



1 - Open Source, Linux, Distributions and OpenLinux

In This Chapter

- Linux - seeing the big picture
- Understanding the Linux advantage
- Looking into the future of Linux
- Examining Linux distributions
- Deciding on OpenLinux

The goal of this chapter is to assist you in understanding where Linux came from, how it got to where it is today, and to offer a peek through the mist into the future of this rapidly growing operating system. We'll discuss Open Source software in general, including the origins of Open Source and an overview of the religious wars that seem to spring up whenever Open Source licensing issues and the associated rhetoric surface.

Next, we examine the relationship between Linux and the rest of the tools that comprise the Linux operating system, as well as the position currently occupied by Linux in the Open Source pantheon. A brief discourse on the major Linux distributions (commonly called "distros" by Linux users) follows, including an overview of overlap and difference among the players, and the myriad ways of acquiring Linux.

We close this chapter with a high pass over the feature set of Caldera's OpenLinux product family, taking a snapshot of the advantages and target markets of this popular set of distributions.

The moment you have been waiting for has arrived. This is why you bought this book in the first place. You want to know all about Linux. You may have tried Linux before and been unsuccessful or frustrated in accomplishing your goals. Then you committed the disk to the round file, or gave it an

honored place among your collection of shelfware. Linux has had a deserved reputation for difficult installs, and less than user-friendly interfaces. The other problem is that there is simply so much information, freely or cheaply available, that it might as well be secret since it's almost impossible for any one individual to fully grasp all the ins and outs of Linux.

This following passage is typical of an archetypal first encounter with Linux. After you read this, take heart. We did all the painful, silly things for you. If your experience was anything like ours, then we understand completely. We have spent years growing up with and learning to work with (or in spite of) the current system software mandated by the politics of business and the market.

We encountered this new operating system called Linux. There must be a trick somewhere here, as you can get it for free. Everyone knows that you get what you pay for, right? But what the heck, we gave this new software a shot. Should be no problem, we figured out how to run and install Windows, Mac OS, DOS - surely this won't be much different. But Linux didn't work. Or it worked, sort of, gave us a few screens full of gibberish that we couldn't understand, and died, leaving cryptic hieroglyphs in place of the regular icons that we meticulously memorized over the years. Free software, such a great deal when downloaded painstakingly over the Internet, didn't have the physical manual that you can have open next to you when the screen doesn't make any sense.

So we did the next bright thing. While the software is free, we could buy a copy that came with a CDROM, and a manual or two, and some email support. We got farther that time, finding out that our hot new video card that we bought to play Quake7000 isn't supported yet. But we got it running, figured out how to make dialup networking work in Linux, got an email from our boss with a MS WinWord document attached. We couldn't read it. Aaaargh. Just how quickly can a body reformat a partition, anyways?

Linux has grown up a lot over the years. The painful, text-only installation routines that ask you questions you can't possibly answer without a degree in system administration are now, for the most part, hidden behind the scenes. Recent versions of Linux do a decent job of auto-detecting and installing the correct drivers for most mainstream hardware. Hardware manufacturers, most especially video card makers, are becoming aware of the fact that Linux users are a force in the market, and most are working like mad to make drivers available for their products.

It isn't quite as easy to install Linux as Windows 98 or 2000, as we shall see in the next two chapters, though the OpenLinux installers bring it very close to that "ideal." However, there are compensations in the knowledge and control that is gained, some obvious and some rather arcane. Let's first take a look at the history behind this wildly successful upstart.

Linux - Seeing the Big Picture

Linux is often called free software, or more correctly: Open Source Software (OSS). Another common phrase you will encounter as you start working and playing in the Linux world is, "Free as in speech, not beer." This aphorism is indicative of the OSS philosophy that software should be free (as in constitutional free speech), not that programmers should starve for their art. There will be more on this philosophy as our work together in the following chapters progresses. For now, travel back into the mists of time with us, and look at the origins of what is now referred to as Free Software (or in a different, more corporate sense) Open Source.

What is Open Source?

There are many tales told of the beginnings of Open Source software. We could pick and choose among these, pleasing some people and angering others - this will probably happen anyway. So be it.

The truth of the matter is that people who write code for a living embody an oddly beautiful combination of art and science. This is exemplified in the pride most programmers take in creating software that is as follows:

- Clean, well documented, and easily maintained.
- Programs that are "tight," and execute quickly.
- Programs that fill a niche in the marketplace (or in the programmer's "toolbox").

Among the goals of programming there are conflicting imperatives. The best program Jane can write may be one that Bob can read and maintain next year, or it may be a blindingly fast implementation of a hash function that she wrote in a moment of revelation. The code is tight, convoluted, and passes the highest bar - it works under all circumstances and has a completely defined interface. It can be understood through careful study, but often cannot be modified without the extreme probability of introducing errors (a.k.a., bugs).

Since the early days of programming, hackers have been sharing their work with each other, building new layers of functionality on the blocks laid down before them. This was the mindset in the halls of Bell Labs where Unix was developed.

The birth and growth of Unix can be laid at Ken Thompson's doorstep. Blending new ideas with ones culled from the Multics OS, using the new 'C' language devised by Dennis Ritchie, Unix slowly spread through Bell Labs running first on the DEC PDP-7, then the PDP-11, and later, VAX boxes which were prevalent during this epoch. The advantage yielded by Unix and C was that the hardware and software tools had advanced to the point where most of an OS could, for the first time, be written in a relatively high level language making the porting of the OS to new hardware relatively straightforward.

Connected via the ARPAnet (Advanced Research Projects Agency Network), the improvements to Unix ebbed and flowed, focusing at Bell Labs and at UC Berkeley, morphing eventually into the two leading Unix versions - ATT System V and BSD (Berkeley Standard Distribution). Warring corporate entities put an end to that era of cooperation in the early 1980s, eventually yielding the fragmentation of the Unix market.

Another nexus of cooperation, and one that continues to bear fruit today, started in the MIT AI Lab. Beginning in the late 1960s, advanced research into so-called machine intelligence was a hot button item and MIT was on the front line. The in-house developed timesharing OS for the PDP-10 was called ITS. The OS and associated tools were freely shared among institutions and corporations. This environment (and philosophy) was adopted by Richard Stallman, and in due course gave life to both the GNU Project and the Free Software Foundation.

There are other names, other places, and other developments that we intend no slight towards as we move forward in our narrative. We bow low before you all - without the work of all the many pioneers in computing, the world would be a startlingly different one today (at least different in that we wouldn't be writing this book for you).

What is GNU?

Richard Stallman (often referred to by his login name and initials, RMS) began the GNU Project in 1981 to provide a non-proprietary suite of operating systems and tools that would counter what he saw as the menacing specter of commercial, proprietary systems and software. The first child of the project was GNU Emacs - the editor that can do virtually everything (see the Emacs section in Chapter 11). If someone wanted a copy of Emacs, Stallman would send it to them for a nominal copying cost. Then the issue of licensing arose. If RMS released his code into public domain, nothing would prevent the corporations from using his code to make a proprietary product. This was not his desire. The solution that surfaced is called "Copyleft", a name for a type of license which turns the tables on standard copyright terms. The original author retains rights to the work, but in distribution, the product must remain freely available.

The non-profit Free Software Foundation (FSF) was formed to provide a structure and organization through which funds could be funneled for free software development, education, and documentation. The FSF took over the distribution duties that had previously occupied much of Stallman's time. The Free Software Foundation is also the keeper of the keys, such as they are. At the FSF Web site (<http://www.fsf.org>), you can find reference to the central document of the Free Software/Open Source world, the GNU General Public License (GPL).

The GNU GPL, although unsupported by case law, seems reasonably stable and is relatively simple to explain. The short form basically states that all derivative works (of a GPL licensed program) are also covered by the GPL. Reasonable copying fees are permitted for the sources. Source code for revisions of GPL code must be made available if binaries are distributed. This is referred to in some sectors, with considerable derision, as viral licensing. We do not seek to provide here an interpretation or definition of the GPL. One very common piece of advice we see on the Usenet is to seek out a

Tip

Resources for locating copies of the GPL are everywhere. In your OpenLinux installation, look for the file `/usr/src/linux/COPYING`, which should be present no matter which installation mode you choose and contains the full text of the GPL. This book has a copy of version 2 of the GPL, in Appendix B. Then there is the canonical reference, <http://www.fsf.org/copyleft/gpl.html>.

copy, read it (all of it) and decide for yourself. We recommend you do so.

There are other licenses that are used in the Open Source community, some of which merely appear to have been written to appease the marketing and legal departments of the originating corporations. Others, such as the Artistic License, are apparently very similar to the GPL. You can learn more about other "approved Open Source" licenses at <http://OpenSource.org>.

Richard Stallman has distinct preferences about the terminology used to describe software of the free type. Some of these appear to be mere semantic follies, but are taken very seriously - as seriously as profits are in other corners of the software industry. "Free Software" (defined as licensed under the GPL) is preferred over "Open Source," as the latter is far more open to interpretation. The combination of Linux kernel and GNU tools (and all of the other public domain and "free" tools) that comprise what you might call Linux, is also often referred to as GNU/Linux.

A Kernel is not an OS

As noted, a kernel by itself does not make the OS. The long-term goal of the GNU project has always been a totally free operating system. Many of the bits and pieces of the OS had been written by the early 1990s. However, an operational kernel for the OS was lacking. The GNU Hurd (one alternative kernel, see <http://www.gnu.org/software/hurd/hurd.html> for more details) was and is in development, but progress remains slow.

In 1991, a Finnish student wrote an OS kernel to replace the Minix System written by Professor Andrew Tannenbaum. That student was Linus Torvalds, and the kernel was called Linux. Linus put word out on the Usenet that he had a minimally functioning OS kernel, and would anyone like to play with it, make bits work better, and add to it, please - a watershed event. Linus became the clearinghouse for revisions and additions to the Linux kernel from around the globe. Within three short years, version 1.0 of Linux was teamed up with several GNU development tools, and released on the world. It was the first operating system ever developed only by interested parties who were not paid for their work. Rather astonishing, really. In the interests of completeness, we note that Linux is a trademark registered in many countries to Linus Torvalds.

This passage is excerpted from one of the initial messages from Linus, to the newsgroup comp.os.minix:

Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.

As I mentioned a month ago, I'm working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02...but I've successfully run bash, gcc, gnu-make, gnu-sed, compress, etc. under it.

Although it is architecturally different from the design of the GNU Hurd, the Linux kernel performs the same basic functions - it provides the low level services that interface all applications with the hardware. Let's contrast that with another popular operating system (or two). In Windows and Mac terminology, the OS includes, incorporates, and requires the graphical operating environment. Applications for these operating systems generally reside in the layer above the window manager, requiring the graphical environment.

In Linux terms, applications are programs that talk to the device drivers that work with the kernel to make the hardware work. For example, the OpenLinux Lizard (Caldera's installation software) is an application that runs on an X-window client, that runs in conjunction with an X-Server that talks to the kernel that interfaces with the hardware. The low-level nature of the Linux base kernel, combined with the availability of the source code gives us a natural segue into our next topic.

Understanding the Linux Advantage

Here's the short form. When you have access to source code, you can inspect the software and adapt it to meet your needs. When you have access to GPL source code, you can change things, add other people's changes to your own code, and share your changes. This gives you more control over your

Tip

These advantages that come with license-mandated access to source code are also applicable to a number of other licenses, such as the BSD, MIT, and Artistic.

Recognized licenses with this feature are listed at

<http://www.opensource.org/licenses/>. There are other licenses derived from these that probably qualify as well.

system and more power; power to leverage other people's changes as they take advantage of your work.

Perhaps that is just a tad too succinct. When you pick up and install Linux for the first time and get it running, you get an incredibly nice feeling of having accomplished something that everyone said was hard. Hmmm. But then you note, every time the system boots (not nearly as frequent as before, since crashes are rare), that your system keeps saying something about a 386 kernel. You have a Pentium or an Athlon, don't you? You've paid for the new CPU and you want to use it. You can do that. You can customize that kernel to include bits that are needed to talk to your specific hardware, compile and install it yourself. When was the last time you compiled the Mac OS kernel? Windows? Never?

The main draw of the Linux system is that it is both customizable and optimizable. For almost every program, from the kernel on up, you have access to the source code. You can tune the software to work just a bit better for your system. Oh. You don't program? You might, before very long (and we mean that in a good way). However, these features still work to your benefit, because there are scads of people out in the world, working on bits of Linux all the time. For fun, for learning, for profit, people are helping to make Linux a stronger, better system. This has several benefits.

Developers are always trying to poke holes in code that already works, trying to make it fail, so that weakness, instability and insecurity can be removed from the running code. There are as many statistics about system stability and reliability as there are people to talk about it. So instead we will resort to this simple thought experiment. Let's say Microsoft will have 5,000 developers and testers work on Windows 2005, a hypothetical product. That costs a lot of money. But the only ones who get to look at the code are those 5,000 people. 10,000 strained, lonely (albeit, extremely well paid) eyeball are looking at a whole lot of code.

Contrast this with Linux development. All kinds of eyeballs are looking at all kinds of code, all the time. Individuals come up with a better idea, and post it to the Kernel mailing list, or mail it to Linus, or simply post it and say, "Play with this, please. I don't know why this bit doesn't work."

This "bazaar" model of software development (with a tip of the hat to Eric Raymond, GPL software developer and author of *The Cathedral and The Bazaar*) leads, perhaps inevitably, to an inherently more stable, more secure system. The principle of enlightened self interest is at play in this arena as well. Jane wants you to know how good she is at this stuff, and she's waiting to hear back from you on how well it works, because she actually makes changes based upon your input. In addition, the skills that a coder hones in working with OS software make for an extremely valuable set of career skills.

This makes for a low cost production model that is optimized for high rates of code evolution towards maximum dependability, stability, and recently, usability. If someone branches (or forks) the code, and does things that people don't like or don't use, then it doesn't make it back into the main tree. If what is developed has wide applicability (and actually works, or is terribly interesting), and then it will be adopted, worked over, and merged back into the main distributions. If something really bad is broken

that causes problems, there are ways to get to the people responsible and get fixes out in a timeframe measured in hours and days, not weeks or months. These are benefits not available from the makers of proprietary operating systems.

Looking into the Future of Linux

Now we visit our crystal ball, and gaze into the future for Linux.

The best estimates we can cull from a multiplicity of sources say that Linux is gaining ground rapidly. In 1999, 25 percent of new server shipments were Linux boxes. There are no hard statistics regarding Linux on the desktop yet, but it is claimed that there were over 15 million Linux users as of January 2000.

There are two major Graphical User Interface projects moving along at a fast clip, several available and evolving desktop office suites for user space applications like work processing, spreadsheets, presentation graphics, and more. There is even the perennial rumor about MS Office for Linux.

Our crystal ball says that Linux will continue to gain ground in server space. The battle for the desktop is up in the air, which is far more than could have been said a year ago. We both use Linux as at least one of our desktop units in our daily work, and can do 98 percent of exactly the same things that a user of Mac or Windows can do, and a bunch of other things that they can't. Competition will be good for Microsoft and for Apple. BeOS is also in the market, and there are rumblings in the BSD space as well. These will be interesting times.

In late 2000 or early 2001, a new Linux kernel, revision 2.4, makes its debut. New versions of the KDE and Gnome GUIs are in the works (or released, by the time this book hits your hands), along with extensive remodels of the associated toolsets and application suites. A discussion on the future of OpenLinux can be found in Chapter 12. There will be upgrades from virtually every Linux vendor. So let's look at distributions.

Examining Linux Distributions

A Linux distribution (or "distro") is, at the very least, a collection of software that incorporates a version of the Linux kernel, versions of the assorted GNU tools, and various other bits and pieces like XFree86, the Open Source version of the X-windows system. Many vendors of distributions offer this collection as a basic version, and charge (extra) money for versions that incorporate or include various non-free programs that would be illegal to "give away."

Understanding distributions

On the face of it, you might suspect that there is very little to differentiate the distributions. In one important sense, you are correct. You are installing Linux, after all. If the version of a program, or if the kernel itself doesn't meet your needs, then you merely need to find the version you want, download it, compile it, and install it. Many of the features we discuss in this book will apply in finer or broader detail to most of the distros currently available.

On the other hand, you may reach the point in using Linux where getting the latest version of X (in the generic sense) means needing the latest version of Y which requires the second-from-last revision of Z. At this point, it makes sense to consider upgrading or overloading your current system.

A survey of the major distributions

Following is a brief pass over the some of the major distributions of Linux available as of this writing. We apologize in advance if we left your favorite out. The players include Caldera, Corel, Debian, Mandrake, Red Hat, Slackware, SuSE, and TurboLinux. In addition, there are many smaller derivative distributions that target specific types of market or user, like Khaos Linux, a highly secured distribution that is currently in active development.

All of these and many of the smaller distributions keep programmers on payroll doing development work for Open Source projects (except perhaps Debian, which in its way IS an Open Source project). Part and parcel of the reputation of a Linux vendor is the relationship that it maintains with the community of coders and users that are its audience, market and goad, rolled up into one.

Caldera

Founded in 1994, Caldera currently puts out a coordinated family of distributions for business and personal use - OpenLinux. eServer 2.3 and eDesktop 2.4 products are current in the lineup, along with other related offerings. Caldera uses the .rpm software-packaging format, in common with many other distributions.

Corel

Corel Linux OS has the simplest install we've run across. Based largely on the Debian distribution, Corel has built a product that integrates well with its other Linux products, WordPerfect Office 2000 for Linux and Corel Draw Graphics Suite for Linux. Corel's packaging uses the .deb format.

Debian

Debian is the perhaps the most conservative of the bunch. As of September 2000, Debian 2.2 is shipping with the 2.2.16 kernel. Debian is widely respected as having an extremely stable, secure by default, distribution. Debian software is packaged in .deb format.

Mandrake

Mandrake has been a popular derivative of Red Hat for several years. The original Mandrake claim to fame is that they provided kernels precompiled and optimized for use on x586 and x686 processors. Now Mandrake is forging its own path, providing unique installation and administration tools, starting with version 7.0.

Red Hat

Red Hat is the apparent market leader in terms of unit shipments. They make a solid product that is used by many of the smaller distributions as a baseline for custom or specialty distributions. Red Hat is a leader in supporting R&D in Open Source development and has (to the best of our knowledge) contributed all of their in-house code to the "cause." This includes their popular software package format, .rpm, and its associated tools.

Slackware

Slackware is among the oldest of distributions, and uses the .tar.gz format for its distribution. Venerable is not necessarily a bad choice of words, yet with version 7.1 released recently, Slackware has upgraded its offering to put it near the front of the pack in terms of software versions. Like all the

distributions, *mentioned here* or not, Slack has its share of vocal adherents.

SuSE

SuSE is based in Germany and expanding rapidly into the global market. SuSE incorporates some leading-edge functionality in their recent 7.0 release, including LVM (Logical Volume Manager) and journaling filesystem ReiserFS. SuSE packages are in .rpm format, and they have a unique package management tool called Yast. [and now, Yast2]

TurboLinux

TurboLinux, from its market-leading base in Asia, is also aggressively growing a global presence. According to the literature, TurboLinux marketing focuses primarily on the server market, where Linux's expansion is most readily apparent. However, there is also a solid workstation product. TurboLinux is an .rpm based distribution.

The difference a distribution makes

The various distributions can be differentiated from each other in four major ways. First, the method and design of system installation is usually the most visible and telling feature by which distributions are distinguished. This is an area of rapid change and cross-pollination, as ideas about how best to help the first time users get Linux up and running fly about the Internet.

Second, distributions set themselves apart by the versions of the various software packages they ship with. A version of an application, service, or kernel that is too old may have known security holes, or lack necessary features. One that is too new may be regarded as be unstable.

Third, the graphical system administration tools that are in active development set several of the distributions apart from the rest. Although system administrators who came up through the Unix school of very hard knocks sometimes disdain the use of graphical admin tools, they are a feature that assist the newer users in bridging the gap from other operating systems to Linux.

Lastly, and of lesser importance, is the selection of the various software packaging methods. There are three primary types of software packaging extant in the Linux milieu. The original and ever-popular .tar and .tar.gz formats are still in use today, and are still the basic method of software distribution outside of getting a program with a specific distribution. There is the .deb format that is unique to the various Debian-based distributions. Lastly, and vastly popular, is the .rpm (or Red Hat Package Manager) package format, which is used, in one variant or another, across many of the most popular distributions. One important feature of package management is the ability to keep track of installed packages, and remove them easily.

Tip

Alien provides conversion between package types. Written and maintained by Joey Hess, Alien freely converts packaged software from one format to another, allowing (for example) a package that was issued in .deb format initially to be quickly converted to .rpm for a correctly documented installation in an OpenLinux system. Alien is available currently at <http://kitenet.net/programs/alien/>. Take advantage of Alien to install packages on your system that will be tracked by the native package database format for your distribution.

The mojo of distribution selection for Linux is this: Ignore the revision of the distribution. Compare the versions of the various major components and ask questions of your local Linux user group to achieve the understanding you desire. Slackware recently underwent extreme version inflation, from 4.0 to 7.0, leapfrogging the Red Hat numbers. The kernel and Xfree86 version numbers tell the tale though. We tell you three times: distribution version numbers are marketing tools.

Getting Linux

Starting sometime in 1999, Linux became one of the easiest piece of software to acquire. There are more ways to get Linux than you can shake a stick at. Almost every distribution is available online, in at least some basic form. You can download the whole distribution and install it from your hard disk. You can download an .iso image and burn your own, bootable installation CDR. In some cases you can actually install Linux over the Internet, using only a small handful of floppies to get the process up and running.

There are online vendors and computer/software bricks-and-mortar stores that sell disk-only distributions for little more than the cost of copying, and you can purchase a buffet-style package that includes several distributions for less than the cost of a movie and popcorn for two.

Of course, you can buy actual "official" distributions from the vendors themselves. Here you pay not only for the relatively nominal copy fee, but usually you get a bound manual, some level of installation support, and a (usually unspecified) portion of your purchase goes towards supporting those programmers who are actually writing the Open Source code that we use, and that you are about to begin using.

Deciding on OpenLinux

Caldera, with its OpenLinux distributions, has been a leader in ease of installation, and in the development of user-friendly administrative tools that buffer new users from the brusque and often confusing command line interface that underlies every pretty screen you see on a Linux system.

At the time of this writing, Caldera's OpenLinux lineup consists of two main Linux distribution versions eDesktop 2.4 and eServer 2.3 - plus a smorgasbord of other ancillary programs primarily targeted at network and server management (for a complete list of Caldera's current product offerings, see <http://www.calderasystems.com/products>).

Note

Of course, by the time you read this, everything remotely associated with computers or computing will no doubt be very different. Since we began writing this book, Caldera has added at least three new products to their existing stable: Tarantella (an applications server), a Technology Preview of the next-generation OpenLinux packages (see Chapter 12), and Volution (a product targeted at managing heterogeneous computer environments). In addition, Caldera recently announced the acquisition of SCO UNIX, which will likely spawn a whole new product line targeted at the corporate server market. Whew! Interesting times ahead?

eDesktop 2.4

eDesktop is a distribution targeted primarily at what might best be termed the "average" Linux user - that is, someone looking for a Linux-based alternative to Windows 98/ME or Mac's OS-X. It's a generic distribution that ships with a diverse range of utilities, desktop applications, and server-side programs such as Samba, Apache, BIND, and so on. In addition, Caldera's retail eDesktop packaging includes a CD-ROM with several commercial offerings - the StarOffice and Applixware office suites, a backup program, MoneyDance (a financial program akin to Quicken or MS Money), and a set of network management tools distributed by Citrix. This same CD-ROM also contains an installer that can be used to install eDesktop from the Windows environment.

eServer 2.3

eServer is targeted primarily at the systems administrator who wishes to install Linux in the role of a network server. This distribution contains all the same server applications that eDesktop has, sans the commercial package options. eServer also ships with a kernel tuned to the server environment (larger RAM support, hardware- and software-level RAID, and so on). In addition to a modified kernel, default configuration options differ between eDesktop and eServer. For example, disk quotas are enabled at[CT5] installation time, PAM (Pluggable Authentication Modules) are included to enhance basic system security, and PHP3 modules are available that enable dynamic Web page generation under Apache.

Support options

Caldera Systems supports their product line through a variety of support options and mechanisms, ranging from an extensive online Knowledge Base, free initial installation support via e-mail and telephone, to enhanced support contracts geared primarily at businesses deploying Linux on a large scale. For phone numbers and package details, see <http://www.calderasystems.com/support>.

If you run into that infamous Catch-22 and can't get online, call 801-443-1000. Support hours are Monday through Friday, 7:00 am to 7:00 pm, MST. Note that free installation and configuration support (details are outlined shortly) is only available for users who purchase retail packages.

Basic installation support (online and phone)

Most installation problems occur as a result of trying to install Linux on unsupported hardware. For this reason, if you encounter difficulties during installation, the first place to look is Chapter 2 of this book, which in turn refers you to Caldera's Hardware Compatibility List (<http://www.calderasystems.com/support/hardware>).

Once you're confident the hardware you're trying to install to is supported, the next place to check for answers - both generic and specific - is Caldera's Knowledge Base (<http://support.calderasystems.com/caldera>). It contains a searchable list of FAQs and solutions to common problems, plus the requisite link to e-mail a specific question directly to a technician. Again, note that the "Personal Assistant" feature is available only if you purchased a Caldera product; a serial number is required on the submitted query form.

Free installation support is available (providing you are using a purchased product and it is registered) for a limited number of initial configuration tasks:

- General installation on supported hardware
- Basic configuration of the graphical system (Xfree86)
- Basic network configuration
- Basic configuration of an e-mail client
- Basic printer configuration
- Basic configuration of a dialup connection
- Basic sound card configuration

To subscribe to one of Caldera's enhanced fee-based support packages, either call the number given in the introduction to this section, or go to <http://www.calderasystems.com/support/programs/enhanced.html>.

Online support

Unless you've purchased a fee-based support package, once your Caldera product is installed and the basics are working correctly, you are pretty much on your own to untangle any subsequent problems - self-induced or otherwise.

Beyond the material contained in this book, there are numerous sources of online help available specific to OpenLinux (an extensive list of both virtual and paper-bound Linux references can be found in Appendix C).

Mailing lists provide an open forum whereby a group of individuals with common interests can share experiences and pose questions. Two such lists run by Caldera are:

- The Announcements List (<http://www.calderasystems.com/support/forums/announce.html>), which is a low-volume source of product announcements, press releases, and security advisories.
- The OpenLinux Users List (<http://www.calderasystems.com/support/forums/users.html>), which is a relatively high-volume list, targeted at users helping other users. Caldera support personnel frequent this list when time permits.
- An additional resource is run by Caldera power users: they operate the Step-by-Step site (<http://linux.nf/>). Many of the most common questions that a user has can be answered by reading the material found here.

If personal contact and discussion is your bag, check to see if there's a Linux Users Group (LUG) in your area. A partial list has been compiled by Caldera, and is located at <http://www.calderasystems.com/support/usergroups/>. There's another LUG site, the Linux User Groups WorldWide, found at <http://lugww.counter.li.org/>.

Tip

If you think you've run across a program bug (that is, you've taken the time to document the error and it is repeatable), Caldera provides a Web page for reporting such incidents. Go to <http://www.calderasystems.com/support/programs/bugs.html>, and click the e-mail link found there. Be sure to provide enough detail that the engineer handling the message can attempt to replicate your findings.

Finally, we strongly recommend a routine scan of Caldera's Security Advisories (<http://www.calderasystems.com/support/security>). While everything posted here will not pertain to your installation or the programs you use, it's important to keep abreast of security issues as they arise. The integrity of any operating system is 50 percent knowledge and 50 percent mental awareness.

Summary

Linux involves a steep learning curve and until you learn enough, Linux is harder to use than other major operating systems. The good news is that due to the hard work of developers, the curve has gotten a lot less daunting in the last year or so, with more progress being made each month. A Linux distribution such as OpenLinux from Caldera is a composite of the Linux kernel and utilities from the GNU Project that work like their Unix counterparts. Additionally there are many, many programs from developers and organizations all over the world that round out the system into a functioning whole. This chapter covered the following points:

- Getting through installation to a functioning GUI environment is the biggest hurdle that Linux users face. New installation programs like Caldera's Lizard Installer make this task much easier than previously.
- Hardware incompatibility hobbles some installations; find the hardware lists for your distribution to locate the source of troublesome problems.
- Linux is an operating system kernel written by Linus Torvalds. Linux is licensed under the GNU General Public License (GPL). The GPL prevents licensed code from being used in proprietary products.
- GNU tools, developed by Richard Stallman and many others under the aegis of the Free Software Foundation and the GNU project, comprise the balance of the central product - a Linux (or GNU/Linux) distribution.
- Linux and other programs are also referred to as Open Source programs. The Open Source Foundation is a clearinghouse for licenses that are compatible with the aims of keeping software free. Of course, bitter arguments result over the varying definitions. The good news is that, disputes aside, Open Source/Free software is generally of extraordinarily high quality, and worthy of use and support.
- There are many distributions of Linux. These "distros" often serve different markets. They are also distinctive in their installation, packaging, and administration tools as well as the specific versions of software components that are bundled together. Caldera's OpenLinux family is one of the market leaders.

- Aside from supporting the development of Free (as in speech) Software, purchasing a retail distribution such as OpenLinux eDesktop 2.4 also provides installation support. This is a good method of getting over the first hump, and starting to use Linux.

#

Copyright 2001, Tom Syroid and Brian Bilbrey. All Rights Reserved.

Brian and Tom's Linux Book is licensed under the OpenContent Licence, V1.0.

This licence permits you to copy and distribute exact replicas of the OpenContent (OC) as you receive it, in any medium, provided you follow the several simple provisions in the license. Details may be found at <http://opencontent.org/opl.shtml>, and at the end of the Preface for this work.

This PDF file was generated by modifying and then printing the original HTML files from the online version using KWord, a program in the KOffice Suite (<http://www.koffice.org/>). Then the output was piped through the PDF generator which is part of the gs-aladdin package (a Ghostscript variant). The underlying distribution is Debian GNU/Linux, Sid (Unstable).