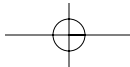
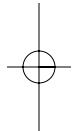


An Overview of Red Hat Linux



Secrets in This Chapter

<i>Linux as a UNIX Platform</i>	6
<i>Posix Compliance</i>	6
<i>Linux Standard Base (LSB)</i>	8
<i>Linux Desktop</i>	8



The world of operating systems changed forever when Linus Torvalds of the University of Helsinki in Finland decided to build a UNIX-like operating system for the PC. What started as a simple task-switching example, with two processes that printed AAAA . . . and BBBB . . . on a dumb terminal, has grown into a full-fledged, multitasking, multiuser operating system that rivals commercially available UNIX systems for Intel 80x86 systems. Many programmers around the world have contributed code and collaborated to bring Linux to its current state. With the release of version 1.0 in March 1994, Linux became an operating system of choice for UNIX enthusiasts, as well as for people looking for a low-cost UNIX platform for a specific purpose, such as developing software or running an Internet host.

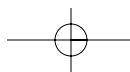
This chapter provides a broad-brushstroke picture of Red Hat Linux, one of several well-known Linux distributions (other well-known Linux distributions are Mandrake, Debian, Slackware, and S.u.S.E.). The chapter describes how you can get the most out of the built-in capabilities of Red Hat Linux, such as networking, developing software, and running applications.

After you overcome your initial fear of the unknown and install Linux, you will see how you can use it to turn your PC into a UNIX workstation. The best part is that you can get Linux for free—just download it from one of several Internet sites (for example, you'll find links to many Linux distributions from the Linux Online website at <http://www.linux.org/dist/>). The best way for beginners and experts alike to get started, though, is to buy a book (such as this one) that comes with a Linux distribution on CD-ROM. This book is your guide to the inner workings of Red Hat Linux. The next chapter shows how to install Red Hat Linux, and subsequent chapters describe specific tasks (such as connecting to the Internet or developing software) that you may want to perform with your Red Hat Linux PC. In addition to many utilities with graphical user interfaces (GUIs), this book provides you the details such as what commands to use and what configuration files to edit.

What Is Red Hat Linux?

Linux is a freely available UNIX-like operating system that runs on a wide variety of systems. Linus Torvalds and other programmers originally developed Linux for the Intel 80x86 processor. Nowadays, Linux is also available for systems based on other processors, such as Intel's new 64-bit Itanium IA64 architecture processor, the Motorola 68000 family, the Alpha AXP processor, the Sun SPARC and UltraSPAC processors, Hewlett-Packard's HP PA-RISC processor, the PowerPC and PowerPC64 processors, ARM family of processors, and the MIPS R4x00 and R5x00 processors. More recently, IBM has announced Linux for its S/390 and zSeries mainframes. This book covers Red Hat Linux for the Intel 80x86 and Pentium processors (these are known as the IA32 architecture processors, or i386, because they support the instruction set of the 80386 processor).

Red Hat Linux is a specific Linux distribution. A Linux distribution is essentially a package consisting of the Linux operating system and a collection of applications, together with an easy-to-use installation program. All Linux distributions include the core Linux operating system (the kernel); the XFree86 X Window System for x86 systems; one or more graphical desktops, such as GNOME and KDE; and a large selection of applications. Everything comes in ready-to-run binary format, but the source code and documentation are also included. By now, each Linux distribution includes so much software that it comes on multiple CD-ROMs. For example, this book comes with two CD-ROMs containing the Publisher's Edition of Red Hat Linux. The source code CD-ROM is not included but is available upon request (see coupon on the final page of this book).



Like many other Linux distributions, Red Hat Linux is a commercial distribution. You can buy Red Hat Linux in computer stores and bookstores. The GNU (which stands for "GNU's Not UNIX") General Public License that applies to Linux allows for such commercial, for-profit distribution, but requires that the software be distributed in source-code form, and stipulates that anyone can copy and distribute the software in source-code form to anyone else.

**insider
insight**

In an effort to unite several different Linux distributions and promote widespread adoption, a number of Linux companies (Conectiva, The SCO Group, SuSE, and Turbolinux) have formed the UnitedLinux group, and in November 2002, the group released version 1.0 of its UnitedLinux product. UnitedLinux is expected to compete with the Red Hat Linux distribution.

Both the Linux kernel and Red Hat Linux have gone through a number of versions. The version numbers are unrelated, but each has particular significance.

Linux Kernel Version Numbers

After Linux version 1.0 was released on March 14, 1994, the loosely organized Linux development community adopted a version-number scheme. Versions 1.x.y and 2.x.y, where x is an even number, are stable versions. The number y is the *patch level*, which is incremented as problems are fixed. Notice that these version numbers are of the form *Major.Minor.Patch*, where *Major* and *Minor* are integers denoting the major and minor version numbers, and *Patch* is another integer representing the patch level.

Versions 2.x.y with an odd x number are beta releases for developers only; they may be unstable, so you should not adopt these versions for day-to-day use. Developers add new features to these odd-numbered versions of Linux.

When this book was written, the latest stable version of the Linux kernel was 2.4.20 (note that information about the latest version of the Linux kernel is available at <http://www.kernel.org/>). This book's companion CD-ROMs contain the latest version of the Linux kernel as of Spring 2003.

**cross
ref**

If you hear about a later version of Linux or about helpful patches (minor corrections) to the current version, you can obtain the patches and rebuild the kernel by following the instructions in Chapter 21. That chapter also describes how you can download the new kernel from Red Hat.

Red Hat Linux Version Numbers

Red Hat assigns the Red Hat Linux version numbers, such as 7.3 or 8.1. They are of the form x.y, where x is the major version and y the minor version. Unlike with the Linux kernel version numbers, there is no special meaning associated with odd and even minor versions. Nowadays if the minor version number is zero, it's simply dropped — as in Red Hat Linux 9. Each version of Red Hat Linux includes specific versions of the Linux kernel and other major components, such as the XFree86, GNOME, KDE, and various applications such as the OpenOffice.org suite.

6 Part I: Setting Up Red Hat Linux ♦♦♦

Red Hat releases new versions of Red Hat Linux on a regular basis. For example, Red Hat Linux 5.2 came out in November 1998, 6.0 in April 1999, 7.0 in September 2000, 8.0 in September 2002, and Red Hat Linux 9 on March 31, 2003. Typically, each new major version of Red Hat Linux provides significant new features. Red Hat Linux 6.x brought in the GNOME and KDE graphical desktops; Red Hat Linux 7.0 offered features such as support for the Universal Serial Bus (USB) keyboard and mice, XFree86 4.0, and strengthened network security with Kerberos. Red Hat Linux 8.0 provided a uniform look-and-feel for both GNOME and KDE desktops, included the OpenOffice.org office suite, and added many graphical configuration tools. Red Hat Linux 9 includes the Common UNIX Printing System (CUPS), Apache 2.0 Web server, and the Native POSIX Thread Library. In all major versions, Red Hat also updates the core components from the kernel to the GNU C Compiler and associated libraries. Often these behind-the-scenes changes to the core operating system provide significant benefits such as support for newer interfaces and a more secure system.

Linux as a UNIX Platform

Like other UNIX systems, Linux is a multiuser, multitasking operating system, which means that it enables multiple users to log in and to run more than one program at the same time.

Secret

UNIX was developed in the early 1970s at AT&T Bell Laboratories. Its development came on the heels of another operating system called MULTICS; developers are said to have come up with the name UNIX by changing the MULT in MULTICS to UN (meaning one). Bell Laboratories continued to develop UNIX and released several versions: System III, followed by System V Release 1, or SVR1, and SVR2, SVR3, and SVR4.

As it maintained and enhanced UNIX, Bell Laboratories distributed source code to educational institutions. The University of California at Berkeley2 (UC Berkeley) was one of the schools that received a copy of UNIX and added many new features to the operating system. Eventually, UC Berkeley released its version of UNIX, called Berkeley Software Distribution (BSD) UNIX. The most widely used versions of BSD UNIX are 4.3 and 4.4 (known as 4.4BSD).

By the time 4.4BSD UNIX came out, UC Berkeley realized that there was very little original Bell Laboratories UNIX code in the source code. Soon, several groups wrote new code to replace the small amount of leftover Bell Laboratories code and adapted BSD UNIX to the Intel 386 processor. This resulted in the FreeBSD and NetBSD versions of freely available BSD UNIX for Intel PCs.

Note that UNIX System V, Release 4—SVR4—combines all features of System V and BSD UNIX.

POSIX Compliance

Linux is designed to comply with IEEE Std 1003.1 1996 Edition (POSIX). This standard defines the functions that applications written in the C programming language use to access the services of the operating system—for tasks ranging from opening a file to allocating memory. On March 8, 1996, the Computer Systems Laboratory of the National Institute of Standards and Technology (NIST), a U.S. government agency, confirmed that

Linux version 1.2.13, as packaged by Open Linux Ltd., conforms to the POSIX standard. To see a list of POSIX-validated products, point your Web browser to <http://www.nist.gov/itl/div897/ctg/posix/finalreg4.htm>. Note that the NIST POSIX testing program ended on December 31, 1997. Of course, POSIX compliance, while commendable, is not synonymous with a high-quality operating system.

Along with POSIX conformance, Linux includes many features of other UNIX standards, such as the System V Interface Document (SVID) and the Berkeley Software Distribution (BSD) version of UNIX. Linux takes an eclectic approach, picking the most-needed features of several standard flavors of UNIX.

**Secret**

POSIX stands for Portable Operating System Interface (abbreviated as POSIX to make it sound like UNIX). The Institute of Electrical and Electronics Engineers (IEEE) began developing the POSIX standards to promote the portability of applications across UNIX environments. POSIX is not limited to UNIX, however. Many other operating systems, such as Hewlett-Packard OpenVMS and Microsoft Windows NT/2000/XP, implement POSIX—in particular, the IEEE Std. 1003.1 1996 Edition, or POSIX.1, which provides a source-level C-language Application Program Interface (API) to the services of the operating system, such as reading and writing files. POSIX.1 has been accepted by the International Organization for Standardization (ISO) and is known as the ISO/IEC 9945-1:1996 standard.

Incidentally, the term POSIX is used interchangeably with the IEEE 1003 and 2003 family of standards. There are several other IEEE standards besides the 1003 and 2003 family—such as 1224 and 1228—that also provide APIs for developing portable applications. For the latest information on all IEEE standards, visit the IEEE Standards Home Page at <http://standards.ieee.org/>. To read summary information about the POSIX standards, visit the IEEE Web page at http://standards.ieee.org/reading/ieee/std_public/description/posix/.

In addition to POSIX (IEEE 1003.1) compliance, Linux supports the IEEE 1003.2 standard, which focuses on the operating system's command interpreter (commonly referred to as the shell) and a standard set of utility programs. If you know UNIX or you've had some exposure to it, you know that UNIX takes a tools-oriented view of the operating system. It provides a tool for almost anything you might want to do, and the shell enables you to combine several tools to perform tasks more complicated than those the basic tools handle. The IEEE 1003.2 standard maintains this tools-oriented view, providing the following features:

- A shell with a specified set of built-in commands and a programming syntax that can be used to write shell programs, or scripts
- A standard set of utility programs—such as `sed`, `tr`, and `awk`—that shell scripts and applications can call. Even the `vi` editor and the mail electronic-mail program are part of the standard set. You will learn more about these utilities in Chapters 8, 10, and 11.
- A set of C functions, such as `system` and `getenv`, that applications can use to access features of the shell
- A set of utilities, such as Perl and Tcl, for developing shell applications

8 Part I: Setting Up Red Hat Linux ♦♦♦

The default Linux shell is called Bash, which stands for Bourne-Again Shell—a reference to the Bourne shell, which has been the standard UNIX shell since the early days of UNIX. Bash incorporates many of the features IEEE 1003.2 requires and then some. It essentially inherits the features and functionality of the Bourne shell. In case of any discrepancy between the Bourne shell and IEEE 1003.2, Bash follows IEEE 1003.2. For stricter IEEE 1003.2 compliance, Bash even includes a POSIX mode.

All in all, Linux serves as a good platform for learning UNIX because it offers a standard set of UNIX commands (the IEEE 1003.2 standard, as well as the best features of both System V and BSD UNIX).

Linux's support for POSIX and other common UNIX system calls (the functions that applications call) makes it an excellent system for software development. Another ingredient of modern workstation software, the X Window System, is also available in Linux in the form of XFree86.

Linux Standard Base (LSB)

Linux has become important enough that there is now a standard for Linux called the Linux Standard Base, or LSB for short. LSB is a set of binary standards that should help reduce variations among the Linux distributions and promote portability of applications. The idea behind LSB is to provide application binary interface (ABI) so that software applications can run on any Linux (or other UNIX) systems that conform to the LSB standard. The LSB specification references the POSIX standards as well as many other standards such as the C programming language standard and the X Window System version 11 release 6 (X11R6). LSB version 1.2 (commonly referred to as LSB 1.2) was released on June 28, 2002. LSB 1.3 went through a public review in early November 2002.

Secret

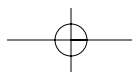
The LSB specification is organized into two parts—a common specification that remains the same across all types of processors and a set of hardware-specific specifications, one for each type of processor architecture. For example, LSB 1.2 has architecture-specific specifications for Intel 32-bit (IA32) and Power PC 32-bit (PPC32) processors. LSB 1.3 adds a specification for the Intel 64-bit (IA64) architecture, in addition to the ones for IA32 and PPC32.

There is an LSB certification program, and by now a number of Linux systems, such as Red Hat Linux 8.0, Mandrake Linux ProSuite 9.0, SuSE Linux 8.1, and UnitedLinux 1.0 LSB, are certified to be LSB 1.2-compliant IA32 runtime environments.

To learn more about LSB, visit <http://www.linuxbase.org/>. The latest list of LSB-certified systems is available at http://www.opengroup.org/lsb/cert/cert_prodlst.tpl.

Linux Desktop

Let's face it—typing cryptic UNIX commands on a terminal is boring. Those of us who know the commands by heart may not realize it, but the installed base of UNIX is not going to increase significantly if we don't make the system easy to use. This is where the X Window System, or X, comes to the rescue.



X provides a standard mechanism for displaying device-independent bitmapped graphics. In other words, an X application can display its graphic output on many different machines that use different methods to display text, graphics, and images on the monitor. X is also a windowing system, meaning it enables applications to organize their output in separate windows. X uses a client/server architecture and works over the network, so you can run X applications on various systems on the network while the output appears in windows that are managed by an X server running on your system.

Although X provides the mechanism for windowed output, it does not offer any specific look or feel for applications. The look and feel comes from GUIs, such as GNOME and KDE, which are based on the X Window System.

**cross
ref**

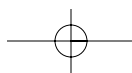
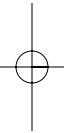
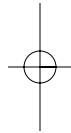
The Red Hat Linux distribution on this book's CD-ROMs comes with the X Window System in the form of XFree86 4.3—an implementation of X11R6 (X Window System version 11, release 6, which is the latest release of X) for 80x86 systems. A key feature of XFree86 is its support for a wide variety of video cards available for today's PCs. As you will learn in Chapter 3, XFree86 supports hundreds of PC video cards, ranging from the run-of-the-mill Super Video Graphics Array (SVGA) to accelerated graphics cards such as the ones based on the 3Dfx, ATI, Intel, Matrox, NVIDIA, and S3 video chipsets. However, XFree86 may not work well on some generic video cards containing variants of popular chipsets such as S3.

As for the GUI, Linux includes two powerful graphical desktop environments: KDE (K Desktop Environment) and GNOME (GNU Network Object Model Environment). When you install Red Hat Linux, you can choose which desktop you want or can install both and switch between the two. GNOME and KDE provide desktops similar to the ones in Microsoft Windows and the Apple Mac OS. GNOME also comes with the Nautilus graphical shell that makes it easy to find files, run applications, and configure your Linux system. With GNOME or KDE, you can begin using your Linux workstation without having to learn UNIX commands. However, if you should ever need to use UNIX commands, all you have to do is open a terminal window and type the commands at the shell prompt.

Red Hat Linux also comes with many graphical applications that run under X. The most noteworthy programs relate to image display and editing. The first is GIMP—the GNU Image Manipulation Program—a program with capabilities on a par with Adobe Photoshop; the second program is ImageMagick.

Another important aspect of the X Window System is that you can run applications across the network because X uses a client/server architecture. The X server runs at the workstation and controls the display, keyboard, and mouse. Client applications send requests to the X server to receive user input and display output. For example, you might run an X application on a server somewhere on the network but view that application's output and interact with it from your Linux desktop that's running a X-based GUI. In other words, with X, your Linux PC becomes a gateway to all the other systems on the network.

Motif is the dominant GUI in the UNIX marketplace, but it's not packaged with Linux because the Open Software Foundation does not distribute Motif for free. Motif has a look and feel similar to Microsoft Windows and includes the Motif Window Manager (MWM) and the Motif toolkit for programmers. You can download OpenMotif for Linux from <http://www.motifzone.net/>. In addition to Open Motif from The Open Group, another option for Motif for Linux is LessTif, a free version of Motif distributed under the GNU



10 Part I: Setting Up Red Hat Linux ♦♦♦

General Public License (visit the LessTif home page at <http://www.lesstif.org> for the latest information on LessTif). Red Hat Linux comes with LessTif and is automatically installed if you select the X Software Development package group during Red Hat Linux installation.

If you need Motif for a project, using a Linux PC with a copy of OpenMotif or LessTif installed is an economical way to set up a software-development platform. If you have a consulting business, or if you want to develop X and Motif software at home, Linux is definitely the way to go.

Along with GNOME and KDE, you get two more options for developing GUI applications in Linux. GNOME comes with a toolkit called Gtk+ (GIMP toolkit), and KDE comes with the Qt toolkit. If you do not want to learn Motif, you may want to use Gtk+ or Qt for your GUI applications.

**cross
ref**

Chapter 3 shows you how to set up XFree86 on your system, and Chapter 8 presents the GNOME and KDE desktops.

Secret

Office productivity software—such as word-processing, spreadsheet, and database applications—is an area in which Linux used to be lacking. This situation has changed, though. Red Hat Linux comes with the OpenOffice.org office-productivity applications. In addition, there are several prominent commercially available office-productivity applications for Linux that are not included on the companion CD-ROMs. Applixware Office—now called Anyware Desktop for Linux—is a good example of productivity software for Linux (<http://www.vistasource.com/products/>). Another well-known productivity-software package is StarOffice from Sun Microsystems (<http://www.sun.com/staroffice/>). CrossOver Office, from CodeWeavers, is a commercially available software package that enables you to install your Microsoft Office applications (only Office 97 or Office 2000, not Office XP) in Linux. Furthermore, many existing software packages (designed for UNIX workstations with the X Window System) can be readily ported to Linux, thanks to Linux's support for portable standards such as POSIX and the X Window System.

Linux Networking

Networking refers to all aspects of data exchange within one computer or between two or more computers, ranging from the physical connection to the protocol for the actual data exchange. A network protocol is the method the sender and receiver agree upon for exchanging data across a network.

Different network protocols are used at different levels of the network. At the physical level—at which the data bits travel through a medium, such as a cable—Ethernet and Asynchronous Transfer Mode (ATM) are two commonly-used protocols. Application programs don't really work at the physical level, however. Instead, they rely on protocols that operate on blocks of data. These protocols include Novell's Internet Packet Exchange (IPX) and the well-known Transmission Control Protocol/Internet Protocol (TCP/IP).

cross
ref

The different levels of network protocols can be represented by a networking model such as the seven-layer Open Systems Interconnection (OSI) reference model, developed by ISO. Chapter 6 includes a discussion of this model.

Standard network protocols such as TCP/IP have been key to the growth of interconnected computers, resulting in local area networks (LANs), as well as wide area networks (WANs). Protocols have enabled these smaller networks to communicate with each other, and we now have interconnected networks that form an internetwork: the Internet.

TCP/IP

The ability to network has been one of the strengths of UNIX since its early days. In particular, the well-known TCP/IP protocol suite has been an integral part of UNIX ever since TCP/IP appeared in BSD UNIX around 1982. By now, TCP/IP is the wide area and local area networking protocol of choice in the global Internet. TCP/IP does not depend on the physical communication media. This media independence enables TCP/IP to work in a wide variety of networks.

Linux supports the TCP/IP protocol suite and includes all common network applications such as Telnet, FTP, and sendmail. At the physical-network level, Linux includes drivers for many Ethernet cards. Token ring is also an integral part of the Linux kernel source; all you have to do is load the token ring driver to enable support for token ring.

cross
ref

You might say that Linux's support for TCP/IP—the dominant protocol suite of the Internet—comes naturally. The rapid development of Linux itself would not have been possible without the collaboration of so many developers from Europe, America, and other parts of the world. That collaboration, in turn, has been possible only because of the Internet. Chapters 13 through 19 show you how to set up TCP/IP networking and to use various servers to offer services such as Web, email, and domain name service.

Linux also includes the Berkeley Sockets programming interface (so named because the Sockets interface was introduced in Berkeley UNIX around 1982), a popular interface for network programming in TCP/IP networks. For those of you with C programming experience, the Sockets interface consists of several C header files and several C functions that you call to set up connections and to send and receive data. Chapter 6 describes sockets.

You can use the Berkeley Sockets programming interface to develop Internet tools such as Web browsers. Because most TCP/IP programs (including those available for free at various Internet sites) use the Sockets programming interface, it is easy to get these programs up and running on Linux because it includes the Sockets interface.

PPP and SLIP

Not everyone has an Ethernet connection to the Internet (although a growing number of us are beginning to have high-speed always-on connections to the Internet, thanks to cable modems or DSL). Most of us still connect to the Internet and communicate by using the TCP/IP protocol over a phone line and a modem. To do this, what you need is access to a server—a system that has an Internet connection and that accepts a dial-in connection from your system.

12 Part I: Setting Up Red Hat Linux ♦♦♦

Commercial outfits known as Internet service providers (ISPs) offer this type of service for a fee. If you don't want to pay for such a connection, find out whether a computer at your place of business provides this access. That option may not be unreasonable, especially if you are doing UNIX software development (for your company) on your Linux PC at home.

When you access the Internet through a server, the server runs the Point-to-Point Protocol (PPP), which works over any serial link, including dial-up connections.

CROSS
REF

Linux supports PPP for dial-up Internet connections. You can also turn your Linux system into a PPP server so that other computers can dial in to your computer and establish a TCP/IP connection over the phone. Chapter 13 explains how to set up a PPP dial-up Internet connection on your Red Hat Linux system.

Cable/DSL and Wireless Networks

If you have high-speed Internet access through cable modem or DSL (Digital Subscriber Line), you can easily hook up a Ethernet-equipped Linux PC to the Internet. For the most part, the configuration of the Linux PC is the same as that for TCP/IP networking. However, for ISPs that use PPP over Ethernet (PPPoE), you may have to do some additional configuring. Linux includes support for PPPoE.

Linux also supports wireless Ethernet cards that you can use to connect laptop PCs to an existing wired Ethernet local area network (LAN). These wireless Ethernet cards conform to the IEEE 802.11b standard, also known as Wi-Fi. If your LAN connects to the Internet through a cable/DSL router and hub, you can extend the LAN by connecting a wireless access point to the hub. Then, any Wi-Fi-equipped laptop or desktop PC can connect to the Internet through the cable or DSL connection.

CROSS
REF

Chapter 13 describes how to configure Linux for a wireless Ethernet network.

File Sharing with NFS

In the Microsoft Windows or Novell NetWare world, you may be familiar with the concept of a file server—a system that maintains important files and allows all other systems on the network to access those files. Storing files on a central server provides for better security and enables convenient backups. Essentially, all PCs on the network share one or more central disks. In Windows and Novell, users see the file server's disk as just another drive, with its own drive letter (such as U). Typically in PC networks, you implement file sharing with Novell NetWare or Microsoft LAN Manager protocols.

File sharing exists in UNIX as well. The Network File System (NFS) provides a standard way for a system to access another system's files over the network. To the user, the remote system's files appear to be in a directory on the local system.

**CROSS
ref**

NFS is available in Linux; you can share your Linux system's directories with other systems that support NFS. The other systems that access your Linux system's files via NFS do not necessarily have to run UNIX; NFS is available for DOS, Windows, OS/2, and NetWare as well. Therefore, you can use a Linux PC as the file server for a small workgroup of PCs that run DOS and/or Windows. Chapter 19 further explores the use of a Linux PC as a file server. Chapter 19 also explains how to use the Samba package to set up your Linux PC as a server in a Windows network.

Linux System Administration

System administration refers to tasks that someone must perform to keep a computer system up and running properly. Now that almost all computers are networked, it's necessary to perform another set of tasks to keep the network up and running. All these tasks are collectively called network administration. A site with many computers probably has a full-time system administrator who takes care of all system-administration and network-administration tasks. Really large sites may have separate system-administration and network-administration personnel. If you are running Linux on a home PC or on a few systems in a small company, you are probably both the system administrator and the network administrator.

**CROSS
ref**

Linux supplies all the basic commands and utilities you need for system and network administration. Chapters 12 and 20 briefly cover some of these commands. Chapter 6 describes some network-administration tools.

GNOME's Nautilus graphical shell comes with many GUI tools that enable you to perform most system-administration and network-administration tasks without having to edit configuration files manually or type cryptic commands. However, you should always learn the key commands and be proficient with a plaintext editor such as vi, for those times when you must use a text-only login and the GUI tools are not available. Additionally, you need to understand the layout of the key configuration files. I cover this type of information throughout this book, even when describing GNOME and KDE, because they too depend on configuration files for correct operation.

System-Administration Tasks

As a system administrator, your tasks typically are the following:

- ♦ **Installing, configuring, and upgrading the operating system and various utilities.** You learn how to install Red Hat Linux and other software packages in Chapter 2. Chapter 3 tells you how to install and configure the X Window System, and Chapter 21 shows you how to upgrade the operating system—the Linux kernel.
- ♦ **Adding and removing users.** As shown in Chapter 18, you can use Nautilus's Red Hat User Manager graphical tool or the `useradd` command to add a new user after you install Linux. If a user forgets a password, you can change the password from the Red Hat User Manager or can use the `passwd` command to change it.

- ♦ **Installing new software.** For the typical Linux software, which you get in source-code form, this task involves using tools such as gunzip (to uncompress the software), tar (to unpack the archive), and make (to build the executable programs). For software, Red Hat distributes in Red Hat Package Manager (RPM) files, use the rpm command to install the software. Chapter 21 describes RPM.
- ♦ **Making backups.** You can use the tar program to archive one or more directories and to copy the archive to a floppy disk (if the archive is small enough) or to a tape (if you have a tape drive). Chapter 20 covers backing up and restoring files and directories.
- ♦ **Managing file systems.** When you want to read an MS-DOS floppy disk, for example, mount that disk's MS-DOS file system on one of the directories of the Linux file system. Use the mount command to do this. You can also use mount an NT file system (NTFS) after installing a kernel module that supports NTFS. You also want to monitor the file system to ensure that users or some errant process have not filled them up.
- ♦ **Monitoring the system's performance.** You have to use a few utilities, such as top (to see where the processor is spending most of its time) and free (to see the amount of free and used memory in the system).
- ♦ **Monitoring the system's integrity.** You want to make sure that no one has tampered with key system files. You can use tools such as Tripwire to perform this task. Chapter 22 covers how to maintain system security.
- ♦ **Starting and shutting down the system.** Although starting the system typically involves nothing more than powering up the PC, you do have to take some care when you want to shut down your Linux system. Use the shutdown command to stop all programs before turning off your PC's power switch. If your system is set up for a graphical login screen, you can perform the shutdown operation by selecting a menu item from the login screen.

Network-Administration Tasks

Typical network-administration tasks are the following:

- ♦ **Maintaining the network configuration files.** In Linux (as well as in other UNIX systems), several text files hold the configuration information for the TCP/IP network. You may have to edit these files to make networking work. You may have to edit one or more of the following files: /etc/hosts, /etc/networks, /etc/host.conf, /etc/resolv.conf, /etc/HOSTNAME, /etc/hosts.allow, /etc/hosts.deny, and the scripts in the /etc/sysconfig/network-scripts directory. You can either edit these files manually or use the graphical Network Configuration tool to configure them.
- ♦ **Setting up PPP.** You may use tools such as wvdial to set up and use PPP connections. You can also use the Red Hat Dialup Configuration Tool to set up PPP connections. Chapter 13 shows you how to work with PPP commands and configuration files.
- ♦ **Monitoring network status.** You have to use tools such as netstat (to view information about active network connections), /sbin/ifconfig (to check the status of various network interfaces), and ping (to make sure that a connection is working).
- ♦ **Securing Internet services.** If your system is connected to the Internet (or if it is on an internal network), you have to secure the system against anyone who

might use one of many Internet services to gain access to your system. Each service—such as email, Web, or FTP—requires running a server program that responds to client requests arriving over the TCP/IP network. Some of these server programs have weaknesses that may enable an outsider to log in to your system—maybe with root privileges. Turn off services you do not need, and edit configuration files to restrict access to those services you are running. Chapter 22 covers network security and how to use commands such as `chkconfig` to turn Internet services on or off.

Windows and Linux

As you probably know, MS-DOS used to be and Microsoft Windows (in its various versions from Windows 95/98 to Windows XP) continues to be the most popular operating system for 80386, 80486, and Pentium PCs. Because Linux started on 80386/80486 PCs, a connection between DOS/Windows and Linux has always existed. Typically, you start the Linux installation with some steps in DOS.

Linux has maintained its connection to DOS/Windows in several ways:

- ♦ Linux supports the older MS-DOS file system called FAT (file allocation table), as well as the newer Windows VFAT (long filenames) and FAT32 file systems. From Linux, you can access MS-DOS and Windows files on a hard disk or a floppy disk.
- ♦ Linux supports read-only access to NTFS file system that is used in Windows NT/2000/XP. You can build (or download) and load a driver module to incorporate the NTFS support.
- ♦ Linux features a set of tools (called `mtools`) that manipulates DOS/Windows files from within Linux.

An ongoing project called WINE is developing a free implementation of Windows for the X Window System under UNIX (see <http://www.winehq.com/>). WINE enables you to run Windows 3.1/95/NT programs. WINE works on some versions of UNIX for the Intel x86 systems, including Linux and FreeBSD.

**cross
ref**

Chapter 12 describes how you can access DOS from Linux and explains the use of the `mtools` utilities.

Software Development in Linux

Of all its potential uses, Linux is particularly well suited to software development. Software-development tools, such as the compiler and libraries, are included because you need them when you rebuild the Linux kernel. If you are a UNIX software developer, you already know UNIX, so you will feel right at home in Linux.

As far as the development environment goes, you have the same basic tools (such as an editor, a compiler, and a debugger) that you might use on other UNIX workstations, such as those from IBM, Sun Microsystems, and Hewlett-Packard (HP). Therefore, if you work by day on one of the mainstream UNIX workstations, you can use a Linux PC at home to

16 Part I: Setting Up Red Hat Linux ♦♦♦

duplicate that development environment at a fraction of the cost. Then, you can either complete work projects at home or devote your time to software you write for fun and share on the Internet.

Just to give you a sense of Linux's software-development support, the following is a list of various features that make Linux a productive software-development environment:

- ♦ GNU's C compiler, `gcc`, which can compile ANSI-standard C programs
- ♦ GNU's C++ compiler (`g++`), which supports ANSI-standard C++ features
- ♦ The GNU compiler for the Java programming language, `gcj`
- ♦ The GNU debugger, `gdb`, which enables you to step through your program to find problems and to determine where and how a program has failed. (The failed program's memory image is saved in a file named `core`; `gdb` can examine this file.)
- ♦ The GNU profiling utility, `gprof`, which enables you to determine the degree to which a piece of software uses your computer's processor time
- ♦ The GNU `make` utility, which enables you to compile and link large programs
- ♦ Concurrent Versions System (CVS) and Revision Control System (RCS), which maintain version information and control access to the source files so that two programmers don't modify the same source file inadvertently
- ♦ The GNU Emacs editor, which prepares source files and even launches a compile-link process to build the program
- ♦ The Perl scripting language, which you can use to write scripts that tie together many smaller programs with UNIX commands to accomplish a specific task
- ♦ The Tool Command Language and its X toolkit (Tcl/Tk), which enable you to prototype X applications rapidly
- ♦ The Python language, an interpreted language comparable to Perl and Tcl (the Red Hat Linux installation program, called `anaconda`, is written in Python)
- ♦ Dynamically linked shared libraries, which allow the actual program files to be much smaller because all the library code that several programs may use is shared, with only one copy being loaded in the system's memory
- ♦ POSIX header files and libraries, which enable you to write portable programs

**CROSS
REF**

Chapter 23 covers software development in Linux. Read Chapters 24 and 25 to learn about Perl and Tcl/Tk programming. Chapter 26 covers Java programming in Red Hat Linux.

Linux as an Internet On-Ramp

Most likely, you have experienced much of what the Internet has to offer: electronic mail, newsgroups, and the Web. So you may be happy to learn that a Red Hat Linux system includes everything you need to access the Internet. In fact, your PC can become a first-class citizen of the Internet, with its own Web server on which you can publish any information you want.

Although Linux includes TCP/IP and supporting network software with which you can set up your PC as an Internet host, there is one catch: First, you have to obtain a physical connection to the Internet. Your Linux PC has to be connected to another node (which can be

another computer or a networking device, such as a router) on the Internet. This requirement is the stumbling block for many people—an Internet connection costs money, the price proportional to the data-transfer rate.

Many commercial ISPs provide various forms of physical connections to the Internet. In the United States, if you are willing to spend between \$15 and \$30 a month, you can get an account on a PPP server. Then you can run PPP software on your Linux system, dial in via a modem, and connect to the Internet at data-transfer rates ranging from 28,800 bits per second (bps) to 56,000 bps, depending on your modem.

Although a dial-up connection is adequate for accessing the Internet, receiving email, and reading news, it may not be adequate if you want your system to provide information to other people through the Web or FTP (File Transfer Protocol). Besides your ISP may not allow you to use the dial-up connection to run a Web server. To set up a useful Web server, you need a connection that is available 24 hours a day, because other systems may try to access your system any time of day. For a few hundred dollars a month, you can get a dedicated connection and make your system a permanent presence on the Internet. Other options that offer higher-capacity Internet connections than dial-up modems are cable modems and Digital Subscriber Lines (DSL). You may also opt to run the server at the ISP's facility—something many people do because it's very convenient.

Another requirement for a business—or for anyone who has a few networked PCs—is connecting a local area network (LAN) to the Internet. You can run Linux on one of the PCs and use it as the Internet gateway to accomplish this task. Typically, you have an Ethernet LAN running TCP/IP connected to all of the PCs on the network, including the Linux machine. The Linux PC sets up a PPP connection to the Internet (via a dial-up or dedicated connection). You then set up the Linux PC to act as a gateway between the Ethernet LAN and the Internet so that the PCs on your LAN can access other systems on the Internet.

CROSS
REF

In Chapter 13, you learn to configure your Linux system to access the Internet.

Summary

After you get Linux going on your PC, you can turn your attention to the work you plan to do with it. Whether you want to develop software or set up your PC as an Internet host, you can use Linux wisely if you know its overall capabilities. Accordingly, this chapter provided an overview of various aspects of Linux, ranging from software development to networking and system administration. In the next chapter, I will show you how to install Red Hat Linux from this book's companion CD-ROMs and get started on using Linux.

By reading this chapter, you learned the following:

- ♦ Linux is a freely available UNIX-like operating system that runs on a wide variety of systems. Red Hat Linux is a specific Linux distribution—a package incorporating the Linux operating system and a huge collection of applications, together with an easy-to-use installation program.
- ♦ Linux developers use a version-number scheme to help you understand what the various versions of Linux kernel—the core operating system—mean. Kernel versions 2.x.y, where x is an even number, are stable versions. The number

represented by *y* is the patch level, which is incremented as problems are fixed. Versions *2.x.y*, where *x* is an odd number, are beta releases for developers only, because these releases may be unstable.

- ♦ POSIX stands for Portable Operating System Interface (abbreviated as POSIX to make it sound like UNIX). The Institute of Electrical and Electronics Engineers (IEEE) began developing the POSIX standards to promote the portability of applications across UNIX environments.
- ♦ Red Hat Linux conforms to a binary standard called Linux Standard Base (LSB), which promotes compatibility among Linux systems so that applications built for one system can run on all LSB-compliant systems with the same processor architecture.
- ♦ This book's Red Hat Linux distribution comes with XFree86 (X Window System Version 11 Release 6 or X11R6), GNOME, and KDE software. After you install XFree86 and GNOME or KDE, you have a graphical user interface (GUI) for Linux. In addition, X enables you to run applications across the network—which means that you can run applications on another system on the network and can have the output appear on your Linux PC's display.
- ♦ Linux effectively supports TCP/IP networking. TCP/IP is the networking protocol of choice on the Internet. Therefore, a Linux PC is ideal as an Internet host, providing services such as FTP and World Wide Web access. You can also use the Linux PC as your Internet ramp by connecting to an Internet service provider through a dial-up, cable, or DSL connection and running a Web browser to surf the Net. You can configure Linux to support wireless Ethernet network cards.
- ♦ The Red Hat Linux distribution on the companion CD-ROM also includes Nautilus with many graphical tools that enable you to perform most system-administration and network-administration tasks from a GUI.
- ♦ Red Hat Linux provides all the software development tools you need to write UNIX and X applications. You'll find the GNU C and C++ compiler for compiling source files, *make* for automating the compiling, the *gdb* debugger for finding bugs, and Concurrent Versions System (CVS) and Revision Control System (RCS) for managing various revisions of a file. Thus, a Linux PC is the software developer's ideal workstation.