

## Chapter 1

# Getting Acquainted with Linux

---

### *In This Chapter*

- ▶ Diving into Linux origins and features
  - ▶ Telling Linux apart from the rest of the operating system pack
  - ▶ Depending on GNU and the GPL
  - ▶ Marveling at the Linux Company (or lack thereof)
  - ▶ Checking out popular Linux distributions
- 

*Ford, you're turning into a penguin. Stop it!*

— Arthur Dent

**W**elcome to the world of Linux, the operating system developed by more than a thousand people around the world! In this chapter, you find out about Linux itself — what it is, where it comes from, and why it gets so much attention in the news these days. Prepare to have your assumptions challenged about how software *must* be developed and sold, and your mind opened to new possibilities.

## *Is Free Really Free?*

Understanding Linux requires a radical shift of thought regarding the way that you acquire and use computer software. (**Note:** By *radical*, I mean getting to the root of the matter, rather than putting on beads and camping out in the administration building.) Your first step toward shifting your mind-set is to alter your general connotation of the word *free* to represent *freedom*, rather than *free lunch*. That's right; you can sell “free” software for a fee . . . and you're encouraged to do so, as long as you relay the same freedom to each recipient of the software.

Don't scratch your head too hard; these concepts are tough to grasp initially, especially when you consider the conditioning you've received from the commercial software industry's marketing departments. Perhaps you don't know that when you purchase most proprietary, shrink-wrapped software, you don't actually *own* the software. Rather, you're granted permission to use the software within the bounds dictated by the licensor.

Linux also has a license. However, the motives and purpose of the license are much different from those of most commercial software. Instead of using a license to restrict use of the software, the GNU General Public License (GPL) that Linux uses ensures that the software will always be open to anyone. No company can ever own or dictate the way in which you use or modify Linux — although they can have their own individual copyrights and trademarks on their various brands of it, like Red Hat and Novell. In essence, you already own Linux, and you can use it for anything you like, as long as you propagate the GPL freedoms to any further recipients of the software.

## *Linux: Revolution or Just Another Operating System?*

*Contrary to popular belief, penguins are not the salvation of modern technology. Neither do they throw parties for the urban proletariat.*

— Anonymous

**Author note:** Cute quote . . . obviously Anonymous has never been to a Linux convention!

Before going any farther into Linux, I need to get some terminology out of the way.

*Tux* is the formal name of the mascot penguin that represents Linux. Rumor has it that Linux's creator, Linus Torvalds, is rather fond of these well-dressed inhabitants of the Antarctic.

An *operating system* is the software that runs your computer, handling all interactions between you and the hardware. Whether you're writing a letter, calculating a budget, or managing your recipes on your computer, the operating system provides the essential air that your computer breathes. Furthermore, an operating system isn't just one program; it consists of hundreds of smaller programs and utilities that allow us humans to use a computer to do something useful. You then run other programs (such as your word processor) on top of the operating system to get everything done.

In recent technological history, Linux has evolved from water-cooler techie chatter to a rock-solid solution for the business enterprise. The same software that was once dismissed as rogue is now being adopted and promoted by industry leaders such as IBM, Hewlett-Packard, Motorola, Microsoft, and Intel. Each of these computer manufacturers has, in some way, determined that Linux provides value for their customers (as well as for their own operations).

Linux has been accused of being “just another operating system.” On the surface, it may appear so, but if you look deeper, you can see that this isn’t so. The Linux project is a flagship leading the current trend toward open source and free (as in freedom, not free beer) software within the computing industry. A rock-solid operating system because of the model under which it was (and continues to be) developed, Linux represents much that is good in software development.

Two fundamental distinctions separate Linux from the rest of the operating-system pack:

- ✓ Linux is licensed under the unique and ingenious *GNU General Public License*, which you can read about in the next section.
- ✓ Linux is developed and maintained by a worldwide team of volunteer and paid programmers, working together over the Internet.

Linux is great for many reasons, including the fact that the folks who built it from the ground up wanted it to be

- ✓ **Multiuser:** More than one user can be logged in to a single computer at one time.
- ✓ **Multiprocess:** True *preemptive multitasking* enables the operating system core to efficiently juggle several programs running at once. This is important for providing multiple services on one computer.
- ✓ **Multiplatform:** Linux currently runs on more than 24 *platforms* (hardware types), including 32- and 64-bit Intel-based PCs, Digital/Compaq Alpha, all variants of the Apple Macintosh, Sun SPARC, the Apple iPod, and even the Microsoft XBox.
- ✓ **Interoperable:** Linux plays nice with most network protocols (languages) and operating systems, allowing you to interact with users and computers running Microsoft Windows, UNIX, Novell Netware, Apple Macintosh computers, and other, more niche groups.
- ✓ **Scalable:** As your computing needs grow, you can rely on Linux to grow with you. The same Linux operating system can run on a tiny electronic photo frame, a desktop computer, or a very large, industrial-strength server system.

- ✓ **Portable:** Linux is mostly written in the C programming language. C is a language created specifically for writing operating system–level software and can be readily *ported* (translated) to run on new computer hardware.
- ✓ **Flexible:** You can configure the Linux operating system as a network host, router, graphical workstation, office productivity PC, home entertainment computer, file server, Web server, cluster, or just about any other computing appliance that you can think of.
- ✓ **Stable:** The Linux *kernel* (the operating system) has achieved a level of maturity that makes most software developers envious. It's not uncommon to hear reports of Linux servers running for years without crashing.
- ✓ **Efficient:** The modular design of Linux enables you to include only the components needed to run your desired services. Even older Pentium computers can utilize Linux and become useful again.
- ✓ **Free!:** To most people, the most intriguing aspect of Linux is the fact that it's often available free of charge. How (the capitalists murmur) can anyone build a better mousetrap with no incentive of direct monetary return?

## So where did Linux come from?

The quickest way to understand Linux is to take a peek at its rich heritage. Although programming of the Linux core started in 1991, the design concepts were based on the time-tested *UNIX* operating system.

UNIX was developed at Bell Telephone Laboratories in the late 1960s. The original architects of UNIX, working back when there were few operating systems, wanted to create one that shared data, programs, and resources both efficiently and securely — something that wasn't available then (and is still sought after now). From there, UNIX evolved into many different versions; its current family tree is so complicated that it looks like a kudzu infestation!

In 1991, Linus Torvalds was a computer science student at the University of Helsinki in Finland. He wanted an operating system that was like the UNIX system that he'd grown fond of at the

university, but both UNIX and the hardware it ran on were prohibitively expensive. A UNIX version called Minix was available for free, but it didn't quite meet his needs. So, as a computer science student, Torvalds studied Minix and then set out to write a new version himself. In his own words (recorded for posterity on the Internet because this was in an early version of an online chat room), his work was "just a hobby, won't be big and professional like GNU."

Writing an operating system is no small task. Even after six months of hard work, Torvalds had made very little progress toward the general utility of the system. He posted what he had to the Internet — and found that many people shared his interest and curiosity. Before long, some of the brightest minds around the world were contributing to Linus's project by adding enhancements or fixing bugs (errors in the code).

In this chapter, I intend to answer that last question for you. I also hope to paint a picture of the open source software development model that created Linux.

## Anatomy of an Open Source Software Project

*Linux isn't a product. Linux is an organic part of a software ecosystem.*

— Michael Robinson, *Netrinsics*

To the casual observer (and some corporate IT decision makers), Linux appears to be a freak mutation — a rogue creature randomly generated by anarchy. How, after all, can something so complex and discipline dependent as a computer operating system be developed by a loosely knit band of volunteer computer geeks from around the world?

Just as science is constantly attempting to classify and explain everything in existence, technology commentators are still trying to understand how the open source approach can create superior software, especially in cases where there is no charge. Often the reasons have much to do with the usual human desire to fill a need with a solution. When a programmer in the Linux world wants a tool, the programmer simply writes one — or bands together with other people who want a similar package, and they write it together.

### GNU who?

Imagine — software created out of need rather than projected profit. Even though UNIX ultimately became expensive proprietary software, the ideas and motives for its creation were originally based on practical needs. What people usually refer to (in the singular) as the *Linux operating system* is actually a collection of software tools that were created with the express purpose of solving specific computing problems.

The speed of Linux's popularity also wouldn't be possible without the vision of a man whom Steven Levy (author of the book *Hackers*) refers to as "The Last of the Great MIT AI-LAB Hackers" — in the original sense of the word *hacker* as someone who plays with code, not the current popular meaning that implies criminal intent. This pioneer and advocate of *freedom* software is Richard Stallman.

The Massachusetts Institute of Technology (MIT) has long held a reputation for nurturing the greatest minds in the technological disciplines. In 1984, Stallman, a gifted student and brilliant programmer at MIT, was faced with a dilemma — sell his talent to a company for a tidy sum of money or donate his gifts to the world. He did what we’d all do . . . right?

Stallman set out on a journey to create a completely free operating system that he would donate to the world. He understands — and continues to live — the original hacker ethic, which declares that information wants to be free. This concept wasn’t new in his time. In the early days of the computing industry, many advancements were made by freely sharing ideas and programming code. Manufacturer-sponsored user groups brought the best minds together to solve complicated problems. This ethic, Stallman felt, was lost when companies began to hoard software as their own intellectual property with the single purpose of profit.

As you may or may not have gathered by this point, widespread and accessible source code is paramount to successful software development. *Source code* is the term for the human-readable text (as opposed to the unreadable cyber-hieroglyphics in an “executable” file) that a programmer types to communicate instructions to the computer.

Writing computer programs in binary is an extremely arduous task. Modern computer software is usually written in a human-friendly language and then *compiled*, or translated, into the computer’s native instruction set. To make changes to this software, a programmer needs access to a program’s source code. Most proprietary software comes only as a precompiled product; the software developer keeps the source code for those programs under lock and key.

After determining that his operating system would be built around the conceptual framework of UNIX, Stallman wanted the project name to distinguish his system from UNIX. So, he chose the recursive acronym *GNU* (pronounced *ga-new*), which means GNU’s not Unix.

To finance the GNU project, Stallman organized the Free Software Foundation (FSF), which sold free (open source) software to help feed the programmers who worked on its continuing development. (Remember, we’re talking *free* as in *free speech*, not *free beer*.) Although this organization and goal of creating a complete operating system was necessary and important, a much more important piece of the puzzle had to be put into place to protect this new software from big-business pirates — a concern still all too relevant today as a former Linux company tries to hijack ownership of decades of volunteer work from thousands of people around the world.

The *GNU General Public License* (GPL) is a unique and creative software license that uses copyright law to protect the freedom of the software user, which is usually the opposite of how a copyright works. Generally, a copyright is an enforceable designation of ownership and restriction from duplication by anyone but the copyright holder. When software is licensed under the GPL,

recipients are bound by copyright law to respect the freedom of anyone else to use the software in any way they choose. Software licensed with the GPL is also known as *copyleft* software (the reverse of right, get it?). Another way to remember the GPL is through its ultimate result: Guaranteed Public for Life.

While Stallman's work set the stage for Linux's rapid climb to popularity, the operating system he and his crew were working on took longer than expected. If you're interested in the completed version, go to [www.gnu.org/software/hurd/hurd.html](http://www.gnu.org/software/hurd/hurd.html).

## *Who's in charge of Linux anyway?*

As an open source project evolves, various people emerge as leaders. This leader is often known as the project's *benevolent dictator*. A person who becomes benevolent dictator has probably spent more time than anyone else on a particular problem and often has some unique insight. Normally, the words *democratic* and *dictator* are never paired in the same sentence, but the open source model is a very democratic process that endorses the reign of a benevolent dictator.

Linus Torvalds is still considered the benevolent dictator of the Linux *kernel* (the operating system's core). He ultimately determines what features are added to the kernel and what features aren't. The community trusts his vision and discretion. In the event that he loses interest in the project, or the community decides that he has gone senile, a new leader will emerge from amongst the very competent people working with him.

## *Einstein was a volunteer*

Someone who is a volunteer or donates time to a project isn't necessarily providing a second-rate effort (or only working on weekends and holidays). In fact, any human resources expert will tell you that people who choose to do a job of their own free will produce the highest quality products.

The volunteers who contribute to open source projects are often leaders in their fields who depend on community collaboration to get useful work done. The open source concept is no stranger to the scientific community. The impartial peer-review process that open source projects foster is critical in validating some new feature or capability as being technically correct.



Those who paint the open source community as copyright violators and thieves often misunderstand — or outright ignore — these vital issues. Open source programmers are very proud of their work *and* are also very concerned about their own copyrights, not wanting their work to be stolen by others — hence licenses such as the GPL. This concern creates an atmosphere with the

greatest respect for copyright. Bandits who claim that they're "just being open source" when they steal other people's hard work are grossly misusing the term to soothe their own consciences.

Many have also pointed out that if copyright is violated in open source, it's easy to tell. Watch the news and notice how often large software corporations are convicted of stealing other people's code and incorporating it into their own work. If the final product is open source, it's easy for anyone to look and make sure nothing stolen is in it. As you might imagine, tracking down such copyright violations is much more difficult in a closed source scheme.

## *Packaging Linux: The Distribution*

What people call a *Linux distribution* is actually the culmination of the GNU project's tools, the Linux kernel, and any number of other open source (and closed source) software projects that sprang up along the way.

Robert Young, cofounder and current chairman of Red Hat, has coined an analogy comparing Linux to ketchup. Essentially, the operating system called Linux — including the GNU tools, Linux kernel, and other software — is a freely available commodity that, like ketchup, different distributors can package and label in different containers. Anyone is encouraged to package and market the stuff, even though the ingredients are fundamentally the same.

Linux is a complex, malleable operating system, and thus it can take on many appearances. The greatest Linux advancement in recent years has been easier installation. After all, the tools that today enable the casual PC user to install Linux weren't originally available. Companies such as Red Hat saw this as an opportunity to add value to an existing product, and the concept took off like gangbusters.

To draw again on the ketchup analogy, various distributions of Linux have a slightly different *flavor* or texture; your distribution preference may be spicy, mild, thick and gooey, or runny. However, you can rest assured that any of the following distributions have the same Linux and GNU heart and soul. Each short description in this list includes a Web address where you can find more information about each project:

- ✔ **Debian GNU/Linux:** This distribution — one of the oldest — is a recognized favorite among advanced technical circles. Historically, it's relatively difficult to install. Easier to install (and use) distributions with Debian underneath are Knoppix, Ubuntu, Xandros, and Linspire (formerly known as Lindows). Of these, Ubuntu is by far the most popular, with a dedicated worldwide fan base. Debian, Knoppix and Ubuntu are free as in free beer, while Xandros and Linspire are both commercial



distributions — though each has a free version you can try — created for those who really are most comfortable in Windows and don't want to have to master the nitty-gritty of using Linux.

Knoppix is particularly interesting as it's a live distribution, meaning that you take a Knoppix CD-ROM or DVD-ROM, place it in the drive, and reboot the machine into a fully operational Linux desktop. This book contains another live distribution option for those who don't want to install Linux on their hard drive. See Chapter 3 for more information. Knoppix, Linspire, Ubuntu, and Xandros are covered on the DVD-ROM that comes with this book.

```
www.debian.org
www.xandros.com
www.ubuntulinux.org
www.knoppix.org
www.linspire.com
```

- ✓ **Mandriva:** This distribution demonstrates the power of the GPL by allowing this competing company to stand on the shoulders of giants. Mandriva (formerly known as Mandrake) was originally based on Red Hat Linux (something that simply could not happen in a closed source environment), but has since become an excellent solution in its own right. Mandriva offers a wide range of commercial products and services as well as its free version.

```
www.mandrivalinux.com
```

- ✓ **Red Hat and Fedora:** Red Hat claims the prize for successfully mass marketing the Linux operating system. Red Hat has validated Linux by packaging the GNU and Linux tools in a familiar method of distribution (shrink-wrapped) and has included value-added features to its product, such as telephone support, training, and consulting services. The commercial version of this distribution is Red Hat Enterprise Linux, and the free version is called Fedora. Fedora is the distribution covered in this book.

```
www.redhat.com
www.fedoraproject.org
```

- ✓ **Slackware:** Of all the more widely recognized surviving Linux distributions, Slackware has been around the longest (in fact, its installation interface remained unchanged until about a year ago). Slackware has a very loyal following, but isn't well known outside of the Linux community. Like Debian in terms of spirit, the Slackware crowd is as respected in Linux circles as the weathered old-timers who share stories of carrying around a shoebox full of diskettes. This distribution is also free.

```
www.slackware.com
```

- ✓ **Novell and SuSE:** (Pronounced *soo-za*) This distribution originally hails from Germany, where it has a very loyal following, and was purchased by Novell, Inc., as part of the company's new focus on Linux. Novell

offers both a commercial distribution in SuSE Enterprise Linux, and a free distribution with openSuSE, an effort similar effort to that of Fedora.

[www.opensuse.org](http://www.opensuse.org)  
[www.novell.com/linux/](http://www.novell.com/linux/)

As you can see, many paths (in the form of distributions) lead to Linux. It's important to note that regardless of which distribution you choose, you're using the same basic ingredients: the GNU tools and the Linux kernel. The major differences you'll encounter among distributions are

- ✓ **Installation programs:** Each distribution has developed its own installation program to help you achieve a running computer system. Some installation programs are designed for the casual computer user (hiding the technical details); others are designed with the seasoned system administrator in mind. Some of the simpler ones offer an “expert mode” for those who want to have more control right from the beginning.
- ✓ **Software versions:** Different distributions may use different versions of the *kernel* (the core of the operating system) and other supporting software packages — which makes for a plethora of versions. Open source projects are dynamic and release new versions regularly, as opposed to the often-sluggish development cycle of traditional commercial software.
- ✓ **Package managers:** Even though one Linux program should be able to run on any distribution, tools called *package managers* keep track of the software on your system and ensure that you have all the required supporting software as well. Distributions are usually dependent on one particular package manager. More recent in the grand scheme of things, package management has also come to involve adding easy-to-use update routines, as well as an easy way to add software without the need of a computer science degree. Chapter 16 provides more information about package management.

It would be impossible to account for *every* possible installation of every Linux distribution. Okay, maybe not literally impossible, but you'd need a forklift to bring your *Linux For Dummies* book home from the bookstore if I did. Consequently, I try to summarize the concepts needed to install any Linux distribution into this one book with enough detail to get you through the process. (As you can imagine, that's a bit of a challenge!)



I chose Fedora as the sample distribution because Red Hat has become a recognized Linux standard, and its Fedora project is specifically aimed at home and small business users who cannot afford (or have no need) to purchase higher-level products. Even better, if you do use Red Hat Enterprise Linux (RHEL) in your office or organization, Fedora is a proving ground for the technologies that will make their way into RHEL.



Check out Appendix B for a full list of the goodies you can find on the DVD.