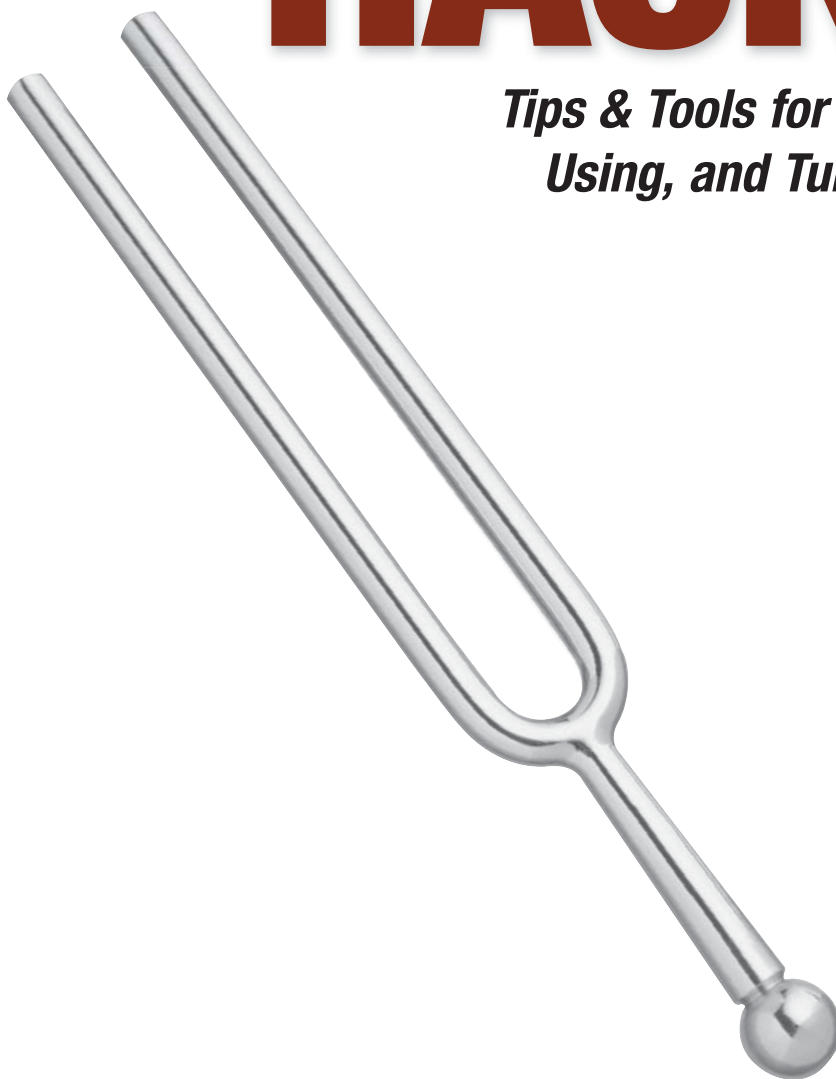

UBUNTU HACKS™

*Tips & Tools for Exploring,
Using, and Tuning Linux*



O'REILLY®

*Kyle Rankin,
Jonathan Oxe & Bill Childers*

Table of Contents

Credits.....	1
About the Authors.....	1
Contributors.....	1
Acknowledgments.....	2
Preface.....	3
Why Ubuntu Hacks?.....	4
How to Use This Book.....	4
How This Book is Organized.....	4
Conventions Used in This Book.....	5
Using Code Examples.....	6
Safari Enabled.....	6
How to Contact Us.....	7
Got a Hack?.....	7
Chapter 1. Getting Started.....	7
Hack 1. Test-Drive Ubuntu.....	8
Hack 2. Get Help.....	10
Hack 3. Make Live CD Data Persistent.....	12
Hack 4. Customize the Ubuntu Live CD.....	14
Hack 5. Install Ubuntu.....	20
Hack 6. Dual-Boot Ubuntu and Windows.....	24
Hack 7. Move Your Windows Data to Ubuntu.....	27
Hack 8. Install Ubuntu on a Mac.....	32
Hack 9. Set Up Your Printer.....	33
Hack 10. Install Ubuntu on an External Drive.....	37
Hack 11. Install From a Network Boot Server.....	41
Hack 12. Submit a Bug Report.....	44
Hack 13. Use the Command Line.....	46
Hack 14. Get Productive with Applications.....	49
Chapter 2. The Linux Desktop.....	53
Hack 15. Get Under the Hood of the GNOME Desktop.....	54
Hack 16. Tweak the KDE Desktop.....	57
Hack 17. Switch to a Lighter Window Manager.....	60
Hack 18. Install Java.....	65
Hack 19. Search Your Computer.....	67
Hack 20. Access Remote Filesystems.....	70
Hack 21. Tweak Your Desktop Like a Pro.....	73
Hack 22. Sync Your Palm PDA.....	76
Hack 23. Sync Your Pocket PC.....	80
Hack 24. Customize the Right-Click Contextual Menu.....	87
Hack 25. Download and Share Files with the Best P2P Software.....	89
Hack 26. Make your own PDFs.....	93
Hack 27. Blog with Ubuntu.....	96
Chapter 3. Multimedia.....	98
Hack 28. Install Multimedia Plug-ins.....	98
Hack 29. Watch Videos.....	100

Hack 30. Play DVDs.....	104
Hack 31. Buy Songs at the iTunes Music Store.....	105
Hack 32. Get a Grip on CD Ripping.....	107
Hack 33. Burn CDs and DVDs.....	110
Hack 34. Automate Audio CD Burning.....	112
Hack 35. Rip and Encode DVDs.....	114
Hack 36. Create a Video DVD.....	117
Hack 37. Connect to a Digital Camera.....	121
Chapter 4. Mobile Ubuntu.....	124
Hack 38. Put Your Laptop to Sleep.....	125
Hack 39. Hibernate Your Laptop.....	127
Hack 40. Prolong Your Battery Life.....	129
Hack 41. Get Proprietary Wireless Cards Working.....	130
Hack 42. Roam Wirelessly.....	133
Hack 43. Make Laptop Settings Roam with Your Network.....	135
Hack 44. Make Bluetooth Connections.....	141
Hack 45. Expand your Laptop.....	145
Hack 46. Hotswap your Laptop's Optical Drive.....	146
Chapter 5. X11.....	147
Hack 47. Configure Multibutton Mice.....	148
Hack 48. Enable Your Multimedia Keyboard.....	149
Hack 49. Configure a Touchpad.....	154
Hack 50. Connect Multiple Displays.....	158
Hack 51. Change the Ubuntu Splash Screen.....	163
Hack 52. Enable 3-D Video Acceleration.....	164
Hack 53. Make Your Fonts Pretty.....	168
Chapter 6. Package Management.....	172
Hack 54. Manage Packages from the Command Line.....	173
Hack 55. Manage Packages with Synaptic.....	175
Hack 56. Manage Packages With Adept.....	180
Hack 57. Install and Remove Standalone .deb Files.....	182
Hack 58. Search for Packages from the Command Line.....	185
Hack 59. Install Software from Source.....	187
Hack 60. Modify the List of Package Repositories.....	190
Hack 61. Cache Packages Locally With Apt-cacher.....	192
Hack 62. Create a Ubuntu Package.....	196
Hack 63. Compile a Source Package.....	199
Hack 64. Convert Non-Ubuntu Packages.....	201
Hack 65. Create Your Own Package Repository.....	203
Hack 66. Convert Debian to Ubuntu.....	205
Chapter 7. Security.....	208
Hack 67. Limit Permissions with sudo.....	208
Hack 68. Manage Security Updates.....	210
Hack 69. Protect Your Network with a Firewall.....	213
Hack 70. Use an Encrypted Filesystem to Protect Your Data.....	220
Hack 71. Encrypt Your Email and Important Files.....	226
Hack 72. Surf the Web Anonymously.....	231
Hack 73. Keep Windows Malware off Your System.....	234

Chapter 8. Administration.....	237
Hack 74. Edit Configuration Files.....	237
Hack 75. Manage Users and Groups.....	240
Hack 76. Mount Any Filesystem.....	244
Hack 77. Control Startup Services.....	247
Hack 78. Build Kernels the Ubuntu Way.....	251
Hack 79. Back Up Your System.....	255
Hack 80. Clone an Installation.....	260
Hack 81. Rescue an Unbootable System.....	262
Hack 82. Check the Captain's Log.....	266
Hack 83. Mount Removable Devices with Persistent Names.....	269
Hack 84. Mount Remote Directories Securely and Easily.....	273
Hack 85. Make Videos of Your Tech-Support Questions.....	276
Hack 86. Synchronize Files Across Machines.....	279
Chapter 9. Virtualization and Emulation.....	284
Hack 87. Run Windows Applications.....	285
Hack 88. Play Windows Games.....	292
Hack 89. Run Ubuntu Inside Windows.....	298
Hack 90. Use Xen to Host Virtual Machines.....	305
Hack 91. Create An Ubuntu/Xen Virtual Machine.....	308
Hack 92. Split Your Machine's Personality.....	315
Chapter 10. Small Office/Home Office Server.....	321
Hack 93. Install and Configure a Ubuntu Server.....	321
Hack 94. Build a File Server.....	323
Hack 95. Administer Your Server Remotely.....	328
Hack 96. Build a Web Server.....	332
Hack 97. Build an Email Server.....	334
Hack 98. Build A Caching Proxy Server.....	337
Hack 99. Build A DHCP Server.....	340
Hack 100. Build a Domain Name Server.....	343

Acknowledgments

AU: could you supply a couple of sentences here--see Wireless Hacks, 2ed for an example

Jonathan

The biggest thanks definitely have to go to my wife Ann and our children Amelia and Thomas, who for several months barely saw me from one week to the next. Writing Ubuntu Hacks has been one of those periods when everything else, including sleep, became secondary to just getting the job done and my family were amazingly supportive and understanding through everything.

Thanks also to my co-authors Kyle Rankin and Bill Childers who so willingly shared their knowledge of all things Linux, and the contributing writers who put in a big effort to supplement the body of the text with their particular areas of expertise. And the whole Ubuntu Hacks circus wouldn't have been possible without our editor, Brian Jepson, acting as ringmaster and keeping all our performances on schedule while even managing to contribute some of his own.

Finally, without the Canonical team there wouldn't be an Ubuntu to hack on, and without Debian there would never have been Canonical, and without the whole Free/Open Source Software community there would never have been Debian so the ultimate thanks have to go to the amazing community that we're all part of. To every person who has ever written Open Source software, or submitted a bug report, or written a how-to, or maintained a Debian package, or stood on a street corner and handed out Ubuntu CDs: Thank you. This book is written in your honour.

Kyle

First I'd like to thank my wife Joy for helping me yet again through the crunch period of this book. I'd also like to thank David Brickner for bringing me on this project along with Brian Jepson for his guidance in editing the book.

This book was the result of a great team effort so many thanks to Bill and Jon for all their hard work to make the book happen and thanks to all the contributing writers.

Finally I'd like to thank Ubuntu's amazing community of users and developers for their hard work in making Ubuntu a success in such a short amount of time.

Bill

I want to kick off the acknowledgments section by thanking and recognizing the most important people in my life—my family. Gillian and Conner, this book is for you. This is why Daddy's been at the keyboard for so many nights and weekends. Special thanks to Kelly for putting up with me while I undertook this project amongst all the other things I do—I love you, honey. You've been a tremendous influence and source of support, and I couldn't have pulled any of this off without you.

Thanks to my parents and grandparents for getting me my first computer and supporting my initial "addiction"—I wouldn't be where I am today without them.

Thanks to all the programmers, documentation people, bug testers, and everyone else who contributes to the Ubuntu and Debian projects. Every one of you should be proud of your work—you've created something truly special. Thanks to the crowd of #linux too, particularly

Jorge, whose ongoing pursuit of shiny stuff led me to run the prerelease of Warty way back when. Also thanks to my fellow writer Kyle—all the stuff we've been through has been a total blast, and I'm looking forward to the future.

Finally, thanks to David Brickner and Brian Jepson for giving me this shot and for editing all my mistakes. [sic]

Preface

The first release of Ubuntu, the Warty Warthog, was made available to the world on 20 October 2004. Less than two years later, Ubuntu is now the number-one most popular Linux version at DistroWatch.com, far ahead of the distribution in second place. Countless articles, reviews, and blog postings have been written about Ubuntu and its sister distros, Kubuntu and Edubuntu. In Macedonia, Ubuntu will be installed in 468 schools and 182 computer labs. In South Africa, HP is going to offer desktops and notebooks with Ubuntu on them. Around the world, hundreds of thousands of people have installed Ubuntu, and in many cases, it was the first Linux distro they'd ever tried. For many of those new Linux users, Ubuntu has been so good that they've switched to Linux. For a Linux distro that's still an infant, this is remarkable stuff!

Why has Ubuntu been so successful? Technically, it's based on Debian, which is an excellent foundation for a Linux distro, but Ubuntu has added a level of finish and polish that has made it a joy to use for newbies, while still a powerhouse for more experienced users. It's incredibly up-to-date; a team of dedicated developers ensures that everything “just works,” with regular updates to the various packages that make up the distro and a roughly six-month release schedule between distros.

But the secret behind the phenomenal success and growth of Ubuntu is really one man: South African Mark Shuttleworth. After founding Thawte, a company providing digital certificates, when he was 22, Shuttleworth sold the company four years later to VeriSign for a large amount of money. After fulfilling his dream of going into space, he decided to fulfill another and build the best Linux distro in the world. In that he has succeeded.

But it's also about principles with Shuttleworth. He has plenty of money, and he wants to do things with his fortune that will change the world. Consequently, Ubuntu will always aim for the highest quality, and it will always be free. The name *Ubuntu* itself is laden with meaning, as it is an African word meaning both “humanity to others” and “I am what I am because of who we all are,” while *Kubuntu* means “towards humanity.” Shuttleworth has promulgated the Ubuntu Code of Conduct, which states that members of the community must practice consideration, respect, and collaboration.

This is a book written by passionate Ubuntu and Kubuntu users who are excited to talk about a powerful, cool distro that meets the needs of novice, intermediate, and experienced users in a wide variety of ways. The hacks in this book cover the essential areas of Ubuntu, and they'll help you maximize your use of the distro. Whether you want to play music and movies, or use Ubuntu on your laptop as you travel, or install just about any software package you could ever want, or run other operating systems inside Ubuntu, we've got it all covered.

We know you'll get a lot out of Ubuntu Hacks, but we also want to encourage you to give back to the community and help grow Ubuntu:

Visit the main Ubuntu and Kubuntu web sites, at <http://www.ubuntu.com> and <http://www.kubuntu.org>. The entire sites are worth exploring in depth, but the Wikis especially offer a wealth of useful information.

Download Ubuntu and offer it to friends, family, and acquaintances. Heck, offer it to total strangers! The more people who try Ubuntu, the more people who will use Ubuntu.

If you don't want to download the distro, you can request free CDs at <https://shipit.ubuntu.com>. Don't be shy—ask and ye shall receive!

If you know how to program, consider becoming a Ubuntu developer. If you don't know how to program, there's still plenty of work you can do. Either way, head over to <http://www.ubuntu.com/developers>. And if you think you have the right stuff, you can even apply for work at <http://www.ubuntu.com/employment>.

Buy some Ubuntu swag from the Ubuntu Shop (<http://www.cafepress.com/ubuntuhop/>), or donate money at <http://www.ubuntu.com/donations>.

Most importantly, tell the world about Ubuntu! Let's get the word out: there's an awesome, free, super-powerful operating system that anyone can use named Ubuntu, and it's made for you.

Why Ubuntu Hacks?

The term *hacking* has a bad reputation in the press. They use it to refer to people who break into systems or wreak havoc with computers as their weapon. Among people who write code, though, the term *hack* refers to a "quick-and-dirty" solution to a problem, or a clever way to get something done. And the term *hacker* is taken very much as a compliment, referring to someone as being *creative*, having the technical chops to get things done. The Hacks series is an attempt to reclaim the word, document the good ways people are hacking, and pass the hacker ethic of creative participation on to the uninitiated. Seeing how others approach systems and problems is often the quickest way to learn about a new technology.

How to Use This Book

You can read this book from cover to cover if you like, but each hack stands on its own, so feel free to browse and jump to the different sections that interest you most. If there's a prerequisite you need to know about, a cross-reference will guide you to the right hack.

How This Book is Organized

This book is divided into ten chapters, organized by subject:

Chapter 1, Getting Started

This chapter shows you how to get started with Ubuntu. Whether you want to give it a whirl with a live CD, or you're ready to jump right in and install Ubuntu on your computer, you'll find what you need here. In addition to getting all the information you need to install Ubuntu on your system, you'll also learn how to get started with the Linux command line, set up your printer, file a bug report, and more.

Chapter 2, The Linux Desktop

You're going to spend a lot of time in front of a mouse, keyboard, and monitor, working with one of the Linux desktops. This chapter helps you get the most out of the GNOME and KDE desktop environments for Linux, and even helps you find out about a few others that are worth checking out. You'll also learn such things as how to get Java set up, how to work with files on remote computers, and how to get Ubuntu talking to handheld computers.

Chapter 3, Multimedia

This chapter gets the music and movies running so you can have some fun in between all the work you get done with Ubuntu. You'll learn how to play nearly any kind of audio and video, and burn files, music, and movies to optical discs.

Chapter 4, Mobile Ubuntu

If you're using Ubuntu on a notebook computer, you're probably going to want to cut a few wires. This chapter helps you get going with various wireless cards. You'll also learn how to get the most out of your laptop, from saving energy to installing add-on cards.

Chapter 5, X11

This chapter shows you how to tweak X11, the windowing system that lurks beneath the shiny veneer of KDE and GNOME. You'll learn how to get your mouse and keyboard working just right, and also how to get X11 configured so it takes full advantage of the graphics adapter in your computer.

Chapter 6, Package Management

To some extent, any Linux distribution is a big collection of packages held together by a whole lot of interesting and useful glue. Ubuntu's great advantage is the quality of those packings and all the testing and improvement that goes into them. This chapter shows you how to work with packages, whether you're installing them, finding new ones from beyond the edges of the Ubuntu universe, or creating your own.

Chapter 7, Security

This chapter shows you how to tighten up security on your system. You'll learn the basics of how the *sudo* command keeps you and your fellow users out of trouble, find out how to protect your network from intruders, and even keep your data safe if one of the bad guys does make it in.

Chapter 8, Administration

Every now and then, you're going to have to take a break from the fun of using Ubuntu and do some administrative tasks. Whether you're adding a new user, tweaking your system's configuration, or doing those backups you should have done long ago, you'll find what you need in this chapter.

Chapter 9, Virtualization and Emulation

This chapter shows you how to run Ubuntu inside of other operating systems, and other operating systems inside of Ubuntu. It's all made possible by a combination of emulation and virtualization, which effectively lets you run a computer inside of a computer.

Chapter 10, Small Office/Home Office Server

Ubuntu isn't just a great desktop operating system; it also makes a fantastic basis for a server. In this chapter, you'll learn everything from doing a basic server install to installing network services such as DNS, mail, Apache, and more.

Conventions Used in This Book

The following is a list of the typographical conventions used in this book:

Italic

Used for emphasis and new terms where they are defined, as well as to indicate Unix utilities, URLs, filenames, filename extensions, and directory/folder names. For example, a path in the filesystem will appear as */usr/local*.

Constant width

Used to show code examples, the contents of files, and console output, as well as the names of variables, commands, and other code excerpts.

Constant

width

bold

Used to highlight portions of code, either for emphasis or to indicate text that should be typed by the user.

Constant width italic

Used in code examples to show sample text to be replaced with your own values.

Gray type

Gray text is used to indicate a cross-reference within the text.



A carriage-return icon is used to indicate a line of code that carries over to the following line because of space limitations. These lines should be entered on one line at the command prompt.

You should pay special attention to notes set apart from the text with the following icons:



This is a tip, suggestion, or general note. It contains useful supplementary information about the topic at hand.



This is a warning or note of caution, often indicating that your money or your privacy might be at risk.

The thermometer icons, found next to each hack, indicate the relative complexity of the hack:

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

ED: Please let me know what the correct author order is

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Ubuntu Hacks* by Bill Childers, Jonathan Oser, and Kyle Rankin. Copyright 2006 O'Reilly Media, Inc., 0-596-52720-9."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari Enabled

When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://safari.oreilly.com>.

How to Contact Us

We have tested and verified the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). As a reader of this book, you can help us to improve future editions by sending us your feedback. Please let us know about any errors, inaccuracies, bugs, misleading or confusing statements, and typos that you find anywhere in this book.

Please also let us know what we can do to make this book more useful to you. We take your comments seriously and will try to incorporate reasonable suggestions into future editions. You can write to us at:

O'Reilly Media, Inc.
1005 Gravenstein Hwy N.
Sebastopol, CA 95472
(800) 998-9938 (in the U.S. or Canada)
(707) 829-0515 (international/local) (707) 829-0104 (fax)

To ask technical questions or to comment on the book, send email to:

bookquestions@oreilly.com

The web site for Ubuntu Hacks lists examples, errata, and plans for future editions. You can find this page at:

<http://www.oreilly.com/catalog/ubuntuhks/>

For more information about this book and others, see the O'Reilly web site:

<http://www.oreilly.com>

Got a Hack?

To explore Hacks books online or to contribute a hack for future titles, visit:

<http://hacks.oreilly.com>

Chapter 1. Getting Started

An operating system takes some getting used to. Whether you are new to Ubuntu or new to Linux itself, there are some basic things you need to get familiar with before you can move on. The hacks in this chapter cover those basics and then some.

The Ubuntu Live CD is a good way to explore Ubuntu without changing anything on your hard drive. This chapter explains how to get up and running with the Live CD, and even shows you how to use it with a memory stick to keep your settings and documents around between reboots. You'll also learn how to install Ubuntu, whether you want to make it the sole operating system on your computer or want to dual-boot between Ubuntu and Windows. You'll even learn how to install Ubuntu on a Macintosh.

This chapter also covers fundamentals such as getting your printer set up, getting help (and helping Ubuntu by submitting bug reports), getting started with the command line, and finding the most important applications you need to start "doing stuff" with Ubuntu.

Hack 1. Test-Drive Ubuntu



Use the Ubuntu Live CD to get to know Linux before installing it on your system. This is simply the fastest and safest way to try out Linux.

Though Linux on the desktop looks and behaves a lot like Windows, the simple fact is it isn't. Your favorite Windows programs probably won't run in Linux, it may be difficult to migrate data from your Windows install [Hack #7], and the years you've spent getting used to how Windows does things will prove mostly useless when it comes to understanding how Linux works. With all of this in mind, wouldn't it be great if you could try out Linux without spending hours or days getting it installed and configured on your system? Well, you can. With the Ubuntu Live CD, you can take Linux for a test-drive to be certain you really want to commit the time and resources to running it full-time. This hack shows you how to download the Ubuntu Live CD and boot your system using it. Other hacks in the book show you how to get around in GNOME [Hack #15] or KDE [Hack #16], the two popular graphical environments that run on top of Linux.

Downloading the Live CD

A *live CD* is a complete installation of Linux that runs entirely from CD. While you are using a live CD, nothing is written to your hard drive, so your Windows or Macintosh installation is not affected in any way. However, because you're running from a CD, you're limited to using only the programs that are installed on the CD, and everything will run a bit slower because CD access is much slower than that of a hard drive. Still, even with these limitations, it's undeniable that a live CD is the easiest way to try out Ubuntu.

You can obtain the Ubuntu Live CD from the main Ubuntu web site (<http://www.ubuntu.com>). There is a convenient Download link that takes you right to the download page to get the latest released version of Ubuntu. This hack, indeed this entire book, was written for the Dapper Drake release, version 6.05, because it is the release that will be supported for the next five years (previous Ubuntu releases were only supported for 12 months). Ubuntu versions are numbered according to the year and month of release; therefore, this version of Dapper Drake was released in May of 2006. Regardless of which version you download, the hacks in this book should be valid for a long time to come.

AU: Please verify dates and version numbers in above paragraph

The file you want to download is the ISO image that corresponds to the computer type you are using. If you're on a PC, this probably means the x86 version, but if you happen to be using a 64-bit AMD or Intel processor, you want to download the 64-bit PC version. Finally, if you're a Mac user, you want to get the PowerPC version. It is unknown at the time of this writing if Ubuntu will support the new Macintoshes with Intel processors.

You can burn the ISO image to disc using any CD burning software you have installed on your computer. Make sure you choose the option that burns the image to disc; don't select the option to burn a data CD that will just copy the image over as a file. The difference is that the former will create a bootable disc, and the latter will not.

Booting the CD

To use a live CD, you typically need do nothing more than boot your computer with the CD already in the optical drive. Most Windows computers these days are preconfigured to boot from a CD or DVD before booting from the hard drive. I fancy this is because users often need

to restore or repair their Windows installation using the OEM-provided restore CD, and this configuration saves a lot of calls to technical support.



If you are using a Mac running OS X, you need to hold down the C key to boot from a CD.

But, if for some reason your Windows computer doesn't want to boot from the CD, the fix is usually quite simple. You need to boot into your computer's BIOS and modify the setting that specifies the boot order. Getting into the BIOS usually requires you to press a key early on in the boot sequence. The key you press depends upon the make of your computer and BIOS, but it is typically displayed on the splash screen that comes up when your computer starts (the one that announces the manufacturer of the computer, not the Windows splash screen). If your splash screen doesn't tell you this information, try one of these keys: Esc, Del, F2, F10, or F12.



On some computers, F12 launches you directly into a boot selection menu, offering options such as booting from hard disk, floppy drive, USB drive, optical drive, or the network. This lets you boot from a different device without making changes to your BIOS configuration.

Once you're in the BIOS, you should look for a menu called Boot or one labeled Advanced Configuration. Under this menu, you should see a setting that allows you to specify that the CD or optical drive boot before the hard disk. There are hundreds of BIOS variants, so I can't be more specific than that, but if you look at every option screen, you will eventually see the setting you need to change as well as instructions for how to do so. Once you've made the change, save it, and then reboot your computer.

Hopefully, this will be the only problem you have booting from the Live CD. If you've got the BIOS configured correctly, shortly after boot you should see a splash screen with the following options:

Run preinstalled live system

This option loads the Live CD environment so you can test-drive Ubuntu. If you don't press any keys within 30 seconds of getting to this screen, this option will automatically execute.

Rescue a broken system

Choose this option to load a minimal Linux environment that you can use to troubleshoot a nonworking Linux installation.

Memory test

You can use this option to run a test of your computer's RAM. Many people don't realize it, but many odd computer problems can be traced to bad RAM modules. If your computer exhibits erratic behavior, such as frequent freezes or an inability to consistently finish booting, your RAM may be the culprit, and running this program may save your hours of frustration.

Boot from the first hard disk

Select this option to continue booting from the hard drive.

Unless you're troubleshooting, about the only other option of interest right now is pressing F2 to select a language. This setting determines the language and keyboard layout that will be used for the rest of the test-drive; the default is English.

Once you've made any necessary language selections, you should use the arrow keys to select "Run preinstalled live system," and press Enter. This begins the loading of Ubuntu. You'll see a lot of messages flash by on the screen and eventually be faced with a text dialog to configure your screen resolution. You can use the tab and arrow keys to move the selection cursor, the spacebar to toggle a selection, and Enter to accept your input and move on to the next screen. You can select multiple entries, depending on what your monitor supports. Ubuntu will use the highest selected and supported resolution as the default.

After this Ubuntu continues to load, and if all goes well, you'll automatically be logged into a GNOME desktop less than a minute later. Depending upon your hardware (network, sound, printer, etc), you may find everything preconfigured and working. If you don't, some of the hardware-configuration hacks later in this book may be useful even in the Live CD environment.

Another Use for the Live CD

The Ubuntu Live CD also includes Windows versions of several open source programs. To access these program installers from within Windows, just insert the live CD while logged in. Within a few seconds the autoloader feature of Windows should display a window that lets you launch each installer. If this doesn't happen you can just open Windows Explorer, navigate to the CD, and use the installers found in the program directory. The programs on the CD are:

OpenOffice.org 2.0

This is a free office suite that includes a word processor, spreadsheet, database, drawing program, and web page creator. OpenOffice.org (the *.org* is really a part of its name, but you can abbreviate it to OOo) can open and save to Microsoft Office formats, which means you may be able to use it in place of that office suite, or at the very least collaborate with others who do. You can learn more about OOo at the OpenOffice.org web site (<http://www.openoffice.org>).

Mozilla Firefox 1.5

Firefox is a web-browsing alternative to Microsoft's Internet Explorer. This secure and feature-rich web browser took the computer world by storm in 2005 and became the first browser to gain market share against IE since the mid-90s. To learn more about Firefox, visit the Mozilla web site (<http://www.mozilla.org>). Pay particular attention to the information about tabs and extensions, two features that can dramatically enhance your browsing experience.

Gaim 1.5.0

Gaim is a multiprotocol instant-messenger program. This means it can connect to multiple networks, such as AOL, MSN, Jabber, and Yahoo! all at the same time, making it easy for you to stay connected to your friends without having to run a separate chat client for each network.

Each of these programs is also part of the Ubuntu Live CD experience, so you can try them out before installing them to Windows. If you like the Live CD so much that you want to keep using it, be sure to check out "[Make Live CD Data Persistent](#)" [Hack #3].

Hack 2. Get Help



Find out where to get more help on using Ubuntu. Forums, Wikis, IRC chat rooms, and a built-in help system stand at the ready.

Everybody needs a place to turn to when he gets stuck. One of the nice things about Ubuntu Linux is the amount of help you can receive, if you know where to look. The development team at Canonical has put together an excellent support infrastructure that includes both free and nonfree support solutions.

Web-based Documentation

Your first stop on the support train should be the Ubuntu Support page at <http://www.ubuntu.com/support>. This page contains links for all the currently possible support methods, both official and unofficial, paid-for and free.

Of course, Ubuntu has excellent documentation. The official documentation effort at <http://help.ubuntu.com> has both a Quick Tour section and a comprehensive Start Guide. The Quick Tour page is a great flyer that advertises the high points of Ubuntu and shows off some screenshots, while the Start Guide is more of an overall how-to document.

The next place to visit if you're stuck should be the Ubuntu Wiki (<https://wiki.ubuntu.com>). The Wiki is extremely comprehensive and is constantly updated by Ubuntu users and developers. As a result, it's typically more up-to-date than the official documentation. One of the side benefits to the Ubuntu Wiki is the Laptop Testing area at <https://wiki.ubuntu.com/LaptopTestingTeam>. If you're about to install Ubuntu on a laptop, you might want to see if your model is on that page, since the Laptop Testing Team puts all their installation notes and tweaks down on that area of the Wiki. It might save you a lot of work and could very well help you get a troublesome feature like wireless or power management working correctly under Ubuntu Linux.

Interactive Help and Support

If you've got a question that you can't find the answer to, you can ask it in either the Ubuntu Forums or the Ubuntu IRC chat room. The Ubuntu Forums at <http://www.ubuntuforums.org> provide a nearly real-time support venue that you can also search. Odds are, if you're having a problem, someone else has already had that problem and asked for help on the forums. If you've got a more urgent issue, or just want instant gratification, you can ask for help in the IRC chat room. The IRC room is located on the *freenode* network (*irc.freenode.net*), and it's called *#ubuntu*. If you've never used IRC before, just click on the Applications menu, select Internet, and launch Xchat. Log in to *irc.freenode.net* and join the *#ubuntu* channel. Once you're online, ask your question, but be sure to provide as much detail as possible for the people in the room. Please note that most of the people there are volunteers who are contributing to the Ubuntu effort by trying to offer support, so be friendly and be prepared to answer questions that anyone in the room may ask in return, since they may need more information to figure out your issue. You might want to lurk in the channel for a while and read the messages that scroll by to get a feel for the tone and flow before you ask your question.

A lot of the work that makes Ubuntu what it is happens on mailing lists. There's a comprehensive list of mailing lists at <https://lists.ubuntu.com/mailman/listinfo>; you can either search the archives of these lists, or you can add yourself to them and post your question there. If you choose to post a question to one of these lists, please show proper etiquette and ensure your question is targeted at the correct mailing list. As with IRC, it's worth spending some time to get familiar with the mailing lists: read some older posts and responses, and pay attention to which questions get answers and which ones don't.

Traditional Pay-per-Incident Support

If you can't get a solution to your problem through the aforementioned free methods, there's always paid-for support through Canonical and other organizations. The page at <http://www.ubuntu.com/support/supportoptions/paidsupport> details the various options open to you for paid support. If you're considering using Ubuntu in a corporate environment, you should become familiar with this page. One thing to note is that with the paid-for support, Canonical has service-level agreements, which is something that you won't get on the free side of the support house.

AU: What are service-level agreements? Are these types of support contracts? Any further clarification needed in previous paragraph?

Whatever your need, the Canonical team and the larger Ubuntu community should have it covered. The support community is widespread, knowledgeable, and ready to help, so don't let a snag in your installation damage your Ubuntu experience!

Hack 3. Make Live CD Data Persistent



Take your desktop with you on a USB stick and access it anywhere with the Ubuntu Live CD

Wouldn't it be handy if you could walk up to any random computer, insert a copy of the Ubuntu Live CD, plug in a USB key, boot it up, and have a fully working system with your own documents, settings, and programs—without modifying the computer in any way?

A little-known feature of the Ubuntu Dapper Drake Live CD allows you to do exactly that. When it starts up, it searches for a volume that has been given the label *casper-cow* and uses it to store documents, themes, and even extra programs that you install. This is far more powerful than just booting up a live CD and mounting a memory stick as your home directory because it's not restricted to just storing your documents. It gives you the flexibility of a fully installed system, while retaining the "go anywhere" feature of a live CD.

You can perform this trick with just about any storage device, including removable USB hard disks and compact flash drives, but for this hack I use a USB memory stick because they're cheap, portable, and commonly available in increasingly large capacities.

Set the Label on Your USB Memory Stick

Connect the USB memory stick to your regular Ubuntu computer. Ubuntu will probably mount it automatically, so the first thing to do is to find the device name that it has been assigned. Open Applications→Accessories→Terminal and type the following at the shell prompt:

```
$ df -h
```

to see a list of mounted volumes. The output should look something like this:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hda3	54G	19G	35G	36%	/
varrun	506M	84K	506M	1%	/var/run
varlock	506M	0	506M	0%	/var/lock
udev	506M	116K	506M	1%	/dev
devshm	506M	0	506M	0%	/dev/shm
/dev/hda1	221M	28M	181M	14%	/boot
/dev/sda1	498M	214M	285M	43%	/media/usbdisk

USB storage devices are emulated as SCSI devices by Linux, and you can see the last device is listed as */dev/sda1*. This means SCSI device A, partition 1. If you have anything on the memory stick that you want to save, now is the time to copy it onto your computer, because you're about to totally erase it.

Once you've backed up your files, it's time to unmount the device:

```
$ sudo umount
```

```
    /dev/sda1
```

Ubuntu is smart enough to figure out if you are “in” the device (either on the command line or using the file browser), so if the system refuses to unmount because the device is still in use, just close any other windows you have open and try again.

Then create a new filesystem with the correct label:

```
$ sudo mkfs.ext3 -b 4096 -L casper-cow  
/dev/sda1
```



You must replace `/dev/sda1` with the actual device name used by your memory stick. If you have other USB devices attached, it is possible that one of them has claimed this device name. If in doubt, run the command `dmesg` right after you plug the memory stick in. You should see a message indicating the name of the device that was used to represent your memory stick.

This will create an `ext3` journaling filesystem, which is a good choice for general purpose use, but if you prefer you can use any filesystem that’s supported by the Live CD. The `mkfs.ext3` command will report some statistics about the new filesystem and then you’re ready to try it out.

Boot the Live CD In Persistent Mode

Plug your USB memory stick into the target machine, power the computer up, and quickly insert the Dapper Drake Live CD. If the computer is not configured to boot from CD-ROM, you may need to press a key (typically DEL or F2) at startup to enter the BIOS settings menu; you then need to change the order of the boot devices to put CD-ROM at the top of the list, and then select the Exit option (the one that saves your changes to the BIOS) from the BIOS menu. The computer will then boot up again and look for the Live CD before attempting to boot from the hard disk. Some computers have a menu (often activated by F12) that let you choose which device to boot from without having to make changes to your BIOS.



If you are using a Mac running OS X, you need to hold down the C key to boot from a CD.

When the Live CD starts up, you will see a menu. Normally you would just press Enter to start the boot process, but instead press F4 to access the Other Options menu that allows you to start up the Live CD in special modes. You’ll see a list of the arguments that will be passed to the kernel on startup: just add a space and type `persistent`; then hit Enter.

That’s it!

Testing Persistence

The computer will now boot from the Live CD in persistent mode, but you won't see anything different. In fact, it can be quite hard to tell if it even worked or not. As a simple test, you can try changing something obvious, such as your desktop picture, and then you can log out and reboot the computer back into persistent mode. If everything worked properly, your desktop picture will still be set as you specified.

Try changing other things on your system such as creating documents or even installing extra software. Changes you make should be preserved even after you reboot the system.

How It Works

The Live CD is a read-only environment, so of course you can't save changes made to the running system straight to the CD. However, when running in persistent mode, the system on the Live CD allows items on your memory stick to override items within the Live CD environment. In the test described in this hack, you changed the desktop image; this caused Ubuntu to save your new desktop picture and settings onto the *casper-cow* device. The next time the Live CD sets the desktop, it detects that a new setting has been stored on the device and applies it instead of the default setting. The Live CD therefore provides the basic data for a complete, functional environment, and any changes you make to that environment are written to the removable device and used to override the default settings.

Hack 4. Customize the Ubuntu Live CD



Rip, burn, and boot to create a personalized version of the Ubuntu Live CD with your choice of software and documents.

The Ubuntu Live CD [Hack #1] contains a complete Ubuntu installation that can run directly from the CD itself, without needing to be installed onto a hard disk. It's ideal for demonstrating Linux on computers with another operating system installed because after you take the CD out and reboot the computer, it returns to exactly the state it was in originally. It's a totally painless way to take Linux for a test run with no risk.

The Live CD is also extremely useful for recovering an unbootable machine: just pop in the Live CD and reboot, and you will have a fully running Linux system from which you can access the internal hard disk, copy files across the network, or do whatever else you need to do to fix the system. And as you saw in "Make Live CD Data Persistent" [Hack #3], you can even use a memory stick to store changes made inside the Live CD environment.

The Ubuntu Live CD starts up a full desktop environment that's functionally identical to a standard Ubuntu installation, but perhaps you want a Live CD that contains specific software or documents to suit your environment. For example, you may want to create a Live CD that boots up a machine as a fully configured router and firewall with no hard disk. Or maybe you want a forensics disk preloaded with virus-scanning and network-analysis tools plus the checksums of important files.

No problem, you can create a customized version of the Ubuntu Live CD configured exactly the way you want it.

Basic Requirements

Building the disk image for the Live CD takes a huge amount of storage so you'll need up to 5 GB of swap plus at least another 3 GB of disk space for storing the image. You'll also need tools for creating and mounting disk images.

Add extra swap

While the disk image is being compressed, *two copies* of it are held entirely in memory, so without a huge amount of swap, you won't be able to do the compression necessary to generate the ISO.

Don't worry if you don't already have a 5 GB swap partition. You can set up a temporary swapfile inside one of your existing partitions without having to reformat. Assuming you have at least 5 GB of space free inside */tmp* (usually in your root partition), you can create the extra swapfile with *dd*:

```
$ sudo dd if=/dev/zero of=/tmp/swap bs=1M count=5000
```

It can take a very long time to create the swapfile, so you'll need to be patient. Once the file itself has been created, you can set up a swap filesystem on it and activate your new swap:

```
$ sudo mkswap /tmp/swap  
$ sudo swapon /tmp/swap
```

You don't need to disable your existing swap first. Linux is smart enough to handle multiple swap partitions simultaneously, so your system should now have a total swap space comprising the new 5 GB swapfile plus your existing swap.

Install the tools

To mount a disk image as a loopback device and generate the ISO for your custom Live CD, you will need the *loop-utils* and *mkisofs* packages, and to work with the *squashfs* compressed image on the Live CD, you'll need the *squashfs-tools* package:

```
$ sudo apt-get install cloop-utils mkisofs squashfs-tools
```

Standard Live CD

While it's possible to build a Live CD from scratch, it's much easier to start by modifying the standard Ubuntu Live CD. You can download the Live CD ISO disk image from <http://cdimage.ubuntu.com/> or use one of the CDs available from Canonical through the ShipIt program (<https://shipit.ubuntu.com/>).

Prepare Original Image

Make sure your locale is set to C to prevent Unicode problems with the build process:

```
$ export LC_ALL=C
```

Mount the original Ubuntu Live CD ISO image as a loopback device:

```
$ mkdir ~/mnt
$ sudo mount dapper-live-i386.iso ~/mnt -o loop
```

This will mount the CD image inside your home directory at `~/mnt`. You can use an alternative location or mount the actual Live CD in your CD-ROM drive if you prefer.

Copy everything from the mounted image into a working directory, but make sure you skip the `filesystem.squashfs` compressed filesystem because you'll need to extract that separately. You can use `rsync` to make it easy:

```
$ rsync --exclude=/casper/filesystem.squashfs -a ~/mnt/ ~/extracted_cd
```

Next extract the compressed filesystem. The Dapper Live CD uses the `squashfs` read-only filesystem, unlike previous Ubuntu Live CDs, which just used `cloop` filesystems. To work with `squashfs`, you will need to load the `squashfs` kernel module:

```
$ sudo modprobe squashfs
```

Now you can mount it and copy it onto your local hard disk:

```
$ mkdir squash
$ sudo mount -o loop ~/mnt/casper/filesystem.squashfs squash
$ sudo cp -a squash extracted_fs
```

Be prepared to wait quite a while for this to run. Once it's finished, you will have a complete, extracted copy of the Live CD image, so you can unmount the original:

```
$ sudo umount ~/mnt
```

Set Up the Target Filesystem

Mount the `proc` and `sys` virtual filesystems into your target:

```
$ sudo mount -t proc proc ~/extracted_fs/proc
$ sudo mount -t sysfs sysfs ~/extracted_fs/sys
```

In a moment, you'll be chrooting into the CD image, so if there are files you will need on your customized CD, the easiest thing to do is mount */home* into it:

```
$ sudo mount -o bind /home ~/extracted_fs/home
```

Then once you are in the chroot, you will have full access to any files stored in your home directory.

Apply Customizations

Use *chroot* to enter the filesystem image:

```
$ sudo chroot ~/extracted_fs/ /bin/sh
```

Now, as far as you're concerned, you're running on a read/write installation of the Live CD. From there, you can use the usual package tools to update programs installed on the Live CD.

Delete unnecessary packages

The default Live CD is fairly full, so if you want to install extra packages, you will probably need to make some room first. If you want some ideas about which packages to remove, you can create a list of installed packages sorted by size using this command:

```
$ dpkg-query -W --showformat='${Installed-Size;10} ${Package}\n' | \\  
    sort -gr | less
```

Be very careful, though, because some packages are essential for the system to work at all. The GNOME Live CD is based on Ubuntu, so if you're looking for inspiration for which packages you can safely remove, you can start by looking at its configuration file, available at <http://cvs.gnome.org/viewcvs/livecd-project/livecd.conf?view=markup>.

Once you've settled on some packages to remove, you can uninstall them using *dpkg*:

```
$ sudo dpkg -r --purge packagename
```

Install additional packages

The regular network-based package tools won't work inside the chroot, so unfortunately it's not as simple as `apt-get install foo` to add packages. There are a number of ways around it, such as copying in a *hosts* file with the addresses of repository servers pre-resolved, because you can't do DNS lookups inside the chroot.

The simplest way, though, is probably just to predownload some packages into your home directory and use *dpkg* to install them after entering the chroot.



One very cool trick to simplify this process is to run the Synaptic package manager on your host system, find and mark the packages you want to install on your Live CD, and then select File→“Generate package download script.” You will then have a script that you can execute to fetch and save the packages locally, storing them in your home directory for access from the chroot.

Customize the home directory

When the Live CD boots, it creates the user’s home directory from scratch each time, using the files in */etc/skel*. If you have specific files you want to include in the home directory, you can put them in *skel*.

Unmount Customized Image

Now that all your changes have been applied, exit the chroot and then unmount the various filesystems:

```
$ exit
$ sudo umount ~/extracted_fs/home
$ sudo umount ~/extracted_fs/sys
$ sudo umount ~/extracted_fs/proc
```

Your customized filesystem is now ready to recompress, but first you need to generate a new manifest file that reflects the changes you have made to the list of installed packages. If you didn’t actually install or remove any software, you can skip this step.

You can’t perform this step using *sudo* (you have to really be running as *root*), so get a *root* shell with `sudo -s`:

```
$ sudo -s
```

Now generate the new manifest:

```
# chroot extracted_fs dpkg-query -W \\  
    --showformat='${Package} ${Version}\\n' \\  
> extracted_cd/casper/filesystem.manifest
```

You can exit the *root* shell now.

Repack the Filesystem

The new *squashfs* filesystem that will go inside the CD needs to be created:

```
$ sudo mksquashfs extracted_fs extracted_cd/casper/filesystem.squash
```

Once again, this stage can take a really long time.

The Live CD also needs to contain a checksum file that can be used to verify the integrity of the compressed filesystem. The checksum needs to be calculated from inside the CD image:

```
$ cd ~/extracted_cd
$ find . -type f -print0 | xargs -0 md5sum > md5sum.txt
```

Build the ISO

Everything up until this point has been architecture-independent, but the final stage of building the ISO itself depends on what type of system you are running:

x86 (i386) and x86_64 (amd64)

Use the following command:

```
$ sudo mkisofs -r -V "Custom Ubuntu 6.04 Live CD" \\  
-cache-inodes \\  
-J -l -b isolinux/isolinux.bin \\  
-c isolinux/boot.cat -no-emul-boot \\  
-boot-load-size 4 -boot-info-table \\  
-o custom-dapper-live-i386.iso extracted_cd
```

PowerPC

For PowerPC it's necessary to download *hfs.map*:

```
$ wget http://people.ubuntu.com/~cjwatson/hfs.map
```

Then build the actual ISO:

```
$ sudo mkisofs -o new_image.iso -chrp-boot \\  
-U -part -hfs -T -r -l -J -A "application_id" \\  
-sysid PPC -V "valid" -volset 4 -volset-size 1 \\  
-volset-seqno 1 -hfs-valid "volume_name_hfs" \\  
-hfs-bless extracted_cd/install \\  
-map hfs.map -no-desktop -allow-multidot extracted_ppc_cd
```

IA64

Use the following command:

```
$ sudo mkisofs -r -V "Custom Ubuntu 6.04 Live CD ia64" \\  
-o custom-dapper-live-ia64.iso -no-emul-boot \\  
-J -b boot/boot.img -c boot/boot.catalog extracted_cd
```

Burn and Boot

You now have an ISO image of your customized Live CD, so burn it to a disc [\[Hack #33\]](#) and give it a try.

More Information and Scripts

The process of creating a customized Live CD is quite manual and laborious, but some of the steps above can be simplified using the *live_cd_tools* scripts that you can find online at <http://wiki.ubuntu.com/LiveCDCustomizationHowTo>. Note, however, that the process for building the Dapper Live CD is a bit different from the older process used by previous releases, such as Breezy, that used a compressed *loopback* filesystem instead of *squashfs*, so make sure you don't use scripts intended for the older process.

Hack 5. Install Ubuntu



Learn how to install Ubuntu on your computer.

If you've given Ubuntu a test-drive [\[Hack #1\]](#), or you're simply ready to dive into it sight unseen, all you need is an installation CD and a computer to install it on, and you can be up and running right away. There are a number of ways you can get an installation CD; if you've got broadband and a CD-R drive, you can probably get your hands on it in under an hour.

System Requirements

Ubuntu will run on just about any current personal computer. If you're using an Intel-compatible PC, it will probably "just work," since the kernel image that Ubuntu uses by default is optimized for the 80386, which means it will also be compatible with systems based on the 486, Pentium, and Pentium Pro, as well as the Pentium II, III, 4, and beyond, including all the other mainstream Intel-compatible CPUs such as the AMD Athlon and Sempron, as well as the Transmeta Crusoe and Efficeon. If your computer can run Windows 95 or later, it can probably run Ubuntu just fine. If you're running an AMD64 system, there is even a special version of Ubuntu you can download.

If you've got a G3, G4, or G5 Macintosh, you'll probably be able to run the PowerPC version of Ubuntu. If it can run Mac OS X, it should be able to run Ubuntu. Mac users should see "[Install Ubuntu on a Mac](#)" [\[Hack #8\]](#) for complete details.

Although you may have a CPU that's compatible with Ubuntu, you may run into some hardware that doesn't want to play along. Wireless network cards can be particularly tricky, but after you get Ubuntu up and running, there are some tricks [\[Hacks #41 and #42\]](#) you can use to get them working. However, because the Ubuntu installer tries to use the network, I strongly urge you to keep an Ethernet cable handy in case you need to plug your system into a wired network for the install. (Early on, the Ubuntu installer will report which network interfaces it was able to activate, so if you don't see your wireless network adapter listed, it's time to use that Ethernet cable.)

Disk space and memory are probably your most important considerations. If you are planning on running the GNOME (Ubuntu) [\[Hack #15\]](#) or KDE (Kubuntu) [\[Hack #16\]](#) desktop environment, your computer will benefit from plenty of RAM and disk space. Consider 2 GB of disk space and 256 MB of RAM to be a comfortable minimum. And with the price of disk space below \$1 per GB and RAM being between \$50 (desktop) and \$100 (laptop) per GB, the more the merrier.

Preserving Your Existing Data

If you've already got Windows or Linux running on your system, and you want to keep it, you should check out "[Dual-Boot Ubuntu and Windows](#)" [[Hack #6](#)], which explains how to set up a dual-boot Ubuntu system. However, if you're just interested in archiving your existing installation so that you can pull files off of it at a later time [[Hack #7](#)], you have some choices:

New hard drive

Since the cost of storage is so low, you might want to pull and replace your current hard drive. Depending on the age of your current machine, that might give you a modest performance improvement, but it also gives you the opportunity to increase your disk space. Once you've done so, you can install your existing hard drive into an external enclosure so that you can access your old files (and use any free space for extra storage). Another option would be to simply buy an external drive and copy all your files onto it before you install Ubuntu.

Burn it to optical media

If your files can fit, burn them to CD or DVD for safekeeping. Linux will be able to read practically any format that you can put on an optical disc if you want to retrieve these files later.

Shrink your existing partition

If you want your data at your fingertips, then you should spend some time deleting everything you can live without from your old operating system: applications, data files, etc. If you're running Windows, make sure you disable hibernation in the control panel's power settings, because hibernation requires a file equal to the size of your installed RAM. Also, if you're not planning to boot the old operating system on a regular basis, disable the paging file. Do everything you can to free up disk space. Then defragment your drive and shrink the partition using a tool such as Partition Magic (<http://www.symantec.com/partitionmagic/>) or the Ubuntu installer's partitioning tool. If you want your old data at your fingertips and you still want to be able to boot up the old OS, be sure to check out "[Dual-Boot Ubuntu and Windows](#)" [[Hack #6](#)].

Move the data to another machine

If you've got another computer with lots of storage, copy the files across the network to a safe spot on that computer. Even if you have a wireless network, consider using a network cable for the transfer, since it will run a lot faster that way.

Getting the Installation CD

Before you install Ubuntu, you'll need to choose which flavor you want: Ubuntu, Kubuntu (Ubuntu with KDE as the default desktop), or Edubuntu (Ubuntu designed for young people). Although you can download a different CD-ROM for each one, you can install their core components [[Hack #54](#)] later; the packages *ubuntu-desktop*, *kubuntu-desktop*, and *edubuntu-desktop* can be installed at any time.



Another flavor of Ubuntu is Xubuntu (the package *xubuntu-desktop*), available at <https://wiki.ubuntu.com/Xubuntu>, which is a variant of Ubuntu that's optimized for older computers. The desktop system is more lightweight, so it's less demanding in terms of memory, CPU, and video-card resources.

Once you've decided which flavor of Ubuntu you want, go to the Ubuntu (<http://www.ubuntu.com>), Kubuntu (<http://www.kubuntu.org>), or Edubuntu (<http://www.edubuntu.org>) home page and follow the download link. Next, you'll need to choose which mirror site to install from (pick one that's close to you geographically), and then choose Intel x86 (for most PC compatibles), AMD64 (64-bit AMD-based PC compatibles), or PowerPC (for Macs, but if you choose this one, you should be in a different hack [[Hack #8](#)]).

On the download site, you'll find links that go directly to the CD image, an ISO file that you can burn to a CD-R using your favorite CD burning utility. Mac OS X includes Disk Utility in the */Applications/Utilities* folder. Linux usually includes the command-line *cdrecord* utility,

- **[download Who Was Dracula?: Bram Stoker's Trail of Blood book](#)**
- [download online The Little Paris Bookshop: A Novel](#)
- [click Ancient Greece: An Illustrated History online](#)
- [download online 3D Printing with SketchUp pdf, azw \(kindle\), epub](#)
- [All Necessary Force \(Pike Logan, Book 2\) book](#)
- [download online Choice or Chance: Understanding Your Locus of Control and Why It Matters](#)

- <http://www.satilik-kopek.com/library/Who-Was-Dracula---Bram-Stoker-s-Trail-of-Blood.pdf>
- <http://www.gateaerospaceforum.com/?library/Nightmare.pdf>
- <http://honareavalmusic.com/?books/Ancient-Greece--An-Illustrated-History.pdf>
- <http://nautickim.es/books/3D-Printing-with-SketchUp.pdf>
- <http://www.satilik-kopek.com/library/All-Necessary-Force--Pike-Logan--Book-2-.pdf>
- <http://hasanetmekci.com/ebooks/The-Collected-Poems-of-W-B--Yeats.pdf>