Summary

The command line is the key to Linux. Even if you prefer GUI tools to text-mode tools, understanding text-mode commands is necessary to fully manage a Linux system. This task begins with the shell, which accepts commands you type and displays the results of those commands. In addition, shells support linking programs together via pipes and redirecting programs' input and output. These features enable you to perform complex tasks using simple tools by having each program perform its own small part of the task. This technique is frequently used with Linux text filters, which manipulate text files in various ways—sorting text by fields, merging multiple files, and so on.

Exam Essentials

Summarize features that Linux shells offer to speed up command entry. The command history often enables you to retrieve an earlier command that's similar or identical to the one you want to enter. Tab completion reduces typing effort by letting the shell finish long command names or filenames. Command-line editing lets you edit a retrieved command or change a typo before committing the command.

Describe the purpose of the `man` command. The `man` command displays the manual page for the keyword (command, filename, system call, or other feature) that you type. This documentation provides succinct summary information that's useful as a reference to learn about exact command options or features.

Explain the purpose of environment variables. Environment variables store small pieces of data—program options, information about the computer, and so on. This information can be read by programs and used to modify program behavior in a way that's appropriate for the current environment.

Describe the difference between standard output and standard error. Standard output carries normal program output, whereas standard error carries high-priority output, such as error messages. The two can be redirected independently of one another.

Explain the purpose of pipes. Pipes tie programs together by feeding the standard output from the first program into the second program's standard input. They can be used to link together a series of simple programs to perform more complex tasks than any one of the programs could manage.

Describe the filter commands. The various simple filter commands allow the manipulation of text. These commands accomplish tasks of various types, such as combining files, transforming the data in files, formatting text, displaying text, and summarizing data.

Summarize the structure of regular expressions. Regular expressions are strings that describe other strings. They can contain normal alphanumeric characters, which match the exact same characters in the string they are describing, as well as several special symbols and symbol sets that match multiple different characters. The combination is a powerful pattern-matching tool used by many Linux programs.

```
[-G gidlist] [-J projidlist] [-t termlist] [-T taskidlist]
[-c ctidlist] [-z zoneidlist] [pattern]
```

The `pkill` command allows you to kill one or more processes based on usernames, user IDs, group IDs, and other features as well as using a matching regular expression. The `pkill` command was introduced in the Solaris operating system, but it has been ported to the Linux environment and is gaining in popularity.

While the `pkill` command is extremely versatile, with that versatility comes danger. Be extremely careful if you're using regular expressions to match process names—it's very easy to match the wrong process names inadvertently!

Summary

Linux provides numerous tools to help you manage software. Most distributions are built around the RPM or Debian package systems, both of which enable installation, upgrade, and removal of software using a centralized package database to avoid conflicts and other problems that are common when no central package database exists. You can perform basic operations on individual files or, with the help of extra tools such as Yum and APT, keep your system synchronized with the outside world, automatically or semiautomatically updating all of your software to the latest versions.

No matter how you install your software, you may need to manage shared libraries. These software components are necessary building blocks of large modern programs, and, in the best of all possible worlds, they operate entirely transparently. Sometimes, though, shared libraries need to be upgraded or the system configuration changed so that programs can find the libraries. When this happens, knowing about critical configuration files and commands can help you work around any difficulties.

Beyond managing packages and libraries, Linux software management involves manipulating processes. Knowing how to manipulate foreground and background processes, adjust process priorities, and kill stray processes can help you keep your Linux system working well.

Exam Essentials

Identify critical features of RPM and Debian package formats. RPM and Debian packages store all of the files for a given package in a single file that also includes information about what other packages the software depends on. These systems maintain a database of installed packages and their associated files and dependencies.

Describe the tools used for managing RPMs. The `rpm` program is the main tool for installing, upgrading, and uninstalling RPMs. This program accepts operations and options that tell it precisely what to do. The Yum utility, and particularly its `yum` command, enables installation of a package and all its dependencies via the Internet rather than from local package files.

Describe the tools used for managing Debian packages. The `dpkg` program installs or uninstalls a single package or a group of packages that you specify. The `apt-get` utility retrieves programs from installation media or from the Internet for installation, and it can automatically upgrade your entire system. The `dselect` program serves as a menu-driven interface to `apt-get`, enabling you to select programs that you want to install from a text-mode menu.

Summarize tools for extracting files and converting between package formats. The `rpm2cpio` program can convert an RPM file to a `cpio` archive, enabling users of non-RPM systems to access files in an RPM. The `alien` utility can convert in any direction between Debian packages, RPMs, Stampede packages, and tarballs. This enables the use of packages intended for one system on another.

Summarize the reasons for using shared libraries. Shared libraries keep disk space and memory requirements manageable by placing code that's needed by many programs in separate files from the programs that use it, enabling one copy to be used multiple times. More generally, libraries enable programmers to use basic "building blocks" that others have written without having to reinvent code constantly.

Describe methods available to change the library path. The library path can be changed system wide by editing the `/etc/ld.so.conf` file and then typing `ldconfig`. For temporary or per-user changes, directories may be added to the path by placing them in the `LD_LIBRARY_PATH` environment variable.

Explain the difference between foreground and background processes. Foreground processes have control of the current terminal or text-mode window (such as an `xterm`). Background processes don't have exclusive control of a terminal or text-mode window but are still running.

Describe how to limit the CPU time used by a process. You can launch a program with `nice` or use `renice` to alter its priority in obtaining CPU time. If a process is truly out of control, you can terminate it with the `kill` command.

As with any filesystems that you want to mount, you must provide mount points—that is, create empty directories—for user-mountable media. Removable media are usually mounted in subdirectories of `/mnt` or `/media`.

Many modern distributions include auto-mount facilities that automatically mount removable media when they're inserted. These tools typically create mount points in `/media` and create icons on users' desktops to enable easy access to the media. This configuration produces effects that are familiar to users of Windows and Mac OS.

The `credentials` option for the `/other/win` mount point in Listing 3.1 deserves greater elaboration. Ordinarily, most SMB/CIFS shares require a username and password as a means of access control. Although you can use the `username=`*`name`* and `password=`*`pass`* options to `smbfs` or `cifs`, these options are undesirable, particularly in `/etc/fstab`, because they leave the password vulnerable to discovery—anybody who can read `/etc/fstab` can read the password. The `credentials=`*`file`* option provides an alternative—you can use it to point Linux at a file that holds the username and password. This file has labeled lines:

```
username=hschmidt
password=yiW7t9Td
```

Of course, the file you specify (`/etc/creds` in Listing 3.1) must be well protected—it must be readable only to `root` and perhaps to the user whose share it describes.

Summary

Most Linux tools and procedures provide a layer around the hardware, insulating you from the need to know too many details. Nonetheless, sometimes you have to dig in and configure hardware directly. Firmware settings can control onboard devices such as hard disk controllers and USB ports. USB and SCSI devices have their own quirks, and USB in particular is quickly evolving.

Hard disks are one class of hardware that's likely to require more attention than most. Specifically, you must know how to create partitions and prepare filesystems on those partitions. These tasks are necessary when you install Linux (although most distributions provide GUI tools to help guide you through this task during installation), when you add a hard disk, or when you reconfigure an existing system. You should also know something about boot managers. These programs help get Linux up and running when you turn on a computer's power, so they're unusually critical to Linux operation.

Filesystem management is basic to being able to administer or use a Linux system. The most basic of these basic tasks are filesystem tasks—the ability to mount filesystems, check their health, and repair ailing filesystems. Once a filesystem is mounted, you may want to check it periodically to see how full it is, lest you run out of disk space.

Exam Essentials

Summarize BIOS and EFI essentials. The BIOS and EFI provide two important functions: First, they configure hardware—both hardware that's built into the motherboard and hardware on many types of plug-in cards. Second, they begin the computer's boot process, passing control on to the boot loader in the MBR or EFI partition in GPT-formatted disks. The BIOS is currently being retired in favor of EFI, which performs these tasks on modern computers.

Describe what files contain important hardware information. There are many files under the `/proc` filesystem. Many of these files have been mentioned throughout this chapter. Familiarize yourself with these files, such as `/proc/ioports`, `/proc/interrupts`, `/proc/dma`, `/proc/bus/usb`, and others.

Explain Linux's model for managing USB hardware. Linux uses drivers for USB controllers. These drivers in turn are used by some device-specific drivers (for USB disk devices, for instance) and by programs that access USB hardware via entries in the `/proc/bus/usb` directory tree.

Summarize how to obtain information about PCI and USB devices. The `lspci` and `lsusb` programs return information about PCI and USB devices, respectively. You can learn manufacturers' names and various configuration options by using these commands.

Identify common disk types and their features. PATA disks were the most common type on PCs until about 2005. Since then, SATA disks, which are more easily configured, have gained substantially in popularity. SCSI disks have long been considered the top-tier disks, but their high price has kept them out of inexpensive commodity PCs.

Describe the purpose of disk partitions. Disk partitions break the disk into a handful of distinct parts. Each partition can be used by a different OS, can contain a different filesystem, and is isolated from other partitions. These features improve security and safety and can greatly simplify running a multi-OS system.

Summarize important Linux disk partitions. The most important Linux disk partition is the root (`/`) partition, which is at the base of the Linux directory tree. Other possible partitions include a swap partition, `/home` for home directories, `/usr` for program files, `/var` for transient system files, `/tmp` for temporary user files, `/boot` for the kernel and other critical boot files, and more.

Describe commands that help you monitor disk use. The `df` command provides a one-line summary of each mounted filesystem's size, available space, free space, and percentage of space used. The `du` command adds up the disk space used by all of the files in a specified directory tree and presents a summary by directory and subdirectory.

Summarize the tools that can help keep a filesystem healthy. The `fsck` program is a front end to filesystem-specific tools such as `e2fsck` and `fsck.jfs`. By whatever name, these programs examine a filesystem's major data structures for internal consistency and can correct minor errors.

Explain how filesystems are mounted in Linux. The `mount` command ties a filesystem to a Linux directory; once the filesystem is mounted, its files can be accessed as part of the mount directory. The `/etc/fstab` file describes permanent mappings of filesystems to mount points; when the system boots, it automatically mounts the described filesystems unless they use the `noauto` option (which is common for removable disks).

EXERCISE 4.2 *(continued)*

5. Type **`find /usr -name startx`**. This search takes longer and, when run as an ordinary user, most likely returns several `Permission denied` error messages. It should also return a single line listing the `/usr/bin/startx` or `/usr/X11R6/bin/startx` program file. Note that this command searches only `/usr`. If you searched `/usr/X11R6`, the command would take less time; if you searched `/`, the command would take more time.
6. Type **`which startx`**. This search completes almost instantaneously, returning the complete filename of the first instance of `startx` the system finds on its path.
7. Type **`type startx`**. Again, the search completes very quickly. It should identify `startx` as an external command stored at `/usr/bin/startx`, `/usr/X11R6/bin/startx`, or possibly some other location.

Summary

File management is basic to being able to administer or use a Linux system. Various commands are useful to both users and administrators for copying, moving, renaming, and otherwise manipulating files and directories. You may also want to set up access controls, both to limit the amount of disk space users may consume and to limit who may access specific files and directories. Finally, Linux provides standards and tools to help you locate files using various criteria.

Exam Essentials

Describe commands used to copy, move, and rename files in Linux. The `cp` command copies files, as in **`cp first second`** to create a copy of `first` called `second`. The `mv` command does double duty as a file-moving and file-renaming command. It works much like `cp`, but `mv` moves or renames the file rather than copying it.

Summarize Linux's directory-manipulation commands. The `mkdir` command creates a new directory, and `rmdir` deletes a directory. You can also use many file-manipulation commands, such as `mv` and `rm` (with its `-r` option), on directories.

Explain the difference between hard and symbolic links. Hard links are duplicate directory entries that both point to the same inode and hence to the same file. Symbolic links are special files that point to another file or directory by name. Hard links must reside on a single filesystem, but symbolic links may point across filesystems.

Summarize the common Linux archiving programs. The `tar` and `cpio` programs are both file-based archiving tools that create archives of files using ordinary file access commands. The `dd` program is a file-copy program; however, when it's fed a partition device file, it copies the entire partition on a very low-level basis, which is useful for creating low-level image backups of Linux or non-Linux filesystems.

Explain the differences between compression utilities. The `gzip`, `bzip2`, and `xz` utilities are compression tools, which reduce a file's size via compression algorithms. They are often used in conjunction with the `tar` command. The `gzip` utility is the oldest compression tool and provides the least compression. The `bzip2` utility provides slightly improved file compression. The `xz` utility is the newest tool, provides the best compression, and is very popular.

Describe Linux's file ownership system. Every file has an owner and a group, identified by number. File permissions can be assigned independently to the file's owner, the file's group, and to all other users.

Explain Linux's file permissions system. Linux provides independent read, write, and execute permissions for the file's owner, the file's group, and all other users, resulting in nine main permission bits. Special permission bits are also available, enabling you to launch program files with modified account features or alter the rules Linux uses to control who may delete files.

Summarize the commands Linux uses to modify permissions. The `chmod` command is Linux's main tool for setting permissions. You can specify permissions using either an octal (base 8) mode or a symbolic notation. The `chown` and `chgrp` commands enable you to change the file's owner and group, respectively. (The `chown` command can do both but can be run only by `root`.)

Describe the prerequisites of using Linux's disk quota system. Linux's disk quota system requires support in the Linux kernel for the filesystem on which quotas are to be used. You must also run the `quotaon` command, typically from a startup script, to enable this feature.

Explain how quotas are set. You can edit quotas for an individual user via the `edquota` command, as in **`edquota larry`** to edit `larry`'s quotas. This command opens an editor on a text file that describes the user's quotas. You can change this description, save the file, and exit from the editor to change the user's quotas.

Summarize how Linux's standard directories are structured. Linux's directory tree begins with the root (`/`) directory, which holds mostly other directories. Specific directories may hold specific types of information, such as user files in `/home` and configuration files in `/etc`. Some of these directories and their subdirectories may be separate partitions, which helps isolate data in the event of filesystem corruption.

Describe the major file-location commands in Linux. The `find` command locates files by brute force, searching through the directory tree for files that match the criteria you specify. The `locate` (or `slocate`) command searches a database of files in publicly accessible directories. The `whereis` command searches a handful of important directories, and `which` searches the path. The `type` command identifies another command as a built-in shell command, a shell alias, or an external command (including the path to that command).

Quit Use the `:q` command to quit the program. As with `:e`, this command won't work unless changes have been saved or you append an exclamation mark to the command (as in `:q!`).

You can combine ex-mode commands such as these to perform multiple actions in sequence. For instance, typing **`:wq`** writes changes and then quits from vi. (`ZZ` is the equivalent of `:wq`.)

Summary

Although Linux distributions are designed to boot painlessly and reliably once installed, understanding the boot process will help you overcome problems and maintain your system. Most Linux systems employ a boot loader known as GRUB (either GRUB Legacy or GRUB 2). These programs both fit themselves into the standard BIOS boot system, enabling the computer to load the Linux kernel. GRUB 2, and some patched versions of GRUB Legacy, also work on EFI-based computers. The kernel then runs the `init` program, which in turn reads various configuration files to boot all of the services that make a running Linux system.

Modifying your GRUB configuration enables you to boot different Linux kernels or non-Linux OSs. You can also pass new boot options to Linux. Once the system is booted, you can use the `dmesg` command or log files to study the boot process in order to verify that it went correctly or to find clues as to why it didn't.

You can use the vi editor to edit your GRUB configuration file, your system initialization scripts and configuration files, or any other plain-text file on your computer. Although vi is old-fashioned in many ways, it's small and is used for emergency boot systems. Every administrator should be familiar with vi, even if it's not their editor of choice for day-to-day operations.

Exam Essentials

Describe how GRUB Legacy is configured and used. GRUB Legacy uses the `menu.lst` or `grub.conf` configuration file in `/boot/grub`. This file contains global and per-image options. Use the `grub-install` program to install the boot loader. When GRUB boots, it presents a menu of OS options that you select using the keyboard arrow keys.

Describe how GRUB 2 is configured and used. GRUB 2 uses the `/boot/grub/grub.cfg` configuration file; however, system administrators are discouraged from editing it directly. Instead, they should rely on automatic configuration scripts and set system-specific defaults in `/etc/defaults/grub` and the files in `/etc/grub.d`. As with GRUB Legacy, you can install GRUB 2 using the `grub-install` program.

Describe the boot process. The CPU runs the firmware, the firmware loads and runs a boot loader, the boot loader loads and runs secondary boot loaders (if needed) and the Linux kernel, the Linux kernel loads and runs the initial system program (`init`), and `init` starts the rest of the system services via startup scripts that are specific to the startup system (SysV, Upstart, systemd, or something more exotic). BIOS-based computers look for boot loaders in various boot sectors, including the MBR of a hard drive or the boot sector of a disk partition or USB flash drive. EFI-based computers look for boot loaders in files on the ESP.

Summarize where to look for boot-time log information. The `dmesg` command prints out logs from the kernel ring buffer, which holds boot-time and other kernel messages. Other useful log information can be found in `/var/log/messages` and other files in `/var/log`.

Summarize the role of `/sbin/init`. The `init` program is responsible for starting many programs and services on your Linux operating system.

Explain the SysV init system. The SysV init system uses a default runlevel specified with a line like `id:2:initdefault:` in the `/etc/inittab` file. Use commands such as `chkconfig`, `update-rc.d`, `ntsysv`, and `systemctl` to change which services are started when switching to specific runlevels. Runlevels 0, 1, and 6 are reserved for shutdown, single-user mode, and rebooting, respectively. Runlevels 3, 4, and 5 are the common user runlevels on Red Hat and most other distributions, and runlevel 2 is the usual user runlevel on Debian systems.

Describe how to change SysV init runlevels. The programs `init` and `telinit` can be used to change to other runlevels. The `shutdown`, `halt`, `poweroff`, and `reboot` programs are also useful when shutting down, rebooting, or switching to single-user mode.

Explain the systemd init system. The systemd init system uses units and targets to control services. The default target is specified by the file `/etc/systemd/system/default.target` and is a link to a target file in the `/lib/systemd/system` folder.

Describe how to change systemd init targets. You use the `systemctl` program to start and stop services as well as to change the target level of the system.

Describe vi's three editing modes. You enter text using insert mode, which supports text entry and deletion. The command and ex modes are used to perform more complex commands or to run outside programs to operate on the text entered or changed in insert mode.

Summary

X is Linux's GUI system. In part because of Linux's modular nature, X isn't a single program; you have your choice of X servers to run on Linux. Fortunately, most Linux distributions use the same X server as all others (X.org-X11). Both X.org-X11 and its main competitor, XFree86, are configured in much the same way, using the `xorg.conf` (for X.org-X11) or `XF86Config` configuration file. Whatever its name, this file consists of several sections, each of which controls one X subsystem, such as the mouse, the keyboard, or the video card. This file also controls X's core fonts system, but you can use a font server in addition to this system, and most modern programs are now emphasizing an entirely new font system, Xft, instead of X core fonts. For this reason, Linux font configuration can be complex.

X's GUI login system uses an XDMCP server, which starts X and manages the X display. Several XDMCP servers are in common use in Linux, the most important being XDM, KDM, GDM, and LightDM. They all perform the same basic tasks, but configuration details differ. (XDM is also less sophisticated than KDM and GDM.) X is a network-enabled GUI, which means that you can use an X server to access programs running on another computer. Doing so requires performing a few steps for each login session. You can also tunnel X accesses through SSH, which greatly improves the security of the connection.

An assortment of tools can help make Linux more accessible to users with visual or motor impairments. You can adjust font size, screen contrast, and other display features to improve legibility; use screen magnifiers to help users read part of a larger screen; or even bypass a visual display entirely and use a screen reader for auditory output or a Braille display for tactile output. On the input side, you can adjust keyboard repeat rates, use sticky keys, or modify the mouse tracking speed and click sensitivity to improve users' ability to input data accurately. You can even have a mouse stand in for a keyboard or vice versa by using the appropriate software.

The second main visual output tool on computers is a printer, and Linux provides sophisticated printer support. The CUPS package manages printers in Linux by accepting local or remote print jobs, passing them through a smart filter for processing, and queuing the jobs so that they print in a reasonable order. Most CUPS configuration is best handled via its own Web interface, but some options (particularly security features) can be set via text configuration files.

Exam Essentials

Name the major X servers for Linux. XFree86 has been the traditional standard Linux X server, but in 2004 X.org-X11 (which was based on XFree86) rapidly gained prominence as the new standard Linux X server. Accelerated-X is a commercial X server that sometimes supports video cards that aren't supported by XFree86 or X.org-X11.

Describe the X configuration file format. The XFree86 and X.org-X11 configuration file is broken into multiple sections, each of which begins with the keyword `Section` and ends with `EndSection`. Each section sets options related to a single X feature, such as loading modules, specifying the mouse type, or describing the screen resolution and color depth.

Summarize the differences between X core fonts, a font server, and Xft fonts. X core fonts are managed directly by X, and they lack modern font features such as font smoothing. Font servers integrate with the X core fonts but run as separate programs and may optionally deliver fonts to multiple computers on a network. Xft fonts bypass the X core font system to provide client-side fonts in a way that supports modern features such as font smoothing.

Explain the role of an XDMCP server. An XDMCP server, such as XDM, KDM, or GDM, launches X and controls access to X via a login prompt—that is, it serves as Linux's GUI login system. XDMCP servers are also network enabled, providing a way to log in remotely from another X server.

Describe X's client-server model. An X server runs on the user's computer to control the display and accept input from the keyboard and mouse. Client programs run on the same computer or on a remote computer to do the bulk of the computational work. These client programs treat the X server much as they treat other servers, requesting input from and sending output to them.

Explain the benefits of using SSH for remote X access. SSH can simplify remote X-based network access by reducing the number of steps required to run X programs from a remote computer. More important, SSH encrypts data, which keeps information sent between the X client and X server secure from prying eyes.

Summarize X accessibility features. You can adjust keyboard and mouse options to help those with motor impairments to use keyboards and mice or to substitute one device for the other. Font size, contrast, and magnification tools can help those with visual impairments. Finally, text readers and Braille displays can enable blind individuals to use a Linux system.

Describe how to set a time zone in Linux. Linux uses a binary file, `/etc/localtime`, to describe the features of the time zone. This file is copied or linked from a repository of such files at system installation, but you can replace the file at any time.

Explain the role of Ghostscript in Linux printing. PostScript is the standard Linux printing language, and Ghostscript converts PostScript into bitmap formats that are acceptable to non-PostScript printers. Thus, Ghostscript is a critical translation step in many Linux print queues, although it's not required for PostScript printers.

Summarize how print jobs are submitted and managed under Linux. You use `lpr` to submit a print job for printing, or an application program may call `lpr` itself or implement its functionality directly. The `lpq` utility summarizes jobs in a queue, and `lprm` can remove print jobs from a queue.

Summary

Routine system administration involves a variety of tasks, many of which center around user management. Adding, deleting, and modifying user accounts and groups are critical tasks that all system administrators must master. Because they are also related to users, you should know where to go to modify the default user environment.

System log and journal files are critical troubleshooting tools that are maintained by the system. You should be able to configure what data is logged to what files and know how to use these log and journal files.

Time management is important in Linux. Setting the Linux clocks (both hardware and software) and configuring NTP to keep the software clock accurate are important tasks. Tools that rely on the time include `cron`, `anacron`, and `at`, which enable the system to run programs in the future. These tools are used for many common system tasks, including rotating log files.

Exam Essentials

Summarize methods of creating and modifying user accounts. Accounts can be created or modified with the help of tools designed for the purpose, such as `useradd` and `usermod`. Alternatively, you can directly edit the `/etc/passwd` and `/etc/shadow` files, which hold the account information.

Describe the function of groups in Linux. Linux groups enable security features to be applied to arbitrary groups of users. Each group holds an arbitrary collection of users, and group permissions can be set on files, giving all group members the same access rights to the files.

Explain the purpose of the skeleton files. Skeleton files provide a core set of configuration files that should be present in users' home directories when those directories are created. They provide a starting point for users to modify their important shell and other configuration files.

Summarize how to configure system logging. System logging is controlled via the `/etc/syslog.conf` file. Lines in this file describe what types of log data, generated by programs, are sent to log files and to which log files the log messages should go.

Describe how log rotation is managed. Log rotation is controlled via the `/etc/logrotate.conf` file (which typically refers to files in `/etc/logrotate.d/`). Entries in these files tell the system whether to rotate logs at fixed intervals or when they reach particular sizes. When a log rotates, it's renamed (and possibly compressed), a new log file is created, and the oldest archived log file may be deleted.

Summarize how to review journal data. The `systemd-journald` service is responsible for journal message data. The daemon is controlled via the `/etc/systemd/journal.conf` configuration file. This journal message data can only be viewed using the `journalctl` utility. To view system journal data, you must either use superuser privileges or be a member of the `systemd-journal` group. Users can view their own user journal data files. To view the entire current journal data file, simply use the `journalctl` command with no parameters. To parse out journal data, use the various filters available with the `journalctl` utility via parameters.

Explain the two types of clocks in *x*86 and *x*86-64 hardware. The hardware clock keeps time when the computer is powered down, but most programs don't use it while the computer is running. Such programs refer to the software clock, which is set from the hardware clock when the computer boots.

Summarize the function of NTP. The Network Time Protocol (NTP) enables a computer to set its clock based on the time maintained by an NTP server system. NTP can function as a tiered protocol, enabling one system to function as a client to an NTP server and as a server to additional NTP clients. This structure enables a single highly accurate time source to be used by anywhere from a few to (theoretically) billions of computers via a tiered system of links.

Explain the difference between system and user `cron` jobs. System `cron` jobs are controlled from `/etc/crontab`, are created by `root`, and may be run as any user (but most commonly as `root`). System `cron` jobs are typically run at certain fixed times on an hourly, daily, weekly, or monthly basis. User `cron` jobs may be created by any user (various security measures permitting), are run under the authority of the account with which they're associated, and may be run at just about any repeating interval desired.

Describe how to configure `anacron` jobs. The `anacron` jobs are controlled from the `/etc/anacrontab` file. The file is checked to see when each listed job was last executed, and it ensures that the designated time period between executions is followed. Time periods are listed in number of days. No time period can be less than one day, and therefore jobs needing to be run more than one time per day should not use `anacron`. Often, `anacron` itself is run from a system startup script or from a `cron` job.

words, 1 × 103KiB/s, or 1,000KiB/s). This can be a useful way to test your network transfer speed, although you'll get more reliable results with files that are several hundred kilobytes or larger in size. In addition to `get`, which retrieves files, you can issue commands such as `put` to upload a file, `ls` or `dir` to display the remote system's directory contents, `cd` to change directories on the remote system, `delete` to remove a file, and `quit` or `exit` to exit from the program. You can use the `help` or `?` command to see a list of available `ftp` commands.

Like Telnet, FTP is a poor choice of protocol for security reasons. The same SSH protocol that can substitute for Telnet can also handle most FTP duties. One important exception exists to the rule to not use FTP, though: using anonymous FTP sites is a common method of distributing public files on the Internet. You can download Linux itself from anonymous FTP sites. These sites typically take a username of `anonymous` and any password (your email address is the conventional reply) and give you read access to their contents. In most cases, you can't upload files to anonymous FTP sites, and you can access only a limited number of files.

You can access public FTP sites using a web browser. Enter a URL that begins with `ftp://`, such as `ftp://downloads.example.org`, and the web browser connects to the site using FTP rather than HTTP.

Summary

Linux is a network-enabled OS, and it relies on its networking features more than do most OSs. This networking is built around TCP/IP, so you should understand the basics of this protocol stack, including IP addresses, hostnames, and routing. Most Linux distributions provide tools to configure networking during system installation, but if you want to change your settings temporarily or permanently, you can do so. Tools such as `ifconfig` and `route` can temporarily change your network configuration, and editing critical files or running distribution-specific utilities enables you to make your changes permanent.

Exam Essentials

Describe the information needed to configure a computer on a static IP network. Four pieces of information are important: the IP address, the netmask (aka the network mask or subnet mask), the network's gateway address, and the address of at least one DNS server. The first two are required, but if you omit either or both of the latter two, basic networking will function, but you won't be able to connect to the Internet or use most DNS hostnames.

Determine when using `/etc/hosts` rather than DNS makes the most sense. The `/etc/hosts` file provides a static mapping of hostnames to IP addresses on a single computer. Therefore, maintaining this file on a handful of computers for a small local network is fairly straightforward, but when the number of computers rises beyond a few or when IP addresses change frequently, running a DNS server to handle local name resolution makes more sense.

Summarize tools you can use to translate between hostnames and IP addresses. The `nslookup` program can perform these translations in both directions using either command-line or interactive modes, but this program has been deprecated. You're better off using `host` for simple lookups and `dig` for more complex tasks.

Describe the function of network ports. Network ports enable packets to be directed to specific programs; each network-enabled program attaches itself to one or more ports, sending data from that port and receiving data directed to the port. Certain ports are assigned to be used by specific servers, enabling client programs to contact servers by directing requests at specific port numbers on the server computers.

Explain when you should use static IP addresses or DHCP. Static IP address configuration involves manually entering the IP address and other information and is used when a network lacks a Dynamic Host Configuration Protocol (DHCP) server or when a computer shouldn't be configured by that server (say, because the computer *is* the DHCP server). DHCP configuration is easier to set up on the client but works only if the network has a DHCP server system.

Explain what the `route` command accomplishes. The `route` command displays or modifies the routing table, which tells Linux how to direct packets based on their destination IP addresses.

Describe some basic network diagnostic tools. The `ping` program tests basic network connectivity, and `traceroute` and `tracepath` perform similar but more complex tests that can help you localize where on a route between two systems a problem exists. The `netstat` utility is a general-purpose network status tool that can report a wide variety of information about your network configuration. Packet sniffers such as `tcpdump` provide detailed information about the network packets "seen" by a computer, which can be a useful way to verify that certain packet types are actually being sent or received.

Learning More about SQL

SQL is a very complex topic, and this chapter can only scratch the surface. For more information, you should read more from various sources. Your own SQL package's documentation can be a good starting point, particularly if you need to use features that are unique to your implementation. Books on SQL, such as Alan Beaulieu's *Learning SQL, 2nd Edition* (O'Reilly, 2009) and Larry Rockoff's *The Language of SQL* (Cengage Learning, 2010), are also worth reading if you need to do more than trivial SQL work.

Summary

Linux administrators must have at least a basic understanding of shell scripts. Many configuration and startup files are in fact shell scripts. Being able to read them, and perhaps to modify them, will help you administer your system. Being able to create new shell scripts is also important because doing so will help you simplify tedious tasks and create site-specific tools by putting together multiple programs to accomplish your goals.

Email server administration is another task with which you must have at least a passing familiarity. Although most Linux systems don't operate as full-blown email servers, most Linux installations do include email servers for processing locally generated email. You can configure email forwarding and perform a few other tweaks without delving too heavily into email server configuration.

The final topic of this chapter, SQL use, will help you manage simple databases stored using the Structured Query Language (SQL). Many programs rely on SQL for their operation, so being able to perform simple SQL queries will help you work with these programs. You may even decide to set up databases to help manage your own tasks, such as tracking computer equipment.

Exam Essentials

Explain the function of environment variables. Environment variables are used to store information on the system for the benefit of running programs. Examples include the `PATH` environment variable, which holds the locations of executable programs, and `HOSTNAME`, which holds the system's hostname.

Describe various shell script components. A shell script combines several commands, possibly including conditional expressions, variables, and other programming features. A shell script must start with a shebang line to let the kernel know it is a script and indicate the shell interpreter to use. Each script component has a specific syntax needed for proper shell execution and logic flow. A shell script is run using various methods. Each method may or may not create a subshell and may or may not require the execution bit to be set.

Describe the purpose of shell aliases. Aliases enable you to create a command "shortcut"—a simple command that can stand in for a different or longer command. Aliases are typically defined in shell startup scripts as a way to create a shortened version of a command. Aliases also allow useful command options to be used as the new default or they create an easier-to-remember version of a command.

Summarize the major SMTP servers for Linux. Sendmail was the most common SMTP server a decade ago, and it is still very popular today. Postfix and Exim are often supplied as the default mail servers on modern distributions, whereas administrators sometimes install qmail. Postfix and qmail use modular designs, whereas sendmail and Exim do not.

Explain the difference between an email alias and email forwarding. An email alias is configured system-wide, typically in `/etc/aliases`. It can set up forwarding for any local address, even if that address doesn't correspond to a real account, and if the system is properly configured, only root may edit `/etc/aliases` and therefore modify aliases. Email forwarding, on the other hand, is handled by the `~/.forward` file in a user's home directory; it is intended as a means for users to control their own email forwarding without bothering the system administrator.

Summarize the structure of a SQL database. Each SQL installation consists of a number of named databases, each of which in turn may contain multiple tables. Each table can be thought of as a two-dimensional array of data. Each row in a table describes some object or concept (inventory items, employees, movies in a personal DVD collection, and so on), and each column in a table holds data about these objects or concepts (model number, salary, or director, for example).

Describe the commands used to enter data in a SQL database. The `INSERT` command inserts a single entry into a database. It requires a table name and a set of values, as in `INSERT INTO movies VALUES('Brazil', 'Terry Gilliam', 1985);`. The `UPDATE` command can be used in a similar way to update an existing entry, but you must use `SET` to specify the column to set and `WHERE` to identify the row or rows to be modified.

Explain the commands used to extract data from a SQL database. The `SELECT` command retrieves data from a SQL database. It can be used with a variety of additional options, such as `FROM`, `JOIN`, and `WHERE`, to identify the table or tables from which data should be retrieved and to locate specific values of interest.

If you omit the `--armor` option, the resulting file is a binary file. To send the binary file through email, you'll need to send it as an attachment. If you include the `--armor` option, the output is ASCII, so you can cut and paste the encrypted message into an email or send it as an attachment.

If you receive a message or file that was encrypted with your public key, you can reverse the encryption by using the `--decrypt` option:

```
$ gpg --out decrypted-file --decrypt encrypted-file
```

You'll be asked to enter your passphrase. The result should be a decrypted version of the original file.

In practice, GPG can be even easier to use than this description may make you think. GPG is primarily used to secure and verify email, so most Linux email clients provide GPG interfaces. These options call `gpg` with appropriate options to encrypt, sign, or decrypt messages. Details vary from one email client to another, so you should consult your email client's documentation for details.

Signing Messages and Verifying Signatures

As noted earlier, GPG can be used to sign messages so that recipients know that they come from you. To do so, use the `--sign` or `--clearsign` option to `gpg`:

```
$ gpg --clearsign original-file
```

The `--sign` option creates a new file with the same name as the original, but with `.gpg` appended to the filename. This file is encrypted using your private key so that it may be decrypted only with your public key. This means that anybody with your public key may read the message, and anybody who can read it knows that it's from you.

The `--clearsign` option works similarly, but it leaves the message text unencrypted and only adds an encrypted signature that can be verified by using your public key. The `--clearsign` option creates a file with a name that ends in `.asc`.

If you receive a signed message, you can verify the signature using the `--verify` option to `gpg`:

```
$ gpg --verify received-file
```

If any of the keys in your keyring can decode the message or verify the signature, `gpg` displays a `Good signature` message. To read a message that was encrypted via the `--sign` option, you must decrypt the message via the `--decrypt` option, as described earlier.

Summary

Maintaining system security is both important and time-consuming. A great deal of security emphasis is on network security. To achieve a high level of network security, properly configuring the server's super daemon and disabling unused servers goes a long way. Attending to passwords and performing miscellaneous tasks to keep your local accounts from becoming security risks are also important security tasks.

Encryption is a hot topic in security. SSH is a protocol and tool that can handle many network encryption tasks by encrypting two-way connections between computers. Typically used as a remote login protocol, SSH can also be used to transfer files or encrypt other protocols. When you want to encrypt data sent to another individual via a tool such as email, you can do so with the help of GPG. This package enables you to encrypt individual files, which can then be attached to or embedded in email messages and decrypted by the recipient.

Exam Essentials

Identify the purpose of a super server. Super servers (also called super daemons), such as `inetd` and `xinetd`, manage incoming network connections for multiple servers. They can add security and convenience features, and they can help to minimize the memory load imposed by seldom-accessed servers.

Explain the function of super server port access controls. Super servers or programs called by them (such as TCP wrappers) can restrict access to ports for the servers they manage. These restrictions occur at a higher level than a firewall's restrictions, and they apply only to the servers managed by the super server.

Summarize the tools that you can use to identify the servers running on a computer. The `netstat` and `lsof` programs both provide options to list all (or a subset of) the open network connections as well as programs that are listening for connections. Remote network scanners, such as Nmap, can probe another computer for open network ports. The `fuser` program can determine the processes currently using a particular network port. Perusal of local configuration files can also provide clues as to what's running on a computer.

Describe why SUID and SGID programs are potentially risky. The set user ID (SUID) and set group ID (SGID) bits tell Linux to run the program as the user or group that owns the file. This is particularly risky when `root` owns the program file because it essentially elevates all users to `root` for the purposes of running the file, making bugs in the program more dangerous and raising the possibility of a clever user abusing the program to acquire full `root` privileges or otherwise wreaking havoc.

Explain why shadow passwords are important. Shadow passwords store password hashes in a file that can't be read by ordinary users, thus making it harder for attackers on the local system to read the hashed passwords and use brute-force attacks to discover other users' passwords. Modern Linux distributions use shadow passwords by default.

Explain how to generate a good password. Ideally, passwords should be random. Failing that, one good approach is to generate a base that's hard to guess and then modify it by adding digits and punctuation, changing the case of some characters, changing letter order, and significantly increasing the length of the password (even with repeated characters).

Explain why SSH is the preferred remote text-mode login tool. The Secure Shell (SSH) protocol provides encryption for all traffic, including both the password exchange and all subsequent data exchanges, whereas older tools, such as Telnet, do not. This makes SSH much safer for the exchange of sensitive data, particularly over untrusted networks such as the Internet.

Identify the most important SSH configuration file. The SSH server is controlled through the `/etc/ssh/sshd_config` file. The SSH client configuration file is `/etc/ssh/ssh_config`; don't confuse the two.

Describe the SSH public and private key files These keys are normally stored in the `/etc/ssh/` directory. Private key files are called `ssh_host_rsa_key`, `ssh_host_rsa1_key`, and `ssh_host_dsa_key`, depending on the encryption algorithm used. Public key files have the same filenames as their private keys, except a `.pub` filename extension is added.

Describe the function of GPG. GPG enables public-key encryption of individual files or email messages. You can use GPG to encrypt sensitive data for transmission over email or other insecure means.