recommended value of 1024 is more than sufficient for existing applications.

STRTHRESH

The maximum total of bytes streams are normally allowed to allocate. When the threshold is passed, users without the appropriate privilege will not be allowed to open streams, push streams modules, or write to streams devices; they will fail with ENOSR (out of streams resources). Users with appropriate privilege will always be allowed to do anything. Note also that the threshold applies to the output side only, thus data coming into the system (for example, the console) is not affected and will continue to work properly. A value of zero means there is no threshold. STRTHRESH should be set to about 1/4 to 1/2 of the total system memory. The default of 2000000 (approximately 2 megabytes) is a good maximum for a 4 megabyte system.

Scheduler Information

The following parameters are defined in the /etc/master.d/kernel file.

MAXCLSYSPRI

Maximum global priority used by the SYS scheduling class for scheduling kernel processes. Changing this changes the range of priorities used to schedule kernel processes and can have a significant effect on the performance of the system. In general, there is no need to change this unless you are adding new scheduling classes or reconfiguring the priorities of other currently configured classes. If it is set to a value below 39, the kernel will automatically set it to 39 at boot time because it needs a range of 40 priorities for the SYS class. (See the "Process Scheduling" chapter for detailed information.)

SYS_NAME — The character string name of the system scheduling class. There is no need to change the default unless you are configuring a different scheduling class with the name SYS.

INITCLASS — Specifies the scheduling class assigned to the init process. This class will be inherited by all processes on the system except descendents of a process whose class has been reset using priocntl(2). Should not be changed without good reason.

XENIX Shared Data

The following parameters are defined in the /etc/master.d/kernel file.

XSDSEGS — Specifies the number of shared data segments in the system. The minimum value is 1, and the default and maximum value is 25.

XSDSLOTS — (XSDSEGS x XSDSLOTS) specifies the maximum number of shared data segment attachments allowed in the system. The minimum value of XSDSLOTS is 1, and the default and value of XSDSLOTS is 1, and the default and maximum value is 3.

High Resolution Timers

The configuration parameters for High Resolution Timers are found in the /etc/master.d/hrt file. They are:

HRTIME — HRTIME is used to define the size of the hrtimes array. The hrtimes array is used for keeping track of sleep and alarm requests for the standard, real-time clock.

HRVTIME — HRVTIME is used to define the size of the itimes array. The itimes array is used for keeping track of the alarm requests for the clocks measuring user process virtual time and a process's virtual time.

MVME332xt Board

The configurable parameters for the MVME332xt board are found in the /etc/master.d/mvme332xt file. They are:

DEFLD	the default line discipline, from 0 to 6
DEFVTIME	the default minimum time to wait for an incoming character in raw mode
DEFVMIN	the default minimum number of incoming characters to wait for in raw mode

STREAMS Log Driver

The configurable parameter for the STREAMS log driver is found in the file /etc/master.d/log. It is:

NLOG	The number of minor devices that are available through the clone interface of the log driver (/dev/log). If an open of /dev/log fails with errno set to ENXIO, this number may need to be increased.

STREAMS Administrative Driver

The configurable parameters for the STREAMS Administrative Driver (SAD) are found in the file /etc/master.d/sad. They are:

NSTRPHASH	The size of the internal hash table. This will probably never need to be changed unless the number of drivers on the system gets very, very large.
NAUTOPUSH	The number of devices that can be configured to be autopushed. If the SAD_SAP ioctl fails with errno set to ENOSR, then this number should be increased.

Time Sharing Scheduler

The following parameter for the Time Sharing Scheduler is found in the `/etc/master.d/ts` file.

`TSMAXUPRI` The range within which users may adjust the user priority of a time-sharing process is `-TSMAXUPRI` to `+TSMAXUPRI`. Configuring higher values gives users more control over the priority of their processes (note that only super-user can raise priority in any case). The default value of 20 provides a degree of control equivalent to what has been available in the past through the `nice`(2) interface.

S5 File System Type

The following parameter for the System V File System is found in the `/etc/master.d/s5` file.

`NINODE` The number of inode entries in the S5 inode table. If `sar -v` shows that table overflows are occurring or if `sar -g` shows `%s5ipf` is greater than 10 percent, then the value should be raised. On the other hand, if `sar -v` consistently shows that the inode table is underutilized, then the value could be lowered. `NINODE` should be greater than `ncsize` which is specified in `/etc/master.d/kernel`. `ncsize` determines the number of inodes used by the directory lookup cache. A general guideline for the value of `NINODE` is 100 S5 inode entries for each megabyte of memory or `ncsize` + 100, whichever is greater.

Fast File System Type

The following parameters are associated with the Fast File System and are found in the `/etc/master.d/ufs` file.

`UFSNINODE` — The number of `ufs` inode table entries. If the main file system is `ufs`, then 100 inode entries per megabyte of memory should be allocated and the number should be greater than the value `ncsize` in `/etc/master.d/kernel`.

`NDQUOT` — The size of the kernel quota table. There is one entry for each user, thus, `NDQUOT` should be more than the maximum number of users that can be logged onto the system. If quotas are in effect, the table entries limit the amount of disk space a user can use. If there are no available entries, the message

```
dquot table full
```

will be printed on the console. If this occurs, the value of `NDQUOT` should be increased.

Profiler

The following parameter is associated with the profiler and is found in the `/etc/master.d/prf` file.

`PRFMAX` — Used by the kernel profiler as a maximum expected number of kernel addresses. This value should be increased if the message

```
too many text symbols
```

is printed when `/usr/sbin/prfld` is run.

Interprocess Communication

Messages

The following tunable parameters are associated with interprocess communication messages. These parameters are defined in the /etc/master.d/msg file. The order in which they are described follows the order in which they are defined in the output of the /usr/sbin/sysdef command.

MSGMAP — Specifies the size of the control map used to manage message segments. Default value is 100. Each entry contains 8 bytes.

MSGMAX — Specifies the maximum size of a message. The default value is 2048. The maximum size is 64 kilobytes -1.

MSGMNB — Specifies the maximum length of a message queue. The default value is 4096.

MSGMNI — Specifies the maximum number of message queues systemwide (id structure). The default value is 50.

MSGSSZ — Specifies the size, in bytes, of a message segment. Messages consist of a contiguous set of message segments large enough to fit the text. The default value is 8. The value of MSGSSZ times the value of MSGSEG must be less than or equal to 131,072 bytes (128 kilobytes).

MSGTQL — Specifies the number of message headers in the system and, thus, the number of outstanding messages. The default value is 40. Each entry contains 12 bytes.

MSGSEG — Specifies the number of message segments in the system. The default value is 1024. The value of MSGSSZ times the value of MSGSEG must be less than or equal to 131,072 bytes (128 kilobytes).

Semaphores

The following tunable parameters are associated with interprocess communication semaphores. These parameters are defined in the `/etc/master.d/sem` file. The order in which they are described follows the order in which they are defined in the output of the `/usr/sbin/sysdef` command.

`NBPW` — The number of bytes per word. This should not be changed since it is used to calculate the sizes of some tunables.

`SEMMAP` — Specifies the size of the control map used to manage semaphore sets. The default value is 10. Each entry contains 8 bytes.

`SEMMNI` — Specifies the number of semaphore identifiers in the kernel. This is the number of unique semaphore sets that can be active at any given time. The default value is 10. Each entry contains 34 bytes.

`SEMMNS` — Specifies the number of semaphores in the system. The default value is 60. Each entry contains 12 bytes.

`SEMMNU` — Specifies the number of undo structures in the system. The default value is 30. The size is equal to 8 x (SEMUME + 2) bytes.

`SEMMSL` — Specifies the maximum number of semaphores per semaphore identifier. The default value is 25.

`SEMOPM` — Specifies the maximum number of semaphore operations that can be executed per `semop`(2) system call. The default value is 10. Each entry contains 8 bytes.

`SEMUME` — Specifies the maximum number of undo entries per undo structure. The default value is 10. The size is equal to 8*(SEMMNU) bytes.

`SEMVMX` — Specifies the maximum value a semaphore can have. The default value is 32,767. The default value is the maximum value for this parameter.

SEMAEM — Specifies the adjustment on exit for maximum value, alias semadj. This value is used when a semaphore value becomes greater than or equal to the absolute value of semop(2), unless the program has set its own value. The default value is 16,384. The default value is the maximum value for this parameter.

Shared Memory

The following tunable parameters are associated with interprocess communication shared memory. These parameters are defined in the /etc/master.d/shm file. The order in which they are described follows the order in which they are defined in the output of the /usr/sbin/sysdef command.

SHMMAX — Specifies the maximum shared memory segment size. The default value is 131,072.

SHMMIN — Specifies the minimum shared memory segment size. The default value is 1.

SHMMNI — Specifies the maximum number of shared memory identifiers system wide. The default value is 100. Each entry contains 112 bytes.

SHMSEG — There is no maximum value enforced for this tunable (it was 15 in the past). The maximum number of shared memory segments that can be attached per process is dependent on the available unused space the process has. So even if a process has fewer than SHMSEG shared memory segments, it may not be able to attach another because of its limited space.

Quick Reference to Performance Management

- To collect kernel profiling data:

```
prfdc
```

- To collect kernel profiling data at the time of invocation only:

```
prfsnap
```

- To collect system activity data automatically:

```
sadc
```

- To collect system activity data on demand:

```
sar
```

The following summarizes `sar` options and their results:

□ `-a`	checks file access operations
□ `-b`	checks buffer activity
□ `-c`	checks system calls
□ `-d`	checks disk activity
□ `-g`	checks page-out and memory freeing activity
□ `-k`	checks kernel memory allocation
□ `-m`	checks interprocess communication
□ `-p`	checks page-in and fault rates
□ `-q`	checks queue activity
□ `-r`	checks unused memory
□ `-u`	checks CPU utilization
□ `-v`	checks system table status
□ `-w`	checks swapping and switching volume
□ `-y`	checks terminal activity

  - □ `-A` reports overall system performance

- To compress an entire file system:

```
/usr/sbin/dcopy fs1 fs2
```

- To compress a single directory:

```
mkdir /var/omail
mv /var/mail /var/omail
chmod 777 /var/omail
cd /var/omail
find . -print | cpio -plm ../mail
cd ..
rm -rf omail
```

- To determine the elapsed time, user time, and system time spent in execution of a command:

```
timex
```

- To enable, disable, or check the status of the sampling mechanism:

```
prfstat
```

- To find large, inefficient directories:

```
find / -type d -size +10 -print
```

NOTE `find` thinks in terms of 512-byte blocks.

- To find inactive files:

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

- To format the data collected by `prfdc` or `prfsnap`:

```
prfpr
```

- To initialize the kernel-recording mechanism:

```
prfld
```

- To move user directories:

```
cd /fsl
find userx usery -print | cpio -pdm /fs2
rm -rf /fsl/userx /fsl/usery
```

- To print out the number of free file blocks and inodes:

```
df
```

- To report disk access location and seek distance:

```
sadp
```

- To terminate a runaway process:

```
kill -9
```

- To summarize file system usage:

```
du
```

- To update the timestamp on the `/stand/system` file:

```
touch/stand/system
```

- To use a shell script to collect and store data in the binary file `/var/adm/sa/sa`***dd***:

```
sal
```

- To use a shell script to collect and store data in the ASCII file `/var/adm/sa/sar`***dd***:

```
sa2
```

11 Print Service

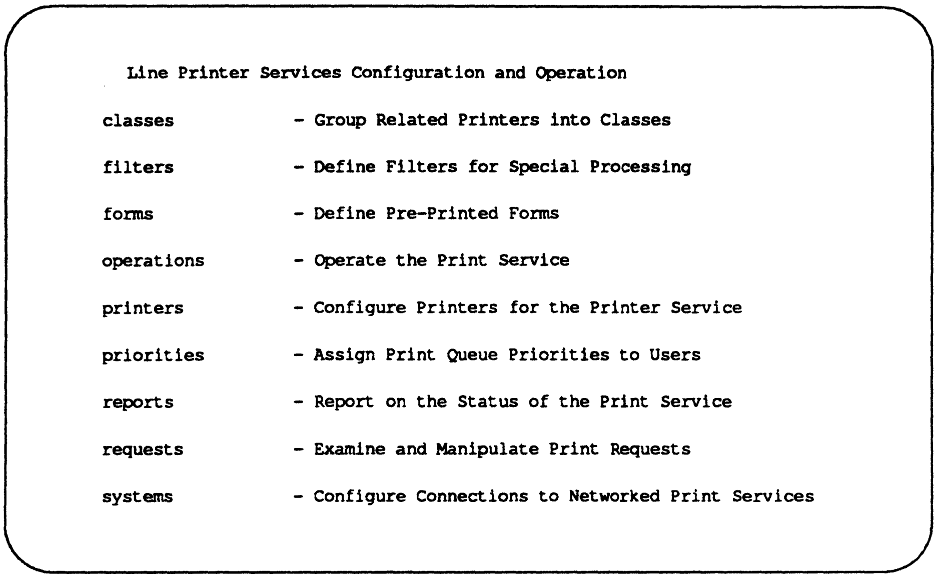
Making Printers Available 11-47

Troubleshooting 11-50

Introduction

This chapter describes the administrative tasks required for setting up and running an LP print service. The UNIX system offers a set of menus that help you do administrative tasks such as these. To invoke the "system administration menu" for the LP print service, type sysadm and select the printers item from the main menu. The following menu will appear on your screen:

Figure 11-1: Main Menu for Print Service



```
    Line Printer Services Configuration and Operation

classes          - Group Related Printers into Classes

filters          - Define Filters for Special Processing

forms            - Define Pre-Printed Forms

operations       - Operate the Print Service

printers         - Configure Printers for the Printer Service

priorities       - Assign Print Queue Priorities to Users

reports          - Report on the Status of the Print Service

requests         - Examine and Manipulate Print Requests

systems          - Configure Connections to Networked Print Services
```

If you prefer not to use the menus, you can perform the same administrative tasks by issuing commands directly to the shell. The following table shows which shell commands are available for doing the tasks listed on the menu.

Figure 11-2: Shell Commands for Print Service Administration

Figure 11-2: Shell Commands for Print Service Administration (continued)

Task Description	Menu Item	Shell Command
Group printers into classes	`classes`	`lpadmin`(1M)
Provide pre-processing software for files to be printed	`filters`	`lpfilter`(1M)
Define pre-printed forms for print requests	`forms`	`lpforms`(1M)

Control (turn on/off) queuing of requests; enable & disable printers; mount forms and fonts; start & stop print service; and report status of printers, classes, & forms	`operations`	`accept` & `reject`
		[see `accept`(1M)],
		enable & `disable`
		[see `enable`(1)],
		`lpadmin`(1M),
		`lpsched` & `lpshut`
		[see `lpsched`(1M)],
		`lpstat`(1)

Configure printers for print service	`printers`	`lpadmin`(1M)
Define levels of priority available to users requesting print jobs	`priorities`	`lpusers`(1M)
Identify active printers, print wheels & character sets, mounted forms, and pending requests	`reports`	`lpstat`(1)

Submit and cancel print requests	`requests`	`lp` & `cancel`
		[see `lp`(1)],
		`lpmove`
		[see `lpsched`(1M)]

Set up communication to remote print service	`systems`	`lpsystem`(1M)

The rest of this chapter describes the work required to set up and maintain print services on a UNIX system with the LP print service utilities. Details about the commands listed above are available in the manual pages for them. Error messages issued by the LP print service are listed in Appendix E.

The information presented in this chapter includes the following:

- A description of how the LP print service works
- References to documentation for installing the print service
- Troubleshooting guidelines
- Instructions for stopping and starting the print service manually
- Instructions for configuring a print service for the unique requirements of your users (such as the need for particular pre-printed forms and filters)
- A list of directories and files delivered as part of the print service
- Instructions for supporting PostScript printers
- Instructions for writing customized filters and interface programs

Overview

The LP print service, originally called the LP spooler, is a set of software utilities that allows you, minimally, to send a file to be printed while you continue with other work. (The term "spool" is an acronym for "simultaneous peripheral output on-line," and "LP" originally stood for Line Printer, but has come to include many other types of printing devices.) The print service has many optional enhancements, however; you can make your print service as simple or as sophisticated as you like.

Components of a Print Service

A print service consists of both hardware and software. You must have at least one computer and one printing device for an LP print service. Beyond that, you may have any number of computers and printing devices; there is no limit to the number of pieces of hardware you may include. The software consists of the LP print service utilities and any filters (programs that process the data in a file before it is printed) that you may provide. Users of your print service may be required to print all their files in the same format, or, if you make different types of printers and/or filters available with your service, they may choose from several formats. You may also offer your users a choice between plain paper and pre-printed forms (such as invoices or checks).

Functions Performed by the Print Service Software

Whether your print service is simple (such as a one-computer/one-printer configuration that prints every file in the same format on the same type of paper) or a sophisticated one (such as a computer network with multiple printers and a choice of printing formats and forms), the LP software helps you maintain it by performing several important functions:

- Scheduling the print requests of multiple users
- Scheduling the work of multiple printers
- Starting programs that interface with the printers
- Filtering users' files (if necessary) so they will be printed properly

- Keeping track of the status of jobs
- Keeping track of forms and print wheels currently mounted and alerting you to mount needed forms and print wheels
- Alerting you to printer problems

Suggestions for LP Print Service Administration

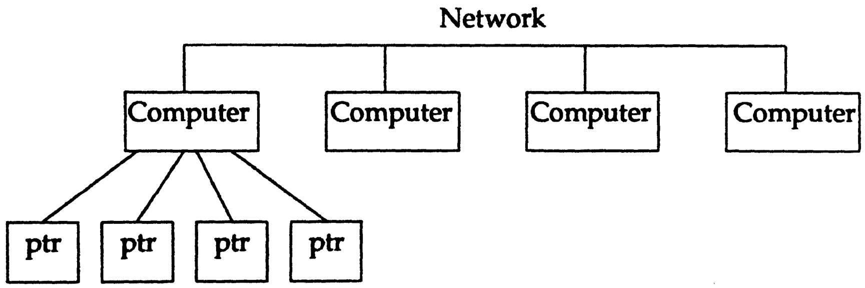
Here are some tips about how to organize your duties as the administrator of an LP print service.

Configuring Your Printer Sites

Where you decide to put your printers and how you decide to connect them to your computers depends on how those printers will be used. (See "Making Printers Accessible Through Your Computer" below for a more detailed description of connection methods.) There are three possible scenarios: (1) users may access printers attached to their own computer; (2) users may access printers attached to a server computer; and (3) users may access remote printers on a network to which their computer belongs.

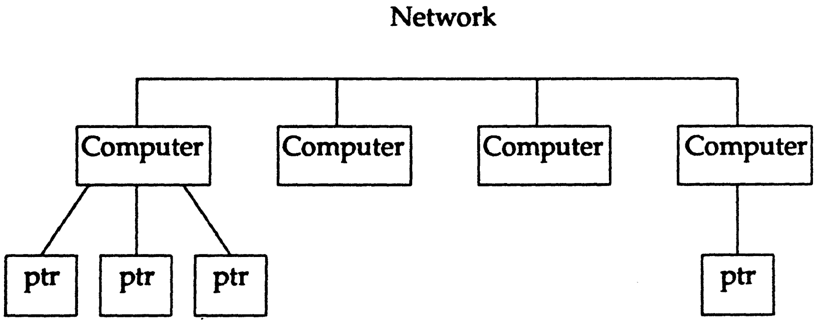
- You may want to connect a particular printer directly to the computer that is the home machine of the users who will use that printer most often. If you do, the type of connection you have will be referred to as a direct connection. An environment that includes more than one computer, each of which has a direct connection to a printer, is said to have a "distributed printing configuration."
- You may want to have all your printers in one physical location, such as a computer center. If so, you might connect them all to one computer. Users on other computers who want to use a printer may access it through a network linking their own computers to the computer serving the printers. An environment in which one computer serves several printers (which can be accessed only through a computer-to-computer network) is described as a "print server configuration." Figure 11-3 shows a sample print server configuration.

Figure 11-3: Print Server Configuration



- You may want to link most of your printers to a dedicated printer server computer, while allowing other printers to be connected to your machine. If so, you can arrange your computers and printers in a network configuration, as shown in Figure 11-4.

Figure 11-4: Network Configuration



Getting Started

Your first tasks are to physically connect your printers to your computers and to install the LP print service utilities from the floppy diskettes on which they were delivered. Once installed, these utilities will be available whenever your UNIX system is brought up. (If for some reason-such as to make a repair-you need to stop the print service without stopping the UNIX system, you will need to stop it manually, and then restart it manually. Instructions for manually stopping and starting the LP print service are provided later in this chapter.)

Even after you've installed the hardware and software, however, printers will not be available for use immediately. Before users can start submitting requests for print jobs, you must complete the following three steps:

- You must configure your printers; that is, you must name the printers and describe their characteristics to the print service. To configure your printers, run the `lpadmin` command. (See "Configuring Your Printers" below for details.)
- You must "register" (with the print service) any printer (or class of printers) that you have configured so that it accepts and queues print requests. To register printers (or classes) run the `accept` command. (See "Making Printers Available" below for details.)
- You must "enable" your printers (that is, you must make your printers active and available to users) by running the `enable` command. (See "Making Printers Available" below for details.)

Installing the LP Print Service

The LP package can be installed when you initially install your system software from tape by answering "all" to the question concerning which packages to install. However, if the LP package was not installed at that time, insert the utlities package tape into your tape drive and install the LP Print Service software by executing the following command:

```
pkgadd -d /dev/rmt/ctape1n lp
```

You will be prompted through the entire installation process.

Sharing Printers

If you have access to other systems via the Remote File Sharing Utilities (RFS), you may want to share printers with those systems by running the print service over RFS. You can do so by following these instructions.

On the server machine:

1. Set up the LP print service on the server machine as you would on any machine. (The server is the computer that does all the spooling.) Make sure that the printer works and that you are able to print text on it.
2. Share /var/spool/lp, /etc/lp, and /var/lp with all the *client(s)* that will be using this printer.
3. In /etc/rfs/auth.info/uid.rules, map the user ID (UID) of lp to itself, so the entry in uid.rules appears as follows: map lp.

On the client machines:

1. Do not run the scheduler on the client machines. On the client machines you need only lp, lpstat, and other LP print service commands.
2. Mount the resources that were shared by the server on the client's /var/spool/lp, /etc/lp, and /var/lp.

On client machines, the -c option of lp should be used for any user file not in a shared resource. This will force a copy of the file to be sent to the server machine. The LP print service cannot access local files that are not in a shared resource.

Controlling Access to the Print Service

Any user can send requests to the LP print service, check the status of requests, and cancel requests. In addition, you may want to give your users the ability to disable and enable a printer by authorizing them to use the `enable` and `disable` commands. The advantage in doing so is that a user with this authority can turn off a malfunctioning printer without calling the administrator. On the other hand, it may not be reasonable, in your printing environment, to allow regular users to disable printers.

During the installation process, the `pkgadd` command will "ask" you whether you want to authorize the users on your system to enable and disable the printer.

For further instructions on authorizing and restricting access to the `enable` and `disable` commands, see "Allowing Users to Enable and Disable a Printer" (in the "Making Printers Available" section) later in this chapter.

Documentation for Installing Printers

The following documentation can be of specific help in the area of printer operation:

- The installation manual that was delivered with your printer
- *Programmer's Guide: Character User Interface (FMLI and ETI)*
- `terminfo`(4) in the *Programmer's Reference Manual*

Configuring Your Printers

Before the LP print service can start accepting print requests, you will have to describe the characteristics of each printer you have. The following is a list of the attributes most commonly defined:

- printer name (mandatory)
- connection method (mandatory for local printers)
- system name (mandatory for access to remote printers and mandatory for allowing remote access to local printers)
- interface program
- printer type
- content types
- printer port characteristics
- character sets or print wheels
- alerting to mount a print wheel
- forms allowed
- printer fault alerting
- printer fault recovery
- restrictions on user access
- inclusion of banner page in output
- printer description
- default printing attributes
- printer class membership
- system default destination
- mounting a form or print wheel
- removing a printer or class

You need to specify very little of this information to add a new printer to the LP print service. The more information you provide, however, the better the printer will satisfy various users' needs.

The descriptions in the sections below will help you understand what this printer configuration information means and how it is used, so that you can decide how to configure your printers. In each section you will also be shown how to specify this information when adding a printer. While you can follow each of the sections in order and correctly configure a printer in several steps, you may want to wait until you've read all the sections before adding a printer, so that you can do it in fewer steps.

Printer Name

The printer name and the connection method (described next) are the only items you must specify to define a new local printer. To define a new remote printer, you must specify the printer name and the system name. The printer name is used to identify the printer, both by you (when you want to change the printer configuration or manage the printer), and by users who want to use the printer. The name may contain a maximum of fourteen alphanumeric characters and underscores.

You may choose any name you like, but it is good practice to choose a name that is meaningful to the users of the LP print service. For example, `laser` is a good name for a laser printer. If you have several laser printers you may name them `laser1`, `laser2`, and so on.

You don't have to try to fit a lot of descriptive information into the name; there is a better place for this information (see the "Printer Description" section below). You also don't have to make the name precisely identify the type of printer; users who need to use a particular type of printer can specify it by type rather than name (see the "Printer Type" section below).

You will use the printer name every time you want to refer to the printer: when adding other configuration information for the printer, when changing the configuration of the printer, when referring to the status of the printer, and when removing the printer. Thus the first thing you must do to add a printer is identify its name. You will do this as shown below; but don't do it yet because you'll also need to specify the connection method.

```
        lpadmin -p printer-name
```

There are no default names; you must name every printer.

Connection Method

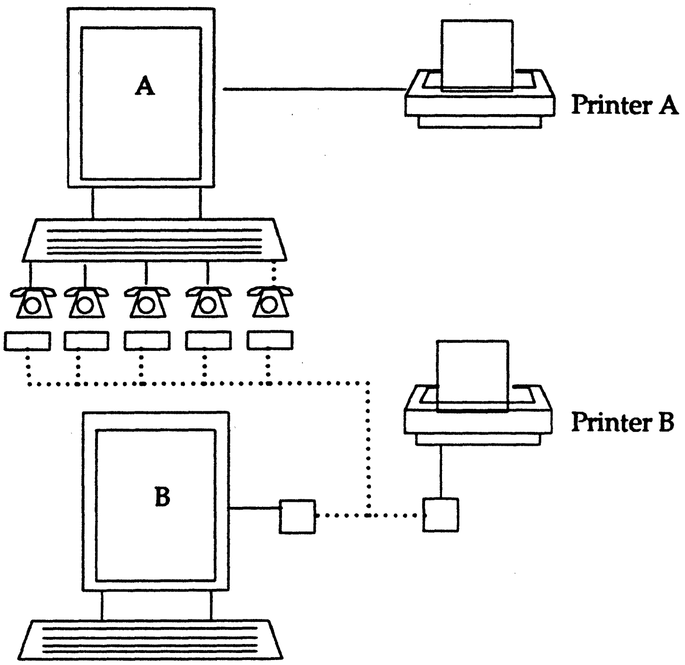
NOTE This section does not apply if you are making a remote printer accessible to users on your system.

The LP print service allows you to connect a printer to your computer in one of the following three ways:

- by connecting the printer directly to your computer
- by connecting the printer directly to a network to which your computer is attached
- by connecting the printer to a modem

Figure 11-5 shows these three types of connections.

Figure 11-5: Methods of Connecting a Printer to a Computer



Computer A accesses printer A through a direct connection, and accesses printer B using modems. Computer B accesses printer B over a local area network. Computer B may be able to access printer A through a remote connection-this is discussed in the next section.

The simplest connection method is by connecting a printer directly to your computer. You may, however, want to connect a printer to a network (so it can be shared with other computers or workstations), or to a modem. Whichever method you use, you must describe it to the LP print service after you've connected the hardware.

To define the connection method for a new printer for your print service, run the `lpadmin` command, specifying a connection method through either the `-v` option for a directly connected printer or the `-U` option for a printer directly connected to a network or a printer connected to a modem.

Direct Connections

The simplest and most common method by which printers are connected to a computer is direct connection. If you use this method, you generally need to specify only two items on the command line when you make the connection: the name of the printer and the name of the connecting port. To connect a printer directly to your computer enter the following command:

```
        lpadmin -p printer-name -v pathname
```

where *pathname* is the name of the special device file representing the printer port. The following are examples of typical names of special device files.

```
        /dev/contty
        /dev/term/11
        /dev/term/12
        /dev/term/13
        /dev/term/14
        /dev/term/15
```

(For details about using these files, see "Printer Port Characteristics" later in this chapter.)

Using a Printer As a Login Terminal

Some directly connected printers can also be used as terminals for login sessions. If you want to use a printer as a terminal, you must arrange for the LP print service to handle it as such. To do so, use the `-l` option to the `lpadmin` command, as follows:

```
        lpadmin -p printer-name -v pathname -l
```

As before, *pathname* is the name of the special file representing the printer port. If the `-l` option is specified, the printer will be disabled automatically whenever the LP print service is started, and therefore will have to be manually enabled before it can be used for printing. For instructions on manually enabling a printer, see "Enabling and Disabling a Printer" (under "Making Printers Available") later in this chapter.

Connections to Networks and Modems

Why would you want to use a printer that is not directly connected to your computer?

- The environment where a printer is located is so far from the computer that a direct connection is not possible or practical. For example, you might have one printer in use with a single terminal at a branch office located a few miles from your main site.
- You may want to share a printer with computers that are not on a common network.

The LP print service establishes a connection to the printer as necessary to print requests; at the end of each request the connection is dropped, making the printer available to the next machine that calls it. Thus the printer gets shared by the users of all the computers, more or less equally.

There are two methods for connecting printers that are not directly connected to your system: attached directly to a network and through a dial-up modem. The LP print service uses the Basic Networking Utilities (BNU) to handle both methods.

When a modem connection is used, the printer must be connected to a dialed modem, and the dial-out modem must be connected to the computer. Whether printers are connected to a modem or directly to a network, the connection must be described to the Basic Networking Utilities. For instructions on describing either type of connection, see the "Network Services" chapter.

To make a printer connected in one of these ways available to your users, enter the following command:

```
lpadmin -p printer-name -U dial-info
```

Dial-info is either the telephone number to be dialed to reach the printer's modem, or the system name entered in the Basic Networking `Systems` file for the printer.

NOTE

The `-U` option provides a way to link a single printer to your print service. It does not allow you to connect with a print service on another system.

A note on printers connected to a modem or directly to a network: if the printer or port is busy, the LP print service will automatically retry later. This retry rate is 10 minutes if the printer is busy, and 20 minutes if the port is busy. These rates are not adjustable. However, you can force an immediate retry by issuing an `enable` command for the printer. If the port or printer is likely to be busy for an extended period, you should issue a `disable` command.

The `lpstat -p` command reports the reason for a failed dial attempt. Also, if you are alerted to a dialing fault (see the "Fault Alerting" section below), the alert message will give the reason for the fault. These messages are identical to the error messages produced by the Basic Networking Utilities (BNU) for similar problems. See the section titled "BNU STATUS Error Messages" in Appendix E ("Error Messages") for an explanation of the reasons for failure.

In summary, to add printers to your system, run the `lpadmin` command, specifying a connection method through one of two options: the `-v` option for a directly connected printer, or the `-U` option for a networked printer.

System Name

NOTE **This section does not apply if you are making only a local printer accessible to users on your system.**

A remote printer is one that is connected to a system other than your local system that you can access only via that remote system. Why might you want to use a remote printer? You might have several computers connected via a high-speed network. You can set up one computer for all the printers.

There are some exceptional cases, however. For example, if only one of the printers has a particular typesetter needed for some print jobs, then users from many systems will want to access it from time to time.

Alternatively, a large community of users on a local area network may want to pool all printers on a single system, where they can share them. When this is done, the system supporting the printers becomes a printer server.

To make accessible a printer that is remote, the name of the system on which the printer resides must be registered with the print service. If the remote printer resides on a System V Release 4 machine, enter

```
lpsystem system-name
```

If the remote printer resides on a BSD machine, enter

```
lpsystem -p bsd system-name
```

NOTE For details about the options available with this command, see `lpsystem(1M)` in the *System Administrator's Reference Manual*.

In either case, after entering the `lpsystem` command, enter the `lpadmin` command, as follows:

```
lpadmin -p printer -s system-name
```

where *printer* is the name by which your users identify the remote printer and *system-name* is the name of the system on which that printer resides. You can usually use the same name used for that printer by the remote system. If the name used by the remote system is the same name used for an existing printer or class on your system, you must use a different name. To assign a different name to a remote printer, enter the following:

```
lpadmin -p local-name -s system-name!remote-name
```

For example, imagine you want your users to have access to a printer called `psjet2` that resides on a remote system called `newyork`. Because you already have a printer called `psjet2` on your own system, you want to give the remote printer a new name on your system: `psjet3`. Request the new name by entering the following:

```
lpadmin -p psjet3 -s newyork!psjet2
```

Before you add a remote printer to your system, be sure communications between your system and the network have been set up properly, and verified. See the *Network User's and Administrator's Guide* for details.

Allowing Remote Users to Access Local Printers

Making the printers on your local system accessible to users on remote systems is a two-step process: you must configure the port monitor on the local system and you must register the system with the LP print service. This section provides instructions for these tasks.

Configuring the Local Port Monitor

If the remote system will need access to printers connected directly to your computer, then you need to configure the local port monitor for the network you share to accept service requests and to notify the LP print service of such requests. For System V machines calling your machines, issue the following command:

```
pmadm -a -p netname -s lp -i root -v `nlsadmin -V`\
-m `nlsadmin -o /var/spool/lp/fifos/listenS5`
```

where *netname* is the name of a network such as `tcp`.

If you expect users on BSD machines to send print requests to your machine, then you need to configure your local port monitor. The output of this command will be a hexadecimal number.

```
pmadm -a -p tcp -s lpd -i root -V `nlsadmin -V`\
-m `nlsadmin -o /var/spool/lp/fifos/listenBSD -A'\xaddress'`
```

Before issuing this command for a BSD machine, however, you need to know its TCP-IP *address*. To get this *address*, run the `-A` option with the `lpsystem` command, as follows:

```
lpsystem -A
```

Adding a System Entry

If you want your system to accept jobs from a remote system (and vice-versa), the print service must know about that system. The `lpsystem` command allows you to register remote systems with the local print service. Run the command as follows:

```
lpsystem system-name
```

where *system-name* is the name of the remote system.

Interface Program

NOTE This section does not apply if you are making a remote printer accessible to users on your system.

This is the program the LP print service uses to manage the printer each time a file is printed. It has several tasks:

- to initialize the printer port (the connection between the computer and the printer)
- to initialize the printer (restore it to a normal state in case a previously printed file has left it in an unusual state) and set the character pitch, line pitch, page size, and character set requested by the user
- to print a banner page
- to run a filter that prepares the file for printing
- to manage printer faults

If you do not choose an interface program, the standard one provided with the LP print service will be used. This should be sufficient for most of your printing needs. If you prefer, however, you can change it to suit your needs, or completely rewrite your own interface program, and then specify it when you add a new printer. See "Customizing the Print Service" later in this chapter for details on how to customize an interface program.

If you are using the standard interface program, you needn't specify it when adding a printer. If, however, you will be using a different interface program on a local printer, you can refer to it either by specifying its full pathname or by referring to another printer using the same interface program.

To identify a customized interface program by name, specify the printer name and the pathname of the interface program, as follows:

```
lpadmin -p printer-name -i pathname
```

To use a customized interface program of another printer, specify the printer names as follows:

```
lpadmin -p printer-name1 -e printer-name2
```

Printer-name$_1$ is the name of the printer you are adding; *printer-name*$_2$ is the name of an existing printer that is using the customized interface program.

Printer Type

A printer type is the generic name for a printer. Typically it is derived from the manufacturer's name, such as `572` for the AT&T 572 Dot Matrix Printer. When you set up your system you can enhance its ability to serve your users by classifying, on the basis of type, the printers available through the print service. Assigning a "type" for each printer is also important because the LP software extracts information about printers from the `terminfo` database on the basis of type. This information includes the list of the printer's capabilities that is used to check the configuration information you supply to the print service. (By checking information provided by you against the capabilities of the printer, the print service can catch any inappropriate information you may have supplied.) The `terminfo` database also specifies the control data needed to initialize a particular printer before printing a file.

You can assign several types to a printer, if your printer is capable of emulating more than one kind of printer. The AT&T 593 Laser Printer, for instance, can emulate an IBM Proprinter XL, an Epson FX86c, and an HP LaserJetII. The `terminfo` database names these types `593ibm`, `583eps`, and `583hp`, respectively. If you specify more than one printer type, the LP print service will use one of them as appropriate for each print request.

NOTE If you specify more than one printer type, you must specify `simple` as the content type.

While you are not required to specify a printer type, we recommend that you do so; when a printer type is specified, better print services can be provided.

To specify a printer type, use the following command line:

```
lpadmin -p printer-name -T printer-type-list
```

If you give a list of printer types, separate the names with commas. If you do not define a printer type, the default `unknown` will be used.

Content Types

While the printer type tells the LP print service what types of printers are being added, the content types tell the LP print service what types of files can be printed directly on each printer. Most printers can print files of two types: the same type as the printer type (if the printer type is defined) and the type `simple`, (meaning an ASCII file) which is the default content type for all printers.

Some printers, though, can accept (and print properly) several different types of files. When adding this kind of printer, specify the names of the content types the new printer accepts by adding these names to the list. (By default, the list contains only one type: `simple`.) If you're adding a remote printer, list the content types that have been established for it by the administrator of the system on which it resides.

To specify the list of content types, enter the following command:

```
lpadmin -p printer-name -I content-type-list
```

The *content-type-list* is a list of names separated by commas or spaces. If you use spaces to separate the names, enclose the entire list (but not the `-I`) in quotes.

Content type names may look a lot like printer type names, but you are free to choose names that are meaningful to you and the people using the printer. (The names `simple` and `any` are recognized as having particular meanings by the LP print service; be sure to use them consistently. The name `terminfo` is also reserved, as a reference to *all* types of printers.) The names must contain no more than fourteen characters and may include only letters, digits, and underscores. The following table lists and describes some accepted content types.

Types	Description
troff	Device independent output from troff
otroff	CAT typesetter instructions generated by BSD or pre-System V troff (old troff)
tex	DVI format files
plot	Plotting instructions for Tektronix displays and devices
raster	Raster bitmap format for Varian raster devices
cif	Output of BSD cifpbt
fortran	ASA carriage control format
postscript	PostScript language
simple	ASCII file

When a file is submitted to the LP print service for printing with the printer specified by the `-d any` option of the `lp` command, the print service searches for a printer capable of handling the job. The print service can identify an appropriate printer through either the content type name or the printer type name. Therefore, you may specify either name (or no name) when submitting a file for printing. If the same content type is printable by several different types of printers, you should use the same content type names when you add those printers. This makes it easier for the people using the printers, because they can use the same name to identify the type of file they want printed regardless of the printing destination.

Most manufacturers produce printers that accept simple ASCII files. While these printers are different types (and thus have different initialization control sequences), they may all be capable of handling the same type of file, which we call `simple`. As another example, several manufacturers may produce printers that accept ANSI X3.64 defined escape sequences. However, the printers may not support all the ANSI capabilities; they may support different sets of capabilities. You may want to differentiate them by assigning different content type names for these printers.

However, while it may be desirable (in situations such as these) to list content types for each printer, it is not always necessary to do so. If you don't, the printer type will be used as the name of the content type the printer can handle. If you have not specified a printer type, the LP print service will assume the printer can print only files of content type `simple`. This may be sufficient if you require users to specify the proper printer explicitly and if files are properly prepared for the printer before being submitted for printing.

The Default Content Type: `simple`

Files of content type `simple` are assumed to contain only two types of characters, printable ASCII characters and the following control characters:

backspace — moves the carriage back one space, except at the beginning of a line

tab — moves the carriage to the next tab stop; by default, stops are spaced every 8 columns on most printers

linefeed — moves the carriage to the beginning of the next line (may require special port settings for some printers—see "Printer Port Characteristics" below)

form feed — moves the carriage to the beginning of the next page

carriage return — moves the carriage to the beginning of the same line (may fail on some printers)

The word "carriage" may be archaic for modern laser printers, but these printers do actions similar to those done by a carriage. If a printer can handle several types of files, including `simple`, you must include `simple` in the content type list; the type `simple` is not automatically added to any list you give. If you *don't* want a printer to accept files of type `simple`, give a blank *content-type-list*, as follows:

```
        lpadmin -p printer-name -I ""
```

Printer Port Characteristics

NOTE This section does not apply if you are making a remote printer available to users on your system.

For Any Computer

Printers connected directly to computers and those connected over some networks require that the printer port characteristics be set by the interface program. These characteristics define the low level communications with the printer. Included are the baud rate; use of XON/XOFF flow control; 7, 8, or other bits per byte; type of parity; and output post-processing. The standard interface program will use the stty command to initialize the printer port, minimally setting the baud rate and a few other default characteristics.

The default characteristics applied by the standard interface program are listed below.

Default	Meaning
9600	9600 baud rate
cs8	8-bit bytes
-cstopb	1 stop bit per byte
-parenb	no parity generation
ixon	enable XON/XOFF flow control
-ixany	allow only XON to restart output
opost	post-process data stream as listed below:
-olcuc	don't map lower-case to upper-case
onlcr	map linefeed into carriage-return/linefeed
-ocrnl	don't map carriage-return into linefeed
-onocr	output carriage-returns even at column 0
nl0	no delay after linefeeds
cr0	no delay after carriage-returns
tab0	no delay after tabs
bs0	no delay after backspaces
vt0	no delay after vertical tabs
ff0	no delay after form-feeds

You may find that the default characteristics are sufficient for your printers. However, printers vary enough that you are likely to find that you have to set different characteristics. See the description of the stty command in the *User's Reference Manual* to find the complete list of characteristics.

If you have a printer that requires printer port characteristics other than those handled by the stty program, you will have to customize the interface program. See the section "Customizing the Print Service" for help.

When you add a new printer, you may specify an additional list of port characteristics. The list you give will be applied after the default list, so that you do not need to include in your list items that you don't want to change. Specify the additional list as follows:

```
        lpadmin -p printer-name -o "stty='stty-option-list' "
```

Note that both the double quotes and single quotes are needed if you give more than one item in the *stty-option-list*.

As one example, suppose your printer is to be used for printing graphical data, where linefeed characters should be output alone, without an added carriage-return. You would enter the following command:

```
        lpadmin -p printer-name -o "stty=-onlcr"
```

Note that the single quotes are omitted because there's only one item in the list.

As another example, suppose your printer requires odd parity for data sent to it. You would enter the following command:

```
        lpadmin -p printer-name -o "stty='parenb parodd cs7' "
```

Character Sets or Print Wheels

NOTE Although your users may use character sets or print wheels that have been mounted on a remote printer (by the administrator of the remote system on which that printer resides), you cannot mount a character set or a print wheel on a remote printer.

Printers differ in the way they can print in different font styles. Some have changeable print wheels, some have changeable font cartridges, others have preprogrammed, selectable character sets.

When adding a printer, you may specify what print wheels, font cartridges, or character sets are available with the printer.

NOTE If you're adding a remote printer and you want your users to be able to use character sets or print wheels that have been mounted by the administrator of the remote system, you must list those character sets and print wheels, just as you would list the character sets and print wheels on a local printer. (See instructions below.)

Only one of these is assumed to apply to each printer. From the point of view of the LP print service, however, print wheels and changeable font cartridges are the same because they require you to intervene and mount a new print wheel or font cartridge. Thus, for ease of discussion, only print wheels and character sets will be mentioned.

When you list the print wheels or character sets available, you will be assigning names to them. These names are for your convenience and the convenience of the users. Because different printers may have similar print wheels or character sets, you should use common names for all printers. This allows a user to submit a file for printing and ask for a particular font style, without regard for which printer will be used or whether a print wheel or selectable character set is used.

If the printer has mountable print wheels, you need only list their names. If the printer has selectable character sets, you need to list their names and map each one into a name or number that uniquely identifies it in the `terminfo` database. Use the following command to determine the names of the character sets listed in the `terminfo` database.

```
        tput -T printer-type csnm 0
```

Printer-type is the name of the printer type in question. The name of the 0th character set (the character set obtained by default after the printer is initialized) will be printed. Repeat the command, using 1, 2, 3, and so on in place of the 0, to see the names of the other character sets. In general, the `terminfo` names should closely match the names used in the user documentation for the printer. However, because not all manufacturers use the same names, the `terminfo` names may differ from one printer type to the next.

NOTE

For the LP print service to be able to find the names in the `terminfo` database, you must specify a printer type. See the section "Printer Type" above.

To specify a list of print wheel names when adding a printer, enter the following command.

```
        lpadmin -p printer-name -S print-wheel-list
```

The *print-wheel-list* is a comma or space separated list of names. If you use spaces to separate the names, enclose the entire list (but not the `-S`) in quotes.

To specify a list of character set names and to map them into `terminfo` names or numbers, enter the following command:

```
        lpadmin -p printer-name -S character-set-list
```

The *character-set-list* is also a comma or space separated list; however, each item in the list looks like one of the following:

> `cs`*N*=*character-set-name*
> *character-set-name*$_1$=*character-set-name*$_2$

The *N* in the first case is a number from 0 to 63 that identifies the number of the character set in the `terminfo` database. The *character-set-name*$_1$ in the second case identifies the character set by its `terminfo` name. In either case the name to the right of the equal sign (=) is the name you may use as an alias of the character set.

NOTE You do not have to provide a list of aliases for the character sets if the `terminfo` names are adequate. You may refer to a character set by number, by `terminfo` name, or by your alias.

For example, suppose your printer has two selectable character sets (sets #1 and #2) in addition to the standard character set (set #0). The printer type is `5310`. You enter the following commands to determine the names of the selectable character sets.

```
tput -T 5310 csnm 1
english
tput -T 5310 csnm 2
finnish
```

The words `english` and `finnish`, the output of the commands, are the names of the selectable character sets. You feel that the name `finnish` is adequate for referring to character set #2,but better names are needed for the standard set (set #0) and set #1. You enter the following command to define synonyms.

```
lpadmin -p printer-name -S "cs0=american,\
english=british"
```

The following three commands will then produce identical results.

```
lp -S cs1 -d any ...

lp -S english -d any ...

lp -S british -d any ...
```

If you do not list the print wheels or character sets that can be used with a printer, then the LP print service will assume the following: a printer that takes print wheels has only a single, fixed print wheel, and users may not ask for a special print wheel when using the printer; and a printer that has selectable character sets can take any `cs`*N* name or `terminfo` name known for the printer.

Alerting to Mount a Print Wheel

NOTE This section does not apply if you are making a remote printer available to users on your system.

If you have printers that can take changeable print wheels, and have listed the print wheels allowed on each, then users will be able to submit a print request to use a particular print wheel. Until it is mounted though (see "Mounting a Form or Print Wheel" in this section), a request for a print wheel will stay queued and will not be printed. You could periodically monitor the number of print requests pending for a particular print wheel, but the LP print service provides an easier way: You can ask to be alerted when the number of requests waiting for a print wheel has exceeded a specified threshold.

You can choose one of several ways to receive an alert.

- You can receive an alert via electronic mail. See the description of the `mailx` command in the *User's Reference Manual* for a description of mail on the UNIX system.
- You can receive an alert written to any terminal on which you are logged in. See the description of the `write` command in the *User's Reference Manual*.
- You can receive an alert through a program of your choice.
- You can receive no alerts.

NOTE If you elect to receive no alerts, you are responsible for checking to see whether any print requests haven't printed because the proper print wheel isn't mounted.

In addition to the method of alerting, you can also set the number of requests that must be queued before you are alerted, and you can arrange for repeated alerts every few minutes until the print wheel is mounted. You can choose the rate of repeated alerts, or you can opt to receive only one alert for each print wheel.

To arrange for alerting to the need to mount a print wheel, enter one of the following commands:

```
lpadmin -S print-wheel-name -A mail -Q requests -W minutes
lpadmin -S print-wheel-name -A write -Q requests -W minutes
lpadmin -S print-wheel-name -A 'command' -Q requests -W minutes
```

The first two commands direct the LP print service to send you a mail message or write the message directly to your terminal, respectively, for each alert. The third command directs the LP print service to run the *command* for each alert. The shell environment currently in effect when you enter the third command is saved and restored for the execution of *command*; this includes the environment variables, user and group IDs, and current directory. The argument *requests* is the number of requests that need to be waiting for the print wheel before the alert is triggered, and the argument *minutes* is the number of minutes between repeated alerts.

If you do not want the print service to issue an alert when a print wheel needs to be mounted, enter the following:

```
lpadmin -S print-wheel-name -A none
```

NOTE If you want mail sent or a message written to another user for an alert, use the third command with the option `-A 'mail` *user-name*' or `-A 'write` *user-name*'. If you do not specify a *login-name*, the mail or message will be sent to your current ID. This may not be your login ID, if you have used the `su` command to change IDs.

When you start receiving repeated alerts, you can direct the LP print service to stop sending you alerts (for the current case only), by executing the following command.

```
lpadmin -S print-wheel-name -A quiet
```

Once the print wheel has been mounted and unmounted again, alerts will start again if too many requests are waiting. Alerts will also start again if the number of requests waiting falls below the `-Q` threshold and then rises up to the `-Q` threshold again, as when waiting requests are canceled, or if the type of alerting is changed.

If *print-wheel-name* is `all` in any of the commands above, the alerting condition will apply to all print wheels for which an alert has already been defined.

If you don't define an alert method for a print wheel, you will not receive an alert to mount it. If you do define a method without the `-W` option, you will be alerted once for each occasion.

Forms Allowed

NOTE For information about how to define, mount, and set up alerting to mount a form, see "Providing Forms" later in this chapter.

You can control the use of preprinted forms on any printer, including remote printers. (Although you cannot mount forms on remote printers, your users may use forms on remote printers.) You may want to do this, for instance, if a printer is not well suited for printing on a particular form because of low print quality, or if the form cannot be lined up properly in the printer.

The LP print service will use a list of forms allowed or denied on a printer to warn you against mounting a form that is not allowed on the printer. However, you have the final word on this; the LP print service will not reject the mounting. The LP print service will, however, reject a user's request to print a file on a printer using a form not allowed on that printer. If, however, the printer is a local printer and the requested form is already mounted, the request will be printed on that form.

If you try to allow a form for a printer, but the printer does not have sufficient capabilities to handle the form, the command will be rejected.

The method of listing the forms allowed or denied for a printer is similar to the method used to list users allowed or denied access to the `cron` and `at` facilities. (See the description of the `crontab` command in the *User's Reference Manual*.) Briefly, the rules are as follows:

- An allow list is a list of forms that are allowed to be used on the printer. A deny list is a list of forms that are not allowed to be used on the printer.
- If the allow list is not empty, only the forms listed are allowed; the deny list is ignored. If the allow list is empty, the forms listed in the deny list are not allowed. If both lists are empty, there are no restrictions on which forms may be used other than those restrictions that apply to a printer of a particular type, such as a PostScript printer for which a license is required.
- Specifying `all` in the allow list allows all forms; specifying `all` in the deny list denies all forms.

You can add names of forms to either list using one of the following commands:

```
lpadmin -p printer-name -f allow:form-list
lpadmin -p printer-name -f deny:form-list
```

The *form-list* is a comma or space separated list of names of forms. If you use spaces to separate names, enclose the entire list (including the `allow:` or `deny:` but not the `-f`) in quotes.

The first command shown above adds names to the allow list and removes them from the deny list. The second command adds names to the deny list and removes them from the allow list. To make the use of all forms permissible, specify `allow:all`; to deny permission for all forms, specify `deny:all`.

If you do not use this option, the LP print service will consider that the printer denies the use of all forms. It will, however, allow you to mount any form, thereby making it implicitly available to use. (See "Mounting a Form or Print Wheel" later in this section for more information.)

Printer Fault Alerting

NOTE This section does not apply if you are making a remote printer accessible to users on your system.

The LP print service provides a framework for detecting printer faults and alerting you to them. Faults can range from simple problems, such as running out of paper or ribbon, or needing to replace the toner, to more serious faults, such as a local power failure or a printer failure. The range of fault indicators is also broad, ranging from dropping carrier (the signal that indicates that the printer is on line), to sending an XOFF, to sending a message. Only two classes of printer fault indicators are recognized by the LP print service itself: a drop in carrier and an XOFF not followed in reasonable time by an XON. However, you can add filters that recognize any other printer fault indicators, and rely on the LP print service to alert you to a fault when the filter detects it.

NOTE For a description of how to add a filter, see the "Filter Management" section in this chapter. For a description of how a filter should let the LP print service know a fault has occurred, see the "Customizing the Print Service" section in this chapter.

You can choose one of several ways to receive an alert to a printer fault:

- You can receive an alert via electronic mail. See the description of the `mailx` command in the *User's Reference Manual* for a description of mail on the UNIX system.
- You can receive an alert written to any terminal on which you are logged in. See the description of the `write` command in the *User's Reference Manual*.
- You can receive an alert through a program of your choice.
- You can receive no alerts.

NOTE If you elect to receive no alerts, you will need a way of finding out about the faults and fixing them; the LP print service will not continue to use a printer that has a fault.

In addition to the method of alerting, you can also arrange for repeated alerts every few minutes until the fault is cleared. You can choose the rate of repeated alerts, or you can choose to receive only one alert per fault.

NOTE Without a filter that provides better fault detection, the LP print service cannot automatically determine when a fault has been cleared except by trying to print another file. It will assume that a fault has been cleared when it is successfully able to print a file. Until that time, if you have asked for only one alert per fault, you will not receive another alert. If, after you have fixed a fault, but before the LP print service has tried printing another file, the printer faults again, or if your attempt to fix the fault fails, you will not be notified. Receiving repeated alerts per fault, or requiring manual re-enabling of the printer (see the "Fault Recovery" section below), will overcome this problem.

To arrange for alerting to a printer fault, enter one of the following commands:

```
lpadmin -p printer-name -A mail -W minutes
lpadmin -p printer-name -A write -W minutes
lpadmin -p printer-name -A 'command' -W minutes
```

The first two commands direct the LP print service to send you a mail message or write the message directly to your terminal, respectively, for each alert. The third command directs the LP print service to run the *command* for each alert. The shell environment currently in effect when you enter the third command is saved and restored for the execution of *command*. The environment includes environment variables, user and group IDs, and current directory. The *minutes* argument is the number of minutes between repeated alerts.

If you do not want the LP print service to issue an alert when a fault occurs, enter the following:

```
lpadmin -p printer-name -A none
```

NOTE

If you want mail sent or a message written to another user when a printer fault occurs, use the third command with the option `-A 'mail` *login-name*`'` or `-A 'write` *login-name*`'`. If you do not specify a *login-name*, the mail or message will be sent to your current ID. This may not be your login ID, if you have used the `su` command to change IDs.

Once a fault occurs and you start receiving repeated alerts, you can direct the LP print service to stop sending you alerts (for the current fault only), by executing the following command:

```
lpadmin -p printer-name -A quiet
```

NOTE

Use the alert type `quiet` only to terminate an active alert; do not specify `quiet` as the alert type for a new printer.

If the *printer-name* is `all` in any of the commands above, the alerting condition will apply to all printers.

If you don't define an alert method, you will receive mail once for each printer fault. If you define a method without the `-W` option, you will be alerted once for each fault.

Printer Fault Recovery

NOTE This section does not apply if you are making a remote printer accessible to users on your system.

When a printer fault has been fixed and the printer is ready for printing again, the LP print service will recover in one of three ways:

- it will continue printing at the top of the page where printing stopped
- it will restart printing at the beginning of the print request that was active when the fault occurred
- it will wait for you to tell the LP print service to re-enable the printer

NOTE The ability to continue printing at the top of the page where printing stopped requires the use of a filter that can wait for a printer fault to be cleared before resuming properly. Such a filter probably has to have detailed knowledge of the control sequences used by the printer so it can keep track of page boundaries and know where in a file printing stopped. The default filter used by the LP print service cannot do this. If a proper filter is not being used, you will be notified in an alert if recovery cannot proceed as you want.

To specify the way the LP print service should recover after a fault has been cleared, enter one of the following commands:

```
lpadmin -p printer-name -F continue
lpadmin -p printer-name -F beginning
lpadmin -p printer-name -F wait
```

These commands direct the LP print service, respectively, to continue at the top of the page, restart from the beginning, or wait for you to enter an `enable` command to re-enable the printer (see the "Enabling and Disabling Printer" section in this chapter for information on the `enable` command).

If you do not specify how the LP print service is to resume after a printer fault, it will try to continue at the top of the page where printing stopped, or, failing that, at the beginning of the print request.

If the recovery is `continue`, but the interface program does not stay running so that it can detect when the printer fault has been cleared, printing will be attempted every few minutes until it succeeds. You can force the LP print service to retry immediately by issuing an `enable` command.

User Access Restrictions

You can control which users are allowed to use a particular printer on your system. For instance, if you're designating one printer to handle sensitive information, you don't want all users to be able to use the printer. Another time you might want to do this is when you're authorizing the use of a high quality printer which produces expensive output.

The LP print service will use a list of users allowed or denied access to a printer. The LP print service will reject a user's request to print a file on a printer he or she is not allowed to use.

NOTE

If your users have access to remote printers, or if users on other systems have access to printers on your system, make sure that the allow and deny lists for those printers on your computer match the allow and deny lists on the remote system where the remote printers reside. If these two sets of lists don't match, your users may receive conflicting messages (some accepting jobs, and others refusing jobs) when submitting requests to remote printers.

The method of listing the users allowed or denied access to a printer is similar to the method used to list users allowed or denied access to the `cron` and `at` facilities, and the method described above in the "Forms Allowed" section. Briefly, the rules are as follows:

- An allow list is a list of users allowed to use the printer. A deny list is a list of users denied access to the printer.
- If the allow list is not empty, only the users listed are allowed; the deny list is ignored. If the allow list is empty, users listed in the deny list are not allowed. If both lists are empty, there are no restrictions on who may use the printer.
- Specifying `all` in the allow list allows everybody access to the printer; specifying `all` in the deny list denies access to everybody except the user `lp` and the super-user (`root`).

You can add names of users to either list using one of the following commands:

```
lpadmin -p printer-name -u allow:user-list
lpadmin -p printer-name -u deny:user-list
```

The *user-list* is a comma or space separated list of names of users. If you use spaces to separate the names, enclose the entire list (including the `allow:` or `deny:` but not the `-u`) in quotes. Each item in the *user-list* may take any of the following forms:

user	*User* on any system
`all`	All users on all systems
local-system`!`*user*	User on *local-system* only
`!`*user*	User on local system only
`all!`*user*	*User* on any system
`all!all`	All users on all systems
system`!all`	All users on *system*
`!all`	All users on local system

The first command shown above adds the names to the allow list and removes them from the deny list. The second command adds the names to the deny list and removes them from the allow list.

If you do not use this option, the LP print service will assume that everybody may use the printer.

Inclusion of Banner Page in Output

Most users want to have the output of each print request preceded by a banner page. A banner page shows who requested the printing, the request ID for it, and when the output was printed. It also allows for an optional title that the requester can use to better identify a printout. Finally, the banner page greatly eases the task of separating a sequence of print requests so that each may be given to the correct user.

Sometimes a user needs to avoid printing a banner page. The likely occasions are when the printer has forms mounted that should not be wasted, such as payroll checks or accounts payable checks. Printing a banner page under such circumstances may cause problems.

Enter the following command to allow users to request no banner page:

```
lpadmin -p printer-name -o nobanner
```

If you later change your mind, you can reverse this choice by entering the following command:

```
lpadmin -p printer-name -o banner
```

If you do not allow a user to skip the banner page, the LP print service will reject all attempts to avoid a banner page when printing on the printer. This is the default action.

Printer Description

An easy way to give users of the LP print service helpful information about a printer is by adding a description of it. This description can contain any message you'd like, including the number of the room where the printer is found, the name of the person to call with printer problems, and so forth.

Users can see the message when they use the `lpstat -D -p` *printer-name* command.

To add a description of a printer, enter the following command.

```
lpadmin -p printer-name -D 'text'
```

The *text* is the message. You'll need to include the quotes if the message contains blanks or other characters that the shell might interpret if the quotes are left out.

Default Printing Attributes

The attributes of a printing job include the page size, print spacing (character pitch and line pitch), and stty options for that job. If a user requests a job to be printed on a particular form, the printing attributes defined for that form will be used for that job. When, however, a user submits a print request without requesting a form, the print service uses one of several sets of default attributes.

- If the user has specified attributes to be used, those attributes will take precedence.
- If the user has not specified attributes, but the administrator has executed the lpadmin -o command, the default attributes for that command will take precedence.
- If neither of the above, the attributes defined in the terminfo database for the printer being used will take precedence.

The LP print service lets you override the defaults for each printer. Doing so can make it easier to submit print requests by allowing you to designate different printers as having different default page sizes or print spacing. A user can then simply route a file to the appropriate printer to get a desired style of output. For example, you can have one printer dedicated to printing wide (132-column) output, another printing normal (80-column by 66-line) output, and yet another printing letter quality (12 characters per inch, 8 lines per inch) output.

You can independently specify four default settings, page width, page length, character pitch, and line pitch. You can scale these to fit your needs: The first two can be given in characters and lines, or inches or centimeters. The last two can be given as characters and lines per inch or per centimeter. In addition, the character pitch can be specified as pica for 10 characters per inch (cpi), elite for 12 cpi, or compressed for the maximum cpi the printer can provide (up to a limit of 30 cpi).

Set the defaults using one or more of the following commands:

```
lpadmin -p printer-name -o width=scaled-number
lpadmin -p printer-name -o length=scaled-number
lpadmin -p printer-name -o cpi=scaled-number
lpadmin -p printer-name -o lpi=scaled-number
```

Append the letter i to the *scaled-number* to indicate inches, or the letter c to indicate centimeters. The letter i for character pitch (cpi) or line pitch (lpi) is redundant. You can also give pica, elite, or compressed instead of a number for the character pitch.

If you don't provide defaults when you configure a printer, then the page size and print spacing will be taken from the data for your printer type in the `terminfo` database. (If you do not specify a printer type, then the type will be `unknown`, for which there is an entry in the `terminfo` database.) You can find out what the defaults will be by first defining the printer configuration without providing your own defaults, then using the `lpstat` command to display the printer configuration. The command

```
lpstat -p printer-name -l
```

will report the default page size and print spacing.

Printer Class Membership

NOTE This section does not apply if you are making a remote printer accessible to users on your system.

It is occasionally convenient to treat a collection of printers as a single class. The benefit is that a user can submit a file for printing by a member of a class, and the LP print service will pick the first printer in the class that it finds free. This allows faster turn-around, as printers are kept as busy as possible.

Classes aren't needed if the only purpose is to allow a user to submit a print request by type of printer. The `lp -T` *content-type* command allows a user to submit a file and specify its type. The first available printer that can handle the type of the file will be used to print it. The LP print service will avoid using a filter, if possible, by choosing a printer that can print the file directly over one that would need it filtered first.

NOTE See the "Providing Filters" section of this chapter for more information about filters.

Classes do have uses, however. One use is to put into a class a series of printers that should be used in a particular order. If you have a high speed printer and a low speed printer, for instance, you probably want the high speed printer to handle as many print requests as possible, with the low speed printer reserved for use when the other is busy. Because the LP print service always checks for an available printer in the order the printers were added to a class, you could add the high speed printer to the class before the low speed printer, and let the LP print service route print requests in the order you wanted.

Until you add a printer to a class, it doesn't belong to one. If you want to do so, use the following command:

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

If the class *class-name* doesn't exist yet, it will be created. If you want to remove a printer from a class without deleting the printer, enter the following command:

```
lpadmin -p printer-name -r class-name
```

The class name may contain a maximum of fourteen alphanumeric characters and underscores.

NOTE Class names and printer names must be unique. Because they are, a user can specify the destination for a print request without having to know whether it's a class of printers or a single printer.

System Default Destination

You can define the printer or class to be used to print a file when the user has not explicitly asked for a particular destination and has not set the LPDEST shell variable. The printer or class must already exist.

Make a printer or class the default destination by entering the following command:

```
lpadmin -d printer-or-class-name
```

If you later decide that there should be no default destination, enter a null *printer-or-class-name* as in the following command.

```
lpadmin -d
```

If you don't set a default destination, there will be none. Users will have to explicitly name a printer or class in each print request, (unless they specify the `-T` *content-type* option) or will have to set the `LPDEST` shell variable with the name of a destination.

Mounting a Form or Print Wheel

NOTE See "Providing Forms" later in this chapter for information about pre-printed forms.

Before the LP print service can start printing files that need a preprinted form or print wheel, you must physically mount the form or print wheel on a printer, and notify the LP print service that you have mounted it. (It is not necessary for a form to be included on the allow list in order to mount it.) If alerting has been set on the form or print wheel, you will be alerted when enough print requests are queued waiting for it to be mounted. (See "Alerting to Mount a Form" below and "Alerting to Mount a Print Wheel" above.)

When you mount a form you may want to see if it is lined up properly. If an alignment pattern has been defined for the form, you can ask that this be repeatedly printed after you've mounted the form, until you have adjusted the printer so that the alignment is correct.

Mounting a form or print wheel involves first loading it onto the printer and then telling the LP print service that it is mounted. Because it is difficult to do this on a printer that's currently printing, and because the LP print service will continue to print files not needing the form on the printer, you will probably have to disable the printer first. Thus, the proper procedure is to follow these three steps:

1. Disable the printer, using the `disable` command.
2. Mount the new form or print wheel as described immediately after this list.
3. Re-enable the printer, using the `enable` command. (The `disable` and `enable` commands are described in the "Enabling and Disabling a Printer" section of this chapter.)

First, physically load the new form or print wheel into the printer. Then enter the following command to tell the LP print service it has been mounted. (The following command line is split into two lines for readability.)

```
lpadmin -p printer-name -M -S print-wheel-name \
        -f form-name -a -o filebreak
```

Leave out the `-S` *print-wheel-name* if you are mounting just a form, or leave out the `-f` *form-name* `-a -o filebreak` if you are mounting just a print wheel.

If you are mounting a form with an alignment pattern defined for it, you will be asked to press the **RETURN** key before each copy of the alignment pattern is printed. After the pattern is printed, you can adjust the printer and press the **RETURN** key again. If no alignment pattern has been defined, you won't be asked to press the **RETURN** key. You can drop the `-a` and `-o filebreak` options if you don't want to bother with the alignment pattern.

The `-o filebreak` option tells the LP print service to add a "formfeed" after each copy of the alignment pattern. The actual control sequence used for the "formfeed" depends on the printer involved and is obtained from the `terminfo` database. If the alignment pattern already includes a formfeed, leave out the `-o filebreak` option.

If you want to unmount a form or print wheel, use the following command:

```
lpadmin -p printer-name -M -S none -f none
```

Leave out the option `-S none` if you just want to unmount a form; likewise, leave out the `-f none` option if you just want to unmount a print wheel.

Until you've mounted a form on a printer, only print requests that don't require a form will be printed. Likewise, until you've mounted a print wheel on a printer, only print requests that don't require a particular print wheel will be printed. Print requests that do require a particular form or print wheel will be held in a queue until the form or print wheel is mounted.

Removing a Printer or Class

You can remove a printer or class if it has no pending print requests. If there are pending requests, you have to first move them to another printer or class (using the `lpmove` command), or cancel them (using the `cancel` command).

Removing the last remaining printer of a class automatically removes the class as well. The removal of a class, however, does not cause the removal of printers that were members of the class. If the printer or class removed is also the system default destination, the system will no longer have a default destination.

To remove a printer or class, enter the following command:

```
lpadmin -x printer-or-class-name
```

If all you want to do is to remove a printer from a class without deleting that printer, enter the following command:

```
lpadmin -p printer-name -r class-name
```

Putting It All Together

It is possible to add a new printer by completing a number of separate steps, shown in the commands described above. You may find it easier, however, to enter one or two commands that combine all the necessary arguments. Below are some examples.

Example 1

Add a new printer called `lp1` (of the type 455) on printer port `/dev/term/13`. It should use the standard interface program, with the default page size of 90 columns by 71 lines, and linefeeds should *not* be mapped into carriage return/linefeed pairs. (The following command line is split into two lines for readability.)

```
lpadmin -p lp1 -v /dev/term/13 -T 455 \
     -o "width=90 length=71 stty=-onlcr"
```

Example 2

Add a new printer called `laser` on printer port `/dev/term/41`. It should use a customized interface program, located in the directory `/usr/doceng/laser_intface`. It can handle three file types—`i10`, `i300`, and `impress`—and it may be used only by the users `doceng` and `docpub`. (The following command line is split into two lines for readability.)

```
lpadmin -p laser -v /dev/term/41 \
     -i /usr/doceng/laser_intface \
     -I "i10,i300,impress" \
     -u "allow:doceng,docpub"
```

Example 3

When you added the `lp1` printer in the first example, you did not set the alerting. Do this now: have the LP print service alert you—by writing to the terminal on which you are logged in—every 10 minutes after a fault until you fix the problem.

```
lpadmin -p lp1 -A write -W 10
```

Examining a Printer Configuration

Once you've defined a printer configuration, you probably want to review it to see if it is correct. If after examining the configuration you find you've made a mistake, just reenter the command that applies to the part that's wrong.

Use the `lpstat` command to examine both the configuration and the current status of a printer. The short form of this command gives just the status; you can use it to see if the printer exists and if it is busy, idle, or disabled. The long form of the command gives a complete configuration listing.

Enter one of the following commands to examine a printer.

```
lpstat -p printer-name
lpstat -p printer-name -l
```

(The second command is the long form.) With either command you should see one of the following lines of output.

```
printer printer-name now printing request-id. enabled since date.

printer printer-name is idle. enabled since date.

printer printer-name disabled since date.
        reason

printer printer-name waiting for auto-retry.
        reason
```

The `waiting for auto-retry` output shows that the LP print service failed in trying to use the printer (because of the *reason* shown), and that it will try again later.

With the long form of the command, you should also see the following output:

```
Form mounted: form-name
Content types: content-type-list
Printer type: printer-type
Description: comment
Connection: connection-info
Interface: pathname
On fault: alert-method
After fault: fault-recovery
Users allowed:
     user-list
Forms allowed:
     form-list
Banner required
Character sets:
     character-set-list
Default pitch: integer CPI, integer LPI
Default page size: scaled-decimal-number wide, scaled-decimal-number long
Default port settings: stty-option-list
```

Making Printers Available

There are two steps in making a printer ready for use after you've defined the printer configuration. First, you must instruct the LP print service to accept print requests for the new printer. To do this, run the `accept` command. Second, you must activate or enable the new printer. To do this, run the `enable` command. These tasks are separate steps because you may have occasion to want to do one but not the other.

Accepting Print Requests for a New Printer

Telling the LP print service to accept print requests for the new printer is done with the `accept` command. You will read more about this command in a later section, "Managing the Printing Load." For now, all you need to know is that you should enter the following command to let this printer be used.

```
accept printer-or-class-name
```

As you can see, this command is needed to let the LP print service start accepting print requests for a class, too. To prevent the print service from accepting any more requests, execute the following command.

```
reject printer-or-class-name
```

Enabling and Disabling a Printer

Because you may want to make sure, before printing begins, that the correct form is loaded in your printer, the correct print wheel or font cartridge is in place, and the printer is on-line, the LP print service will wait for an explicit signal from you before it starts printing files. Once you have verified that all the necessary components are in place, you can request the beginning of printing by issuing the `enable` command for a particular printer, as follows:

```
enable printer-name
```

If you want to enable several printers simultaneously, list the printers (separating the names with spaces) on the same line as the `enable` command. Don't enclose the list in quotes.

Disabling a printer stops further print requests from being printed. (It does not, however, stop the LP print service from accepting new print requests for the printer.) From time to time you may want to disable a printer. For example, you may want to interrupt a print request, or you may want to change a form or print wheel, in which case you should disable the printer first. Normally, disabling a printer also stops the request that's currently being printed, placing it back in the queue so it can be printed later. You can, however, have the LP print service wait until the current request finishes, or even cancel the request outright.

To disable a printer, enter one of the following commands:

```
disable -r "reason" printer-name
disable -W -r "reason" printer-name
disable -c -r "reason" printer-name
```

The first command disables the printer, stopping the currently printing request and saving it for printing later. The other commands also disable the printer, but the second one makes the LP print service wait for the current request to finish, while the third cancels the current request.

NOTE The -c and -W options are not valid when the disable command is run to stop a remote printer because, when run for a remote printer, disable stops the transferring (rather than the actual printing) of print requests.

The *reason* is stored and displayed whenever anyone checks the status of the printer. You can omit it (and the -r option) if you don't want to specify a reason.

Several printers can be disabled at once by listing their names in the same line as the disable command.

NOTE You can only enable or disable local printers; the loading of forms, print wheels, and cartridges in a remote printer and the enabling of that printer are the responsibility of the administrator of the remote system. You can, however, enable or disable the transfer of print requests to the remote system on which a printer is located. Only individual printers can be enabled and disabled; classes cannot.

Allowing Users to Enable and Disable a Printer

You may want to make the enable and disable commands available for use by other users. This availability is useful, for instance, if you have a small organization where anyone who spots a problem with the printer should be able to disable it and fix the problem. This is *not* a good idea if you want to keep others from interfering with the proper operation of the print services.

If you want to allow others access to the enable and disable commands, use a standard UNIX system feature called the "setuid bit." By assigning ownership of these commands to the user lp (this should have been done automatically when you installed the software), and by setting the setuid bit, you can make sure that anyone will be allowed to use the enable and disable commands. Clearing the bit removes this privilege.

To allow everybody to run enable and disable, enter the following two commands:

```
chown lp /usr/bin/enable /usr/bin/disable
chmod u+s /usr/bin/enable /usr/bin/disable
```

The first command makes the user lp the owner of the commands; this step should be redundant, but it is safer to run the command than to skip it. The second command turns on the setuid bit.

To prevent others from running enable and disable, enter the following command:

```
chmod u-s /usr/bin/enable /usr/bin/disable
```

Troubleshooting

Here are a few suggestions of what to do if you are having difficulty getting a printer to work.

No Output (Nothing Is Printed)

The printer is sitting idle; nothing happens. First, check the documentation that came with the printer to see if there is a self-test feature you can invoke; make sure the printer is working before continuing.

There are three possible explanations when you don't receive any output: (1) the printer might not be connected to the computer; (2) the printer might not be enabled; or (3) the baud rate for the computer and the printer might not be set correctly. The rest of this section describes each of these situations in detail.

Is the Printer Connected to the Computer?

The type of connection between a computer and a printer may vary. Refer to the manual that came with your printer for details.

Is the Printer Enabled?

The printer must be "enabled" in two ways: First, the printer must be turned on and ready to receive data from the computer. Second, the LP print service must be ready to use the printer. If you receive error messages when setting up your printer, follow the "fixes" suggested in the messages. When the printer is set up, issue the commands

```
accept printer-name
enable printer-name
```

where *printer-name* is the name you assigned to the printer for the LP print service. Now submit a sample file for printing:

```
lp -d printer-name file-name
```

Is the Baud Rate Correct?

If the baud rate (the rate at which data are transmitted) is not the same for both the computer and the printer, sometimes nothing will be printed (see below).

Illegible Output

The printer tries printing, but the output is not what you expected; it certainly isn't readable. There are five possible explanations for this situation: (1) the baud rate for the printer might not match the baud rate for the computer; (2) the parity setting of the computer might be incorrect; (3) the tabs might be set incorrectly; or (4) the printer type might not have been set correctly. The rest of this section describes each of these situations in detail.

Is the Baud Rate Correct?

Usually, when the baud rate of the computer doesn't match that of the printer, you'll get some output but it will not look at all like what you submitted for printing. Random characters will appear, with an unusual mixture of special characters and unlikely spacing.

Read the documentation that came with the printer to find out what its baud rate is. It should probably be set at 9600 baud for optimum performance, but that doesn't matter for now. If it isn't set to 9600 baud, you can have the LP print service use the correct baud rate (by default it uses 9600). If the printer is connected via a parallel port, the baud rate is irrelevant.

To set a different baud rate for the LP print service, enter the following command:

```
lpadmin -p printer-name -o stty=baud-rate
```

Now submit a sample file for printing (explained earlier in this section).

Is the Parity Setting Correct?

Some printers use a "parity bit" to ensure that the data received for printing has not been garbled in transmission. The parity bit can be encoded in several ways; the computer and the printer must agree on which one to use. If they do not agree, some characters either will not be printed or will be replaced by other characters. Generally, though, the output will look approximately correct, with the spacing of "words" typical for your document and many letters in their correct place.

Check the documentation for the printer to see what the printer expects. The LP print service will not set the parity bit by default. You can change this, however, by entering one of the following commands:

```
        lpadmin -p printer-name -o stty=oddp
        lpadmin -p printer-name -o stty=evenp
        lpadmin -p printer-name -o stty=-parity
```

The first command sets odd parity generation, the second sets even parity. The last command sets the default, no parity.

If you are also setting a baud rate other than 9600, you may combine the baud rate setting with the parity settings, as in the sample command below.

```
        lpadmin -p printer-name -o "stty='evenp 1200'"
```

Tabs Set Correctly?

If the printer doesn't expect to receive tab characters, the output may contain the complete content of the file, but the text may appear in a chaotic looking format, jammed up against the right margin (see below).

Correct Printer Type?

See "Wrong Character Set or Font" below.

Legible Printing, but Wrong Spacing

The output contains all of the expected text and may be readable, but the text appears in an undesirable format: double spaced, with no left margin, run together, or zig-zagging down the page. These problems can be fixed by adjusting the printer settings (if possible) or by having the LP print service use settings that match those of the printer. The rest of this section provides details about solving each of these types of problems.

Double Spaced

Either the printer's tab settings are wrong or the printer is adding a linefeed after each carriage return. (The LP print service has a carriage return added to each linefeed, so the combination causes two linefeeds.) You can have the LP print service not send tabs or not add a carriage return by using the `stty -tabs` option or the `-onlcr` option, respectively.)

```
lpadmin -p printer-name -o stty=-tabs
lpadmin -p printer-name -o stty=-onlcr
```

No Left Margin/Runs Together/Jammed Up

The printer's tab settings aren't correct; they should be set every 8 spaces. You can have the LP print service not send tabs by using the `-tabs` option.

```
lpadmin -p printer-name -o stty=-tabs
```

Zig Zags Down the Page

The `stty onlcr` option is not set. This is set by default, but you may have cleared it accidentally.

```
lpadmin -p printer-name -o stty=onlcr
```

A Combination of Problems

If you need to use several of these options to take care of multiple problems, you can combine them in one list, as shown in the sample command below. Include any baud rate or parity settings, too.

```
lpadmin -p printer-name -o "stty='-onlcr -tabs 2400'"
```

Correct Printer Type?

See below.

Wrong Character Set or Font

If the wrong printer type was selected when you set up the printer with the LP print service, the wrong "control characters" can be sent to the printer. The results are unpredictable and may cause output to disappear or to be illegible, making it look like the result of one of the problems described above. Another result may be that the wrong control characters cause the printer to set the wrong character set or font.

If you don't know which printer type to specify, try the following to examine the available printer types. First, if you think the printer type has a certain name, try the following command:

```
tput -T printer-type longname
```

(This may not work on early versions of System V.) The output of this command will appear on your terminal: a short description of the printer identified by the *printer-type*. Try the names you think might be right until you find one that identifies your printer.

If you don't know what names to try, you can examine the `terminfo` directory to see what names are available. Warning: There are probably many names in that directory. Enter the following command to examine the directory.

```
ls -R /usr/share/lib/terminfo/*
```

Pick names from the list that match one word or number identifying your printer. For example, the name `495` would identify the AT&T 495 Printer. Try each of the names in the other command above.

When you have the name of a printer type you think is correct, set it in the LP print service by entering the following command:

```
lpadmin -p printer-name -T printer-type
```

Dial Out Failures

The LP print service uses the Basic Networking Utilities to handle dial out printers. If a dialing failure occurs and you are receiving printer fault alerts, the LP print service reports the same error reported by the Basic Networking software for similar problems. (If you haven't arranged to receive fault alerts, they are mailed, by default, to the user `lp`.) See the "Basic Networking Utilities Error Messages" section of Appendix E in this guide for an explanation of the failure reasons.

Idle Printers

There are several reasons why you may find a printer idle and enabled but with print requests still queued for it:

- The print requests need to be filtered. Slow filters run one at a time to avoid overloading the system. Until a print request has been filtered (if it needs slow filtering), it will not print. Use the following command to see if the first waiting request is being filtered.

  ```
  lpstat -o -l
  ```

- The printer has a fault. After a fault has been detected, printing resumes automatically, but not immediately. The LP print service waits about five minutes before trying again, and continues trying until a request is printed successfully. You can force a retry immediately by enabling the printer as follows:

  ```
  enable printer-name
  ```

- A dial out printer is busy or doesn't answer, or all dial out ports are busy. As with automatic continuation after a fault, the LP print service waits five minutes before trying to reach a dial out printer again. If the dial out printer can't be reached for an hour or two (depending on the reason), the LP print service finally alerts you to a possible problem. You can force a retry immediately by enabling the printer as follows:

  ```
  enable printer-name
  ```

Networking Problems

You may encounter several types of problems while trying to get files printed over a network: (1) requests being sent to remote printers may back up in the local queue; (2) requests sent to remote printers may be backed up in the remote queue; or (3) a user may receive contradictory messages about whether a remote printer has accepted a print request. The rest of this section describes each of these situations and suggests how to resolve them.

Jobs Backing Up in the Local Queue

There are a lot of jobs backing up in the local queue for a remote printer. There are three possible explanations:

- The remote system is down or the network between the local and remote systems is down. To resolve this problem, run the `reject` command for all the remote printers on your system, as follows:

  `reject` *printer-name*

  This will stop new requests for those printers from being added to the queue. Once the system comes up again, and jobs start being taken from your queue, type `accept` *printer* to allow new jobs to be queued.
- The remote printer is disabled on the local system.
- The underlying System V network software was not set up properly. For details, see `lpsystem`(1M).

Jobs Backing Up in the Remote Queue

The remote printer has been disabled.

Conflicting Messages About the Acceptance/Rejection of Jobs

A user enters a print request and is notified that the system has accepted it. The job is sent to a remote system and the user receives mail that the job has been rejected. This may be happening for one of two reasons. First, the local computer may be accepting requests while the remote computer is rejecting requests.

Second, the definition of the remote printer on the local computer may not match the definition of that printer on the remote computer. Specifically, the definitions of print job components such as filters, character sets, print wheels, and forms are not the same on the local and remote systems. Identical definitions of these job components must be registered on both the local and the remote systems, if local users are to be able to access remote printers.

Providing Forms

A form is a sheet of paper, on which text or graphical displays have already been printed, that can be loaded into a local printer (that is, a printer on your system) for use in place of plain stock. Common examples of forms include company letterhead, special paper stock, invoices, blank checks, vouchers, receipts, and labels.

Typically, several copies of a blank form are loaded into a printer, either as a tray of single sheets or as a box of fan-folded paper. An application is used to generate data that will be printed on the form, thereby filling it out.

The LP print service helps you manage the use of preprinted forms, but does not provide your application any help in filling out a form; this is solely your application's responsibility. The LP print service, however, will keep track of which print requests need special forms mounted and which forms are currently mounted. It can alert you to the need to mount a new form.

This section tells you how you can manage the use of preprinted forms with the LP print service. You will see how you can

- define a new form
- change the print service's description of an existing form
- remove the print service's description of a form
- examine the print service's description of a form
- restrict user access to a form
- arrange alerting to the need to mount a form
- inform the print service that a form has been mounted

Defining a Form

When you want to provide a new form, the first thing you have to do is define its characteristics. To do so, enter information about each of the nine required characteristics (page length, page width, and so on) as input to the `lpforms` command (see below for details). The LP print service will use this information for two purposes: to initialize the printer so that printing is done properly on the form, and to send you reminders about how to handle that form. Before running the `lpforms` command, gather the following information about your new form:

Page length The length of the form, or of each page in a multi-page form. This can be expressed as the number of lines, or the size in inches or centimeters.

Page width The width of the form, expressed in characters, inches, or centimeters.

Number of pages
The number of pages in a multi-page form.

The LP print service uses this number with a filter (if available) to restrict the alignment pattern to a length of one form. (See the description of alignment patterns below.) If no filter is available, the LP print service does not truncate the output.

Line pitch A measurement that shows how closely together separate lines appear on the form. It can be expressed in either lines per inch or lines per centimeter.

Character pitch
A measurement that shows how closely together separate characters appear on the form. It can be expressed in either characters per inch or characters per centimeter.

Character set choice
The character set, print wheel, or font cartridge that should be used when this form is used. A user may choose a different character set for his or her own print request when using this form, or you can insist that only one character set be used.

Ribbon color If the form should always be printed using a certain color ribbon, then the LP print service can remind you which color to use when you mount the form.

Comment Any remarks that might help users understand what the form is, when it should be used, and so on.

Alignment pattern
A sample file that the LP print service uses to fill one blank form. When mounting the form, you can print this pattern on the form to align it properly. You can also define a content type for this pattern so that the printer service knows how to print it.

NOTE The LP print service does not try to mask sensitive information in an alignment pattern. If you do not want sensitive information printed on sample forms - very likely the case when you align checks, for instance - then you should mask the appropriate data. The LP print service keeps the alignment pattern stored in a safe place, where only you (that is, the user `lp` and the super-user `root`) can read it.

When you've gathered this information about the form, enter it as input to the `lpforms` command. You may want to record this information first in a separate file so you can edit it before entering it with `lpforms`. You can then use the file as input instead of typing each piece of information separately after a prompt. Whichever method you use, enter the information in the following format:

```
Page length: scaled-number
Page width: scaled-number
Number of pages: integer
Line pitch: scaled-number
Character pitch: scaled-number
Character set choice: character-set-name[,mandatory]
Ribbon color: ribbon-color
Comment:
informal notes about the form
Alignment pattern: [content-type]
alignment-pattern
```

Although these attributes are described in detail on the previous page, a few points should be emphasized here. First, the phrase `[mandatory]` is optional and, if present, means that the user cannot override the character set choice in the form. Second, *content-type* can be given optionally, with an alignment pattern. If this attribute is given, the print service uses it to determine, as necessary, how to filter and print the file.

With two exceptions, the information in the above list may appear in any order. The exceptions are the alignment pattern (which must always appear last) and the *comment* (which must always follow the line with the `Comment:` prompt). If the *comment* contains a line beginning with a key phrase (such as `Page length`, `Page width`, and so on), precede that line with a > character so the key phrase is hidden. Be aware, though, that any initial > will be stripped from the comment when it is displayed.

Not all of the information has to be given. When you don't specify values for the items listed below, the values shown beside them are assigned by default.

Item	Default
Page length	66 lines
Page width	80 columns
Number of pages	1
Line pitch	6 per inch
Character pitch	10 per inch
Character set choice	any
Ribbon color	any
Comment	(no default)
Alignment pattern	(no default)

To define the form, use one of the following commands

```
lpforms -f form-name -F file-name
lpforms -f form-name -
```

where *file-name* is the full path for the file.

The first command gets the form definition from a file; the second command gets the form definition from you, through the standard input. A *form-name* can be anything you choose, as long as it contains a maximum of fourteen alphanumeric characters and underscores.

If you need to change a form, just reenter one of the above commands. You need only provide information for items that must be changed; items for which you don't specify new information will stay the same.

Removing a Form

The LP print service imposes no fixed limit on the number of forms you may define. It is a good idea, however, to remove forms that are no longer appropriate. If you don't, users will see a long list of obsolete forms when choosing a form, and may be confused. In addition, because the LP print service must occasionally look through all the forms listed before performing certain tasks, the failure to remove obsolete forms may require extra, unnecessary processing by the print service.

To remove a form, enter the following command:

```
        lpforms -f form-name -x
```

Restricting User Access

If your system has a form that you don't want to make available to everyone, you can limit its availability to selected users. For example, you may want to limit access to checks to the people in the payroll department or accounts payable department.

The LP print service restricts the availability of a form by using the lists (provided by you) of users allowed or denied access to that form. If a user is not allowed to use a particular form, the LP print service will reject his or her request to print a file with it.

The method used to allow or deny users access to a form is similar to the method used to allow or deny users access to the `cron` and `at` facilities. (See the description of the `crontab` command in the *User's Reference Manual*.) Briefly, the rules are as follows:

- An allow list is a list of users who are allowed to use the form. A deny list is a list of users who are not allowed to use the form.
- If the allow list is not empty, only the users listed are allowed; the deny list is ignored. If the allow list is empty, the users listed in the deny list are not allowed to to use the form. If both lists are empty, there are no restrictions on who may use the form.
- Specifying `all` in the allow list allows everybody to use the form; specifying `all` in the deny list allows no one except the user `lp` and the super-user (`root`) to use the form.

If users on your system are to be able to access forms on a remote printer, it's necessary for all the users included on the allow list for the local system to be included on the allow list for the remote system, as well.

If, on the other hand, a local user is to be denied permission to use forms on a remote printer, it's not necessary for the deny lists on both the local and remote print services to include that user. By being included in only one of these deny lists, a user can be denied access to remote forms. As a courtesy to your users, however, it's a good idea to make sure that any local users who are included in a deny list on a remote system are included in the corresponding deny list on your local system. By doing this you can make sure that whenever a user on your system requests a form that he or she is not authorized to use, he or she is immediately informed that permission to use the form is being denied. If the local print service does not "know" that a user is denied permission to use a particular remote form, there will be a delay before the user receives a "permission denied" message from the remote system.

You can add names of users to either list, using one of the following commands:

```
        lpforms -f form-name -u allow:user-list
        lpforms -f form-name -u deny:user-list
```

The *user-list* is a comma or space separated list of names of users. If you use spaces to separate the names, enclose the entire list (including the `allow:` or `deny:` but not the `-u`) in quotes. Each item in the list can include a system name, as shown under "User Access Restrictions" earlier in this chapter. The first command adds the names to the allow list and removes them from the deny list. The second command adds the names to the deny list and removes them from the allow list. Specifying `allow:all` will allow everybody; specifying `deny:all` will deny everybody.

If you do not add user names to the allow or deny lists, the LP print service will assume that everybody may use the form.

Alerting to Mount a Form

If you define more forms than printers, you will obviously not be able to print files on all the forms simultaneously. This means that some print requests may be held in a queue until you mount the forms they need. How will you know when to mount a particular form? One method would be to monitor, periodically, the number of print requests pending for that form. The LP print service, however, provides an easier way: You can ask to be alerted when the number of requests waiting for a form has exceeded a specified threshold.

You can choose one of several ways to receive an alert.

- You can receive an alert via electronic mail. (See the description of the `mailx` command in the *User's Reference Manual* for a description of mail on the UNIX system.)
- You can receive an alert written to any terminal on which you are logged in. (See the description of the `write` command in the *User's Reference Manual*.)
- You can receive an alert through a program of your choice.
- You can receive no alerts.

NOTE If you elect to receive no alerts, you are responsible for checking to see whether any print requests haven't been printed because the proper form isn't mounted.

In addition to the method of alerting, you can also set the number of requests that must be queued before you are alerted, and you can arrange for repeated alerts every few minutes until the form is mounted. You can choose the rate of repeated alerts, or choose to receive only one alert for each form.

To arrange for alerting to the need to mount a form, enter one of the following commands:

```
lpforms -f form-name -A mail -Q requests -W minutes
lpforms -f form-name -A write -Q requests -W minutes
lpforms -f form-name -A 'command' -Q requests -W minutes
```

The first two commands direct the LP print service to send you a mail message or write the message directly to your terminal, respectively, for each alert. The third command directs the LP print service to run the *command* for each alert. The shell environment in effect when you enter the third command is saved and restored for the execution of *command*; this includes the environment variables, user and group IDs, and the current directory.

In each command line, the argument *requests* is the number of requests that need to be waiting for the form to trigger the alert, and the argument *minutes* is the number of minutes between repeated alerts.

NOTE

If you want mail sent or a message written to another user for an alert, use the third command with the option `-A 'mail` *login-name*`'` or `-A 'write` *login-name*`'`. If you do not specify a login-name, the mail or message will be sent to your current ID. This may not be your login ID, if you have used the `su` command to change IDs.

If you want the print service to issue no alert when the form needs to be mounted, execute the following command:

```
lpforms -f form-name -A none
```

When you start receiving repeated alerts, you can direct the LP print service to stop sending you alerts (for the current case only) by issuing the following command:

```
lpforms -f form-name -A quiet
```

NOTE

Use the alert type `quiet` only to terminate an active alert; do not specify `quiet` as the alert type for a new printer.

Once the form has been mounted and unmounted again, alerts will resume if too many requests are waiting. Alerts will also start again if the number of requests waiting falls below the `-Q` threshold and then rises up to the `-Q` threshold again. This happens when waiting requests are canceled, and when the type of alerting is changed.

If *form-name* is `all` in any of the commands above, the alerting condition applies to all forms for which an alert has not already been defined.

If you don't define an alert method for a form, you will not receive an alert to mount it. If you define a method without the `-W` option, you will be alerted once for each occasion.

Mounting a Form

Refer to the section "Mounting a Form or Print Wheel" under "Configuring Your Printers" in this chapter.

Examining a Form

Once you've defined a form to the LP print service, you can examine it with one of two commands, depending on the type of information you want to check. The `lpforms` command displays the attributes of the form. (The display produced by `lpforms` can be used as input; you may want to save it in a file for future reference.) The `lpstat` command displays the current status of the form.

Enter one of the following commands to examine a defined form.

```
lpforms -f form-name -l lpforms -f form-name -l > file-name
lpstat -f form-name lpstat -f form-name -l
```

The first two commands present the definition of the form; the second command captures this definition in a file, which can be used later to redefine the form if you inadvertently remove the form from the LP print service. The last two commands present the status of the form, with the second of the two giving a long form of output, similar to the output of `lpforms -l`:

```
Page length: scaled-number Page width: scaled-number Number of
pages: integer Line pitch: scaled-number Character pitch:
scaled-number Character set choice: character-set[,mandatory]
Ribbon color: ribbon-color Comment: comment Alignment pattern:
[content-type] content
```

To protect potentially sensitive content, the alignment pattern is not shown if the `lpstat` command is used.

Providing Filters

This section explains how you can manage the use of filters with the LP print service. You will see how you can

- define a new filter
- change a filter
- remove a filter
- examine a filter

The "Customizing the Print Service" section at the end of this chapter describes how you can write a filter. First, let's see what a filter is and how the LP print service can use one.

What Is a Filter?

A filter is a program that you can use for any of three purposes:

- To convert a user's file from one data format to another so that it can be printed properly on a given printer
- To handle the special modes of printing that users may request with the `-y` option to the `lp` command (such as two-sided printing, landscape printing, draft or letter quality printing)
- To detect printer faults and notify the LP print service of them, so that the print service can alert you

Not every filter can perform all three tasks. Given the printer-specific nature of these three roles, the LP print service has been designed so that these roles can be implemented separately. This separation allows you or a printer manufacturer (or another source) to provide filters without having to change the LP print service.

A default filter is provided with the LP print service to provide simple printer fault detection; it does not convert files or handle any of the special modes. It may, however, be adequate for your needs.

Let's examine the three tasks performed by filters more closely.

Task 1: Converting Files

For each printer (local or remote) you can specify what file content types it can print. When a user submits a file to print on any printer, and specifies its content type, the print service will find a printer that can handle files of that content type. Because many applications can generate files for various printers, this is often sufficient. However, some applications may generate files that cannot be printed on your printers.

By defining and creating a filter that converts such files into a type that your printers can handle, you can begin to support more applications in the LP print service. (The LP print service comes with a few filters for converting various types of files into PostScript.) For each filter you add to the system, you must specify one or more types of input it can accept and the type of output it can produce (usually only one).

When a user specifies (by executing `lp -T`) a file content type that no printer can handle, the print service tries to find a filter that can convert the file into an acceptable type. If the file to be printed is passed through a filter, the print service will then match the output type of that filter with a printer type or the input type of another filter. The LP print service will continue to match output types to input types in this way, thus passing a file through a series of filters, until the file reaches a printer that accepts it.

Below are some examples.

Example 1

The user Chris has run a spreadsheet program and has generated a file containing a copy of a spreadsheet. Chris now wants to print this file using the LP print service. You have only AT&T model 455 printers on your system. Fortunately, the spreadsheet application understands how to generate output for several printers, and Chris knows it's necessary to request output that can be handled by the AT&T 455. When Chris submits the file for printing, the LP print service queues it for one of the printers; no filter is needed.

Example 2

A user named Marty has created a graphic image that can be displayed on a Tektronix 4014 terminal. Marty now wants to print this image, but all of the printers are PostScript printers. Fortunately, your system provides a filter called `posttek` that converts Tektronix type files to PostScript. Because you set the printer type to PostScript, the LP print service recognizes that it can use the `posttek` filter to convert Marty's output before printing it.

Task 2: Handling Special Modes

Another important role that filters can perform is the handling of special printing modes. Each filter you add to the filter table can be registered to handle special modes and other aspects of printing:

special modes
printer type
character pitch
line pitch
page length
page width
pages to print
character set
form name
number of copies

A filter is required to handle the special modes and printing of specific pages; the LP print service provides a default handling for all the rest. However, it may be more efficient to have a filter handle some of the rest, or it may be that a filter has to know several of these aspects to fulfill its other roles properly. A filter may need to know, for example, the page size and the print spacing if it is going to break up the pages in a file to fit on printed pages. As another example, some printers can handle multiple copies more efficiently than the LP print service, so a filter that can control the printer can use the information about the number of copies to skip the LP print service's default handling of multiple copies.

We'll see below how you can register special printing modes and other aspects of printing with each filter.

Task 3: Detecting Printer Faults

Just as converting a file and handling special printing modes is a printer-specific role, so is the detecting of printer faults. The LP print service attempts to detect faults in general, and for most printers it can do so properly. The range of faults that the print service can detect by itself, however, is limited. It can check for hang-ups (loss of carrier, the signal that indicates the printer is on-line) and excessive delays in printing (receipt of an XOFF flow-control character to shut off the data flow, with no matching XON to turn the flow back on). However, the print service can't determine the cause of a fault, so it can't tell you what to look for.

A properly designed filter can provide better fault coverage. Some printers are able to send a message to the host describing the reason for a fault. Others indicate a fault by using signals other than the dropping of a carrier or the shutting off of data flow. A filter can serve you by detecting more faults and providing more information about them than you would otherwise receive.

Another service a filter can provide is to wait for a printer fault to clear and then to resume printing. This service allows for more efficient printing when a fault occurs because the print request that was interrupted does not have to be reprinted in its entirety. Only a real filter, which has knowledge of the control sequences used by a printer, can "know" where a file breaks into pages; thus only such a filter can find the place in the file where printing should resume.

The LP print service has a simple interface that allows a filter to send you fault information and to restart printing if it can. The alerting mechanism (see the "Printer Fault Alerting" section under "Configuring Your Printers" in this chapter) is handled by the LP print service; the interface program that manages the filter takes all error messages from the filter and places them in an alert message that can be sent to you. Thus you'll see any fault descriptions generated by the filter. If you've set the printer configuration so that printing should automatically resume after a fault is cleared, the interface program will keep the filter active, so that printing can pick up where it left off.

Will Any Program Make a Good Filter?

It is tempting to use a program such as `troff`, `nroff`, or a similar word-processing program as a filter. However, the `troff` and `nroff` programs have a feature that allows references to be made in a source file to other files, known as "include files." The LP print service does not recognize include files; it will not enqueue any that are referenced by a source file when that file is in a queue to be printed. As a result, the `troff` or `nroff` program, unable to access the include files, may fail. Other programs may have similar features that limit their use as filters.

Here are a few guidelines for evaluating a program for use as a filter:

- Only programs capable of reading data from standard input and writing data to standard output may be used as filters.

- Examine the kinds of files users will submit for printing that will require processing by the program. If they stand alone (that is, if they do not reference other files that the program will need), the program is probably okay.

  Check also to see if the program expects any files other than those submitted by a user for printing. If it does, those files must be in the directory of the person using the filter, or they must be readable by all users authorized to use the filter. The latter prerequisite is necessary because filters are run with the user ID and group ID of the user who submitted the print request.

- If referenced files are permitted in the files submitted for printing, or if the program will need files other than those submitted by a user, then the program, unable to access the additional files, is likely to fail. We suggest you don't use the program under consideration as a filter; instead, have users run the program before submitting files for printing.

Referenced files that are always specified by full pathnames *may* be okay, but only if the filter is used for local print requests. When used on requests submitted from a remote machine for printing on your machine, the filter may still fail if the referenced files exist only on the remote machine.

Filters for Your System

The LP print service is delivered with several filters. As you add, change, or delete filters, you may overwrite or remove some of these original filters. If necessary, you can restore the original set of filters (and remove any filters you have added), by running the following command:

```
lpfilter -f all -i
```

Defining a Filter

When adding a new filter, the first thing you must do is to define the characteristics of its use. To do this, issue the `lpfilter` command with arguments that specify the values of the following filter characteristics:

- the name of the filter (that is, a command name)
- the types of input it will accept

- the types of output it will produce
- the types of printers to which it will be able to send jobs
- the names of specific printers to which it will send jobs
- the "type" of the filter (whether it's a `fast` filter or a `slow` filter)
- options

Each of these characteristics is described below.

`Command:` This is the full path of the filter program. If there are any fixed options that the program always needs, include them here.

`Input types:` This is the list of file content types that the filter can process. The LP print service doesn't impose a limit on the number of input types that can be accepted by a filter, but most filters can take only one. Several file types may be similar enough so that the filter can deal with them. You can use whatever names you like here, using a maximum of fourteen alphanumeric characters and dashes (not underscores). Because the LP print service uses these names to match a filter with a file type, you should follow a consistent naming convention. For example, if more than one filter can accept the same input type, use the same name for that input type when you specify it for each filter. These names should be advertised to your users so they know how to identify the type of a file when submitting that file for printing.

`Output types:` This is the list of file types that the filter can produce as output. For each input type the filter will produce a single output type, of course; the output type may vary, however, from job to job. The names of the output types are also restricted to fourteen alphanumeric characters and dashes.

These names should either match the types of printers you have on your system, or match the input types handled by other filters. The LP print service groups filters together in a shell pipeline if it finds that several passes by different filters are needed to convert a file. It's unlikely that you will need this level of sophistication, but the LP print service allows it. Try to find a set of filters that takes (as input types) all the different files your users may want printed, and converts those files directly into types your printers can handle.

`Printer types:`

This is a list of printer types into which the filter can convert files. For most filters this list will be identical to the list of output types, but it can be different.

For example, you may have a printer that is given a single type for purposes of initialization (see the "Printer Type" section under "Printer Management" in this chapter), but which can recognize several different types of files. In essence this printer has an internal filter that converts the various types into one with which it can deal. Thus, a filter may produce one of several output types that match the "file types" that the printer can handle. The filter should be marked as working with that printer type.

As another example, you may have two different models of printers that are listed as accepting the same types of files. However, due to slight differences in manufacture, one printer deviates in the results it produces. You label the printers as being of different printer types, say A and B, where B is the one that deviates. You create a filter that adjusts files to account for the deviation produced by printers of type B. Because this filter is needed only for those printer types, you would list it as working only on type B printers.

For most printers and filters you can leave this part of the filter definition blank.

`Printers:` You may have some printers that, although they're of the correct type for a filter, are in other ways not adequate for the output that the filter will produce. For instance, you may want to dedicate one printer for fast turn-around; only files that the printer can handle without filtering will be sent to that printer. Other printers, of identical type, you allow to be used for files that may need extensive filtering before they can be printed. In this case, you would label the filter as working with only the latter group of printers.

In most cases a filter should be able to work with all printers that accept its output, so you can usually skip this part of the filter definition.

`Filter type:` The LP print service recognizes "fast" filters and "slow" filters. Fast filters are labeled "fast" because they incur little overhead in preparing a file for printing, and because they must have access to the printer when they run. A filter that is to detect printer faults has to be a fast filter. A filter that uses the `PRINTER` keyword as a filter option must be installed as a fast filter.

Slow filters are filters that incur a lot of overhead in preparing a file and that don't require access to a printer. The LP print service runs slow filters in the background, without tying up a printer. This allows files that don't need slow filtering to move ahead; printers will not be left idle while a slow filter works on a file if other files can be printed simultaneously.

Slow filters that are invoked by modes (via the `-y` option), must be run on the computer where the print request was issued. The LP print service can't pass values for modes to remote machines. It can, however, match a file content type (specified after the `-T` option of the `lp` command) to a content type on a remote machine. Therefore, if you want to activate special modes on a remote machine, you must do so by specifying content types that will allow the LP print service to match input types and output types.

`Options:` Options specify how different types of information should be transformed into command line arguments to the filter command. This information may include specifications from a user (with the print request), the printer definition, and the specifications implemented by any filters used to process the request.

There are thirteen sources of information, each of which is represented by a "keyword." Each option is defined in a "template," a statement in the following format:

keyword pattern=replacement.

This type of statement is interpreted by the LP print service to mean "When the information referred to by *keyword* has the value matched by *pattern,* take the *replacement* string, replace any asterisks it contains with the *pattern* specified or expand any regular expressions it contains, and append the result to the command line."

The options specified in a filter definition may include none, all, or any subset of these thirteen keywords. In addition, a single keyword may be defined more than once, if multiple definitions are required for a complete filter definition. (See "Defining Options with Templates" below.)

When you've gathered enough information to define the above characteristics of your filter, you are ready to run the `lpfilter` command, using your data as arguments. Because there are so many arguments, and because some of them may need to be entered more than once (with different values), we recommend you record this information first in a separate file and edit it, if necessary. You can then use the file as input to the `lpfilter` command and avoid typing each piece of information separately.

Whether you store the information in a file or enter it directly on the command line, use the following format:

```
Command: command-pathname [options] Input types: input-type-list
Output types: output-type-list Printer types: printer-type-list
Printers: printer-list Filter type: fast or slow Options:
template-list
```

The information can appear in any order. Not all the information has to be given. When you do not specify values for the items listed below, the values shown beside them are assigned by default.

Item	Default
`Command:`	(no default)
`Input types:`	`any`
`Output types:`	`any`
`Printer types:`	`any`
`Printers:`	`any`
`Filter type:`	`slow`
`Options:`	(no default)

As you can see, the default values define a very flexible filter, so you probably have to supply at least the input and output type(s). When you enter a list, you can separate the items in it with blanks or commas, unless it is a *templates-list;* items in a *templates-list* must be separated by commas.

Defining Options With Templates

A template is a statement in a filter definition that defines an option to be passed to the filter command based on the value of one of the characteristics of the filter. A filter definition may include more than one template. Multiple templates may be entered on a single line and separated with commas, or they may be entered on separate lines, preceded by the `Options:` prefix.

The format of a template is as follows:

keyword pattern = *replacement*

The *keyword* identifies the type of option being registered for a particular characteristic of the filter.

Let's look at an example of how an option is defined for a particular filter. Suppose you want to have the print service scheduler assign print requests to filters on the basis of the following criteria:

- If the type of `OUTPUT` to be produced by the filter is `impress`, then pass the `-I` option to the filter.
- If the type of `OUTPUT` to be produced by the filter is `postscript`, then pass the `-P` option to the filter.

To specify these criteria, provide the following templates as options to the `lpfilter` command.

```
Options: OUTPUT impress=-I, OUTPUT postscript=-P
```

NOTE

If the Options: line becomes too long, put each template on a separate line, as follows:

```
Options: OUTPUT impress=-I
Options: OUTPUT postscript=-P
```

In both templates, the keyword is defined as OUTPUT. In the first template, the value of *pattern* is impress and the value of the *replacement* is -I. In the second template, the value of *pattern* is postscript and the value of the *replacement* is -P.

Template Keywords

The following thirteen *keywords* are available for defining Options in a filter definition:

Characteristic	*keyword*	Possible *patterns*	Example
Content type (input)	INPUT	*content-type*	troff
Content type (output)	OUTPUT	*content-type*	postscript
Printer type	TERM	*printer-type*	att495
Printer name	PRINTER	*printer-name*	lp1
Character pitch	CPI	*scaled-decimal*	10
Line pitch	LPI	*scaled-decimal*	6
Page length	LENGTH	*scaled-decimal*	66
Page width	WIDTH	*scaled-decimal*	80
Pages to print	PAGES	*page-list*	1-5,13-20
Character set	CHARSET	*character-set*	finnish
Form name	FORM	*form-name*	invoice2
Number of copies	COPIES	*integer*	3
Special modes	MODES	*mode*	landscape

To find out which values to supply for each type of template (that is, for the *pattern* and *replacement* arguments for each *keyword*), see the source of information listed below.

- The values for the INPUT and OUTPUT templates come from the file type that needs to be converted by the filter and the output type that has to be produced by the filter, respectively. They'll each be a type registered with the filter.

- The value for the `TERM` template is the printer type.
- The value for the `PRINTER` template is the name of the printer that will be used to print the final output.
- The values for the `CPI`, `LPI`, `LENGTH`, and `WIDTH` templates come from the user's request, the form being used, or the default values for the printer.
- The value for the `PAGES` template is a list of pages that should be printed. Typically, it is a comma separated list of page ranges, each of which consists of a dash separated pair of numbers or a single number (such as 1-5,6,8,10 for pages 1 through 5, 6, 8, and 10). However, whatever value was given in the `-P` option to a print request is passed unchanged.
- The value for the `CHARSET` template is the name of the character set to be used.
- The value for the `FORM` template is the name of the form requested by the `-f` option of the `lp` command.
- The value of the `COPIES` template is the number of copies that should be made of the file. If the filter uses this template, the LP print service will reduce to 1 the number of copies of the filtered file *it* will have printed, since this "single copy" will really be the multiple copies produced by the filter.
- The value of the `MODES` template comes from the `-y` option of the `lp` command (the command used to submit a print request). Because a user can specify several `-y` options, there may be several values for the `MODES` template. The values will be applied in the left-to-right order given by the user.

The *replacement* part of a template shows how the value of a template should be given to the filter program. It is typically a literal option, sometimes with the place-holder `*` included to show where the value goes. The *pattern* and *replacement* can also use the regular expression syntax of `ed(1)` for more complex conversion of user input options into filter options. All of the regular expression syntax of `ed(1)` is supported, including the `\(` ... `\)` and `\n` constructions, which can be used to extract portions of the *pattern* for copying into the *replacement*, and the `&`, which can be used to copy the entire *pattern* into the *replacement*.

NOTE

If a comma or an equals sign (=) is included in a *pattern* or a *replacement*, escape its special meaning by preceding it with a backslash (\). Note that some regular expressions include commas that will have to be escaped this way. A backslash in front of any of these characters is removed when the *pattern* or *replacement* is used.

The following examples show how this works.

Example 1

You provide the following filter definition for a filter called `col`.

```
Input types:    N37, Nlp, simple
Output types:   simple
Command:        /usr/bin/col
Options:        TERM 450 = -b, MODES expand = -x
Options:        INPUT simple = -p -f
```

NOTE

If you provide more than one definition (that is, more than one line) for any filter characteristic other than `OPTIONS`, only the second definition will be used by the print service.

After you have "registered" this definition with the print service by entering it as input with the `lpfilter` command, users' print requests will be handled as follows:

- If a user enters the command

  ```
  lp -y expand report.dec10
  ```

  the filter command will be run with the following arguments:

  ```
  /usr/bin/col -x -p -f
  ```

- If a user enters the command

  ```
  lp -T N37 -y expand report.dec10
  ```

  the filter command will be run with the following arguments:

  ```
  /usr/bin/col -x
  ```

  Qualifier: The default printer is not of type 450.

- If a user enters the command

  ```
  lp -y expand -T 450 report.dec10
  ```

  the filter command will be run with the following arguments:

  ```
  /usr/bin/col -b -x
  ```

Example 2

The filter program is called `/usr/lib/lp/postscript/dpost`. It takes one input type, `troff`, produces an output type called `postscript`, and works with any printer of type `PS` (for PostScript). You've decided that your users need give just the abbreviations `port` and `land` when they ask for the paper orientation to be portrait mode and landscape mode, respectively. Because these options are not intrinsic to the LP print service, users must specify them using the `-y` option to the `lp` command.

The filter definition would look like this:

```
Input types: troff
Output types: postscript
Printer types: PS
Filter type: slow
Command: /usr/lib/lp/postscript/dpost
Options: LENGTH * = -l*
Options: MODES port = -pp, MODES land = -pl
```

A user submitting a file of type `troff` for printing on a PostScript printer (type `PS`), with requests for landscape orientation and a page length of 60 lines, would enter the following command:

```
lp -T troff -o length=60 -y land -d any
```

Then this filter would be invoked by the LP print service to convert the file as follows:

```
/usr/lib/lp/postscript/dpost -l60 -ol
```

Example 3

You add the following option template to the previous example:

```
Options:  MODES group\=\([1-9]\) = -n\1
```

This template is used to convert a `MODES` option of the form

```
-y group=number
```

into filter options

```
-nnumber
```

So if a user gives the following command

```
lp -y group=4
```

the `dpost` command would include the following options:

```
-n4
```

For additional examples, run the command

```
lpfilter -f filter -l
```

where *filter* is the name of the factory installed PostScript filters. (For a list of PostScript filters, see "PostScript Printers" later in this chapter.)

Command to Enter

Once a filter definition is complete, enter one of the following commands to add the filter to the system.

```
lpfilter -f filter-name -F file-name lpfilter -f filter-name -
```

The first command gets the filter definition from a file, and the second command gets the filter definition from the standard input. A *filter-name* can be any string you choose, with a maximum of fourteen alphanumeric characters and underscores.

If you need to change a filter, just reenter one of the same commands. You need only provide information for those items that must be changed; items for which you don't specify new information will stay the same.

Removing a Filter

The LP print service imposes no fixed limit on the number of filters you can define. It is a good idea, however, to remove filters no longer applicable, to avoid extra processing by the LP print service which must examine all filters to find one that works in a given situation.

To remove a filter, enter the following command:

```
lpfilter -f filter-name -x
```

Examining a Filter

Once you've added a filter definition to the LP print service, you can examine it by running the `lpfilter` command. The output of this command is the filter definition displayed in a format that makes it suitable as input. You may want to save this output in a file that you can use later to redefine the filter if you inadvertently remove the filter from the LP print service.

To examine a defined filter, enter one of the following commands:

```
lpfilter -f filter-name -l lpfilter -f filter-name -l >file-name
```

The first command presents the definition of the filter on your screen; the second command captures this definition in a file for future reference.

Restoring Factory Defaults

The software is shipped from the factory with a default set of filters. If, after changing them, you want to restore some or all of them, enter the following command:

```
lpfilter -f filter-name -i
```

Replace *filter-name* with the name of the filter to restore, or the word `all` to restore all the default filters.

A Word of Caution

Adding, changing, or deleting filters can cause print requests still queued to be canceled. This is because the LP print service evaluates all print requests still queued, to see which are affected by the filter change. Requests that are no longer printable, because a filter has been removed or changed, are canceled (with notifications sent to the people who submitted them). There can also be delays in the responses to new or changed print requests when filters are changed, due to the many characteristics that must be evaluated for each print request still queued. These delays can become noticeable if there is a large number of requests that need to be filtered.

Because of this possible impact, you may want to make changes to filters during periods when the LP print service is not being used much.

Managing the Printing Load

Occasionally you may need to stop accepting print requests for a printer or move pending print requests from one printer to another. There are various reasons why you might want to do this, such as the following:

- the printer needs periodic maintenance
- the printer is broken
- the printer has been removed
- you've changed the configuration so that the printer is to be used differently
- too many large print requests are queued for one printer and should be spread around

If you are going to make a big change in the way a printer is to be used, such as stopping its ability to handle a certain form, changing the print wheels available for it, or disallowing some users from using it, print requests that are currently queued for printing on it will have to be moved or canceled. The LP print service will attempt to find alternate printers, but only if the user doesn't care which printer is to be used. Requests for a specific printer won't be automatically moved; if you don't move them first, the LP print service will cancel them.

If you decide to take a printer out of service, to change its configuration, or to lighten its load, you may want to move print requests off it and reject additional requests for it for awhile. To do so, use the `lpmove` and `reject` commands. If you do reject requests for a printer, you can accept requests for it later, by using the `accept` command.

Rejecting Requests for a Printer or Class

To stop accepting any new requests for a printer or class of printers, enter the following command.

```
reject -r "reason" printer-or-class-name
```

You can reject requests for several printers or classes in one command by listing their names on the same line, separating the names with spaces. The *reason* will be displayed whenever anyone tries to print a file on the printer. You can omit it (and the `-r`) if you don't want to specify a reason.

Although the `reject` command stops any new print requests from being accepted, it will not move or cancel any requests currently queued for the printer. These will continue to be printed as long as the printer is enabled.

Accepting Requests for a Printer or Class

After the condition that led to rejecting requests has been corrected or changed, enter one fo the the following commands to start accepting new requests.

```
accept printer-name
accept class-name
```

Again, you can accept requests for several printers or classes in one command by listing their names on the same line.

You will always have to use the `accept` command for a new printer or class after you have added it, because the LP print service does not initially accept requests for new printers or classes.

Moving Requests to Another Printer

If you specify `-d any` when you run the `lp` command to queue a job, the print service schedules the job for a particular printer. If another becomes available first, the job is sent to the latter printer. If a job is scheduled for a given printer and you run `lpmove` to get jobs off that printer, that job will be moved off and the destination will change from `any` to the printer you've specified on the `lpmove` command line. Users may not have intended this side effect. If not, run the following command:

```
lp -i request-ID -d any
```

This command will change the destination for the requested job to the original destination: `any` (that is, any available printer).

If you have to move requests from one printer or class to another, enter one of the following commands

```
lpmove request-id printer-name lpmove printer-name1 printer-name2
```

You can give more than one request ID before the printer name in the first command.

The first command above moves the listed requests to the printer *printer-name*. The second command tries to move *all* requests currently queued for *printer-name*$_1$ to *printer-name*$_2$. If some requests cannot be printed on the new printer, they will be left in the queue for the original printer. When the second command is used, the LP print service also stops accepting requests for *printer-name*$_1$ (the same result you would obtain by running the command `reject` *printer-name*$_2$).

Examples

Here are some examples of how you might use these three commands:

Example 1

You've decided it is time to change the ribbon and perform some preventive maintenance on printer `lp1`. First, to prevent the loss of print requests already queued for `lp1`, you move all requests from printer `lp1` to printer `lp2`.

```
        lpmove lp1 lp2
```

After the requests are moved, make sure the LP print service does not print any more requests on `lp1` by disabling it.

```
        disable lp1
```

Now you may physically disable the printer and start working on it.

Example 2

You've finished changing the ribbon and doing the other work on `lp1`; now it's time to bring it back into service. Execute the following commands in any order:

```
        accept lp1 enable lp1
```

See the "Enabling and Disabling a Printer" section under "Making Printers Available" in this chapter.

Example 3

You notice that someone has queued several large files for printing on the printer laser1. Meanwhile laser2 is idle because no one has queued requests for it. Move the two biggest requests (laser1-23 and laser1-46) to laser2, and reject any new requests for laser1 for the time being.

```
lpmove laser1-23 laser1-46 laser2 reject -r "too
busy--will reopen later" laser1
```

Managing Queue Priorities

The LP print service provides a simple priority mechanism that users can use to adjust the position of a print request in the queue. Each print request can be given a priority level by the user who submits it; this is a number from 0 to 39, with *smaller* numbers indicating *higher* levels of priority. Requests with higher priority (smaller numbers) are placed ahead of requests with lower priority (larger numbers).

Thus, for example, a user who decides that her print request is of low priority can assign it a larger value when she submits the file for printing. Another user who decides that his print request is of high priority can assign it a smaller value when he submits the file for printing.

A priority scheme this simple would not work if there were no controls on how high one can set the priority. You can define the following characteristics in this scheme:

- Each user can be assigned a priority limit. One cannot submit a print request with a priority higher than his or her limit, although one can submit a request with a lower priority.
- A default priority limit can be assigned for the balance of users not assigned a personal limit.
- A default priority can be set. This is the priority given print requests to which the user does not assign a priority.

By setting the characteristics according to your needs, you can prevent lower priority printing tasks (such as regular printing by most staff members) from interfering with higher priority printing tasks (such as payroll check printing by the accounting staff).

You may find that you want a critical print request to print ahead of any others, perhaps even if it has to preempt the currently printing request. You can have the LP print service give "immediate" handling to a print request, and you can have it put on "hold" another print request. This will allow the first request to be printed and will delay the second print request until you allow it to be "resumed."

The `lpusers` command lets you assign both priority limits for users and priority defaults. In addition, you can use the `lp -i` *request-id* `-H hold` and `lp -i` *request-id* `-H immediate` commands to put a request on hold or to move it up for immediate printing, respectively. These commands are discussed in detail below.

Setting Priority Limits

To set a user's priority limit, enter the following command.

```
        lpusers -qpriority-level -u user-name
```

You can set the limit for a group of users by listing their names after the -u option. Separate multiple names with a comma or space (enclose the list in quotes if you use a space, though). The argument *priority-level* is a number from 0 to 39. As mentioned before, the lower the number the higher the priority, or, in this case, the priority limit.

If you want to set a priority limit for the remaining users, enter the following command:

```
        lpusers -qpriority-level
```

This sets the default limit; the default applies to those users for whom you have not set a personal limit, using the first `lpusers` command.

If you later decide that someone should have a different priority limit, just reenter the first command above with a new limit. Or, if you decide that the default limit is more appropriate for someone who already has a personal limit, enter the following command:

```
        lpusers -u user-name
```

Again, you can do this for more than one user at a time by including a list of names. Using the `lpusers` command with just the -u option removes users' personal priority limits and puts the "default limit" into effect for those users.

Setting a Default Priority

To set the default priority (the priority level assigned to print requests submitted without a priority), use the following command:

```
        lpusers -dpriority-level
```

Don't confuse this default with the "default limit." This default is applied when a user doesn't specify a priority level; the default limit is applied if you haven't assigned a limit for a user—it is used to limit the user from requesting too high a priority.

NOTE If the default priority is greater than the limit for a user, the limit is used instead.

If you do not set a default priority, the LP print service will use a default of 20.

Examining the Priority Limits and Defaults

You can examine all the settings you have assigned for priority limits and defaults by entering the following command.

```
lpusers -l
```

Moving a Request Around in the Queue

Once a user has submitted a print request, you can move it around in the queue to some degree:

- you can adjust the priority to any level, regardless of the limit for the user (who may adjust it only up to his or her limit)
- both you and the user can put it on hold and allow other requests to be printed ahead of it
- you can put it at the head of the queue for immediate printing

Use the `lp(1)` command to do any of these tasks.

Changing the Priority for a Request

If you want to change the priority of a particular request that is still waiting to be printed, you can assign a new priority level to it. By doing so, you can move it in the queue so that it is ahead of lower priority requests, and behind requests at the same level or of higher priority. The priority limit assigned to the user (or the default priority limit) has no effect because, as the administrator, you can override this limit.

Enter the following command to change the priority of a request.

```
lp -i request-ID -q new-priority-level
```

You can change only one request at a time with this command.

Putting a Request on Hold

Any request that has not finished printing can be put on hold. This will stop its printing, if it is currently printing, and keep it from printing until you resume it. A user may also put his or her own request on hold and then resume it, but may not resume a print request that has been put on hold by the administrator.

To place a request on hold, enter the following command:

```
lp -i request-ID -H hold
```

Enter the following command to resume the request:

```
lp -i request-ID -H resume
```

Once resumed a request will continue to move up the queue and will eventually be printed. If printing had already begun when you put it on hold, it will be the next request printed.

Moving a Request to the Head of the Queue

You can move a print request to the head of the queue where it will be the next one eligible for printing. If you want it to start printing immediately but another request is currently being printed, you may interrupt the first request by putting it on hold, as described above.

Enter the following command to move a print request to the head of the queue:

```
lp -i request-ID -H immediate
```

Only you, as the administrator, can move a request in this way; regular users cannot use the `-H immediate` option.

NOTE

If you set more than one request for immediate printing, the requests will be printed in the reverse order set; that is, the request moved to the head of the queue most recently will be printed first.

Starting and Stopping the LP Print Service

Under normal operation, you should never have to start or stop the LP print service manually. It is automatically started each time the UNIX system is started, and stopped each time the UNIX system is stopped. If, however, you need to stop the LP print service without stopping the UNIX system as well, you can do so by following the procedure described below.

Stopping the LP print service will cause all printing to cease within seconds. Any print requests that have not finished printing will be printed in their entirety after the LP print service is restarted. The printer configurations, forms, and filters in effect when the LP print service is stopped will be restored after it is restarted.

NOTE

To start and stop the LP print service manually, you must be logged in as either the user `lp` or the super-user (`root`).

Manually Stopping the Print Service

To stop the LP print service manually, enter the following command:

```
lpshut
```

The message

```
Print services stopped.
```

will appear, and all printing will cease within a few seconds. If you try to stop the LP print service when it is not running, you will see the message

```
Print services already stopped.
```

Manually Starting the Print Service

To restart the LP print service manually, enter the following command:

```
lpsched
```

The message

```
Print services started.
```

will appear. It may take a minute or two for the printer configurations, forms, and filters to be reestablished, before any saved print requests start printing. If you try to restart the LP print service when it is already running, you will see the message

```
Print services already active.
```

NOTE The LP print service does not have to be stopped to change printer configurations or to add forms or filters.

Directories and Files Used by the LP Print Service

This section lists the directories and files used by the LP print service. You can use this list to see if any files are missing or if the ownership or access permissions have changed. Normal operation of the LP print service should not cause any problems. However, if you do notice any discrepancies, there may be a security breach on your system.

At the end of this section is a description of the script used to clean out the request log (`/var/lp/logs/requests`) periodically. You may want to change this script to have the file cleaned out more or less frequently, or to condense the information into a report. See "Cleaning Out the Request Log" later in this section.

The various LP print service files and directories are found under the main directories listed below. These main directories should have the access permissions and ownerships shown.

Permissions	Owner	Group	Directory or File
drwxrwxr-x	lp	lp	/var/spool/lp
drwxrwxr-x	lp	lp	/var/lp
drwxrwxr-x	lp	lp	/etc/lp
drwxr-xr-x	root	other	/usr/lib/lp

You can check this by entering the following command:

```
ls -ld /var/spool/lp /var/lp /etc/lp /usr/lib/lp
```

Under these directories you should see only the files and directories shown in the table on the next few pages. You can generate a similar table for comparison by entering this command:

```
ls -lR /var/spool/lp /var/lp /etc/lp /usr/lib/lp
```

Permissions	Owner	Group	Directory or File
/var/spool/lp:			
-rw-rw-r--	lp	lp	SCHEDLOCK
drwxrwxr-x	lp	lp	admins
lrwxrwxrwx	lp	lp	bin -> /usr/lib/lp/bin
lrw-rw-r--	lp	lp	default -> /etc/lp/default
drwxrwxr-x	lp	lp	fifos
lrwxrwxr-x	lp	lp	logs -> /var/lp/logs
lrwxrwxr-x	lp	lp	model -> /usr/lib/lp/model
drwxrwxr-x	lp	lp	requests
drwxrwxr-x	lp	lp	system
lrwxrwxrwx	lp	lp	temp -> /var/spool/lp/tmp/sfsti
drwx--x--x	lp	lp	tmp
-rw-r--r--	lp	lp	users -> /etc/lp/users
/var/spool/lp/admins:			
lrwxrwxrwx	lp	lp	lp -> /etc/lp
/var/spool/lp/fifos:			
prw-rw-rw-	lp	lp	FIFO
prw-------	root	other	listenBSD
prw-------	root	other	listenS5
drwxrwx--x	lp	lp	private
drwxrwx-wx	lp	lp	public
/var/spool/lp/fifos/private:			
pr--------	*user*	*group*	*systemPID*
⋮			
/var/spool/lp/fifos/public:			
pr--------	*user*	*group*	*systemPID*
⋮			

Permissions	Owner	Group	Directory or File
/var/spool/lp/requests:			
drwxrwx---	lp	lp	*system1*
drwxrwx---	lp	lp	*system2*
⋮			
drwxrwx---	lp	lp	*systemN*
/var/spool/lp/requests/*systemK*:			
-rw-rw----	lp	lp	*id1*-0
-rw-rw----	lp	lp	*id2*-0
⋮			
-rw-rw----	lp	lp	*idN*-0
/var/spool/lp/system:			
-rw-rw-r--	lp	lp	cstatus
-rw-rw-r--	lp	lp	pstatus
/var/spool/lp/tmp:			
drwxrwxr-x	lp	lp	*system1*
drwxrwxr-x	lp	lp	*system2*
⋮			
drwxrwxr-x	lp	lp	*systemN*
/var/spool/lp/tmp/*systemK*:			
-rw-------	lp	lp	*idN*-0
-rw-------	lp	lp	*idN*-1
-rw-------	lp	lp	*idN*-2
⋮			
-rw-------	lp	lp	*idN*-*M*
-rw-------	lp	lp	F*idN*-1
-rw-------	lp	lp	F*idN*-2
⋮			

Permissions	Owner	Group	Directory or File
-rw-------	lp	lp	F*idN-M*
-rw-------	lp	lp	*idN*
-rw-------	lp	lp	A-*K*
-rw-------	lp	lp	F-*K*
-rw-------	lp	lp	P-*K*
/var/lp/logs:			
-rw-rw----	lp	lp	lpsched
-rw-rw----	lp	lp	requests
-rw-rw-r--	root	other	lpNetLog
-rw-rw-r--	root	other	p*PID*
-rw-rw-r--	root	other	c*PID*
/etc/lp:			
-rw-rw-r--	lp	lp	Systems
drwxrwxr-x	lp	lp	classes
-rw-rw-r--	lp	lp	filter.table
-rw-rw-r--	lp	lp	filter.table.i
drwxrwxr-x	lp	lp	forms
drwxrwxr-x	lp	lp	interfaces
lrwxrwxrwx	lp	lp	logs -> /var/lp/logs
drwxrwxr-x	lp	lp	printers
drwxrwxr-x	lp	lp	printwheels
-rw-rw-r--	lp	lp	users
/etc/lp/classes:			
-rw-rw-r--	lp	lp	*class1*
-rw-rw-r--	lp	lp	*class2*
⋮			
-rw-rw-r--	lp	lp	*classN*
/etc/lp/forms:			
drwxrwxr-x	lp	lp	*form1*

Permissions	Owner	Group	Directory or File
drwxrwxr-x	lp	lp	*form2*
⋮			
drwxrwxr-x	lp	lp	*formN*
/etc/lp/forms/*formK*:			
-rwxrwx---	lp	lp	alert.sh
-rw-rw----	lp	lp	alert.vars
-rw-rw----	lp	lp	align_ptrn
-rw-rw-r--	lp	lp	allow
-rw-rw-r--	lp	lp	comment
-rw-rw-r--	lp	lp	deny
-rw-rw-r--	lp	lp	describe
/etc/lp/interfaces:			
-rwxrwxr-x	lp	lp	*printer1*
-rwxrwxr-x	lp	lp	*printer2*
⋮			
-rwxrwxr-x	lp	lp	*printerN*
/etc/lp/printers:			
drwxrwxr-x	lp	lp	*printer1*
drwxrwxr-x	lp	lp	*printer2*
⋮			
drwxrwxr-x	lp	lp	*printerN*
/etc/lp/printers/*printerK*:			
-rwxrwx---	lp	lp	alert.sh
-rw-rw----	lp	lp	alert.vars
-rw-rw-r--	lp	lp	comment
-rw-rw-r--	lp	lp	configuration
-rw-rw-r--	lp	lp	forms.allow

Permissions	Owner	Group	Directory or File
`-rw-rw-r--`	lp	lp	`forms.deny`
`-rw-rw-r--`	lp	lp	`residentfonts`
`-rw-rw-r--`	lp	lp	`users.allow`
`-rw-rw-r--`	lp	lp	`users.deny`
`/etc/lp/pwheels:`			
`drwxrwxr-x`	lp	lp	*printwheel1*
`drwxrwxr-x`	lp	lp	*printwheel2*
⋮			
`drwxrwxr-x`	lp	lp	*printwheelN*
`/etc/lp/pwheels`/*printwheelK*:			
`-rwxrwx---`	lp	lp	`alert.sh`
`-rw-rw----`	lp	lp	`alert.vars`
`/usr/lib/lp:`			
`dr-xr-xr-x`	lp	lp	`bin`
`---x--x--x`	lp	lp	`lpNet`
`---x--x---`	lp	lp	`lpdata`
`---s--x--x`	root	lp	`lpsched`
`drwxrwxr-x`	lp	lp	`model`
`/usr/lib/lp/bin:`			
`-r--r--r--`	lp	lp	`alert.proto`
`-r-xr-xr-x`	lp	lp	`drain.output`
`-r-xr-xr-x`	lp	lp	`lp.cat`
`-r-xr-xr-x`	lp	lp	`lp.set`
`-r-xr-xr-x`	lp	lp	`lp.tell`
`-r-xr-xr-x`	lp	lp	`slow.filter`
`/usr/lib/lp/model:`			
`-rwxrwxr-x`	lp	lp	`standard`

The italicized names—*printerN*, *formN*, *classN*, *printwheelN*, *idN*, and *systemN*—are placeholders for a single printer, form, class, print wheel, request ID, and UNIX system name, respectively (*idN* is just the numeric part of the request ID.) There will be one set of these directories and files for each active printer, form, class, print wheel and request configured on your system, and for each system used for remote printing. The italicized letter *K* is a placeholder for an internal number; the `A-`*K*, `F-`*K*, and `P-2`*K*, files are used to store alert messages.

The ownership and permissions of the *id*N-*M* request files under the `/var/spool/lp/tmp/`*system* directory will change during the life of a print request, alternating between the user who submitted the request and the `lp` ID.

The two directories under the `/var/spool/lp/fifos` directory contain named pipes used to communicate between the LP print service and commands such as `lpadmin`, `lpstat`, and `lp`. These directories must have the permission flags and ownership shown if communication with the LP print service is to work. Every entry below these directories is given a unique name formed by combining the name of the system (the node name) and the process ID of the command. The uniqueness of the entry names prevents two or more people from accidentally sharing the same communications path.

Cleaning Out the Request Log

The directories `/var/spool/lp/tmp/`*system* and `/var/spool/lp/requests/`*system* contain files that describe each request that has been submitted to the LP print service. Each request has two files (one in each directory) that contain information about the request. The information is split to put more sensitive information in the `/var/spool/lp/requests/`*system* directory where it can be kept secure: the request file in the `/var/spool/lp/tmp/`*system* is safe from all except the user who submitted the request, while the file in `/var/spool/lp/requests/`*system* is safe from all users, including the submitting user.

These files remain in their directories only as long as the request is in the queue. Once the request is finished, the information in the files is combined and appended to the file `/var/lp/logs/requests`. This file is not removed by the LP print service, but can be cleaned out periodically, using, for instance, the `cron` facility. (See the description of the `crontab` command in the *User's Reference Manual.*)

The default `crontab` entry provided with the LP print service is shown below.

```
13 3 * * * cd /var/lp/logs; if [ -f requests ]; then
  /usr/bin/mv requests xyzzy; /usr/bin/cp xyzzy requests; >xyzzy;
  /usr/lbin/agefile -c2 requests; /usr/bin/mv xyzzy requests; fi
```

(This is one line in the crontab but is split into several lines here for readability.) What this entry does, briefly, is "age" the file, changing the name to requests-1, and moving the previous day's copy to requests-2. The number 2 in the -c option to the agefile program keeps the log files from the previous two days, discarding older log files. By changing this number you can change the amount of information saved. On the other hand, if you want the information to be saved more often, or if you want the file to be cleaned out more often than once a day, you can change the time when the crontab entry is run by changing the first two numbers. The current values, 13 and 3, cause cleaning up to be done at 3:13 A.M. each day.

The default crontab entry supplied is sufficient to keep the old print request records from accumulating in the spooling file system. You may want to condense information in the request log to produce a report on the use of the LP print service, or to aid in generating accounting information. You can produce a different script that examines the file and extracts information just before the clean up procedure.

The request log has a simple structure that makes it easy to extract data from it using common UNIX shell commands. Requests are listed in the order they are printed, and are separated by lines showing their request IDs. Each line below the separator line is marked with a single letter that identifies the kind of information contained in that line. Each letter is separated from the data by a single space. See the following table for details.

Letter	Content of line
=	This is the separator line. It contains the request ID, the user and group IDs of the user, the total number of bytes in the original (unfiltered) files, and the time when the request was queued. These items are separated by commas and are in the order just named. The user ID, group

Letter	Content of line
	ID, and sizes are preceded by the words uid, gid, and size, respectively.
C	The number of copies printed.
D	The printer or class destination or the word any.
F	The name of the file printed. This line is repeated for each file printed; files were printed in the order given.
f	The name of the form used.
H	One of three types of special handling: resume, hold, and immediate. The only useful value found in this line will be immediate.
N	The type of alert used when the print request was successfully completed. The type is the letter M if the user was notified by mail, or W if the user was notified by a message to his or her terminal.
O	The -o options.
P	The priority of the print request.
p	The list of pages printed.
r	This single letter line is included if the user asked for "raw" processing of the files (the -r option of the lp command).
S	The character set or print wheel used.
s	The outcome of the request, shown as a combination of individual bits expressed in hexadecimal form. While several bits are used internally by the print service, the most important bits are listed below: 0x0004 Slow filtering finished successfully. 0x0010 Printing finished successfully.

Letter	Content of line
	`0x0040` The request was canceled. `0x0100` The request failed filtering or printing.
`T`	The title placed on the banner page.
`t`	The type of content found in the file(s).
`U`	The name of the user who submitted the print request.
`x`	The slow filter used for the request.
`Y`	The list of special modes to give to the filters used to print the request.
`y`	The fast filter used for the request.
`z`	The printer used for the request. This will differ from the destination (the `D` line) if the request was queued for any printer or a class of printers, or if the request was moved to another destination by the LP print service administrator.

PostScript Printers

PostScript is a general purpose programming language, like C or Pascal. In addition to providing the usual features of a language, however, PostScript allows a programmer to specify the appearance of both text and graphics on a page.

A PostScript printer is a printer equipped with a computer that runs an interpreter for processing PostScript language files. When a PostScript printer receives a file, it runs that file through the interpreter and then prints it. Unless special provisions have been made by the manufacturer, files submitted to a PostScript printer must be written in the PostScript language.

Why would you want to use a PostScript printer? PostScript provides excellent facilities for managing text and graphics and combining them. Graphics operators facilitate the construction of geometric figures which can then be positioned and scaled with any orientation. The text capabilities allow the user to specify a number of different fonts that can be placed on a page in any position, size, or orientation. Because text is treated as graphics, text and graphics are readily combined. Moreover, the language is resolution and device independent, so that draft copies can be proofed on a low-resolution device and the final version printed in higher resolution on a different device.

Applications that support PostScript, including word-processing and publishing software, will create documents in the PostScript language without intervention by the user. Thus, it is not necessary to know the details of the language to take advantage of its features. However, standard files that many applications produce cannot be printed on a PostScript printer because they are not described in the language. The LP print service provides optional filters to convert many of these files to PostScript so that users may take advantage of PostScript and continue to use their standard applications, such as `troff`.

How to Use a PostScript Printer

When the PostScript printers and filters have been installed, LP manages PostScript files like any others. If `psfile` is a file containing a PostScript document and psprinter has been defined to LP as a PostScript printer, the command

```
lp -d psprinter -T postscript psfile
```

will schedule the print request and manage the transmission of the request to the PostScript printer.

Support of Non-PostScript Print Requests

Because PostScript is a language and PostScript printers are expecting print requests written in that language, some applications may produce standard print requests that may not be intelligible to PostScript printers. The following are examples of print requests that may not be interpreted by some PostScript printers.

Content Type	Type of Print Request
`troff`	Print output from the `troff` command.
`simple`	Print an ASCII ("simple") text file.
`dmd`	Print the contents of a bit-mapped display from a terminal such as an AT&T 630.
`tek4014`	Print files formatted for a Tektronix 4014 device.
`daisy`	Print files intended for a Diablo 630 ("daisy-wheel") printer.
`plot`	Print plot-formatted files.

Filters are provided with the LP print service to translate print requests with these formats to the PostScript language. For example, to convert a file containing ASCII text to PostScript code, the filter takes that text and writes a program around it, specifying printing parameters such as fonts and the layout of the text on a page.

Once the PostScript filters are installed, they will be invoked automatically by the LP print service when a user specifies a content-type for a print request with the `-T` option. For example, if a user enters the command

```
lp -d psprinter -T simple report2
```

the ASCII file `report2` (a file with an ASCII or "simple" format) will be converted to PostScript automatically, as long as the destination printer (`psprinter`) has been defined to the system as a PostScript printer.

Additional PostScript Capabilities Provided by Filters

The filters previously described also take advantage of PostScript capabilities to provide additional printing flexibility. Most of these features may be accessed through the "mode option" (invoked by the `-y` option) to the `lp` command. These filters allow you to use several unusual options for your print jobs. The following list describes these options and shows the option you should include on the `lp` command line for each one.

`-y reverse` Reverse the order in which pages are printed.

`-y landscape` Change the orientation of a physical page from portrait to landscape.

`-y x=`*number*`, y=`*number*
Change the default position of a logical page on a physical page by moving the origin.

`-y group=`*number*
Group multiple logical pages on a single physical page.

`-y magnify=`*number*
Change the logical size of each page in a document.

`-o length=`*number*
Select the number of lines in each page of the document.

`-P` *number* Select, by page numbers, a subset of a document to be printed.

`-n` *number* Print multiple copies of a document.

NOTE

If these filters are to be used with an application that creates PostScript output, make sure that the format of the application conforms to the format of the PostScript file structuring comments. In particular, the beginning of each PostScript page must be marked by the comment

`%%Page:` *label ordinal*

where *ordinal* is a positive integer that specifies the position of the page in the sequence of pages in the document, and *label* is an arbitrary page label.

For example, say you have a file called `report2` that has a content type `simple` (meaning that the content of this file is in ASCII format). You want to print six pages of this file (pages 4-9) with two logical pages on each physical page. Because one of the printers on your system (`psprinter`) is a PostScript printer, you can do this by entering the following command:

```
lp -d psprinter -T simple -P 4-9 -y group=2 myfile
```

The filter which groups these logical pages will try to position the pages on the physical page to maximize space utilization. Thus when you specify `group=2`, the pages will be printed side by side, so that the physical page will be landscape orientation. Landscape mode, which controls the orientation of the logical page rather than the physical page, would cause the logical pages to be positioned one on top of the other when combined with the `group=2` option.

The Administrator's Duties

Support of PostScript printers is similar to support of other printers, in that the printers must be defined to the system with the `lpadmin` command and the appropriate software must be installed to manage them. PostScript printers may require some additional effort in supporting fonts.

Installing and Maintaining PostScript Printers

PostScript printers, like other printers, are installed with the `lpadmin` command.

NOTE

The printer-type and content-type of a PostScript printer must be consistent with the printer type used in PostScript filters. Therefore you should install your PostScript printers with a printer-type of PS or PSR, and a content-type of PS.

The printer types PS and PSR serve two functions. First, they cause LP to activate the `postio` filter to communicate with the printer. Second, the standard interface shell creates a PostScript banner page for printers with printer type PS or PSR. The banner page is printed last if the printer-type is PSR, and the pages of the document are printed in reverse order. The printer type is specified with the `-T` option in the `lpadmin` command.

As part of the installation, you may want to install fonts on the printer or downloadable fonts on the computer. See "Installing and Maintaining PostScript Fonts" later in this chapter for details.

Installing and Maintaining PostScript Filters

PostScript filters are provided with UNIX System V/68 or V/88 Release 4 and are installed during regular installation. This installation covers the majority of situations. In certain circumstances, however, you may find it helpful to change the filter descriptions and install the filters differently. To help you do this, this section provides describes the location and function of these filters.

PostScript filters are contained in the directory

```
/usr/lib/lp/postscript
```

NOTE There are two types of filters: fast filters and slow filters. For definitions of these types, see `lpfilter`(1M) in the *System Administrator's Reference Manual* and "Defining a Filter" earlier in this chapter.

A prerequisite of communication between any system and a PostScript printer is the presence of the `postio` filter on the system. This program is the only mandatory PostScript filter that communicates directly with the PostScript printer. The following filters allow other types of documents to be translated to PostScript and to be printed on a PostScript printer.

File Content Type	Filter
`simple`	`postprint`
`troff`	`dpost`
`daisy`	`postdaisy`
dmd(AT&T 630)	`postdmd`
`tek4014`	`posttek`
`plot`	`postplot`

The following filters perform special functions:

Function	Filter
Communicate with printer	`postio`
Download fonts	`download`
Reverse or select pages	`postreverse`
Matrix gray scales	`postmd`

Installing and Maintaining PostScript Fonts

One of the advantages of PostScript is its ability to manage fonts. Fonts are stored in outline form, either on the printer or on a computer that communicates with a printer. When a document is printed, the PostScript interpreter generates each character as needed (in the appropriate size) from the outline description of it. If a font required for a document is not stored on the printer being used, it must be transmitted to that printer before the document can be printed. This transmission process is called "downloading fonts."

Fonts are stored and accessed in several ways.

- Fonts may be stored permanently on a printer. These "printer resident" fonts may be installed in ROM on the printer by the manufacturer. If the printer has a disk, fonts may be installed on that disk by you (that is, by the print service administrator). Most PostScript printers are shipped with thirty-five standard fonts.
- A font may be "permanently-downloaded" by being transmitted to a printer with a PostScript "exitserver" program. A font downloaded in this way will remain in the printer's memory until the printer is turned off. Memory allocated to this font will reduce the memory available for PostScript print requests. Use of exitserver programs requires the printer system password and may be reserved for the printer administrator. This method is useful when there is continual use of a font by the majority of print requests serviced by that printer.
- Fonts may be prepended to a user's print request by the user, and be transmitted as part of the user's print request. When the user's document has been printed, the space allocated to the font is freed for other print requests. The font is stored in the user's directory. This method is preferable for fonts with more limited usage.

- Fonts may be stored on a system shared by many users. These fonts may be described as "host-resident." This system may be a server for the printer or may be a system connected to the printer by a network. Each user may request fonts in the document to be printed. This method is useful when there are a large number of available fonts or when there is not continual use of these fonts by all print requests. If the fonts will be used only on printers attached to a server, they should be stored on the server. If the fonts are to be used by users on one system, who may send jobs to multiple printers on a network, they may be stored on the users' system.

The LP print service allows you to manage fonts in any of these ways. It provides a special download filter to manage fonts using the last method described above.

The LP print service supplies `troff` width tables for the thirty-five standard PostScript fonts which reside on many PostScript printers, for use by the `troff` program.

Managing Printer-Resident Fonts

Most PostScript printers come equipped with fonts resident in the printer ROM. Some printers have a disk on which additional fonts are stored. When a printer is installed, the list of printer-resident fonts should be added to the font-list for that printer. These lists are kept in the printer administration directories. For a particular printer, this list is contained in the file

`/etc/lp/printers/`*printer-name*`/residentfonts`

where *printer-name* is the name of the printer. When fonts are permanently downloaded to the printer, the font names should be added to this file. This will prevent fonts from being downloaded when they are already on the printer, a time-consuming procedure. If the printer is attached to a remote system, this list should include fonts which reside on that system and are available for downloading to the printer. This prevents fonts from being transmitted unnecessarily across a network. These files must be edited manually; that is, with the help of a text editor such as `vi`.

Installing and Maintaining Host-Resident Fonts

Some fonts will be resident on the host and transmitted to the printer as needed for particular print requests. As the administrator, it's your job to make PostScript fonts available to all the users on a system. To do so, you must know how and where to install these fonts, using the guidelines described previously. Because fonts are requested by name and stored in files, the LP print service keeps a map file that shows the correspondence between the names of fonts and the names of the files containing those fonts. Both of these must be updated when fonts are installed on the host.

Install host-resident PostScript fonts by doing the following:

- Copy the font file to the appropriate directory.
- Add to the map table the name of the font and the name of the file in which it resides.
- If you are using `troff`, you must create new width tables for this font in the standard `troff` font directory.

Where Are Fonts Stored?

The fonts available for use with PostScript printers reside in directories called `/usr/share/lib/hostfontdir/`*typeface*`/`*font* where *typeface* is replaced by a name such as `palatino` or `helvetica`, and *font* is replaced by a name such as `bold` or `italic`.

Adding an Entry to the Map Table

Also within the `hostfontdir` directory, you (the administrator) must create and maintain a map table that shows the correspondence between the name assigned to each font by the foundry (the company that created the font) and the name of the file in which that font resides. For example, to map the font called "Palatino Bold," add the following line to the map table:

```
Palatino-Bold /usr/share/lib/hostfontdir/palatino/bold
```

(The map table itself is in the file `/usr/share/lib/hostfontdir/map`).

Once this entry exists in the map table on your system, your users will be able to have a Palatino Bold font used in their print jobs. When they submit, for printing, a file containing a request for this font, the LP print service will prepend a copy of the file

```
/usr/share/lib/hostfontdir/palatino/bold
```

to that file before sending it to the printer.

Downloading Host-Resident Fonts

The creators of the PostScript language anticipated that users would want to download fonts to printers. The *PostScript Language Reference Manual* (by Adobe Systems, Inc., Addison-Wesley Publishing Co., Inc., 1985) states the following:

> "...programs that manage previously generated PostScript page descriptions, such as 'printer spooler' utilities, may require additional information about those page descriptions. For example, if a page description references special fonts, a spooler may need to transmit definitions of those fonts to the PostScript printer ahead of the page description itself.
>
> To facilitate these and other operations, [PostScript] defines a standard set of *structuring conventions* for PostScript programs."

The download filter relies on these structuring conventions to determine which fonts must be downloaded.

When the LP PostScript document contains a request for fonts not loaded on the printer, the download filter manages this request. This filter is invoked as a "fast filter"; it downloads fonts automatically if the fonts reside on the same system as the printer. The download filter may also be used to send fonts to a remote printer. To do this, you may create a new filter table entry which calls the download filter as a "slow" filter through the -y option. Alternatively, you may force selection of this filter by changing the input-type.

The download filter does five things:

- It searches the PostScript document to determine which fonts have been requested. These requests are documented with the following PostScript structuring comments:

  %%DocumentFonts: *font1 font2* ...

  in the header comments.

- It searches the list of fonts resident on that printer to see if the requested font must be downloaded.
- If the font is not resident on the printer, it searches the host-resident font directory (by getting the appropriate file name from the map table) to see if the requested font is available.
- If the font is available, the filter takes the file for that font and prepends it to the file to be printed.
- The filter sends the font definition file and the source file (the file to be printed) to the PostScript printer.

Customizing the Print Service

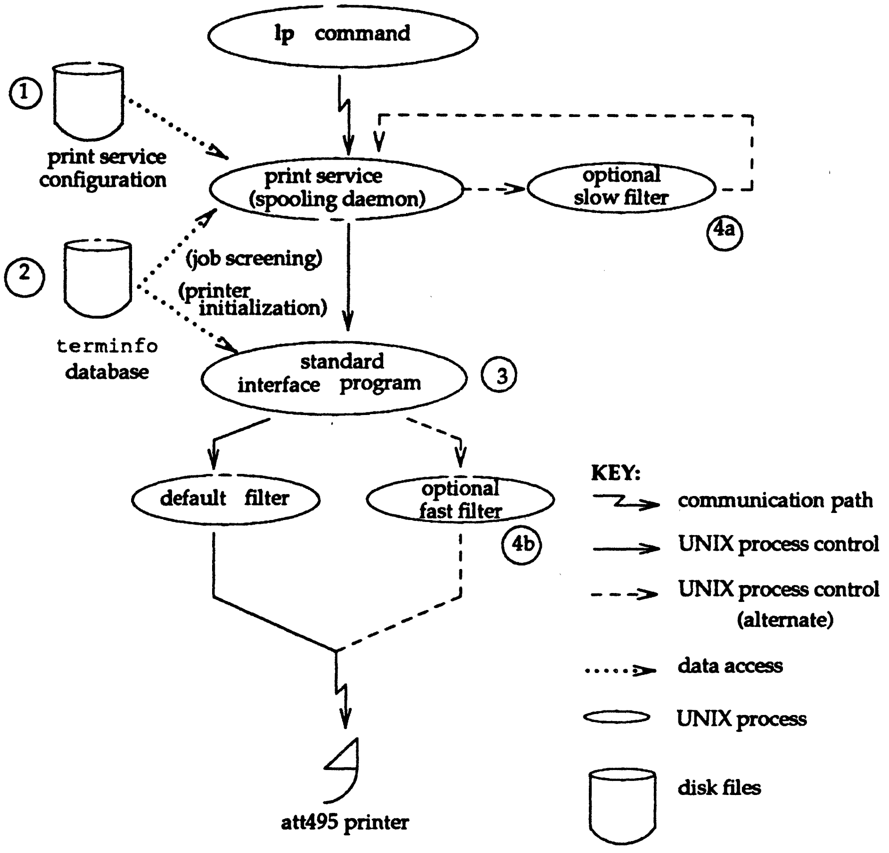
Although the LP print service has been designed to be flexible enough to handle most printers and printing needs, it doesn't handle every possible situation. You may buy a printer that doesn't quite fit into the way the LP print service handles printers, or you may have a printing need that the standard features of the LP print service don't accommodate.

You can customize the LP print service in a few ways. This section tells you how you can

- adjust the printer port characteristics,
- adjust the `terminfo` database,
- write an interface program, and
- write a filter.

The diagram in Figure 11-6 gives an overview of the processing of a print request.

Figure 11-6: How LP Processes Print Request `lp -d att495` *file*



Each print request is sent to a "spooling daemon" that keeps track of all requests. The daemon is created when you start the LP print service. This UNIX system process is also responsible for keeping track of the status of printers and slow filters; when a printer finishes printing a user's file, the daemon starts it printing another request (if there is one queued).

To customize the print service, adjust or replace some of the pieces shown in Figure 11-6. (The numbers are keyed to the diagram.)

1. For most printers, you need only change the printer configuration stored on disk. The earlier sections of this chapter explain how to do this. Configuration data that are relatively dependent on the printer include the printer port characteristics: baud rate, parity, and so on.

2. For a printer that is not represented in the terminfo database, you can add a new entry that describes its capabilities. The terminfo database is used in two parallel capacities: screening print requests to ensure that those accepted can be handled by the desired printer, and setting the printer in a state where it is ready to print a request.

   For instance, if the terminfo database does not contain an entry for a printer capable of setting a page length requested by a user, the spooling daemon will reject the request. On the other hand, if it does contain an entry for such a printer, then the same information will be used by the interface program to initialize the printer.

3. For particularly difficult printers, or if you want to add features not provided by the delivered LP print service, you can change the standard interface program. This program is responsible for managing the printer: it prints the banner page, initializes the printer, and invokes a filter to send copies of a user's files to the printer.

4a. and 4b.
To provide a link between the applications used on your system and the printers, you can add slow and fast filters. Each type of filter can convert a file into another form, mapping one set of escape sequences into another, for instance, and can provide special setup by interpreting print modes requested by a user. Slow filters are run separately by the daemon, to avoid tying up a printer. Fast filters are run so their output goes directly to the printer; thus they can exert control over the printer.

Adjusting the Printer Port Characteristics

You should make sure that the printer port characteristics set by the LP print service match the printer communication settings. The standard printer port settings have been designed to work with typical files and many printers, but they won't work with all files and printers. This isn't really a customizing step, because a standard feature of the LP print service is to allow you to specify the port settings for each printer. However, it's an important step in getting your printer to work with the LP print service, so it's described in more detail here.

When you add a new printer, read the documentation that comes with it so that you understand what it expects from the host (the LP print service). Then read the manual page for the `stty(1)` command in the *User's Reference Manual*. It summarizes the various characteristics that can be set on a terminal or printer port.

Only some of the characteristics listed in the `stty(1)` manual page are important for printers. The ones likely to be of interest to you are listed below (but you should still consult the `stty(1)` manual page for others).

`stty` Option	Meaning
`evenp`	Send even parity in the 8th bit
`oddp`	Send odd parity in the 8th bit
`-parity`	Don't generate parity; send all 8 bits unchanged
`110 - 38400`	Set the communications speed to this baud rate
`ixon`	Enable XON/XOFF (also known as START/STOP or DC1/DC3) flow control
`-ixon`	Turn off XON/XOFF flow control
`-opost`	Don't do any "output post-processing"
`opost`	Do "output post-processing" according to the settings listed below
`onlcr`	Send a carriage return before every linefeed

`-onlcr`	Don't send a carriage return before every linefeed
`ocrnl`	Change carriage returns into linefeeds
`-ocrnl`	Don't change carriage returns into linefeeds
`-tabs`	Change tabs into an equivalent number of spaces
`tabs`	Don't change tabs into spaces

When you have a set of printer port characteristics you think should apply, adjust the printer configuration as described in the section "How to Define Printer Ports and Printer Port Characteristics" under "Printer Management" in this chapter. You may find that the default settings are sufficient for your printer.

Adjusting the `terminfo` Database

The LP print service relies on a standard interface and the `terminfo` database to initialize each printer and establish a selected page size, character pitch, line pitch, and character set. Thus, it is usually sufficient to have the correct entry in the `terminfo` database to add a new printer to the LP print service. Several entries for popular printers are delivered in the standard `terminfo` database.

Each printer is identified in the `terminfo` database with a short name; this kind of name is identical to the kind of name used to set the TERM shell variable. For instance, the AT&T model 455 printer is identified by the name `455`. The "Acceptable Terminal Names" section of Appendix G ("Setting Up the Terminal") in the *User's Guide* describes how to determine a correct TERM variable for a user's terminal; you can use it as a guide for picking a known name for your printer.

If you cannot find a `terminfo` entry for your printer, you should add one. If you don't, you may still be able to use the printer with the LP print service but you won't have the option of automatic selection of page size, pitch, and character sets, and you may have trouble keeping the printer set in the correct modes for each print request. Another option to follow, instead of updating the `terminfo` entry, is to customize the interface program used with the printer. (See the next section for details on how to do this.)

There are hundreds of items that can be defined for each terminal or printer in the `terminfo` database. However, the LP print service uses fewer than 50 of these. The following table lists the items that need to be defined (as appropriate for the printer) to add a new printer to the LP print service.

`terminfo` item	Meaning
Booleans:	
`cpix`	Changing character pitch changes resolution
`daisy`	Printer needs operator to change character set
`lpix`	Changing line pitch changes resolution
Numbers:	
`bufsz`	Number of bytes buffered before printing
`cols`	Number of columns in a line
`cps`	Average print rate in characters per second
`it`	Tabs initially every # spaces
`lines`	Number of lines on a page
`orc`	Horizontal resolution in units per character
`orhi`	Horizontal resolution in units per inch
`orl`	Vertical resolution in units per line
`orvi`	Vertical resolution in units per inch
Strings:	
`chr`	Change horizontal resolution
`cpi`	Change number of characters per inch
`cr`	Carriage return
`csnm`	List of character set names
`cud1`	Down one line
`cud`	Move carriage down # lines
`cuf`	Move carriage right # columns
`cuf1`	Carriage right
`cvr`	Change vertical resolution

terminfo item	Meaning
ff	Page eject
hpa	Horizontal position absolute
ht	Tab to next 8-space tab stop
if	Name of initialization file
iprog	Path name of initializing program
is1	Printer initialization string
is2	Printer initialization string
is3	Printer initialization string
lpi	Change number of lines per inch
mgc	Clear all margins (top, bottom, and sides)
rep	Repeat a character # times
rwidm	Disable double wide printing
scs	Select character set
scsd	Start definition of a character set
slines	Set page length to # lines per page
smgl	Set left margin at current column
smglp	Set left margin
smgr	Set right margin at current column
smgrp	Set right margin
smglr	Set both left and right margins
smgt	Set top margin at current line
smgtp	Set top margin
smgb	Set bottom margin at current line
smgbp	Set bottom margin
smgtb	Set both top and bottom margins
swidm	Enable double wide printing
vpa	Vertical position absolute

To construct a database entry for a new printer, see details about the structure of the terminfo database in the terminfo(4) manual page in the *Programmer's Reference Manual*.

Once you've made the new entry, you need to compile it into the database using the tic(1) program (available in the Terminal Information Utilities). Just enter the following command:

```
tic file-name
```

file-name is the name of the file containing the terminfo entry you have crafted for the new printer.

NOTE The LP print service gains much efficiency by caching information from the terminfo database. If you add or delete terminfo entries, or change the values that govern pitch settings, page width and length, or character sets, you should stop and restart the LP print service so it can read the new information.

How to Modify the Interface Program

NOTE If you have an interface program that you have used with the LP Spooling Utilities before UNIX System V Release 3.2, it should still work with the LP print service. Note, though, that several -o options have been "standardized," and will be passed to every interface program. These may interfere with similarly named options used by your interface.

If you have a printer that is not supported by simply adding an entry to the terminfo database, or if you have printing needs that are not supported by the standard interface program, you can furnish your own interface program. It is a good idea to start with the standard interface program, and change it to fit, rather than starting from scratch. You can find a copy of it under the name

```
/var/spool/lp/model/standard
```

What Does the Interface Program Do?

Any interface program is responsible for doing the following tasks:

- Initializing the printer port, if necessary. The generic interface program uses the stty command to do this.
- Initializing the physical printer. The generic interface program uses the terminfo database and the TERM shell variable to get the control sequences to do this.
- Printing a banner page, if necessary.
- Printing the correct number of copies of the request content.

An interface program is not responsible for opening the printer port. The LP print service opens the port, a process which includes calling a "dial-up" printer, if one is used to connect the printer. The printer port connection is given to the interface program as standard output, and the printer is identified as the "controlling terminal" for the interface program so that a "hang-up" of the port will cause a `SIGHUP` signal to be sent to the interface program.

A customized interface program must not terminate the connection to the printer or "uninitialize" the printer in any way.

How Is the Interface Program Used?

When the LP print service routes an output request to a printer, the interface program for the printer is invoked as follows:

```
/var/spool/lp/admins/lp/interface/P ID user title copies \
        options file1 file2 ...
```

Arguments for the interface program are:

P	printer name
id	request ID returned by the `lp(1)` command
user	logname of the user who made the request
title	optional title specified by the user
copies	number of copies requested by the user
options	blank-separated list of options specified by the user or set by the LP print service
file$_1$, *file*$_2$, ...	full pathnames of the files to be printed

When the interface program is invoked, its standard input comes from `/dev/null`, its standard output is directed to the printer port, and its standard error output is directed to a file that will be given to the user who submitted the print request.

The standard interface recognizes the following values in the blank-separated list in *options*.

`nobanner` This option is used to skip the printing of a banner page; without it, a banner page is printed.

`nofilebreak` This option is used to skip page breaks between separate data files; without it, a page break is made between each file in the content of a print request.

`cpi=`*decimal-number*$_1$
`lpi=`*decimal-number*$_2$

These options specify a format of *decimal-number*$_1$ columns per inch and *decimal-number*$_2$ lines per inch, respectively. The standard interface program extracts from the `terminfo` database the control sequences needed to initialize the printer to handle the character and line pitches.

The words `pica`, `elite`, and `compressed` are acceptable replacements for *decimal-number*$_1$ and are synonyms, respectively, for `10` columns per inch, `12` columns per inch, and as many columns per inch as possible.

`length=`*decimal-number*$_1$
`width=`*decimal-number*$_2$

These options specify the length and width, respectively, of the pages to be printed. The standard interface program extracts from the `terminfo` database the control sequences needed to initialize the printer to handle the page length and page width.

`stty='`*stty-option-list*'

The *stty-option-list* is applied after a default *stty-option-list* as a set of arguments to the `stty` command. The default list is used to establish a default port configuration; the additional list given to the interface program is used to change the configuration as needed.

`lpd='`*argument-list*'

This option is used internally by the `lpsched` command; you can ignore it.

`flist='`*file-list*'

This option is used internally by the `lpsched` command; you can ignore it.

The above options may be specified by the user when issuing a print request. Alternatively, they may be specified by the LP print service from defaults given by the administrator either for the printer (cpi, lpi, length, width, stty) or for the preprinted form used in the request (cpi, lpi, length, width).

Additional printer configuration information is passed to the interface program in the following shell variables:

TERM=*printer-type*
: This shell variable specifies the type of printer. The value is used as a key for getting printer capability information from the terminfo database.

FILTER=*'pipeline'*
: This shell variable specifies the filter to use to send the request content to the printer; the filter is given control of the printer.

CHARSET=*character-set*
: This shell variable specifies the character set to be used when printing the content of a print request. The standard interface program extracts from the terminfo database the control sequences needed to select the character set.

A customized interface program should either ignore these options and shell variables or should recognize them and treat them in a consistent manner.

Customizing the Interface Program

Make sure that the custom interface program sets the proper stty modes (terminal characteristics such as baud rate and output options). The standard interface program does this, and you can follow suit. Look for the section that begins with the shell comment

```
## Initialize the printer port
```

Follow the code used in the standard interface program. It sets both the default modes and the adjusted modes given by either the LP print service or the user with a line such as the following:

```
stty mode options 0<&1
```

This command line takes the standard input for the stty command from the printer port. An example of an `stty` command line that sets the baud rate at 1200 and sets some of the option modes is shown below.

```
stty -parenb -parodd 1200 cs8 cread clocal ixon 0<&1
```

One printer port characteristic not set by the standard interface program is hardware flow control. The way that this is set will vary, depending on your computer hardware. The code for the standard interface program suggests where this and other printer port characteristics can be set. Look for the section that begins with the shell comment

```
# Here you may want to add other port initialization
code.
```

Because different printers have different numbers of columns, make sure the header and trailer for your interface program correspond to your printer. The standard interface program prints a banner that fits on an 80-column page (except for the user's title, which may be longer). Look in the code for the standard interface program for the section that begins with the shell comment

```
## Print the banner page
```

The custom interface program should print all user related error messages on the standard output or on the standard error. The messages sent to the standard error will be mailed to the user; the messages printed on the standard output will end up on the printed page where they can be read by the user when he or she picks up the output.

When printing is complete, your interface program should exit with a code that shows the status of the print job. Exit codes are interpreted by the LP print service as follows:

Code	Meaning to the LP print service
0	The print request has been completed successfully. If a printer fault has occurred, it has been cleared.
1 to 127	A problem has been encountered in printing this particular request (for example, too many non-printable characters, or the request exceeds the printer capabilities). The

	LP print service notifies the person who submitted the request that there was an error in printing it. This problem will not affect future print requests. If a printer fault had occurred, it has been cleared.
128	Reserved for internal use by the LP print service. Interface programs must not exit with this code.
129	A printer fault has been encountered in printing the request. This problem will affect future print requests. If the fault recovery for the printer directs the LP print service to wait for the administrator to fix the problem, the LP print service will disable the printer. If the fault recovery is to continue printing, the LP print service will not disable the printer, but will try printing again in a few minutes.
greater than 129	These codes are reserved for internal use by the LP print service. Interface programs must not exit with codes in this range.

As the table shows, one way of alerting the administrator to a printer fault is to exit with a code of 129. Unfortunately, if the interface program exits, the LP print service has no choice but to reprint the request from the beginning when the fault has been cleared. Another way of getting an alert to the administrator (that does not require the entire request to be reprinted) is to have the interface program send a fault message to the LP print service but wait for the fault to clear. When the fault clears, the interface program can resume printing the user's file. When the printing is finished, the interface program can give a zero exit code just as if the fault had never occurred. An added advantage is that the interface program can detect when the fault is cleared automatically, so that the administrator doesn't have to enable the printer.

Fault messages can be sent to the LP print service using the `lp.tell` program. This is referenced using the `$LPTELL` shell variable in the standard interface code. The program takes its standard input and sends it to the LP print service where it is put into the message that alerts the administrator to the printer fault. If its standard input is empty, `lp.tell` does not initiate an alert. Examine the standard interface code immediately after these comments for an example of how the `lp.tell` (`$LPTELL`) program is used:

```
# Here's where we set up the $LPTELL program to capture
# fault messages.

# Here's where we print the file.
```

If the special exit code 129 or the `lp.tell` program is used, there is no longer a need for the interface program to disable the printer itself. Your interface program can disable the printer directly, but doing so will override the fault alerting mechanism. Alerts are sent only if the LP print service detects the printer has faulted, and the special exit code and the `lp.tell` program are its main detection tools.

If the LP print service has to interrupt the printing of a file at any time, it will "kill" the interface program with a signal `TERM` (trap number 15; see `kill`(1) and `signal`(2) in the *User's Reference Manual* and *Programmer's Reference Manual*, respectively). If the interface program dies from receipt of any other signal, the LP print service assumes that future print requests won't be affected, and continues to use the printer. The LP print service notifies the person who submitted the request that the request has not been finished successfully.

When the interface is first invoked, the signals `HUP`, `INT`, `QUIT`, and `PIPE` (trap numbers 1, 2, 3, and 13) are ignored. The standard interface changes this so that these signals are trapped at appropriate times. The standard interface interprets receipt of these signals as warnings that the printer has a problem; when it receives one, it issues a fault alert.

How to Write a Filter

A filter is used by the LP print service each time it has to print a type of file that isn't acceptable by a printer. A filter can be as simple or as complex as needed; there are only a few external requirements:

- The filter should get the content of a user's file from its standard input and send the converted file to the standard output.

- A `slow` filter can send messages about errors in the file to standard error. A `fast` filter should not, as described below. Error messages from a `slow` filter are collected and sent to the user who submitted the file for printing.
- If a `slow` filter dies because of receiving a signal, the print request is stopped and the user who submitted the request is notified. Likewise, if a `slow` filter exits with a non-zero exit code, the print request is stopped and the user is notified. The exit codes from `fast` filters are treated differently, as described below.
- A filter should not depend on other files that normally would not be accessible to a regular user; if a filter fails when run directly by a user, it will fail when run by the LP print service.

The "Filter Management" section earlier in this chapter describes how to add a filter to the LP print service.

If you want your filter to detect printer faults, you must also fulfill the following requirements:

- If possible, the filter should wait for a fault to be cleared before exiting. Additionally, it should continue printing at the top of the page where printing stopped after the fault clears. If the administrator does not want this contingency followed, the LP print service will stop the filter before alerting the administrator.
- It should send printer fault messages to its standard error as soon as the fault is recognized. It does not have to exit, but can wait as described above.
- It should *not* send messages about errors in the file to standard error. These should be included in the standard output stream, where they can be read by the user.
- It should exit with a zero exit code if the user's file is finished (even if errors in the file have prevented it from being printed correctly).
- It should exit with a non-zero exit code *only* if a printer fault has prevented it from finishing a file.
- When added to the filter table, it must be added as a `fast` filter. (See the "Defining a Filter" section in this chapter for details.)

Quick Reference to LP Print Service Administration

These commands are found in the `/usr/lib` directory. (If you expect to use them frequently, you might find it convenient to include that directory in your `PATH` variable. To use the administrative commands, you must be logged in either as `root` or as `lp`. (`lp` is a system login, use of which requires a password.) For a description of how to set up a password for a system login, see the "Security" chapter.

You'll also probably need to use the commands for disabling and enabling a printer and the rest of the user commands.

- Activating a printer:

  `enable(1)`

- Canceling a request for a file to be printed:

  `cancel(1)`

- Sending a file (or files) to a printer:

  `lp(1)`

- Reporting the status of the LP print service:

  `lpstat(1)`

- Deactivating a specified printer(s):

  `disable(1)`

- Permitting job requests to be queued for a specific destination:

  `/usr/sbin/accept(1M)`

- Preventing jobs from being queued for a specified destination:

  `/usr/sbin/reject` - described on the `accept(1M)` manual page

- Setting up or changing printer configurations:

  `/usr/sbin/lpadmin(1M)`

- Setting up or changing filter definitions:

  `/usr/sbin/lpfilter(1M)`

- Setting up or changing preprinted forms:

  `/usr/sbin/lpforms(1M)`
- Mounting a form:

  `/usr/sbin/lpadmin(1M)`
- Moving output requests from one destination to another:

  `/usr/sbin/lpmove` - described on the `lpsched(1M)` manual page

  See `lpsched(1M)`.
- Starting the LP print service scheduler:

  `/usr/lib/lp/lpsched(1M)`
- Stop the LP print service scheduler

  `/usr/sbin/lpshut(1M)` - described on the `lpsched(1M)` manual page
- Setting or changing the default priority and priority limits that can be requested by users of the LP print service:

  `/usr/sbin/lpusers(1M)`

12 Process Scheduling

Introduction

The UNIX system scheduler determines when processes run. It maintains process priorities based on configuration parameters, process behavior, and user requests; it uses these priorities to assign processes to the CPU.

System V Release 4.0 gives users absolute control over the order in which certain processes run and the amount of time each process may use the CPU before another process gets a chance.

By default, the Release 4 scheduler uses a time-sharing policy like the policy used in previous releases. A time-sharing policy adjusts process priorities dynamically in an attempt to provide good response time to interactive processes and good throughput to processes that use a lot of CPU time.

The System V Release 4.0 scheduler offers a real-time scheduling policy as well as a time-sharing policy. Real-time scheduling allows users to set fixed priorities on a per-process basis. The highest-priority real-time user process always gets the CPU as soon as it is runnable, even if system processes are runnable. An application can therefore specify the exact order in which processes run. An application may also be written so that its real-time processes have a guaranteed response time from the system.

For most UNIX environments, the default scheduler configuration works well and no real-time processes are needed: administrators should not change configuration parameters and users should not change scheduler properties of their processes. However, when the requirements for an application include strict timing constraints, real-time processes sometimes provide the only way to satisfy those constraints.

NOTE

Real-time processes used carelessly can have a dramatic negative effect on the performance of time-sharing processes.

This chapter is addressed to administrators of systems that include the System V Release 4.0 scheduler. There are at least two reasons why administrators should understand the scheduler:

- The scheduler has an overriding effect on the performance and perceived performance of a system. The default scheduler is tuned to perform well in representative work environments, but you must understand how it operates to know whether you can reconfigure it to better suit local needs.

- A bug in a real-time program or a malicious real-time user can lock out all other processing, including kernel processing. Users need root permission to create real-time processes, and presumably only trustworthy users have this permission. But administrators still should be aware that the scheduler functions introduce new ways to cause trouble, and should be prepared for accidents and for misuse of these functions.

For programming information on the scheduler, see the *Programmer's Guide: System Services and Application Packaging Tools*. The primary user command for controlling process scheduling is `priocntl`(1), which is described in the *User's Reference Manual*. The primary function call for controlling process scheduling is `priocntl`(2), which is described in the *Programmer's Reference Manual*.

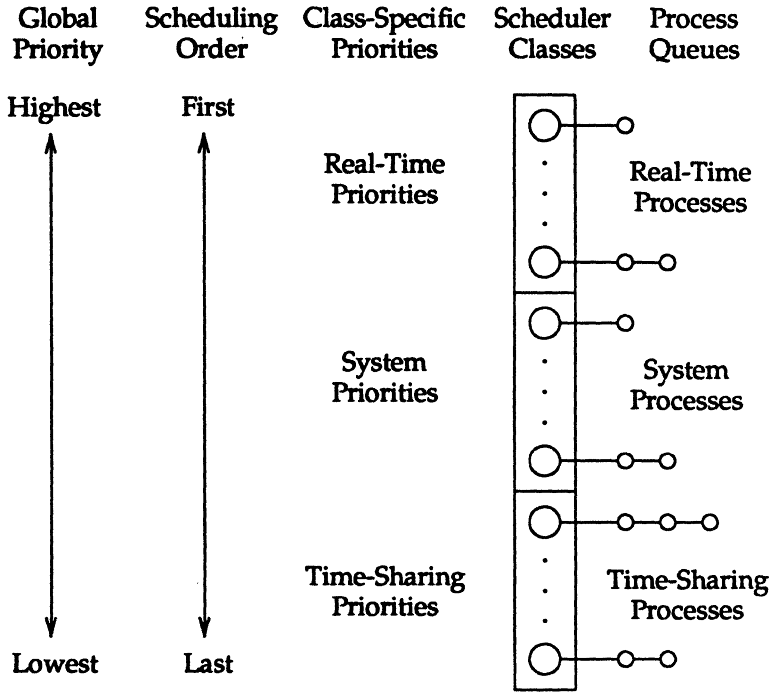
The rest of this chapter is organized as follows:

- The "Overview of the Process Scheduler" tells what the scheduler does and how it does it. It also introduces scheduler classes.
- "Configuring the Scheduler" describes how you can control the scheduler using tunable parameters and the two scheduler parameter tables: `ts_dptbl`(4) for time-sharing parameters and `rt_dptbl`(4) for real-time parameters.
- "Changing Scheduler Parameters with `dispadmin`" tells how to display or change scheduler parameters in a running system. `dispadmin` changes do not survive a reboot. To make permanent changes in scheduler configuration, you must change the scheduler parameter tables in the `master.d` directory.

Overview of the Process Scheduler

The following figure shows how the System V Release 4.0 process scheduler works:

Figure 12-1: The System V Release 4.0 Process Scheduler



When a process is created, it inherits its scheduler parameters, including scheduler class and a priority within that class. A process changes class only as a result of a user request. The system manages the priority of a process based on user requests and a policy associated with the scheduler class of the process.

In the default configuration, the initialization process belongs to the time-sharing class. Because processes inherit their scheduler parameters, all user login shells begin as time-sharing processes in the default configuration.

The scheduler converts class-specific priorities into global priorities. The global priority of a process determines when it runs—the scheduler always runs the runnable process with highest global priority. Numerically higher priorities run first. Once the scheduler assigns a process to the CPU, the process runs until it uses up its time slice, sleeps, or is preempted by a higher-priority process. Processes with the same priority run round-robin.

Administrators specify default time slices in the configuration tables, but users may assign per-process time slices to real-time processes.

You can display the global priority of a process with the `-cl` options of the `ps(1)` command. You can display configuration information about class-specific priorities with the `priocntl(1)` command and the `dispadmin(1M)` command.

By default, all real-time processes have higher priorities than any kernel process, and all kernel processes have higher priorities than any time-sharing process.

NOTE As long as there is a runnable real-time process, no kernel process and no time-sharing process runs.

The following sections describe the scheduling policies of the three default classes.

Time-Sharing Class

The goal of the time-sharing policy is to provide good response time to interactive processes and good throughput to CPU-bound processes. The scheduler switches CPU allocation frequently enough to provide good response time, but not so frequently that it spends too much time doing the switching. Time slices are typically on the order of a few hundred milliseconds.

The time-sharing policy changes priorities dynamically and assigns time slices of different lengths. The scheduler raises the priority of a process that sleeps after only a little CPU use (a process sleeps, for example, when it starts an I/O operation such as a terminal read or a disk read); frequent sleeps are characteristic of interactive tasks such as editing and running simple shell commands. On the other hand, the time-sharing policy lowers the priority of a process that uses the CPU for long periods without sleeping.

The default time-sharing policy gives larger time slices to processes with lower priorities. A process with a low priority is likely to be CPU-bound. Other processes get the CPU first, but when a low-priority process finally gets the CPU, it gets a bigger chunk of time. If a higher-priority process becomes runnable during a time slice, however, it preempts the running process.

The scheduler manages time-sharing processes using configurable parameters in the time-sharing parameter table `ts_dptbl`. This table contains information specific to the time-sharing class.

System Class

The system class uses a fixed-priority policy to run kernel processes such as servers and housekeeping processes like the paging daemon. The system class is reserved for use by the kernel; users may neither add nor remove a process from the system class. Priorities for system class processes are set up in the kernel code for those processes; once established, the priorities of system processes do not change. (User processes running in kernel mode are not in the system class.)

Real-Time Class

The real-time class uses a fixed-priority scheduling policy so that critical processes can run in predetermined order. Real-time priorities never change except when a user requests a change. Contrast this fixed-priority policy with the time-sharing policy, in which the system changes priorities in order to provide good interactive response time.

Users with super-user privileges can use the `priocntl` command or the `priocntl` system call to assign real-time priorities.

The scheduler manages real-time processes using configurable parameters in the real-time parameter table `rt_dptbl`. This table contains information specific to the real-time class.

Configuring the Scheduler

The default configuration includes both the time-sharing and the real-time scheduler classes. The time-sharing class is tuned for representative UNIX system workloads. Such workloads have a high proportion of interactive processes, which sleep early and often. The real-time class is configured for applications that need it.

For traditional time-sharing uses such as software development, office applications, and document production, real-time processes may be unnecessary. In addition, they may be undesirable. First, they consume memory that cannot be paged: the u-blocks of real-time processes are never paged out. Second, they introduce new ways to cause performance problems: a high-priority real-time process can block out all other processing. In a computing environment where only time-sharing is needed, you may want to remove the real-time scheduler class from your configuration, as described below in the section "Changing Scheduler Configuration."

On the other hand, if a machine is running an application that has strict requirements on the order in which processes must run, then the real-time scheduler class provides the only way to guarantee that those requirements are met.

NOTE

Real-time processes can have a dramatic negative effect on time-sharing performance.

This section describes the parameters and tables that control scheduler configuration, and tells how to reconfigure the scheduler. A basic assumption is that your workload is reasonable for your system resources, such as CPU power, primary memory, and I/O capacity. If your workload does too much computation or too much I/O for your hardware, reconfiguring the scheduler will not help. See the "Performance Management" chapter for more information.

Default Global Priorities

The following table shows the scheduling order and global priorities for each scheduler class.

Scheduling Order	Global Priority	Scheduler Class
first	159	
	.	
	.	Real-Time
	.	
	100	
	99	
	.	
	.	System
	.	
	60	
	59	
	.	
	.	Time-Sharing
	.	
last	0	

When your system is built, it constructs this information from the tunable parameters and scheduler parameter tables described in the following sections. Although you are not forced to configure scheduler classes to produce consecutive, non-overlapping global priorities like the default priorities, we recommend that you do so for the sake of simplicity. Likewise, we recommend that you make all real-time global priorities greater than the global priorities of all other classes. These conventions simplify scheduler configuration, and they should be able to accommodate any requirements on the scheduler.

Kernel processes such as the swapper and the paging daemon run in the system scheduler class. Kernel processes must compete with user processes for CPU time, and in the default configuration all real-time processes have higher priorities than all system processes. Therefore, real-time applications must be written carefully to ensure that the kernel gets the processing time it needs. Also, if you reconfigure the scheduler, make sure that the system class gets enough priority over the time-sharing class to give kernel processes the CPU time they need.

Tunable Parameters

This section describes the tunable parameters that control scheduler configuration. These parameters are specified in files in the /etc/master.d directory.

The following parameters are specified in the kernel file:

- MAXCLSYSPRI is the maximum global priority of processes in the system class. When the kernel starts system processes, it assigns their priorities using MAXCLSYSPRI as a reference point.

  NOTE MAXCLSYSPRI must be 39 or greater, because the kernel assumes it has at least that great a range of priorities below MAXCLSYSPRI. If you request a MAXCLSYSPRI below 39, it is changed to 39.

  The most important system processes get global priorities at or near MAXCLSYSPRI; the least important system processes get global priorities at or near (MAXCLSYSPRI - 39). The default value of MAXCLSYSPRI is 99, which gives all system processes higher priorities than all user processes.

- INITCLASS is the scheduler class assigned to the init process. This scheduler class is inherited by all descendants of init, which normally include all user login shells. By default, INITCLASS is TS; that is, all login shells are time-sharing processes in the default configuration.
- SYS_NAME is the character string name of the system scheduler class. The default value of SYS_NAME is SYS.

The following parameters are specified in the ts file, which controls the time-sharing policy:

- TSMAXUPRI specifies the range within which users may adjust the priority of a time-sharing process using the priocntl system call: the valid range is -TSMAXUPRI to +TSMAXUPRI. The default value of TSMAXUPRI is 20. (Configuring a value of 20 emulates the behavior of the older, less general scheduler interfaces nice and setpriority, which continue to work as in all previous releases to System V Release 4.)

  The value of TSMAXUPRI is independent of the configured number of global time-sharing priorities, though we recommend configuring at least 40 time-sharing priorities, as explained below in the section on ts_dptbl. In the default configuration, there are 60 time-sharing priorities, but users may adjust their priorities only within a range of -20 to +20. The system may use the remaining priorities depending on process behavior.

- NAMETS specifies the character string name of the time-sharing scheduler class. This name is returned by the priocntl system call and it is assigned to the tunable parameter INITCLASS to specify the default scheduler class for user processes. The default value of NAMETS is TS.

The following parameter is specified in the rt file, which controls the real-time policy:

- NAMERT specifies the character string name of the real-time scheduler class. The default value of NAMERT is RT.

Real-Time Parameter Table rt_dptbl

The scheduler uses rt_dptbl(4), the real-time scheduler (or dispatcher) parameter table, to manage real-time processes. A default version of rt_dptbl is delivered with the system, and an administrator may change it to suit local needs. rt_dptbl is specified in the rt file in the master.d directory. It is built into the kernel as part of system configuration if the system file contains the line

```
INCLUDE:RT
```

You may adjust the size and values of rt_dptbl depending on the applications on your system. Here is part of a simple rt_dptbl:

rt_glbpri	rt_qntm
100,	100,
101,	80,
102,	60,
103,	40,
104,	20,
105,	10

- The rt_glbpri column contains global priorities (the priorities that determine when a process runs). Higher numbers run first.

- The rt_qntm column contains the default time slice (or quantum) associated with the priority in the rt_glbpri column; this is the maximum amount of time a process with this priority may use the CPU before the scheduler gives another process a chance. This time slice is specified in clock ticks. (The system clock ticks HZ times per second, where HZ is a hardware-dependent constant defined in the param.h header file.)

The highest priority specified in this table is 105, so processes with priority 105 always run before any other processes. If it does not sleep, a process with priority 105 runs for 10 clock ticks before the scheduler looks for another process to run. (Because 105 is the highest priority, a process at this priority would be preempted after its time slice only if there were another process with priority 105.) Processes at priority 104 run for 20 clock ticks, and so on. The lowest real-time priority specified in this table is 100; a process with priority 100 runs for 100 clock ticks.

The default real-time priority is the lowest priority configured in rt_dptbl. This is the priority assigned to a process if it is changed to a real-time process and no priority is specified. This is also the priority assigned to the init process and all its children if INITCLASS is set to RT.

Though rt_dptbl contains default time slices for real-time priorities, users with the appropriate privilege can set real-time priority and time slice independently. Users can specify any time slice they want for a real-time process, including an infinite time slice. The system assumes that real-time processes voluntarily give up the CPU so other work can get done.

Time-Sharing Parameter Table ts_dptbl

The scheduler uses ts_dptbl(4), the time-sharing scheduler (or dispatcher) parameter table, to manage time-sharing processes. A default version of ts_dptbl is delivered with the system, and an administrator may change it to suit local needs. Save a backup of the default version of ts_dptbl. ts_dptbl is specified in the ts file in the master.d directory. It is automatically built into the kernel as part of system configuration.

You may change the size and values of ts_dptbl depending on your local needs, but only experienced administrators should make such changes. The default values have a long history of good performance over a wide range of environments. Changing the values is not likely to help much, and inappropriate values can have a dramatic negative effect on system performance.

If you do decide to change `ts_dptbl`, we recommend that you include at least 40 time-sharing global priorities. A range this large gives the scheduler enough latitude to distinguish processes based on their CPU use, which it must do to give good response to interactive processes. The default configuration has 60 time-sharing priorities. Here is part of a simple `ts_dptbl`:

glbpri	qntm	tqexp	slprt	mxwt	lwt
0,	100,	0,	1,	5,	1,
1,	90,	0,	2,	5,	2,
2,	80,	1,	3,	5,	3,
3,	70,	1,	4,	5,	4,
4,	60,	2,	5,	5,	5,
5,	50,	2,	6,	5,	6,
6,	40,	3,	7,	5,	7,
7,	30,	3,	8,	5,	8,
8,	20,	4,	9,	5,	9,
9,	10,	4,	9,	5,	9,

- The `glbpri` column contains global priorities (the priorities that determine when a process runs). Higher numbers run first.

  In the table above, the global priorities run from a high of 9 to a low of 0.

- The `qntm` column contains the time slice (or quantum) associated with the priority in the `glbpri` column; this is the maximum amount of time a process with this priority may use the CPU before the scheduler gives another process a chance. This time slice is specified in clock ticks. (The system clock ticks `HZ` times per second, where `HZ` is a hardware-dependent constant defined in the `/usr/include/sys/param.h` header file.)

  In the table above, time slices run from 10 clock ticks for the highest-priority processes to 100 clock ticks for the lowest-priority processes.

- The `tqexp` column determines the new process priority for a process whose time slice expires before it sleeps. If a process at the priority in the `glbpri` column uses its whole time slice without sleeping, the scheduler changes its priority using the `tqexp` column as an index back into `ts_dptbl`: the new priority is the global priority in the `tqexp` position in `ts_dptbl`. (In the default configuration, the index of an entry in `ts_dptbl` happens to match the global priority of that entry. However, this match is not necessary.)

  It is usually reasonable to lower the priority of a time-sharing process whose time slice expires, because the process is too CPU-bound for its current priority. A long, CPU-intensive process is an extreme example of such a process, and its priority should usually be lowered in favor of processes that sleep after a little CPU use, which are more likely to be interactive processes.

In the table above, process priorities are cut roughly in half when a time slice expires. The lowest priority (0) stays at 0, priority 1 is reduced to 0, priorities 2 and 3 are reduced to 1, and so on.

- The `slprt` column gives the priority assigned to a process when it returns from a sleep. A process may sleep voluntarily, as it does when it makes certain system calls, or involuntarily, as it does when the kernel puts it to sleep after a page fault, for example. It is usually reasonable to raise the priority of a process that has slept. By sleeping, it has shown desirable behavior (it gave up the CPU, so we reward it by giving it a higher priority when it awakes).

  In the table above, process priorities are incremented by 1 after they sleep, except that the highest time-sharing priority (9) stays the same.

- The `mxwt` column specifies the number of seconds a process can remain runnable before having its priority changed. (The priority is changed using the `lwt` column. See the explanation below.)

  In the table above, all priorities are recalculated after a wait of 5 seconds.

- The `lwt` column gives the new priority for a process that is runnable for `mxwt` seconds without getting its full time slice. It is usually reasonable to raise the priority of a process that is not getting any CPU time.

  In the table above, process priorities are incremented by 1 when they have been runnable for 5 clock ticks, except that the highest time-sharing priority (9) stays the same.

The default global priority of a time-sharing process is the priority in the middle of `ts_dptbl`. This is the priority assigned to a process if it is changed to a time-sharing process with default parameters. This is also the priority initially assigned to the `init` process if `INITCLASS` is set to `TS`. The descendants of `init`, normally including all login shells and other user processes, inherit its class and current scheduler parameters.

Kernel-Mode Parameter Table ts_kmdpris

The scheduler uses the kernel-mode parameter table ts_kmdpris to manage sleeping time-sharing processes. A default version of ts_kmdpris is delivered with the system, and there is seldom a reason to change it. ts_kmdpris is specified in the ts file in the master.d directory. It is automatically built into the kernel as part of system configuration.

NOTE The kernel assumes that it has at least 40 priorities in ts_kmdpris. It panics if it does not.

The kernel-mode parameter table is a one-dimensional array of global priorities. The kernel assigns these priorities to sleeping processes based on their reasons for sleeping. If a user process sleeps because it is waiting for an important resource, such as an inode, it sleeps at a priority near the high end of the ts_kmdpris priorities, so that it may get and free the resource quickly when the resource becomes available. If a user process sleeps for a less important reason, such as a wait for terminal input, it sleeps at a priority near the low end of the ts_kmdpris priorities.

The default kernel-mode parameter table is simply a one-dimensional array of the integers from 60 through 99, which means that time-sharing processes sleep at priorities between the default real-time priorities and the default time-sharing priorities.

In the default configuration, the priorities in ts_kmdpris happen to be exactly the same as the priorities used by system class processes, because the tunable parameter MAXCLSYSPRI is the same as the highest priority in ts_kmdpris. This overlap is designed to be consistent with the scheduler behavior of previous releases of the UNIX system, because these priorities produce good performance in most environments. But the overlap is not necessary. The System V Release 4.0 scheduler introduces a logical separation between the priorities of system processes and sleeping time-sharing processes; an administrator may configure a machine so that the two sets of processes have different ranges of global priorities.

Changing Scheduler Configuration

Changing scheduler configuration requires changing one or more of the tunable parameters or the configuration tables rt_dptbl, ts_dptbl, and ts_kmdpris. You can change any of these by changing the appropriate file in the master.d directory and rebuilding the kernel as described in the "Performance Management" chapter. Changes made in this way are permanent. This is the only way to change the size of the configuration tables.

See the "Changing Scheduler Parameters with dispadmin" section for a way to make temporary changes on a running system.

Removing a Scheduler Class

For systems that do not need real-time processes, it may make sense to remove the real-time class, thereby making it impossible to create real-time processes. By not having real-time processes, you avoid their non-pageable u-blocks and you avoid the possibility of a runaway process monopolizing the machine. To remove the real-time scheduler class:

- Replace the INCLUDE:RT line from the /etc/system file with:

```
EXCLUDE:RT
```

- Rebuild the kernel.

There should be no reason to remove the time-sharing scheduler class. An application can put its crucial processes into the real-time class, and thereby ensure that they always run before any time-sharing process. However, if you have a compelling reason to remove the time-sharing class, the following provides a way to do it:

- Replace the INCLUDE:TS line from the /etc/system file with:

```
EXCLUDE:TS
```

- In the kernel file in master.d, change INITCLASS to RT. This makes init and all its descendants real-time processes.
- Rebuild the kernel.

Installing a Scheduler Class

By default, both the time-sharing and the real-time scheduler classes are installed. Therefore, you need to install a class only if you first remove it.

To re-install the time-sharing class:

- Make sure that the `TS` module is in the `/boot` directory.
- Insert the `INCLUDE:TS` line in the `system` file. (The `TS` module is automatically configured unless it is explicitly `EXCLUDE`d.)
- Build the kernel.

To re-install the real-time class:

- Make sure that the `RT` module is in the `/boot` directory.
- Insert the `INCLUDE:RT` line in the `system` file. (The `RT` module is not configured unless it is explicitly `INCLUDE`d.)
- Build the kernel.

When you re-install a scheduler class, you should also check the value of the tunable parameter `INITCLASS` to make sure that your configuration is assigning the default scheduler class you want.

Changing Scheduler Parameters with `dispadmin`

The `dispadmin`(1M) command allows you to change or retrieve scheduler information in a running system. Changes made using `dispadmin` do not survive a reboot. To make permanent configuration changes, you must change the scheduler parameter tables in the `master.d` directory as described in the section above on configuration. However, you can use `dispadmin` to get an effect equivalent to changing configuration tables by calling `dispadmin` from a startup script that changes the configuration automatically at boot time.

The `dispadmin` command has three forms:

- `dispadmin -l` lists the configured scheduler classes.
- `dispadmin -g [-r` *res*`] -c` *class* gets scheduler parameters for the specified class. By default, time slices are printed in milliseconds. You may optionally retrieve time slices at a resolution specified by the `-r` option.
- `dispadmin -s` *config_file* `-c` *class* sets scheduler parameters for the specified class from *config_file* (your configuration file).

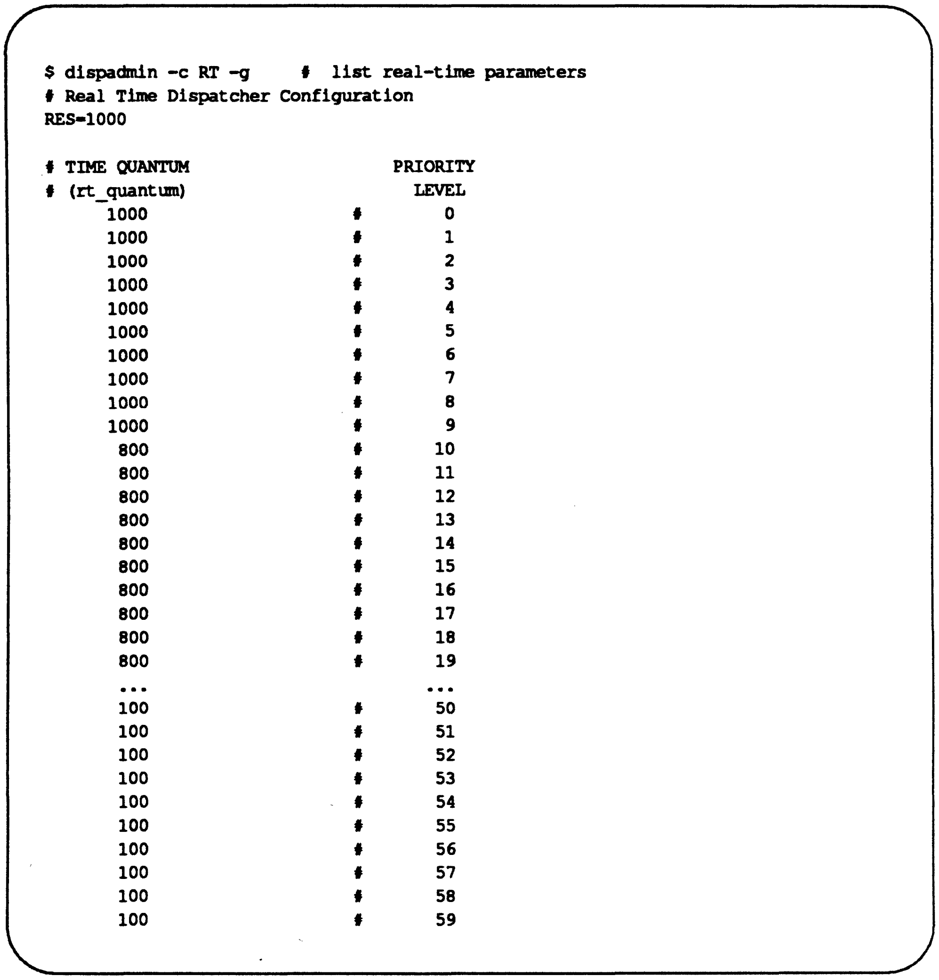
Here is the output of the `-l` option for the default configuration.

```
$ dispadmin -l
CONFIGURED CLASSES
==================

SYS     (System Class)
TS      (Time Sharing)
RT      (Real Time)
```

The `-g` option gets current scheduler parameters for the specified class and writes them to the standard output. Scheduler parameters are class specific. The parameters for the default classes are described in the sections above on the scheduler parameter tables; `ts_dptbl` holds time-sharing parameters and `rt_dptbl` holds real-time parameters.

The following screen shows part of the output of `dispadmin -g` for the real-time class:



```
$ dispadmin -c RT -g     # list real-time parameters
# Real Time Dispatcher Configuration
RES=1000

# TIME QUANTUM                    PRIORITY
# (rt_quantum)                      LEVEL
      1000                  #         0
      1000                  #         1
      1000                  #         2
      1000                  #         3
      1000                  #         4
      1000                  #         5
      1000                  #         6
      1000                  #         7
      1000                  #         8
      1000                  #         9
       800                  #        10
       800                  #        11
       800                  #        12
       800                  #        13
       800                  #        14
       800                  #        15
       800                  #        16
       800                  #        17
       800                  #        18
       800                  #        19
       ...                          ...
       100                  #        50
       100                  #        51
       100                  #        52
       100                  #        53
       100                  #        54
       100                  #        55
       100                  #        56
       100                  #        57
       100                  #        58
       100                  #        59
```

The following screen shows part of the output of dispadmin -g for the time-sharing class:

```
$ dispadmin -c TS -g      # list time-sharing parameters
# Time Sharing Dispatcher Configuration
RES=1000

# ts_quantum  ts_tqexp  ts_slpret  ts_maxwait ts_lwait  PRIORITY LEVEL
       1000       0        10          5        10         #      0
       1000       0        11          5        11         #      1
       1000       1        12          5        12         #      2
       1000       1        13          5        13         #      3
       1000       2        14          5        14         #      4
       1000       2        15          5        15         #      5
       1000       3        16          5        16         #      6
       1000       3        17          5        17         #      7
       1000       4        18          5        18         #      8
       1000       4        19          5        19         #      9
        800       5        20          5        20         #     10
        800       5        21          5        21         #     11
        800       6        22          5        22         #     12
        800       6        23          5        23         #     13
        800       7        24          5        24         #     14
        800       7        25          5        25         #     15
        800       8        26          5        26         #     16
        800       8        27          5        27         #     17
        800       9        28          5        28         #     18
        800       9        29          5        29         #     19
        ...      ...      ...         ...       ...              ...
        100      40        55          5        55         #     50
        100      41        55          5        55         #     51
        100      42        56          5        56         #     52
        100      43        56          5        56         #     53
        100      44        57          5        57         #     54
        100      45        57          5        57         #     55
        100      46        58          5        58         #     56
        100      47        58          5        58         #     57
        100      48        59          5        59         #     58
        100      49        59          5        59         #     59
```

By default, dispadmin reports time slices in milliseconds. If you specify the -r *res* option, dispadmin reports time slices in units of *res* intervals per second. For example, a *res* of 1000000 reports time slices in microseconds (millionths of a second).

The -s *config_file* option uses *config_file* to set scheduler parameters for the specified class. The configuration file must be in the class-specific format produced by the -g option. The meanings of the parameters are described in the sections above on the scheduler parameter tables; ts_dptbl holds time-sharing parameters and rt_dptbl holds real-time parameters.

The following examples show how to set the parameters for the default classes as specified in the configuration files rt_config and ts_config. The examples presuppose that these two files are in the correct formats.

```
$ dispadmin -c RT -s rt_config     #  set real-time parameters

$ dispadmin -c TS -s ts_config     #  set time-sharing parameters
```

The files that specify the new scheduler parameters must have the same number of priority levels as the current table that is being overwritten. To change the number of priority levels, you must change the ts file or the rt file in the master.d directory as described in the section above on configuration.

13 Restore Service

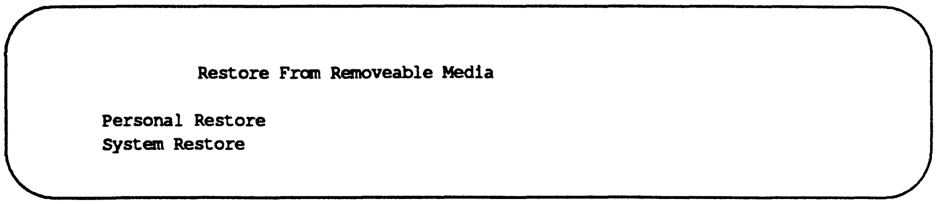
DRAFT COPY
February 1, 1992
File: Crestore

Introduction

This chapter tells you how to restore backed up files, directories, file systems, data partitions, disks, and partitioning information from archive volumes. You can do any of these functions by selecting the appropriate task from a series of menus provided for administration. To access the system administration menu for using the restore service, type

```
        sysadm restore_service/basic
```

The following menu will appear on your screen:



```
                Restore From Removeable Media

        Personal Restore
        System Restore
```

If you prefer not to use the menus, you can perform the same tasks by executing shell-level commands instead. The following table shows the shell commands that correspond to the tasks on the menu.

Task to Be Performed	*sysadm* Task	Shell Command
Restore from backup archives	`restore`	`cpio(1)`

Each task listed above is explained fully later in this chapter. In addition, the *System Administrator's Reference Manual* and *User's Reference Manual* provide information on the shell commands.

Some of the `sysadm` menus do not offer all the functionality available with the corresponding shell-level commands. If you are not an experienced administrator, however, you may find it easier to use the menus, which provide prompts and help messages that are not available with shell commands.

Discussion of System Restores

This chapter includes a section on performing full and partial system restores. These functions are separate from the restore service that is available to all users. They must be performed by the administrator in firmware mode. See the section entitled "System Restores."

Overview of Restore Operations

The restore service enables you to retrieve copies (archives) of files, directories, file systems, data partitions, disks, and partitioning information that have been preserved on archive media such as tapes. There are various types of restore operations; the type you do depends on the type of archive you have. Therefore, to fully understand restore operations, you should familiarize yourself with the "Backup Service" chapter before reading this chapter.

On large computer systems, the tasks associated with the restore service are usually performed by two people: a system administrator and a computer operator. (On small systems, one person may serve as both the administrator and the operator.) The system administrator's job is

- to establish backup policies based on factors such as available resources, the needs of users, and management directives
- to decide which storage media are used for restore operations and how these media are organized and used
- to issue restore requests for entire file systems, data partitions, directories, and files

The computer operator's job is

- to check the status of pending restore requests
- to respond to system prompts when a restore request requires assistance
- to insert and remove storage media

Users on the system may not perform restore operations themselves. They may submit requests for restore operations to the operator and periodically check the status of operations in progress.

How Restore Requests Are Identified

The restore service assigns a job ID only if a restore request cannot be satisfied automatically and needs operator assistance. A job ID is an alphanumeric string consisting of the word `rest`, a dash, and a numeric ID. When a job ID is assigned to a restore request for multiple individual files, the restore service identifies each file in the request by appending a letter to the job ID for each file. This labeling convention is useful because it allows individual parts of a request to be handled separately. For example, a user might request a restore of the files `myfile.1`, myfile.2, and myfile.3. The job IDs for these files are rest-4592a, rest-4592b, and rest-4592c, respectively. The user decides that it is not necessary to restore myfile.3 so he or she cancels the request for that file. The operator can then cancel rest-4592c without affecting the restore requests for the other two files.

Note that if these three files are contained in a directory, and if the directory is the object to be restored, then only one job ID is assigned to the directory restore no matter how many files are in that directory.

How Restore Operations Are Performed

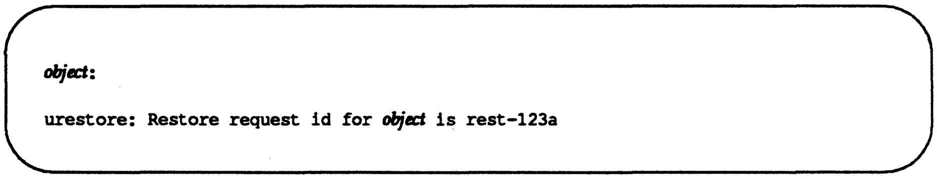
The restore process begins when someone, either a user or an administrator, requests that a particular object be restored. This is done by issuing either the restore or urestore shell command or by using the restore menu item.

Once a restore request is issued, the restore service first looks at the backup history log to see if a backup has been performed for the object to be restored and determines if the archive is online.

If all this is true, the user receives the following immediate message:

```
Attempting restore from online archive
```

If the restore service cannot find an archive of that object for any reason, the user will receive a message similar to the following:



```
object:

urestore: Restore request id for object is rest-123a
```

This could happen if the archive is not online or if the restore service cannot find an archive with that information.

In the latter case the operator must then determine which archive might contain the object to be restored. Some restores might require multiple archives.

NOTE

If the operator finds an archive produced on an earlier System V system, it may or may not be possible to restore that object with the Release 4.0 restore service, depending on the command used for the backup. If the backup was done with the cpio command, the operator can restore the archive the same way he or she would restore any Release 4.0 archive. However, if the volcopy command was used for the backup, the item cannot be restored using the Release 4.0 restore service. Instead, it must be restored by using the menus for the restore service which created it. (The restore service menu can be accessed by typing sysadm old_sysadm and then selecting the filemgmt item from the Main Menu.)

If the necessary archives are mounted, the restore operation proceeds automatically. Otherwise a mail message is sent to the operator indicating that a restore request is pending. This message reads as follows:

```
There is a pending restore request from user_login.
```

The operator can periodically look up pending restore requests that need servicing by issuing the rsstatus command or by using the restore_status menu item.

The last step of the restore process is to service pending restore requests. The operator loads the appropriate storage medium and then either issues the rsoper command or selects the respond item from the restore_service menu.

How the Restore Service Works

When the restore service tries to see if the archive is online, it does so by first looking to see if the object is listed in the history log. The restore service then tries to identify the exact archive that contains the object.

If the restore service is processing a request to restore a file, it looks through the archives available for all dates until just before the date from which the object is to be restored. The restore service chooses the most recent archive but not one that is more recent than the requested date.

On the other hand, if the restore service is processing a request to restore any object other than a file, it looks through the archives to find the most recent full file backup performed before the date from which the object is to be restored.

The restore service will restore that full file backup and update it with any incremental file backups performed up to the requested restore date.

As stated before, if the archive is found online, the restore proceeds automatically. Otherwise mail is sent to the operator and the request waits for service.

As explained in the "Backup Service" chapter, an archive is created by backing up requested objects using one of six default backup methods. The method chosen for the backup operation affects the type of restore that can be done and the level of difficulty of that restore operation. For example, data partitions can be restored only from full data partition backup archives.

Preparing for Restore Jobs

When an administrator is establishing restore policies for a computer site, one of the choices is who should be responsible for servicing restore requests. If the administrator decides to delegate this job to an operator, the `rsnotify` command can be used to route service requests to the person selected. For example, the command

```
        rsnotify -u user
```

causes mail to be sent to *user* when a request is made. If `rsnotify` is invoked without options, the login of the assigned operator is displayed. The display also includes the date the operator was assigned. If `rsnotify -u` *user* is invoked when an operator has been assigned, the existing assignment is replaced by the new one.

Once an operator has been assigned, the operator receives mail about pending jobs and can scan for pending restore requests with the `rsstatus` command or the `status` item from the `restore_service` menu.

If no operator is assigned, `mail` messages about restore requests are sent to `root` instead.

Using the Restore Service

All users can request restores of any files and directories that they own, but only administrators can request restores of data partitions, file systems or entire disks. Because of this division of functionality there are two commands for requesting restores. The `urestore` command can be used by anyone to request a restore of one of their own files or directories. (An administrator can use this `urestore`—from the the root prompt—for any file or directory, regardless of who owns it.) The `restore` command can be used only by an administrator. `restore` is used to request a restore of file systems, disks or data partitions, or to display partitioning information.

By default the restore service retrieves the latest version of an object. However, any archived version can be restored by identifying the desired version on the command line.

Directory or File Restores

When issuing the `urestore` command, you must have read permission for the parent directories of the file or directory you are having restored. You must have write permission as well for the immediate parent directory. In addition, if the request is made by anyone other than the administrator or the operator, that person must have owned the file or directory at the time the archive was made. The following options are available with the `urestore` command:

`-c` *job_ID* Cancels a previously issued restore request identified by *job_ID*.

`-d` *date* Restores a file system or directory as of *date*, which may or may not be the date of the latest archive version. The value of *date* is the same ten-character string used to specify a month, day, hour, minute, and year with the `date(1)` command: *MMddhhmmyy*.

`-m` If the restore cannot be carried out immediately, this option notifies the person requesting the restore (via `mail`) when the request has been completed.

`-n` When issued with the `-D` or `-F` option (one of which must be used), the `-n` option displays a list of all archived versions of a file or directory contained in the backup history log. The `-n` option does not restore the file or directory.

`-o` *target* Restores the archive to the *target* location.

`-D` *directory* Restores *directory* and all the files underneath it.

`-F` *file_name* Restores *file_name*.

Restoring Other Disk Objects

An administrator or operator can restore a data partition, a file system partition, or an entire disk by using the `restore` command, along with the following options:

`-A` *partdev* Initiates a restore of the entire disk *partdev*.

`-P` *partdev* Initiates a restore of the data partition *partdev*.

`-S` *oname* Initiates a restore of the file system partition *oname*.

`-n` When issued with the `-A`, `-P`, or `-S` option (one of which must be used), the `-n` option displays a list of all archived versions (of the disk object) contained in the backup history log. The `-n` option does not restore any disk objects.

partdev takes the form `/dev/rdsk/c?d?s?.` `If` `-A` is specified, the full disk method is used to repartition the disk.

CAUTION Restoring a disk overwrites any data on the disk and resets all partitioning information for the disk to that of the archive.

Restoring the Backup and Restore Services

When restoring any file system containing the backup and restore services, you must restore the following components:

- the backup and restore shell level commands
- the backup methods found in the directory `/etc/bkup/method`

- the system-supplied backup and restore tables found in the directory /etc/bkup or any backup and restore tables you have defined

On most systems, these components reside in the root directory (/) and in the /usr file system, but they may be located in any file system.

The following is the procedure for restoring the backup and restore services:

1. Change the system state to single-user mode by typing `init S`.
2. Mount the tape containing the Essential Utilities. These utilities include the commands for the backup and restore services.
3. When the program asks if you want to reformat your disk, type `yes` (if you want default partitioning). You will need to have your reformatting specifications at hand so you can provide this information. (See "Full System Restore: Changing the Disk Partition Sizes.")
4. Restore the most recent copy of the backup history log for your system; this may be the system-supplied log or a log you have created. To restore the most recent copy of the system-supplied log, type

   ```
   urestore -F /etc/bkup/bkhist.tab /etc/device.tab \
   /etc/dgroup.tab
   ```

   To restore the most recent copy of your custom backup history log, type

   `urestore -F` *pathname*

   where *pathname* is the correct pathname to your backup history log.
5. To restore the /usr file system, type

   ```
   restore -S /usr
   ```

   Note that to do these functions, you must run `restore` rather than the `urestore` command.
6. Restore the lost file systems and data partitions. Use the backup history log display to determine the labels of archive volumes that contain the information to be restored.

Specific Archive Version Restores

You may want to restore a specific archive version of an object other than the most recent version, especially if the most recent version has been corrupted.

This is done by invoking `restore -n` with the `-A`, `-P`, or `-S` option. This command displays a list of all archived versions of the object by date. The `-n` option displays the version list only; it does not invoke the restore request. From the list, you can select the date of the correct version to be restored and then with the `restore` command include the `-d` *date* option, where the value of *date* is the same ten-character string used to specify a month, day, hour, minute, and year with the `date(1)` command: *MMddhhmmyy.*

Restoring an Object to a New Location

By default, the restore service restores an object to its original location. However, there may be times when you don't want to restore an object to its original location.

You can restore an object to a new location by running the `restore` command with the `-o` *target* option, where *target* is the name of the new destination location. Either command redirects the restore to a new location and thus avoids overwriting the contents of the previous location.

Checking the Status of Restore Requests

You can check the status of restore requests by using options to any of three commands: `restore`, `rsstatus`, or `ursstatus`.

The `-s` or `-v` options can be used with either the `restore` or `urestore` command. The `-s` option displays a "." for each 100 blocks transferred from the archive. The `-v` option displays the name of each directory or file as it is transferred from the archive. Note that these options take effect only when the required archive volume is on line and the restore operation is processing.

Furthermore, both the `rsstatus` and `ursstatus` commands provide a list of pending restore requests. The `rsstatus` command can be issued only by an administrator or operator. The `ursstatus` command can be used by anyone to view the status of any restore he or she has requested. `ursstatus` lists pending file and directory restore requests for the invoking user. `rsstatus` lists all the pending restore requests in detail.

Displaying the Status Table

The information gathered by either of these commands is recorded in the `/etc/bkup/rsstatus.tab` table. To display the contents of that table, invoke either command without options. An `rsstatus` report contains the following fields:

`Job_ID` The job ID of the restore request

`Login` The login name of the person requesting the restore

`File` The full pathname of the file to be restored

`Date` The specified date from which an object should be restored. (The restore service selects archives made on a backup date as near as possible to the date specified.)

`Target` The full pathname of the location where the restored object should be placed

`Backup_Date` The date of the last backup performed before the date specified in a restore request

`Method` The method used to perform the backup

`Dtype` The destination device, such as a cartridge tape, on which the backup archive was created

`Labels` The labels for the archive volumes to be restored

The `ursstatus` report contains only the first five fields.

The `rsstatus` report appears in the following format on your screen:

```
Jobid        Login  File          Date     Target        Bkp date  Method   Dtype

rest-11111a  user1  /home/user1   Dec 27   /home/user1   Sun Dec   incfile  ctape1
                    /oam/ISSUES   1989     /bkISSUES     27 1987
                                  17:00:
                                  00
rest-32984c  user1  /home/user1   Dec 27   /home/user1   Sun Dec   incfile  ctape1
                    /oam/bkrs     1989     /oam/bkrs     27 1987
                                  17:00:
                                  00
rest-04818a  user2  /home/user2   Dec 27   /home/user2   Sun Dec   ffile    ctape1
                    /rsstatus.c   1989     /rsstatus.c   27 1987
                                  17:00:
                                  00
rest-04818a  user2  /home/user2   Dec 27   /home/user2   Sun Dec   ffile    ctape1
                    /myfile.c     1989     /myfile.c     27 1987
                                  17:00:
                                  21
rest-64978a  root   /usr          Dec 27   /dev/rd       Mon Jan   ffile    ctape1
                                  1989     sk/c8d0       18 1988
                                  17:00:   s2
                                  50
```

Customizing the Display of the Status Table

The last section described the restore status table and the default display of information in it. If you do not want to see a default report, however, you can customize both the contents and the format of the status report.

The default display provides all the available information about pending restore requests. You can restrict the fields displayed by using one of the following options to the `rsstatus` command: `-d` [*dtype*], `-j`, or `-u`.

`-d [`*dtype*`]` Restricts the display to restore jobs that can be satisfied by a specific device. *dtype* specifies the device type (ctape1).

The *dtype* argument is optional. When would an administrator want to use this argument? Specifying a device type may be helpful, for example, if an administrator's access to a particular device (such as a cartridge tape drive) is limited and he or she therefore has little time in which to do restore operations on cartridge tape.

In this situation an administrator may need to ignore, temporarily, operations being done on devices that are available at any time, and concentrate on monitoring those operations that require a cartridge tape drive. By specifying `ctape1` as an argument to the `-d` option, the administrator can display status reports for only those operations being done on cartridge tape.

`-d ctape1` Displays the status of all restore requests that can be satisfied by inserting cartridge tapes into a cartridge tape drive.

In another instance, an operator may have mounted a certain set of archive volumes and might want to verify that all restores that can be satisfied by those volumes will be performed.

`-d:bk0010,bk0011,bk0012`
Displays the status of all requests for restore operations from an archive on the volumes with the labels specified (`bk0010`, `bk0011`, and `bk0012`).

`-j` *job_IDs* Displays the status of operations with the specified restore job IDs

`-u` *users* Displays the status of restore operations requested by the specified user logins

You can specify all three options (`-d`, `-j`, and `-u`) on the same command line. When you do, the report displayed will contain only those entries that satisfy all three specifications.

In the default `rsstatus` display, each field has a title and a specific length. Data entries that exceed the specified field length wrap to the next line within the field. You can change the format of the `rsstatus` display by using the `-h, -f`, or `-s` option to the `rsstatus` command.

`-h` Suppresses the headers (titles) for a display. This option is useful when the contents of a display are to be filtered by another process.

`-s` Suppresses field wrap on the display so the data in the display appear in a single line. This option is often used with the `-f` option.

-f *field_separator*

Specifies a field separator to be used when field wrap is suppressed. The value of *field_separator* is the character that will appear as the field separator in the display. For clarity, do not choose a character that is likely to appear in a field. For example, do not use a colon as a field separator if the display will contain dates in which a colon is used to separate hours from minutes.

Servicing Pending Restore Requests

Restore operations that cannot be performed immediately are posted to the restore status table and are considered "pending." Pending restore jobs must be serviced by an operator who received mail about the pending request and services it through the `rsoper` command. An operator can usually satisfy a pending restore request by locating and installing the correct archive volume.

You can service a pending restore job, by completing the following procedure.

1. Display the restore status table by running `rsstatus` with the desired options. Note the name of the object to be restored and the label of the archive volume for it.

   NOTE If there is no label information in the status table, you may have to determine which archive volume contains the backup from which to restore by guessing and relying on other information. For example, the date of a backup may allow you to identify an archive volume.

2. Mount the archive volume.
3. Invoke `rsoper -d` *ddev* where *ddev* is the name of the device that is to read the archive. *ddev* takes the following form:

   ddevice[*:dchar*][*:dlabels*]

   dlabel must be specified if there is more than one label. It must include a device name and it may include device characteristics and volume labels.

NOTE **If the history log has been removed, all of the *ddev* fields (*ddevice, dchar, dlabels*) must be specified.**

4. After the volume is processed, check the restore status with the `rsstatus` command.

The restore service compares the information you have entered on the `rsoper` command line with the information in the restore status table and on the archive volume. If the information on the command line matches the information in the restore status table, the restore operation begins. If the information on the command line does not match that in the restore status table, the information on the command line predominates and the restore operation begins. Then the restore service attempts to resolve any restore requests that can be satisfied by the archive volume described by `-d` *ddev*.

Options to `rsoper`

There are several options that can be specified with `rsoper`. These options are used only if you need to override the information in the restore status table. Archives made with the UNIX System V Release 4.0 backup methods contain the information needed for all restore operations except those performed with the `-d` option. Thus, the following options will probably be needed only for archives made on systems running earlier releases of the UNIX system. (These archives may not contain all the information required by the Release 4.0 restore service.)

Basic Options

Three options allow you to describe the archive volume being mounted for use by a restore operation: `-t`, `-o`, and `-m`.

`-t`
Tells the service that the volume inserted in the destination device contains a table of contents for the archive.

`-o` *oname:odev*
Specifies the originating file system partition or data partition to be restored. *oname* is the name of the originating file system; the value of *oname* may be null. *odev* is the device name of the originating file system or data partition.

-m *method* Specifies that the first archive volume in the destination device was created by the backup *method* specified.

The following example illustrates the use of some of these options. Suppose the backup history log no longer contains a log of the backup operations for which archives are being requested. To obtain the desired data, you must request an incremental file restore from the /usr2 file system. To request this, enter

```
rsoper :/dev/rmt/ctape1 arc.dec79.c -m incfile -o /usr2
```

The /usr2 file system archive is found on cartridge tapes.

Restricting Restores

By default, when rsoper is invoked, the restore service attempts to complete any restore requests on the archive volume mounted. However, you can restrict restore jobs to those with specific job-IDs by using the -j option. You can also restrict restore jobs to specific user logins through the -u option.

Removing and Canceling Restore Jobs

Entries in the restore status table may be deleted for either of two reasons:

- to remove a restore request that has been satisfied
- to cancel a job that cannot be completed or cannot be serviced at all

To remove an entry for a pending restore request, type

```
rsoper -r job_IDs
```

where *job_IDs* is a list of pending restore requests that have been serviced. This command notifies the people who issued these requests that the restore operations have been done successfully and that the entries for them have been removed from the status table.

To cancel a request, type

```
rsoper -c job_IDs
```

where *job_IDs* is a list of pending restore requests to be canceled. This command notifies the people who issued these requests that the requests cannot be serviced and have been canceled.

System Restores

Full and partial system restores are altogether different from other types of restore operations. System restore operations are not done by issuing the `restore` command or any related commands; rather, they are done by rebooting the Essential Utilities cartridge tape.

Partial system restores are required when a portion of the system has become corrupted or the administrator has forgotten the root password. Full system restores are necessary when there is a new system or disk, or when you need to increase the size of the core file system data partition.

- A partial system restore replaces (overwrites) the core system files on a hard disk with those originally distributed. These files include the Essential Utilities; user files are not affected by a partial system restore.
- A full system restore erases everything on a hard disk and then loads the core system files.

To request either a full or partial system restore, you must be in single-user state, and you must mount `/usr`.

NOTE A full system restore erases everything on a disk. If you use this procedure to change partition sizes, do a complete file disk backup first.

Partial System Restore

This type of restore is useful if you have forgotten your root password or if your system has been corrupted.

During a partial system restore, certain system files are overwritten, including the terminal configuration and password files. (These files are needed to rebuild your previous system configuration.)

Use the following procedure to do a partial system restore:

1. Change directory to the root directory.
2. Change the system state to firmware mode (system state 5) by typing `init 5`.

3. Insert the first Essential Utilities cartridge tape into the cartridge tape drive.
4. Boot the operating system by selecting option `0` from the instructions displayed on your screen.
5. Initiate a partial system restore by choosing option 2 from the instructions displayed on your screen.
6. Follow the instructions on the screen to remove and insert the Essential Utilities cartridge tape. When the cartridge tape has been copied, the system will reboot from the hard disk.
7. When the system is ready, you must rebuild the system files. You can do this in either of two ways: (1) you can follow the displayed instructions to access the `sysadm syssetup` menu; or (2) you can use your own procedure (see Step 8).
8. Recover, from backup copies, any system files not automatically saved, and reconfigure the system as necessary. System files automatically saved during this procedure are found in `/var/old` and include the following:

```
/bin/ed
/bin/red
/dgn/edt_data
/etc/TIMEZONE
/etc/bkup/bkexcept.tab
/etc/bkup/bkhist.tab
/etc/bkup/bkreg.tab
/etc/bkup/rsnotify.tab
/etc/checklist
/etc/cron/.proto
/etc/cron/at.allow
/etc/cron/cron.allow
/etc/cron/queuedefs
/etc/defaults/*
/etc/device.tab
/etc/dgroup.tab
/etc/disk.tab
/etc/group
/etc/init.d/*
/etc/inittab
/etc/master.d/*
/etc/motd
/etc/passwd
/etc/profile
/etc/rc2.d/S18setuname
/etc/rstrat.dat
/etc/saf/token/_config
/etc/saf/token/_pmtab
/etc/saf/_sactab
/etc/saf/_sysconfig
/etc/shadow
/etc/shutdown.d/*
/etc/system
/etc/ttydefs
/etc/vfstab
/usr/lib/uucp/*
/usr/sadm/sysadm/menu/main.menu
/var/sadm/bkup/*/bkexcept.tab
/var/sadm/bkup/*/bkreg.tab
/var/sadm/install/admin/default
/var/sadm/install/contents
/var/spool/cron/crontabs/adm
/var/spool/cron/crontabs/root
/var/spool/cron/crontabs/sys
/var/spool/cron/crontabs/sysadm
```

If you are doing this restore operation because of a forgotten root password, copy all the above files back into their proper places and then add the new root password by typing

```
passwd root
```

If you are doing this restore operation because of system corruption, examine each of the files listed above carefully before putting them back in your system. One of these files may have caused the corruption.

After this procedure has been completed, all software packages should be reinstalled to insure that the system configuration is properly restored.

Full System Restore: Using the Default Disk Partition Size

A full system restore erases everything on a hard disk and then loads the core system files.

NOTE Because all files on hard disk are destroyed during a full system restore, a full system restore should not be performed until you have backed up (onto cartridge tape) all files you want to keep.

This procedure allows you to perform a full system restore while maintaining the default disk partition sizes. Skip to the next section if you want to perform a full system restore that changes the disk partition size.

The following is the procedure for doing a full system restore (using the default disk partition size):

1. Change directory to the root directory.
2. Change the system state to firmware mode (system state 5) by typing `init 5`.
3. Boot the operating system by selecting option `0` from the instructions displayed on your screen.
4. Initiate a full system restore by choosing option `1` from the instructions displayed on your screen.
5. Type `y` after the following prompt, as shown:

```
Use the default hard disk partitioning?
   [y, n, help]: y
```

6. Follow the instructions displayed on the screen to remove and insert the Essential Utilities cartridge tape. When the cartridge tape has been copied, the system will reboot from the hard disk.
7. When the system is ready, you must rebuild the system files. You can do this in either of two ways: (1) you can follow the displayed instructions to access the `sysadm syssetup` menu; or (2) you can use your own procedure.

Full System Restore: Changing the Disk Partition Sizes

This procedure differs from the one above because it allows you to change the disk partition sizes.

Before you begin, make sure you have calculated the number of blocks to be allocated to the various partitions, find out how much space is left on each disk, and determine the current disk partition sizes.

To determine how much space is available on a disk, run the `df -t` command; to determine the current disk partition size, issue the `prtvtoc` command and specify each disk for which you want to know the size. (Sizes reported by the `prtvtoc` command are not exact; this command rounds up all disk partition sizes reported.)

Enter these commands as follows:

```
df -t
prtvtoc /dev/rdsk/c8d0s6
prtvtoc /dev/rdsk/c8d1s6
```

Record the results.

To do a full system restore (changing the disk partition size), complete the following procedure:

1. Change directory to the root directory.
2. Change the system state to firmware mode (system state 5) by typing `init 5`.
3. Boot the operating system by selecting option `0` from the instructions displayed on your screen.
4. Initiate a full system restore by choosing option `1` from the instructions displayed on your screen.
5. Type `n` at the following prompt as shown:

   ```
   Use the default hard disk partitioning?
       [y, n, help]: n
   ```

6. The next prompt is

   ```
   How many blocks for the swap partition?
       [ (range 3500 through max) quit help ](default n):
   ```

Enter the number of blocks you want to allocate (*max* will vary according to disk size and *n* will depend on how your disk is partitioned). To determine how many blocks to allocate, use the figures you worked out beforehand. Remember, if you increase the default number of blocks in one partition, there will be fewer blocks available for the remaining partitions.

7. The system then prompts you for similar information about the next partition on the disk. Continue supplying information until you have partitioned all the blocks.

8. When the system is ready, you must rebuild the system files. You can do this in either of two ways: (1) you can follow the instructions displayed on the screen to access the `sysadm syssetup` menu; or (2) you can use your own procedure (see Step 9).

9. Recover, from backup copies, any system files not automatically saved, and reconfigure the system as necessary. The system files that are automatically saved (in `/var/old)` during this procedure include the following:

/bin/ed
/bin/red
/dgn/edt_data
/etc/TIMEZONE
/etc/bkup/bkexcept.tab
/etc/bkup/bkhist.tab
/etc/bkup/bkreg.tab
/etc/bkup/rsnotify.tab
/etc/checklist
/etc/cron/.proto
/etc/cron/at.allow
/etc/cron/cron.allow
/etc/cron/queuedefs
/etc/defaults/*
/etc/device.tab
/etc/dgroup.tab
/etc/disk.tab
/etc/group
/etc/init.d/*
/etc/inittab
/etc/master.d/*
/etc/motd
/etc/passwd
/etc/profile
/etc/rc2.d/S18setuname
/etc/rstrat.dat
/etc/saf/*token*/_config
/etc/saf/*token*/_pmtab
/etc/saf/_sactab
/etc/saf/_sysconfig
/etc/shadow
/etc/shutdown.d/*
/etc/system
/etc/ttydefs
/etc/vfstab
/usr/lib/uucp/*
/usr/sadm/sysadm/menu/main.menu
/var/sadm/bkup/*/bkexcept.tab
/var/sadm/bkup/*/bkreg.tab
/var/sadm/install/admin/default
/var/sadm/install/contents
/var/spool/cron/crontabs/adm
/var/spool/cron/crontabs/root
/var/spool/cron/crontabs/sys
/var/spool/cron/crontabs/sysadm

Quick Reference to the Restore Service

The Administrator's Tasks

- Assigning an operator to service restore operations:

  ```
  rsnotify -u user
  ```

  where *user* is the login name of the operator.

- Displaying the name of the operator assigned to service restore operations:

  ```
  rsnotify
  ```

- Initiating a restore of a data partition:

  ```
  restore -P partdev
  ```

  where *partdev* is the name of a data partition.

- Initiating a restore of a file system partition:

  ```
  restore -S oname
  ```

  where *odevice* is the name of the file system partition to be restored.

- Initiating a restore of an entire disk:

  ```
  restore -A partdev
  ```

  where *partdev* is the name of the disk to be restored.

- Restoring an object from archive volumes of a particular date:

  `restore -d` *date* or `urestore -d` *date*

  where *date* is the date of the archive to be used for the restore. The value of *date* is the same ten-character string used to specify a month, day, hour, minute, and year with the `date(1)` command: *MMddhhmmyy*.

- Restoring an object to a new disk location:

  `restore -o` *target* or `urestore -o` *target*

  where *target* is the complete pathname of the destination location on the disk.

The Operator's Tasks

- Displaying a list of all archived versions of an object:

  `restore -n` or `urestore -n`

- Displaying the complete contents of the restore status table:

  ```
  rsstatus
  ```

- Displaying a list of restore jobs that could be satisfied by a specified device type or archive volume:

  `rsstatus -d` [*dtype*][*:dlabels*]

  where *dtype* is a description of a device type (e.g., ctape1) and *dlabels* is the label of a particular archive volume (such as `bk0010`, `bk0011`).

- Displaying the status of particular `fdisk`, `fimage`, `ffile`, or `fdp` restore jobs:

  `rsstatus -j` *job_IDs*

  where *job_IDs* is a list of one or more job IDs for requested restores.

- Displaying the status of restore requests made by specific users:

  `rsstatus -u` *users*

  where *users* is a list of one or more user login names.

- Removing a pending restore request from the restore status table and designating it canceled:

  `rsoper -c` *job_IDs*

  where *job_IDs* is a list of one or more job IDs for requested restores.

- Removing a pending restore request from the restore status table and designating it complete:

  `rsoper -r` *job_IDs*

  where *job_IDs* is a list of one or more job IDs for requested restores.

- Servicing a pending restore request:

  `rsoper -d` *ddev*

  where *ddev* describes the device to be used to read the archive containing the file system or data partition to be restored.

- Suppressing field wrap and specifying an output field separator on the restore status display:

```
rsstatus -f c
```

where *c* is the character that will appear as the field separator in the display of the restore status table.

- Suppressing headers on the restore status display:

```
rsstatus -h
```

Non-Privileged Tasks

- Canceling a previously requested restore:

```
urestore -c job_ID
```

where *job_ID* is the job ID of a job to be canceled.

- Displaying the status of particular `incfile` restore jobs:

```
ursstatus -j job_IDs
```

where *job_IDs* is a list of one or more job IDs for requested restores.

- Initiating a restore of a directory:

```
urestore -D directory
```

where *directory* is the name of the directory to be restored.

- Initiating a restore of a file:

```
urestore -F filename
```

where *filename* is the name of the file to be restored.

14 Security

Introduction

The UNIX operating system provides extensive features to maintain system security. However, no computer system is secure unless good standards of administration and use are established and followed. This chapter provides details about security for UNIX System V/68 or V/88 Release 4 and describes computer center practices that can enhance the security of your computer operations.

There is no individual administration menu dedicated to system security. However, there are several tools to help maintain security that are described in this chapter. Some of these are available through the `systemsetup` and `users` menus under `sysadm`(1M). If you prefer to work at the shell level you can do so instead of using the `sysadm` interface. The following table shows the shell commands that correspond to the tasks in the `sysadm` interface. Not all security-related commands discussed in this chapter have corresponding `sysadm` menu items.

Figure 14-1: Menus and Shell Commands for Performing Some Security Related Tasks

Action to be Performed	`sysadm` Task*	Shell Command
Display password aging for a user	`users/password`	`passwd -s` *user*
Change password for user	`users/password`	`passwd` *user*
Turn on aging, set *max*	`users/password`	`passwd -x` *max user*
Turn on aging, set *min*	`users/password`	`passwd -n` *min user*
Turn off aging	`users/password`	`passwd -x -1` *user*
Lock a user's login	`users/password`	`passwd -l` *user*
Set password for administrative or system logins	`systemsetup/password`	`passwd` *admlogin-ID*

* Many of these actions may be done via the commands available for the administration of users. See `sysadm`(1M) in the *System Administrator's Reference Manual* for all available options and an explanation of their capabilities.

Each task listed above is explained later in this chapter. In addition, the *System Administrator's Reference Manual*, and *User's Reference Manual* provide manual pages for these shell commands.

Overview of Security Administration

Security is an aspect of the operation of your computer that must always be kept in mind. A machine connected to phone lines or a local network has the potential for intruders. Even an isolated machine is subject to idle browsing by its legitimate users. Consider the possible loss if a file is altered or destroyed, or if the wrong person sees it.

This chapter covers:

- Important security concepts and guidelines for the UNIX system—how to control user and group access to directories and files
- Logins and passwords—what precautions to take when changing a password (some of which may be required by company policies) and how to assign or change a password
- Assigning special administrative passwords—what are administrative logins and why they should be protected
- Logging login attempts—how to start a procedure to track unsuccessful attempts to log in to the system
- Set-UID and set-GID—how to determine if unauthorized programs conditioned to execute via an administrative login exist
- A quick reference section that lists the security-related commands discussed in this chapter

Suggestions for Making Your System Secure

The security of any system is ultimately the responsibility of all who have access to it. As the administrator of your system, you need to consider the following:

- Restrict physical access to your computer (especially if it is a small machine) so that someone does not simply walk off with it.
- Set the access permissions to directories and files so that they can be accessed only as needed by the owner, group, or others. Publicly writable directories are a security hazard. Allow them only if you have a good reason.
- Assign passwords to all logins and change them regularly. You can force them to be changed regularly by implementing password aging. Do not pick obvious passwords: six-to-eight character nonsense strings using letters and numbers are recommended over recognizable words. Remove or lock logins that are no longer needed.
- Do not keep sensitive information on a system with dial-up ports; the security of any system with dial-up ports is difficult to guarantee.
- Users who make use of the `su` command to become `root`, or any other user, can compromise the security of your system by accessing files belonging to other users without the other users' knowledge. For this reason, a log is kept on the use of the su(1) command. Check the file `/var/adm/sulog` to monitor use of this command. The format of `/var/adm/sulog` is described in Appendix B, "Directories and Files."
- Keep in mind that login directories, user profile files, and files in `/sbin`, `/usr/sbin`, and `/etc` that are writable by others are security give-aways.
- Encrypt sensitive data files. The `crypt(1)` command together with the encryption capabilities of the editors (`ed` and `vi`) provide better protection for sensitive information. The Security Administration Utilities package (domestic customers only) must be installed before you can run `crypt(1)`.
- Do not leave a logged-in terminal unattended, especially if you are logged in as `root`. If you must be away from your terminal, log off before leaving.
- Place an appropriate `umask` command in the system profile (`/etc/profile`) to set a default security level for file creation.

- As system administrator, use full pathnames for critical commands (for example /usr/bin/su instead of su).
- Don't mount a medium (such as a cartridge tape) unless the contents are trusted. These file systems may contain set-user-ID or Trojan horse (undesirable gift) programs.
- Don't add packages or programs from untrusted sources. This is the most common way of spreading computer viruses.
- For more information on network security and dialup passwords, see the "Network Services" chapter.

NOTE

The X display manager xdm, does check for valid login names and passwords. However, xdm does not perform all the other functions of /bin/login (such as doing checks on password aging) and should not be construed as a graphical interface for /bin/login. See the xdm(1) manual page for a description of xdm features and capabilities.

The xdm script /usr/lib/X11/xdm/Xstartup can be edited to permit or deny logins by superuser by invoking or commenting-out, as desired, the call to the deny_root function. As distributed, the Xstartup script allows login by superuser through xdm.

Logins and Passwords

To log in to the UNIX system, a user must enter both a login name and a password. Although logins are publicly known, passwords must be kept secret, known only to their owners. To enhance the security of your system and data, we recommend that you ask your users to change their passwords occasionally. For a high level of security, normal users should do so about every 6 weeks. System administration logins (such as `root` and `sys`) should be changed monthly or whenever a person having the root password leaves the company or is reassigned. Although voluntary compliance with this practice is desired, the UNIX operating system provides a mechanism to force compliance. This mechanism is called password aging.

Choosing a Password

Most security breakins of computer systems involve guessing the person's password. While the `passwd`(1) command has some criteria for making sure the password is hard to obtain using mechanical means, a clever person can sometimes guess a password just by knowing something about the person and habits.

- Bad choices: names of family members or pets, car license numbers or telephone numbers, Social Security number, employee number, names related to a person's hobbies or interest, words currently popular in the media (such as slang from TV shows), seasonal themes (such as "turkey" in November or "superbowl" in January). Also, any variations on this by substitution or addition of a special character.
- Good choices: puns, words in a foreign language, a word reversed (yekrut for turkey), or a nonsense word made up of the first letter of every word in a phrase (Mhallifwwas - Mary had a little lamb, its fleece was white as snow).
- Add a non-alphabetic character in the middle of the password (be careful about magic characters such as `#` and `@` and control characters). Substitute a number for a similar letter (for example `0` for `o`, `3` for `e`, `1` for `l` or `i`).
- Remember that the clever person is aware of the above as well.

Password Aging

The password aging mechanism forces users to change their passwords on a periodic basis. Provisions are made to prevent a user from changing a new password before a specified interval. Password aging is selectively applied to logins by using the `passwd(1)` command. If you require more access control than what is provided by password aging, you can also change `/etc/profile` to require a second access code as part of the login process (see the "User and Group Management" chapter).

The password aging information requires setting the following parameters for each login:

min the minimum number of days required between password changes

max the maximum number of days the password is valid

warn the number of days before the password must change that a warning message will begin appearing to the user

As a result of using `passwd(1)`, the following parameter will also have changed:

lastchanged the number of days between January 1, 1970, and the date that the password was last modified

Displaying Password Information

Password and aging information can be displayed using the `-s` option of the `passwd` command. For example, if you type:

```
passwd -s sms
```

the following information will appear if there is password aging.

```
sms  PS  06/23/90  14  84  7
```

If password aging is not turned on, only the first two fields will appear. The six fields contain the following information:

- login name (sms)

- password status (PS)
  The following strings may appear:

Status	
Type	Symbol
No password for this login	NP
Login is locked	LK
Anything else	PS

- date the password was last changed (06/23/90)
- minimum number of days after *lastchanged* before the user can change the password (14)
- maximum number of days after *lastchanged* until the user will be forced to change the password (84)
- number of warning days before the password must be changed (7)

Thus, the information obtained for this example shows that there is a password for the login sms that cannot be changed before July 6 and that must be changed by September 15, 1990. On September 8, 1990, this user will begin seeing a warning message that the password will expire and should be changed.

To display the password status and aging information for all users on your system, use the -a option to the passwd command, instead of specifying individual logins:

```
passwd -s -a
```

Only a privileged user can use the -a option for the passwd command.

Sample passwd Commands

Password administration can be set up in a variety of ways to meet the needs of different organizations. Some examples are discussed in the following sections.

1. Change someone's password

   ```
   passwd login_name
   ```

   Because this command is run by the administrator, no prompt for the old password is given. Instead, as a privileged user, the administrator is prompted to enter the new password. The password is not displayed as it is typed. The command requires you to enter the password twice to assure it is typed accurately.

2. Turn on aging, set *max* to 84 and *min* to 7 days, respectively.

   ```
   passwd -x 84 -n 7 login_name
   ```

3. Force a user to change the password at the next login session.

   ```
   passwd -f login_name
   ```

4. Lock a password, set *max* to 7 and *min* to 10 days.

   ```
   passwd -x 7 -n 10 login_name
   ```

   Because *min* is greater than *max*, the password is locked and cannot be changed but the user can still log into the system. Only `root` can change this password.

5. Turn off aging by setting *max* to negative one.

   ```
   passwd -x -1 login_name
   ```

6. Warn the user starting 14 days before the password is set to expire that a new password must be chosen.

   ```
   passwd -w 14 login_name
   ```

   Starting `14` days before *max*, the user will see the message:

   ```
   Your password will expire in 14 days
   ```

   Each day, the number will decrease until the password expires or the user changes the password.

For more information, see `passwd`(1) in the *User's Reference Manual*.

Dial-up Passwords

A dial-up password is an additional password you must enter before you are allowed access to the system. This built-in UNIX system capability can be added to enhance the system's security.

A dial-up password can be changed only by the system administrator, who, to ensure the integrity of the system, should change the password about once a month, usually on the first of the month.

Creating a Dial-up Password

When you first establish a dial-up password, make certain to remain logged in on at least one terminal while testing the password on a different terminal. If you make a mistake while installing the extra password and log off to test the new password, you might not be able to log back on. If you are still logged in on another terminal, you can go back and fix your mistake.

To institute dial-up password protection on your system, you need to create two files:

- /etc/dialups, containing a list of the terminal devices on which dial-up password will be required, and
- /etc/d_passwd, containing the encrypted password and the login programs that require the user to enter a password before they can be invoked.

The modes of the files should be 600, and they should have owner and group set to root.

The /etc/dialups file is a list of terminal devices. It should look similar to this:

```
/dev/term/21
/dev/term/22
/dev/term/23
```

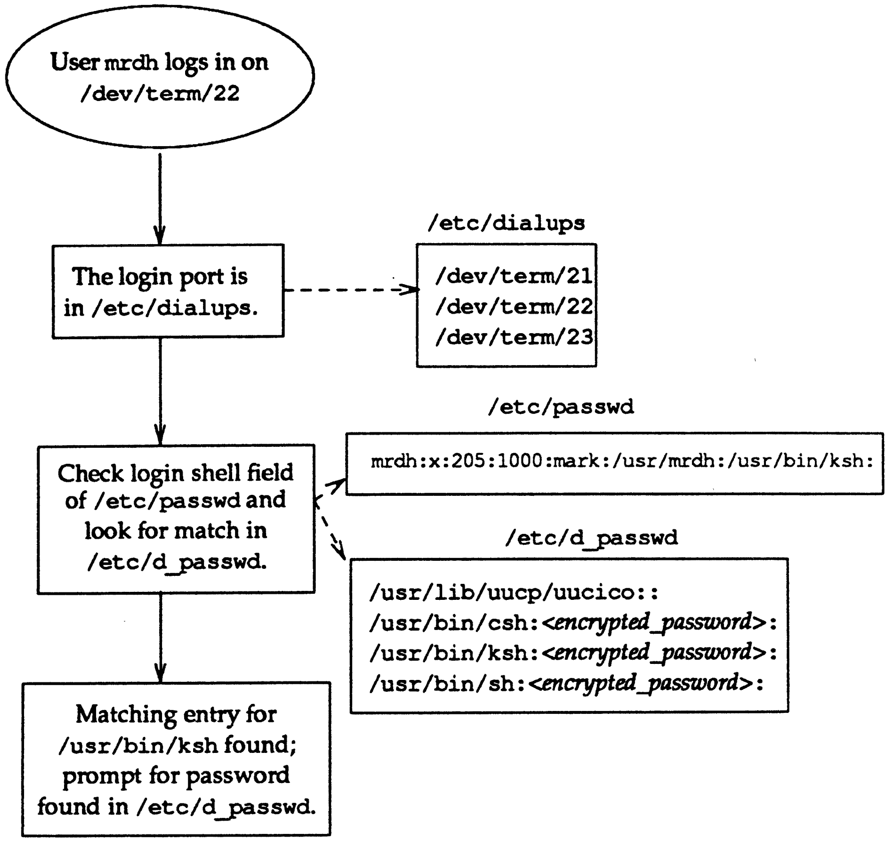
It lists all ports that require the extra security provided by a dial-up password. These are actually modem ports on the system.

The /etc/d_passwd file looks similar to this:

```
/usr/lib/uucp/uucico::
/usr/bin/csh:encrypted_password:
/usr/bin/ksh:encrypted_password:
/usr/bin/sh:encrypted_password:
```

When a user attempts to log in on any of the ports listed in /etc/dialups, the login program looks at /etc/d_passwd and may prompt the user for a second password. Whether a second password is asked for or not depends upon the login shell that is specified in the shell field of the user's login entry in the /etc/passwd file, and whether or not this login shell has an entry in /etc/d_passwd. The basic sequence is best illustrated by the following figure.

Figure 14-2: Basic Dialup Password Sequence


User mrdh logs in on /dev/term/22
The login port is in /etc/dialups.
Check login shell field of /etc/passwd and look for match in /etc/d_passwd.
Matching entry for /usr/bin/ksh found; prompt for password found in /etc/d_passwd.
/etc/dialups
/etc/passwd
/etc/d_passwd

```
/dev/term/21
/dev/term/22
/dev/term/23
```

```
mrdh:x:205:1000:mark:/usr/mrdh:/usr/bin/ksh:
```

```
/usr/lib/uucp/uucico::
/usr/bin/csh:<encrypted_password>:
/usr/bin/ksh:<encrypted_password>:
/usr/bin/sh:<encrypted_password>:
```

Because most users will be running a shell when they log in, all shell programs should have entries in /etc/d_passwd. Such programs include uucico, sh, ksh, and csh. If some users run something else as their login shell, include it in the file, too.

Each entry in /etc/d_passwd has two fields, demarcated by semicolons. The first field is the login program that will require a dial-up password. The second field contains the encrypted password and will be discussed later.

In our example above, the uucico program does not have anything in this field. This allows remote systems to call your system, via uucp, without having to know the dial-up password. The uucp subsystem is relatively secure, if properly administered, and usually does not need a dial-up password. However, if you are concerned about security here, it would be wise to require a password for uucp, too. The dial-up password need not be the same for uucp as for ksh, sh, etc.

The entry for /usr/bin/sh defines the default dial-up password. If the user's login program is not found in /etc/d_passwd, or if the login shell field in /etc/passwd is null, this password entry will be used.

If there is no entry for /usr/bin/sh, users whose shell field in /etc/passwd is null or does not match any entry in /etc/d_passwd will not be prompted for a dial-up password.

Note that the /etc/d_passwd file could be used to temporarily disable dial-up logins by putting an entry such as:

```
/usr/bin/sh:*:
```

by itself in the file.

A dial-up password can be created by following these steps

1. Using useradd or sysadm's "add user" menu, add a "dummy" user, say dummy.
2. Give it a password with the passwd(1) command or the sysadm form.
3. Capture the encrypted password from /etc/shadow by typing

   ```
   grep dummy /etc/shadow > dummy.temp
   ```

4. Using userdel or the appropriate sysadm form, delete the dummy user.
5. Edit dummy.temp and delete all fields except the encrypted password. Fields are delimited with a colon (:) and the password is the second field.
6. Edit the /etc/d_passwd file and read the encrypted password from your dummy.temp file in as the password field.

You should follow these steps every time you change the dial-up password.

Locking Unused Logins

If a login is not used or needed, you should do one of two things:

- use userdel(1M) to delete the login (see the "User and Group Management" chapter)
- disable (lock) the login

A login is locked by running the passwd command with the -l option.

```
    passwd -l login_name
```

The string LK is displayed in the password field when the passwd -s command is issued. This shows that the login is locked; the user will not be allowed to log in. To unlock this login, the administrator must run passwd for the user.

Login Authorization

UNIX System V/68 or V/88 Release 4 provides two other login control mechanisms: login deactivation and expiration.

Login Expiration

As an administrator, you may want to create a login that can only be used for a short time. After this time, this login would "expire" and the owner would no longer be able to use it to log in without the administrator's intervention. To set the expiration date, type

```
    useradd -e mm/dd/yy login_name
```

where *mm/dd/yy* is an absolute date:

mm is a one or two-digit number representing the month (1-12)

dd is a one or two-digit number representing the day of the month (1-31) *yy* is a two-digit number representing the year (00-99)

Setting or changing the expiration date may also be done with `usermod`(1M).

If a login's expiration date must be extended, you can do so by running `usermod` again with a new date as the argument to `-e`.

Deactivating a Login

It may be useful to know that a user has not logged into the system for a while. By setting the "inactive" field, a login will be considered inactive if a user has not logged in for a set number of days. If the user does not log in for this number of days, the login becomes inactive and the user will be prevented from logging in until the administrator resets the login. To set an inactive field, type:

```
usermod -f n login_name
```

where *n* is the number of days after *lastlogin* after which the login will be considered inactive.

Setting the inactive field may also be done when adding a new user to your system by using `useradd`(1M).

If a login has become inactive, it can be reactivated by completing the following steps:

1. Check the current inactive value

   ```
   logins -a -l login_name
   ```

   The first number on the second line of the output of this command is the inactive field.

2. Set the inactive field to zero.

   ```
   usermod -f 0 login_name
   ```

3. Have the affected user log in again so that the *lastlogin* date will be updated.

4. Reset the inactive field once again.

   ```
   usermod -f n login_name
   ```

   You can use the value found in Step 1 for *n*.

Displaying Login Information

The amount of warning time, number of days before inactivation and the expiration date can be displayed using the `logins` command. This command displays a variety of information about the users on your system. Of interest here are the `-a` and `-x` options. For information on other options and uses of this command, see the "User and Group Management" chapter and the `logins(1M)` command in the *System Administrator's Reference Manual*.

Login Logging

A logging mechanism exists that logs unsuccessful attempts to access your UNIX system. After a person makes five consecutive unsuccessful attempts to log in, all these attempts are logged in the file /var/adm/loginlog.

loginlog

To turn on the mechanism that logs unsuccessful attempts to access the system, the administrator must create the file /var/adm/loginlog. If this file exists and five consecutive unsuccessful login attempts occur, all are logged in loginlog(4) and then login sleeps for 20 seconds before dropping the line. If a person makes fewer than five unsuccessful attempts, none of them are logged.

If loginlog does not exist, five failed login attempts will still cause the system to sleep for 20 seconds and drop the line, but nothing will be logged.

The default status is for this text file not to exist and for logging to be off. To enable logging, create the log file with read and write permission for root only.

1. Reset the default file creation privileges in a separate shell.

   ```
   /bin/sh
   umask 066
   ```

2. Create the loginlog file.

   ```
   > /var/adm/loginlog
   ```

3. Set the group to sys.

   ```
   chgrp sys /var/adm/loginlog
   ```

4. Change the ownership of the file to root.

   ```
   chown root /var/adm/loginlog
   ```

5. Return from the newly created shell level.

   ```
   exit
   ```

This file may grow in size quickly. To use this information and to prevent the file from getting too large, it is important to check and to clear the contents of the `loginlog` file occasionally. A large number of lines in a short amount of time in this file may suggest an attempt to break into the system. For more information about this file, see `loginlog`(4) in the *System Administrator's Reference Manual*.

Last Login Time

When a user logs into the system, the time the login was last used will be displayed. We recommend that users check this time to make sure that it corresponds to the time they actually did log in. If it does not, an unauthorized use of that user's login may have occurred.

Recording `su` Use

One way to record all use of the `su` command is to print a message on the system console each time the command is run. To do this, add the line

```
        CONSOLE=/dev/console
```

to `/etc/default/su`.

Special Administrative and System Logins

There are two familiar ways to access the system: via either a conventional user login or the root login. If these were the only two ways to access the system, however, effective use of the system would have to be curtailed (because root would own many directories) or many users would have to know the root password (a bad security risk) or the system would be wide open (because root would own few directories). All these conditions are undesirable.

The solution to a good mix of system use and system security is available to you with the use of special system logins and administrative commands that can be password-protected (see the "System Setup" chapter for information on doing this). There are two types of special logins:

administrative These commands, which are also logins, perform functions that might be needed by the users on your computer.

system These logins allow privileges to be split into smaller domains so that fewer people have access to the entire system.

We recommend that all the following logins be password protected.

Figure 14-3: Administrative Logins and Uses

Function	UID	Use
setup	0	Set up the computer. Once the machine is set up, you do not want anyone doing it again without your knowledge.
sysadm	0	Allows access to administrative functions that do not require a user to log in as root.
powerdown	0	Power the computer down.
checkfsys	0	Begin a file system check on the specified file systems.
makefsys	0	Make a new file system on the specified media.
mountfsys	0	Mount the specified file system for use.
umountfsys	0	Unmount the specified, previously mounted file system.
sync	67	Synchronize disk cache and disks.
nobody	60001	Root users accessing the system via an NFS mounted file system.

Figure 14-3: Administrative Logins and Uses (continued)

Function	UID	Use
noaccess	60002	Users accessing the system via an NFS mounted file system whose UID on the client system is invalid on the server system.

The commands above allow access to selected directories and system functions. They may be used as login names at the `login` prompt as well as commands. If you log in to the system with one of these names, the system executes the command after login and exits to the `login` prompt once you quit or complete the function performed by the command.

Most of these administrative functions allow a user access to critical portions of the operating system. Therefore, it is recommended that you assign passwords to the commands above. Once you assign passwords to these commands, any user attempting to log on to your computer using one of these commands as a login (and any user attempting to execute one of these commands from the shell) is prompted for the password.

It is recommended that the passwords to the following system logins be distributed on a need to know basis.

Figure 14-4: System Logins and Uses

Login	UID	Use
root	0	Has no restrictions and overrides all other logins, protections, and permissions. It allows the user access to the entire system. The password for the `root` login should be very carefully protected.
powerdown	0	Shuts machine down.
daemon	1	System daemon login; controls background processing.
bin	2	Owns most of the commands.

Figure 14-4: **System Logins and Uses** (continued)

Login	UID	Use
sys	3	Owns many system files.
adm	4	Owns certain administrative files.
uucp	5	Owns the object and spooled data files for uucp.
listen	7	Network listener service.
nuucp	10	Used by remote machines to log in to the system and start file transfers.
service	20	Customer service login.
lp	71	Owns the object and spooled data files for lp.

Assigning Special Administrative Passwords

After you have set up your system you should assign passwords to the special administrative and system logins. To do so, refer to the "System Setup" chapter.

Password Recovery

Limiting the number of people that know the `root` password is an important part of maintaining system security. Ideally, few people will know the password for this privileged login. However, when fewer people know the `root` password, the chances of losing or forgetting this password will increase. Make every effort to remember or discover the root password before performing this procedure.

Forgotten Root Password Recovery

If you cannot recover your `root` password, call your support hotline.

File Protection

Because the UNIX operating system is a multi-user system, you usually do not work alone in the file system. System users can follow pathnames to various directories and read and use files belonging to one another, as long as they have permission to do so.

If you own a file, you can decide who has the right to read it, write in it (make changes to it), or, if it is a program, to execute it. You can also restrict permissions for directories. When you grant execute permission for a directory, you allow the specified users to change directory to it and list its contents with the `ls(1)` command. Only the owner or a privileged user can define the following:

- which users have permission to access data
- which types of permission they have (that is, how they are allowed to use the data)

This section introduces types of files and discusses file protection.

File Types

When you display the contents of a directory with the `ls -l` command the first column of output describes the "mode" of the file. This information tells you not only what type of file it is, but who has permission to access it. This first field is 10 characters long. The first character defines the file type and can be one of the following types:

Figure 14-5: File Types

File Types	
Type	Symbol
Text, programs, etc.	-
Directories	d
Character special	c
Block special	b
FIFO (named pipe) special	p
Symbolic links	l

For more information on these types, see the "Storage Device Management" chapter or `ls(1)` in the *User's Reference Manual*.

File Access Permissions

In the first field of the `ls -l` output, the next nine characters are interpreted as three sets of three bits each. The first set refers to the owner's permissions; the next to permissions of members in the file's group; and the last to all others. Within each set, the three characters show permission to read, to write, and to execute the file as a program, respectively. For a directory, "execute" permission is interpreted to mean permission to search the directory for a specified file.

The permissions are as follows:

Figure 14-6: File Access Permissions

Permissions	
Explanation	Symbol
The file is readable.	r
The file is writable.	w
The file is executable.	x
This permission is *not* granted.	-
Mandatory locking will occur during access. (The set-group-ID bit is on and the group execution bit is off.)	l
The set-user-ID or set-group-ID bit is on, and the corresponding user or group execution bit is also on.	s
The set-user-ID bit is on and the user execution bit is off.	S
The sticky and the execution bits for `other` are on.	t
The sticky bit is turned on, and the execution bit for `other` is off.	T

Figure 14-7: Directory Access Permissions

Permissions	
Explanation	Symbol
The directory is readable.	r
The directory may be altered (files may be added or removed).	w
The directory may be searched. (This permission is required to `cd` to the directory.)	x
File removal from a writable directory is limited to the owner of the directory or file unless the file is writable.	t

For more information, refer to `ls(1)` and chmod(1) in the *User's Reference Manual*.

Setting a default `umask`

When a file is created its default permissions are set. These default settings may be changed by placing an appropriate `umask` command in the system profile (`/etc/profile`).

Figure 14-8: `umask(1)` Settings for Different Security Levels

Level of Security	umask	Disallows
Permissive	0002	w for others
Moderate	0027	w for group, rwx for others
Severe	0077	rwx for group and others

Set-UID and Set-GID

The set-user identification (set-UID) and set-group identification (set-GID) bits must be used carefully. These bits are set through the chmod(1) command and can be specified for any executable file. When any user runs an executable file that has either of these bits set, the system gives the user the permissions of the owner (or group) of the executable.

System security can be compromised if a user copies another program onto a file with `-rwsrwxrwx` permissions. For example, if the switch user (`su`) command has the write access permission allowed for others, anyone can copy the shell onto it and get a password-free version of `su` with no `sulog` entry being made. Experience has shown that people who have had root permissions once, tend to keep such a file around "just in case." The following paragraphs provide a few examples of command lines that can be used to identify the files with a set-UID. A vigilant system administrator will check the system for potential problems periodically and investigate any unusual occurrences.

For more information about the set-UID and set-GID bits, see chmod(1) and chmod(2).

Check Set-UIDs

The following command line lists all set-UID programs owned by `root`. The results are saved in a file in `/tmp`. All mounted paths are checked by this command starting at `/`. Any surprises in the output should be investigated. Search time is dependent on the number of entries in the directory to be searched.

This program can be run for `sys`, `bin`, and `mail`, as well.

```
# find / -user root -perm -4000 -exec ls -ldb {} \; > /tmp/ckprm
# cat /tmp/ckprm
-r-sr-xr-x   1 root     bin         38836 Aug 10 16:16 /usr/bin/at
-r-sr-xr-x   1 root     bin         19812 Aug 10 16:16 /usr/bin/crontab
---s--x--x   1 root     sys         46040 Aug 10 15:18 /usr/bin/ct
-r-sr-xr-x   1 root     sys         12092 Aug 11 01:29 /usr/lib/mv_dir
-r-sr-sr-x   1 root     bin         33208 Aug 10 15:55 /usr/lib/lpadmin
-r-sr-sr-x   1 root     bin         38696 Aug 10 15:55 /usr/lib/lpsched
---s--x---   1 root     rar         45376 Aug 18 15:11 /usr/rar/bin/sh
-r-sr-xr-x   1 root     bin         12524 Aug 11 01:27 /usr/bin/df
-rwsr-xr-x   1 root     sys         21780 Aug 11 01:27 /usr/bin/newgrp
-r-sr-sr-x   1 root     sys         23000 Aug 11 01:27 /usr/bin/passwd
-r-sr-xr-x   1 root     sys         23824 Aug 11 01:27 /usr/bin/su
#
```

In this example, an unauthorized user (rar) has made a personal copy of /usr/bin/sh and has made it set-UID to root. This means that rar can execute /usr/rar/bin/sh and become the privileged user.

If you want to save this output for future reference, move the file out of /tmp.

Check Set-UIDs by File System

The command line entry in the example below shows the use of the ncheck command to examine the /usr file system (/dev/dsk/c1d0s2, assuming a single-disk system with default partitioning) for files with a set-UID. The normal output of the ncheck -s command includes special files. The -F tells ncheck that it should expect an s5 File System Type. If you are using some other file system type, refer to the "File System Administration" chapter. The output of the modified ncheck is used as an argument to the ls command. In this example, the complete pathnames for the files start with /usr. /usr is not part of the ncheck output but must be added (using sed(1)) for the ls to work. The use of the ls command is possible only if the file system is mounted.

```
# ls -l `ncheck -F s5 -s /dev/dsk/c1d0s2 | cut -f2 | sed 's:^:/usr:'`
-r-sr-xr-x   1 root     bin        72579 Mar  3 07:25 /usr/bin/at
-r-sr-xr-x   1 root     bin        33608 Mar  3 07:25 /usr/bin/atq
-r-sr-xr-x   1 root     bin        23040 Mar  3 07:25 /usr/bin/atrm
-r-sr-xr-x   1 root     bin        28424 Mar  3 07:25 /usr/bin/crontab
---s--x--x   1 root     uucp       74762 Mar  6 11:15 /usr/bin/ct
---s--x--x   1 uucp     uucp       83346 Mar  6 11:15 /usr/bin/cu
-r-sr-xr-x   1 root     bin        29370 Mar  3 10:44 /usr/bin/df
-r-xr-sr-x   1 bin      sys        11990 Mar 14 12:34 /usr/sbin/fusage
-r-xr-sr-x   1 bin      sys        36068 Mar  3 01:37 /usr/bin/ipcs
-r-sr-xr-x   1 root     bin        34514 Mar  3 10:46 /usr/bin/login
-r-xr-sr-x   2 bin      mail       88724 Mar  3 10:46 /usr/bin/mail
-r-xr-sr-x   1 bin      mail       85034 Mar  3 10:54 /usr/bin/mailx
-rwsr-xr-x   1 root     sys         8718 Mar  3 10:44 /usr/bin/newgrp
-r-sr-sr-x   1 root     sys        21154 Mar  3 10:44 /usr/bin/passwd
-r-sr-xr-x   1 root     bin        24202 Mar  3 10:46 /usr/bin/ps
-r-xr-sr-x   2 bin      mail       88724 Mar  3 10:46 /usr/bin/rmail
-rwsr-xr-x   1 root     sys        17526 Mar  3 10:44 /usr/bin/sacadm
-r-xr-sr-x   1 bin      sys        39508 Mar  3 02:50 /usr/sbin/sadp
-r-sr-xr-x   1 root     root       35128 Mar 14 13:07 /usr/bin/su
---s--x--x   1 uucp     uucp       78668 Mar  6 11:15 /usr/bin/uucp
---s--x--x   1 uucp     uucp       36628 Mar  6 11:15 /usr/bin/uuglist
---s--x--x   1 uucp     uucp       32254 Mar  6 11:16 /usr/bin/uuname
---s--x--x   1 uucp     uucp       77550 Mar  6 11:16 /usr/bin/uustat
---s--x--x   1 uucp     uucp       81424 Mar  6 11:16 /usr/bin/uux
-r-xr-sr-x   1 bin      tty        14438 Mar  3 10:47 /usr/bin/write
-r-sr-xr-x   1 root     bin        20106 Mar  3 11:13 /usr/lib/mail/mail_pipe
-rwxr-sr-x   1 bin      mail        3268 Mar  3 11:04 /usr/lib/mailx/rmmail
-r-sr-xr-x   1 root     sys        15864 Mar  3 10:52 /usr/lib/mv_dir
---s--x--x   1 root     bin        26801 Mar  3 02:46 /usr/lib/pt_chmod
-r-xr-sr-x   1 bin      sys        16682 Mar  3 02:52 /usr/lib/sa/sadc
---s--x--x   1 uucp     uucp      150868 Mar  6 11:17 /usr/lib/uucp/uucico
---s--x--x   1 uucp     uucp       51748 Mar  6 11:17 /usr/lib/uucp/uusched
---s--x--x   1 uucp     uucp       93294 Mar  6 11:18 /usr/lib/uucp/uuxqt
-r-sr-xr-x   1 root     sys        23824 Mar 11 01:27 /usr/rar/bin/su
-r-xr-sr-x   1 bin      tty        17488 Mar  3 10:43 /usr/sbin/wall
-r-xr-sr-x   1 bin      sys        11274 Mar  3 09:25 /usr/sbin/whodo

#
```

In this example, the /usr/rar/bin/su should be investigated.

Security Audit

After the system has been fully configured, the system administrator should perform a check for SETUID/SETGID files and devices on `root` and `/usr` using one of the above procedures. The output from this should be saved on some medium (for example, on a cartridge tape) and printed in hard-copy. Periodically, rerun the procedure, compare its results with the previous output, and investigate changes such as additions, deletions, or changes in size or date.

This is an example output:

```
# make sure /usr is mounted!
ls -l `ncheck -F s5 -s /dev/dsk/c8d0s0 | cut -f2 | grep -v dev` > sec.audit
ls -l `ncheck -F s5 -s /dev/dsk/c8d0s2 | cut -f2 | sed -e 's:^:/usr:'` >> sec.audit
```

Quick Reference to Security Procedures

The following commands allow `root` to do the tasks described in this chapter.

- Display password and aging information for a user.

  `passwd -s login_name`

- Display password and aging information for all users.

  `passwd -s -a`

- Change a user's password.

  `passwd` *login_name*

- Turn on aging, set *min* and *max.*

  `passwd -x` *max* `-n` *min login_name*

- Force a user to change the password at the next login session.

  `passwd -f` *login_name*

- Turn off password aging.

  `passwd -x -1` *login_name*

- Set the warning period before a user's password is to expire to one week.

  `passwd -w 7` *login_name*

- Lock a user's login.

  `passwd -l` *login_name*

- Set the expiration date for a user.

  `usermod -e` *mm/dd/yy login_name*

- Set the number of days a login is allowed to be inactive before it is locked.

  `usermod -f` *n login_name*

- Display inactive, warning, and expiration login information for a user.

  ```
  logins -a -l login_name
  ```

- Turn on login logging.

  (See the five-step procedure in the section "Login Logging.")

- List all set-UID programs owned by root.

  ```
  find / -user root -perm -4000 -exec ls -ldb {} \;
  ```

- List all files with a set-UID for the root file system.

  ```
  ncheck -s /dev/dsk/c1d0s0
  ```

- List all files with a set-UID for the `usr` file system.

  ```
  ncheck -s /dev/dsk/c1d0s2
  ```