The Ubuntu User Guide – Version 06.02 by A.Y. Siu

Table of Contents

About This Guide

The Purpose of this Guide

The Ubuntu User Guide is not by any means comprehensive—as such, there are a list of some important links in this document. You should visit those links for more information. However, this guide does fill a niche that I believe needs to be filled—an easily printable PDF version of a user guide for the Linux distribution called Ubuntu. I found PDFs like this quite useful when trying Mandriva and Mepis, and I believe one should be available for Ubuntu.

Version Numbers

Ubuntu is constantly under development and helpful links appear and change every month. As such, it's very important to pay attention to the version number of this document. The version number of the User Guide, similarly to how Ubuntu itself does its version numbers, is based on the year and the month. So, the first version, created in November 2005 was version 05.11. I deliberately avoided using typical software version numbering because I want people to know just how outdated the PDF they're using is. So if you're reading a PDF that's version 06.02 (February 2006), and it's May 2006, you know your User Guide is three months old and possibly out of date.

The User Guide is Unofficial

I am, just like you, a user. I don't officially represent Mark Shuttleworth, Canonical, or Ubuntu. They are not responsible for the contents of this guide and may not even know of the guide's existence. Likewise, being a user myself, I am not legally liable for the content here. **Use these instructions at your own risk**. I don't say this because I believe the instructions are misleading or incorrect (though, they may be—either because of my ignorance or because of this document's outdatedness). I just don't want people blaming me or suing me because they followed some instruction and accidentally screwed up their computers.

Once again, I am not responsible for any of the instructions in here and what they may do to your computer. **Use this guide at your own risk!**

Copyright for This Document

What is Ubuntu?

Ubuntu is an operating system based on the Linux kernel.[1] There are many operating systems based on Linux, and Ubuntu is only one of hundreds of these. I don't recommend Ubuntu to everyone, but here are some things that make Ubuntu unique among Linux distributions:

- Ubuntu is guaranteed to be **cost-free**. Other Linux distributions tend to be cost-free as well, but there exists no subscription or member fee that gives you extra in Ubuntu. There is no “enterprise edition.” Ubuntu is Ubuntu, and it's free. Not only is the operating system itself free—Canonical will actually ship Ubuntu CDs (worldwide) and pay for the shipping as well.[2] Granted, these freely shipped CDs can take around two months to arrive, but you can't really complain if you're not paying for postage.

- Ubuntu and all the software it includes is free in two ways—they're all cost-free and **non-proprietary**. You won't be bogged down with all sorts of licensing issues, and if you know something about programming, you can take a look at the source code of the applications and modify them as you see fit. On the one hand, this is an advantage, as you won't have limits on how many computers you put Ubuntu on. On the other hand, you may rely (much more than you may be aware of) heavily on proprietary software and wonder why you can't immediately do something you used to be able to do.

- Ubuntu has **regular release cycles**. Other popular Linux distributions (or versions) generally update as frequently as possible, but Ubuntu is dedicated to having new releases every six months—in April and October of each year.

- Ubuntu tries to make the installation of the operating system as **simple** as possible—one user (at least initially—you can add more users later), one password, one application per task, one CD for the entire operating system.

- The **forums** (http://www.ubuntuforums.org) have quick response times, helpful users and staff, and a lot of good customization tips and tricks. It's a friendly, supportive environment, in accordance with Ubuntu's philosophy of being “Linux for Human Beings”—humanity to others. The forums are run entirely by volunteers and fellow users. They are not paid employees of Canonical.

Some people view these characteristics as advantages. Others view them as disadvantages.

---

1 I've written an absolute beginner's guide to Linux in general. If this guide seems over your head, and you just want a basic rundown of Linux, please visit my website at http://www.psychocats.net/essays/linuxguide.php

2 https://shipit.ubuntu.com (the Ubuntu download page now says “For a limited time we will send CDs to you at no cost, too,” so this deal with the free shipping may not last forever.

Even though Ubuntu comes with a lot of productivity software—an office suite, a music player, a Photoshop-like graphics editor, an instant messaging program, an email client, an internet browser, etc.—many people like their *proprietary* software to “just work” out of the box. That won't happen in Ubuntu. If you want to play commercial DVDs, have MP3 support, or view Flash movies in your internet browser, you'll have to enable proprietary software that Ubuntu does not include by default. Several other Linux distributions do have proprietary formats built into them, though: Mepis, Blag, or Linspire.

Regular release cycles generally mean improved software. Regular improvements breed instability, though. Even if you upgrade from Windows 2000 to Windows XP or Mac OS X Panther to Mac OS X Tiger, you may notice some glitches here and there. Likewise, if you upgrade your Ubuntu operating system *every six months*, you may find the upgrade process a bit wearying, especially if you upgrade on or before the official release day. For those who like a Linux distribution with a lot of stability and infrequent upgrades, Debian may be a better choice than Ubuntu.

I think a lot of Ubuntu advocates will agree with me when I say you should use the operating system that best suits your needs. Ubuntu may be that, but there are other Linux distributions out there, and you may be better off with a non-Linux operating system (a Windows or Mac operating system, for example).

Different Versions of Ubuntu

Ubuntu has CDs[3] available for x86 (this will apply to most Windows PCs), AMD, and PowerPC (most Macintosh PCs before January 2006) architectures. As of this writing, there is no Ubuntu distribution suitable for the newly-released Apple Intel computers.

You may notice there are different versions of Ubuntu—one called Ubuntu, one called Kubuntu, and one called Edubuntu.

Ubuntu is the original version of Ubuntu with a Gnome desktop environment. Kubuntu is the same as Ubuntu but with a KDE desktop environment.

If you want to read more about KDE and Gnome—what they are, the differences between them—Google them. If you don't know where to begin, I recommend you just start with Ubuntu, as it has better documentation than the KDE version, *and* you can always easily install KDE on Ubuntu with a single command or a few clicks.[4]

Kubuntu and Ubuntu are the same operating system. They simply have different desktop environments.

I don't know much about Edubuntu, but it's supposed to be tailored for schools and educators.

---

3 There are DVD images available as well. The process is very similar, and the DVDs contain more than the CDs (obviously), but I'm not sure what the difference is in terms of content.

4 The command, if you have a terminal open, is *sudo apt-get install kubuntu-desktop*. If you have Kubuntu and want to install Gnome, the command is *sudo apt-get install ubuntu-desktop*.

How to Get Ubuntu

Order Free CDs
You can go to https://shipit.ubuntu.com/ and order your free CDs from Canonical. Apparently, they won't be offering these free CDs forever, but as yet there's no “end date” on the service.

Download and Burn
You can also, if you have broadband internet and a CD burner, download various CD images for Ubuntu at http://www.ubuntulinux.org/download/. Both Nero and Easy CD Creator have options to burn ISOs as disk images. Mac OS X's Disk Utility can also do this. If you want a free CD burning program for Windows, you can also try CDBurnerXP: http://www.cdburnerxp.se/help/english/burniso

If you download the ISO (disk image) of Ubuntu, you may want to verify that the image came through intact and didn't get corrupted during download. For more information on how to verify disk images' integrity, visit this link: http://www.linuxiso.org/viewdoc.php/verifyiso.html. You can also Google for the keywords *md5sum* and *checksum*.

Do not burn the ISO as data, and do not extract the ISO, even if the file appears as an extractable type file (WinRar, for example, may identify the ISO as a rar file).

When you burn the CD, try to burn it at as slow a speed as possible (4x or 8x, for example). This, along with verification of the CD image integrity will help to ensure that the CD you eventually use will work properly. You may burn at a faster speed if you wish, but don't complain if the CD freezes in the middle of the installation.

Booting the CD
In order to use the live CD or installer CD, you may need to set your computer to boot from the CD-ROM or DVD-ROM drive.

For **Windows users**, this involves going into the BIOS settings. When your computer boots up, there's usually a key you can press to get into the BIOS settings—this key may be Delete, F1, F2, Escape, or some other F# key. Pay attention to what allows you to “Enter Setup” or just experiment.

In the BIOS setup, there should be an option to change the boot order. Ask the computer to look to boot from CD before it tries to boot from the hard drive.

For **Mac users**, you can usually force the computer to boot from the CD drive by holding down the *c* key during boot-up.

How to Try Ubuntu

There are two ways to “try” Ubuntu without taking the full dive into wiping out your entire hard drive and installing it as your only operating system.

Live CD

A live CD allows you to try out an entire operating system without affecting your hard drive at all. It boots from your hard drive and runs completely off the CD itself and your computer's memory (RAM). This means two things:

1. You can safely get a Ubuntu experience without affecting your current operating system or any of your files. This “test” will give you an idea of what the interface is like and also how well Ubuntu will recognize your hardware (sound, video, internet, etc.). If Ubuntu doesn't recognize more than two of your hardware pieces out of the box, you may want to look for another Linux distribution. Sure, you may be able to fix some of those problems, especially with help from the Ubuntu Forums (http://www.ubuntuforums.org), but it'll be a hard fight. One or two things that are off may be easily fixable.

2. The “test drive” of Ubuntu will run a lot more slowly than a regular installation of it would run. Hard drive data can be read faster than CD data, and the memory on your computer is being used for a lot of temporary files Ubuntu loads in for the live session. During a regular installation, the memory is all completely available for better performance.

If you like the live CD, chances are a permanent installation may be worth your time.

Dual Boot

Some people just dive right in and install Ubuntu straight to their computers. Others want to have their Mac or Windows OS available as another boot option. I don't have any experience with the Mac dual boot (and, to be perfectly honest, the documentation for PowerPC Ubuntu is scant), but it's possible.

The best dual boot guide I've seen for Ubuntu and Windows is this one: http://users.bigpond.net.au/hermanzone/

It has screenshots and explanations that walk you through a dual boot step by step, including the shrinking of the Windows partition and the creation of the necessary Linux partition(s).

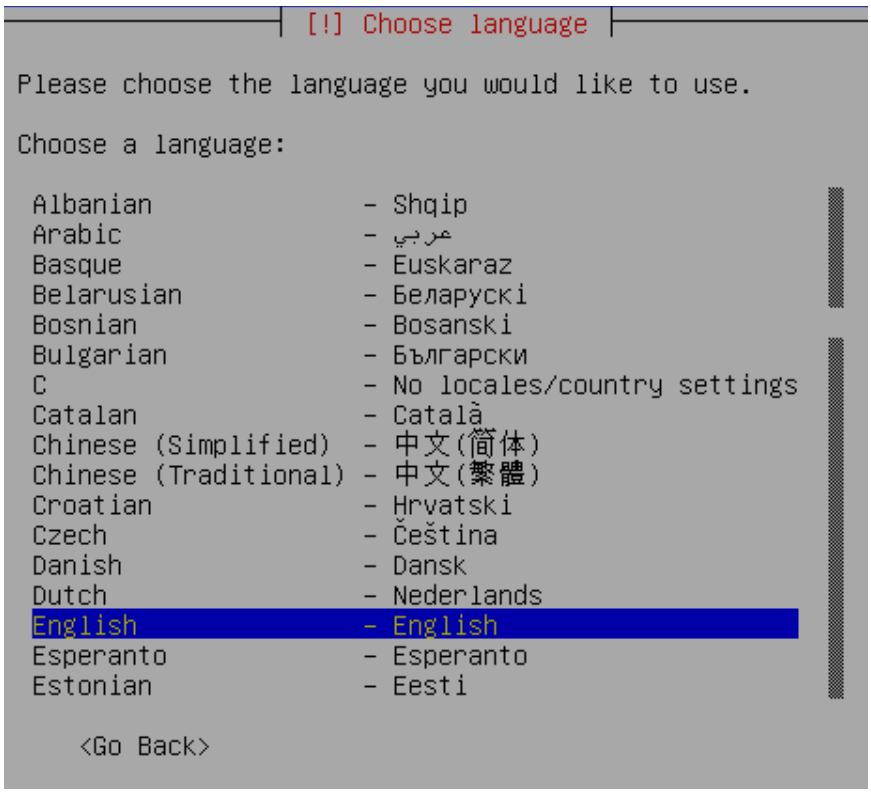
Installing Ubuntu

I won't walk you through the entire installation process, but I'll highlight a few things.

At this first boot screen, you have several options. If you press F1, you can get some advanced installation options (this can be handy, for example, if you're experiencing problems during the installation that do not have to do with a corrupt disk—e.g., the screen resolution is off).

For most people, pressing *Enter* here is the best choice for the default installation. **Do not do a *server* install or *expert* install unless you absolutely know what you're doing.** Beginners and even most intermediate users should just press *Enter* and do a regular installation.

You may notice a bunch of gibberish scroll by on the screen. Do not worry if this happens. **This is normal**.



You'll then be asked a series of questions that are pretty simple to answer: What's your language? What's your time zone? What's your username? What's your password?

<Tab> moves between items; <Space> selects; <Enter> activates buttons

Please pay attention to the keyboard navigation instructions. You may be put off at first by the text-only graphical installer, but it is still a graphical installer—it just isn't pretty, and you have to use the keyboard instead of the mouse.

To clarify a bit:

<Tab> moves between item groups, but within groups, you can move up and down between items using the up and down arrows. For example, in the above screenshot for language selection, you press up to go to *Dutch*, press down to go to *Esperanto*, and press tab to go *<Go Back>*.

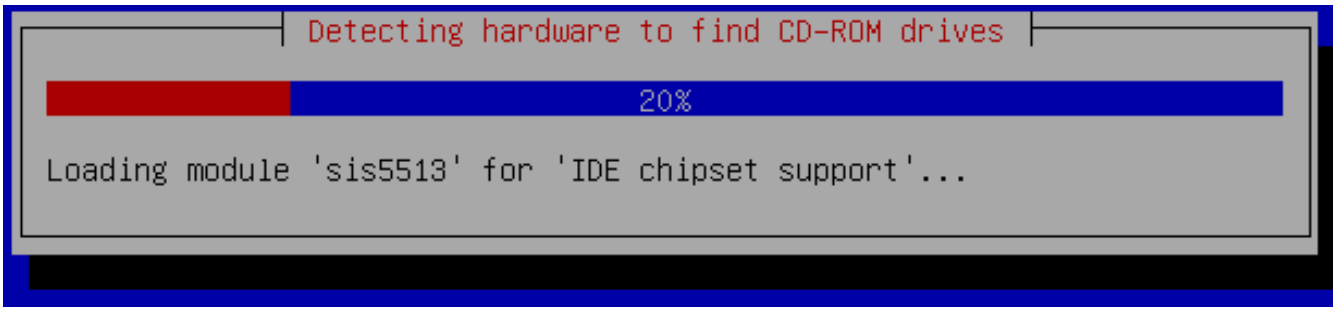
<Space> "selecting" means it decides whether something is checked or unchecked (*ticked* for you British out there)—think of <Space> as the toggle key.

<Enter> basically means "I've made my decision here—let's move on to the next screen."

Depending on how slow your computer is, you may occasionally get what appears to be a “blue screen of death” (especially to former Windows users). Do not panic if this appears. Everything's working just fine.

Of course, if the screen stays that way for more than a few minutes, then you should be worried.

People have already made the developers aware of this occasional lack of feedback in the installer. Hopefully that will be improved upon for Ubuntu's April 2006 release (nicknamed *Dapper Drake*).



Ordinarily, though, you will get feedback with a progress bar indicating how much of the current activity is finished.

As I said before, most of the installation's questions are pretty easy to answer. There are a few questions that may cause some confusion, though.

UTC

You'll get a question about time zones, Greenwich Mean Time, and UTC.

If you're planning to dual boot, do not use UTC.

If you're not planning to dual boot, go ahead and use it.

Only after you decide to use UTC or not do you actually decide (on the next screen) which time zone to select.

The time zone you can easily change later. Whether you use UTC or not *can* be changed later, but the process is a bit more involved.

Partitioning

If you are doing a dual boot with Windows, there's too much involved for me to go into here. You should follow this tutorial: http://users.bigpond.net.au/hermanzone/

If you want to erase the entire hard drive and make Ubuntu the only operating

system on your computer, choose the *Erase Entire Hard Drive* option.

Grub

The boot loader determines which operating system the computer boots to. Even if you're booting only one operating system, you still need a boot loader. I would highly recommend installing Grub to the MBR (Master Boot Record). Otherwise, you'll need to do some tricky maneuvering to get your computer booted properly to Ubuntu.

Username and Password

Initially, you'll be asked for only one user with one password. If you're used to other Linux distributions, you may be shocked to not be asked for a *root* password. That's because Ubuntu does not enable *root* by default. It makes the first user (the one created during the installation process) have *sudo* privileges. That means the first user can temporarily make herself administrator of the computer by typing *sudo* in front of commands.

(Mac users should feel right at home with sudo, as it is the same security model Mac OS X uses.)

Subsequent users will, by default, not have *sudo* privileges unless you deliberately go out of your way to give them such access. You can also enable a separate *root* account if you wish.

For more information on *root* and *sudo* (including their respective advantages and disadvantages as security models), visit https://wiki.ubuntu.com/RootSudo

For users unfamiliar with the whole *root* phenomenon, the bottom line is that you create one user during the installation. This user will need to enter her password to make system-wide changes and edit configuration files. Otherwise, for security purposes, she will have access to only her own personal files, preferences, and settings.

The rest of the installation process

After you answer all the questions, you'll have to reboot your computer to install the rest of the packages (you'll be asked to eject the installer disk, and you won't need the installer disk for the rest of the installation).

If you did not install Grub to the MBR, you will have to find some way (via a boot floppy or some other boot loader) to boot to Ubuntu to finish the package installation process.

The package installation can take a while. On my computer, it took about twenty minutes.

Once that finishes, you should get to a login screen. This is where you will enter your newly created username and password.

Tweaking Ubuntu After Installation

Before you can use Ubuntu, you may have to tweak a few things. I won't go over everything here, but I will point you in a few directions for things I can't cover here.

First of all, as I said before, to a lot of people, proprietary software is important, even when using a free (in licensing and cost) operating system. There are ways to install all of these things manually, but there are also some third-party projects that enable you to do have some of the typically-used software semi-automatically installed. Two of the most popular of these projects are **Easy Ubuntu** and **Automatix**. You can get more information on these projects at the Ubuntu Forums (http://www.ubuntuforums.org).

There is also a wonderful one-page guide on Ubuntu tweaks that's for Ubuntu version 5.04 (nicknamed *Hoary Hedgehog*), but it hasn't been updated for Ubuntu version 5.10 (nicknamed *Breezy Badger*) as of this writing. Nevertheless, apart from some of the codecs information and the repository information, most of the information in that guide should work for version 5.10 as well. The guide is located at http://www.ubuntuguide.org and is a one-page (very long) HTML document with internal links to pages.

A new one-page guide for version 5.10 is also available: http://makuchaku.info/amnesty/

I won't go over how to install all of the proprietary codecs and popular software here, but I will cover some frequently asked questions (post-install) and some general how-to-use Ubuntu interface steps.

The Filesystem

First of all, you should become at least a little bit familiar with how files and folders are organized in Linux (and, really, any Unix-like system). I don't know everything about the folder hierarchy, but I'll highlight a few important things for you.

/boot has the boot information, including the Grub configuration file.

/etc has a lot of settings for things, including which software repositories you use and what other partitions or drives you have “mounted” (available for browsing).

/home this is the only directory you will have access to without using your password to gain temporary administrator privileges. All of your files (unless you're storing them on a separate partition—for example, you may put them on a FAT32 partition to share with Windows) will live here, along with your settings and preferences, inside a folder called */home/username*. So if your username is *cooljenny*, your files and settings will be in the */home/cooljenny* folder.

At first, it may not appear to you that your settings aren't in there at all. They're actually in hidden folders. In Ubuntu, if you press *Control-H*, you'll see them all. In Kubuntu, if you press *Alt-V* and then *H*, you'll see them all.
For example, the settings folder for Firefox, for our hypothetical user, would be in */home/cooljenny/.mozilla*
The settings folder for icons would live in */home/cooljenny/.icons*

Sometimes people will refer to */home/username* as *~/*
So, */home/cooljenny/.mozilla* could also be *~/.mozilla*

By the way, many people will recommend you create a separate */home* partition during installation. You would create this if you anticipate reinstalling the operating system often (either because you experiment a lot and screw up the installation or because you want to keep up with the six-month releases and prefer a clean installation to an upgrade) but don't want to have to redo all your settings and preferences every time.

/media and **/mnt** are where your media (CDs, DVDs, USB drives, etc.) and mounted partitions would go.

/root is the */home* folder for *root* or *sudo* and has its own settings. For example, if you choose to have items open with a single-click for your regular user but do not specify so using *sudo*, you will have to double-click files to open them when browsing as *root*.

/usr is where a lot of stuff is stored that you'll be using. That doesn't mean you'll actually navigate to this folder, but if you install software, it'll usually install to this folder (/usr/bin is a typical place for executable binaries to go), and a lot of icons that are system-wide icons will be in /usr/share/icons or /usr/share/pixmaps.

Benefits of the Command-Line/Terminal

A lot of new users are afraid of using the command-line (I was one of them when I first started using Linux). For some reason, the idea of typing in commands instead of pointing and clicking on buttons with a mouse seems archaic and scary to people who haven't used the command-line since MS-DOS in the 1980s.

I'm not advocating we use the command-line for everything, but there are a couple of benefits to the command-line:

- Sometimes it's just faster. If I wanted to install Thunderbird graphically, it'd take me at least five clicks (and a lot of waiting) to do. If I wanted to install it via the command-line, it would take only one command: *sudo apt-get install mozilla-thunderbird*. Of course, you would have to know what the command is—it isn't easily discoverable, but it is useful.

- More importantly, it makes it easier for others to help you. If you're on the

> Ubuntu Forums (or any kind of online help venue), it's a lot easier for someone to say, “Here, type this command in” than to describe in detail what you should click on, and then what to click on in the subsequent dialogue. In fact, you don't even need to type at all. If someone gives you a command to type in, just copy it and paste it into the terminal.

To find the terminal:
in Hoary (Ubuntu 5.04), go to Applications > System Tools > Terminal
in Breezy (Ubuntu 5.10), go to Applications > Accessories > Terminal.
in Kubuntu, click on the "KMenu" and go to System > Konsole (Terminal Program)

If you're thinking, *This is ridiculous. I shouldn't have to use the command-line at all*, then Ubuntu may not be the Linux distribution you want to install. You have several options:
1. You can have someone else install and configure it for you
2. You can use another distribution (Mepis or Linspire)
3. You can not use Linux at all.

Some helpful commands to know:

startx – If you happen to end up at a command-prompt without any graphics, you can log in and try typing this command to get back to the graphical (or “x”) system.

sudo dpkg-reconfigure xserver-xorg – If you're having screen resolution problems that can't be solved via point-and-click, you can try this command. It'll walk you through setting up your video settings again.

xkill – Kills a misbehaving application. Once this command is run, the mouse cursor will become a skull and crossbones. Any window you click on after that will close immediately. Kubuntu has a keyboard shortcut built in for this already: *Control-Alt-Escape*.

killall gnome-panel – Refreshes the Gnome panel. Particularly helpful if you've added new menu items that don't appear or have changed icons. The Kubuntu equivalent is **killall kicker**.

top – A quick way to see what applications are using the most resources at any given time.

man – Gives you the **man**ual for an application. For example, if you want to learn how to use Wine, you would type *man wine*.

rm – Deletes a file. For example, if you had a file called putmeinthetrash, you could type in *rm putme** to delete it.

ls – Lists the files and folders in a directory.

cd – Changes directories (that's right—just as it does in MS-DOS).

Editing Files as Root/Sudo

Most of the time, once you're all set up, you'll rarely have to edit files as *root*. Right after you install Ubuntu, though, you'll find you probably have to edit a bunch of configuration files, and you'll need administrative (or *root*) privileges to do so.

There are two good ways to do this:

1. Open up a terminal and type ***sudo*** before each command you use to edit a file. For example, if you want to edit the /boot/grub/menu.lst file, you can type *sudo gedit /boot/grub/menu.lst* (which uses the Gnome text editor Gedit), *sudo kwrite /boot/grub/menu.lst* (which uses the KDE text editor Kwrite), or *sudo nano /boot/grub/menu.lst* (which uses the text editor Nano—available in both Ubuntu and Kubuntu).

2. If you prefer to do it graphically, you can run the command ***gksudo nautilus*** (for Gnome) or ***kdesu konqueror*** (for KDE) and a file browser window will appear that has *root* (or administrator) privileges. When you close that window, you'll be a regular user again (and, in fact, you'll be a regular user for just about everything outside of that window, even before you close it). You can run the command from the *Run* dialogue (Alt-F2), or you can create a launcher in the panel (right-click on the panel and add a custom application launcher to it.

Either way (#1 or #2), you'll be prompted for your user password, and that will last for 15 minutes for you to act as the administrator of the computer and then return to being a regular user again.

If you're using the terminal, though, you'll have preface every root command with the word *sudo*—you'll be prompted for your password only the first time, though.

A couple more things you should know about configuration files:

- Always back them up before editing them. When I give you commands for editing them, I will always include the command for backing them up as well.

- A # sign in front of the line means the line has been “commented out”—it essentially means the line doesn't exist any more, as far as Ubuntu's concerned. It exists only to be read by you as having formerly existed.

Getting to Know Grub

Grub controls the boot menu when you first start up the computer. It's particularly important to get to know Grub if you're dual-booting.

To edit Grub, you should type the following commands (for future examples, I'm going to use *nano* as the editing command—seeing as how it works in both Kubuntu and Ubuntu. You're welcome to use *kwrite* or *gedit* if you wish, though).

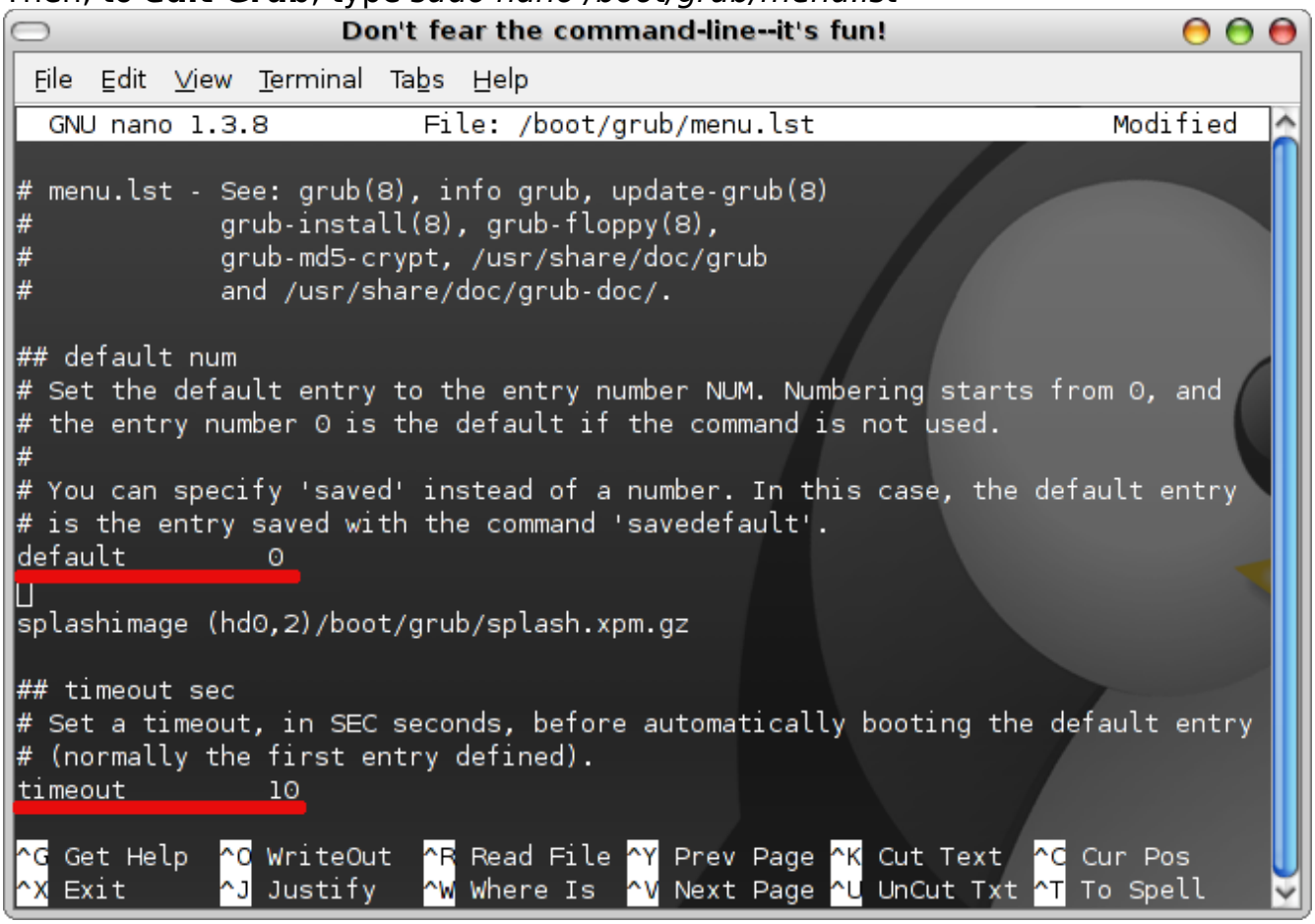
The two things people most frequently want to do with Grub are

1. changing the default boot option (Grub will default to booting from Ubuntu, but some people want Grub to default to Windows)
2. changing the time before a boot option is selected automatically.

Before doing anything, let's **back Grub up** first. Open up a terminal and type this command:

sudo cp /boot/grub/menu.lst /boot/grub/menu.lst_backup

Then, to **edit Grub**, type *sudo nano /boot/grub/menu.lst*



There are really only two things you'll need to change here. To **change the default boot option**, change *default 0* to be *default 1* (or 2 or 3...). Keep in mind that the numbering starts at 0. That means the first entry is 0. The second entry is 1. The third entry is 2. The fourth entry is 3, and so forth.

Do you see another line that says *timeout 10*? That means you have ten seconds to decide which boot option you want to choose before Grub automatically selects the default option. If you want the time to be **longer**, you can make it *timeout 15* (fifteen seconds); if you want it **shorter**, you can make it *timeout 5* (five seconds). Do not change this to *timeout 0*, especially if your default boot option is Windows (not Ubuntu).

I keep talking about “entries” and “boot options.” If you scroll down a bit in the menu.lst file, you'll see something that looks similar to this:

```
title           Ubuntu
root            (hd0,7)
kernel          /boot/vmlinuz-2.6.12-10-386 root=/dev/hda8 ro quiet splash
initrd          /boot/initrd.img-2.6.12-10-386
savedefault
boot
```

That's an entire **entry**. I'd recommend, unless you know what you're doing, that you leave entries alone. The only part that's safe to modify is what's after the word *title*. That's just what you see as the name of the boot option. That won't affect how the system sees the boot option.

If you do know a little bit of what you're doing, there are a couple of added parameters you may be interested in, particularly in the *kernel* line. Right now the parameters are *ro, quiet, splash*.

Splash means that a pretty Ubuntu logo shows up during boot time. Some people prefer the scrolling white text on a black screen. If you prefer that, take *splash* out.

If you notice that the **screen resolution for your login screen** is too high or low, you may want to tack on a *vga=791* parameter to make the screen resolution 1024 x 768 (a fairly typical/safe resolution). So the line above would be changed to read

```
kernel      /boot/vmlinuz-2.6.12-10-386 root=/dev/hda8 ro quiet splash vga=791
```

If there are entries you want to **comment out** (i.e., you don't want them to appear as options during boot-up—maybe it's too cluttered), then you can put # signs in front of the entry:

```
#title          Ubuntu
#root           (hd0,7)
#kernel         /boot/vmlinuz-2.6.12-10-386 root=/dev/hda8 ro quiet splash
#initrd         /boot/initrd.img-2.6.12-10-386
#savedefault
#boot
```

Finally, save your work in Nano (Control-X), confirm the save (y), and exit (Enter).

Mounting Windows Partitions

If you want the quick and dirty version of this, visit http://www.ubuntuguide.org/#windows

I'll be going into quite a bit more detail here.

For some strange reason, Knoppix (a version of Linux that is primarily a live CD and recovery CD) and one of its derivatives (Mepis) allow you to automount partitions simply by clicking on them. This feature has yet to be implemented in Ubuntu. Ubuntu will automount external media (external hard drives, MP3 players, etc.), though.

So, if you have an NTFS and/or FAT32 partition you want mounted, you'll have to follow these directions. Keep in mind that, as of this writing, NTFS is reliably read-only in Linux. If you want to read from and write to a Windows partition, make sure it's formatted as FAT32.

First, you have to find out what your Windows partitions are called. To do so, type this command in a terminal: *sudo fdisk -l*

This is what that command lists for my computer:

Disk /dev/hda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders
*Units = cylinders of 16065 * 512 = 8225280 bytes*

Device Boot	*Start*	*End*	*Blocks*	*Id*	*System*
/dev/hda1 *	***1***	***1911***	***15350076***	***7***	***HPFS/NTFS***
/dev/hda2	*1912*	*18494*	*133202947+*	*f*	*W95 Ext'd (LBA)*
/dev/hda3	*18495*	*19457*	*7735297+*	*83*	*Linux*
/dev/hda5	***1912***	***14763***	***103233658+***	***b***	***W95 FAT32***
*/dev/hda6 **	*14764*	*16434*	*13422276*	*83*	*Linux*
/dev/hda7	*18363*	*18494*	*1060258+*	*82*	*Linux swap*
/dev/hda8	*16435*	*18362*	*15486628+*	*83*	*Linux*

Partition table entries are not in disk order

I've put the relevant entries in bold. These are the two important things to note: NTFS is /dev/hda1 and FAT32 is /dev/hda5.

Now, we need to make a mount point for each one. A mount point is how it will appear in your system—it'll be almost as if it is another folder on your system, even though it's not really a folder... it's a partition. One way to think of mounting a partition is to think of it as an alias (to use Mac terminology) or a shortcut (to use Windows terminology).

We'll create the mount points with the following commands:

sudo mkdir /xp
sudo mkdir /documents

/xp is the NTFS partition where my Windows XP partition lives.
/documents is the FAT32 partition where I keep documents (music, files, pictures) to share between Windows and Ubuntu.

Next, we need to specify how these get mounted. We want /xp to be read-only and /documents to be read-write. To do this, we edit the /etc/fstab file:

sudo cp /etc/fstab /etc/fstab_backup
sudo nano /etc/fstab

This is what pops up with **the to-be-added lines in bold**:

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/hda8       /               ext3    defaults,errors=remount-ro 0       1
/dev/hda5       /documents      vfat    umask=000       0       0
/dev/hda1       /xp             ntfs    nls=utf8,umask=0222     0       0
/dev/hda7       none            swap    sw              0       0
/dev/hdc        /media/cdrom0   udf,iso9660 user,noauto     0       0
/dev/hdd        /media/cdrom1   udf,iso9660 user,noauto     0       0
```

Once you're finished, save (Control-X), confirm (y), and exit (Enter).

Finally, we need to have the changes actually take effect. The most reliable way to do this is to reboot the computer. If you're really impatient, though, you can type this command in: *sudo mount -a*.

Now if you navigate to /xp, you should see your C:\ partition. If you navigate to the /documents folder, you'll see the documents you share with Windows.[5]

For a more detailed explanation of what /etc/fstab is and what each of its parameters means, visit http://www.tuxfiles.org/linuxhelp/fstab.html

5 A similar process works for Macintosh partitions as well, using *hfs* or *hfsplus* instead of *vfat* or *ntfs*.

Installing Software in Ubuntu

Here are the main ways to install software in order of ease (easiest to hardest):

apt-get/Synaptic/Adept

All Debian-based distros having something called *apt-get* which allows you to draw from a set of online repositories (stored in the /etc/apt/sources.list file) that house packages (i.e., programs/software).[6] The *apt-get* command does several things at once—it downloads the appropriate files, downloads all their dependencies, and installs all of them. A single command installs the software. You don't have to download a separate installer file or unzip or go through a wizard or reboot. For example, if I wanted to install Thunderbird, I'd type these commands in a terminal:

sudo apt-get update
sudo apt-get install mozilla-thunderbird

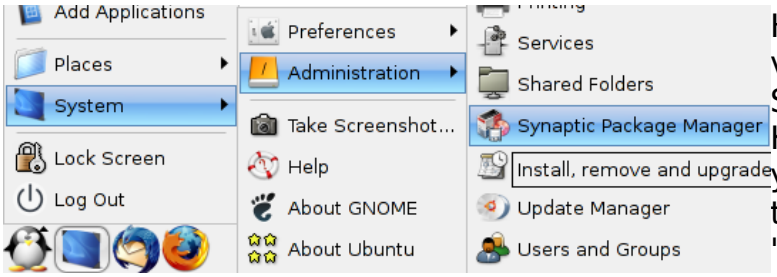
The first command looks both at what I have installed and what's available in the repositories. The second command downloads the packages needed for Thunderbird and installs them.

Another great thing about apt-get is the ability to install several different packages at once. For example, if I wanted to install not only Thunderbird but Firefox, GIMP, Inkscape, Juk, and Wine, I could type in these commands:

sudo apt-get update
sudo apt-get install mozilla-thunderbird firefox gimp inkscape juk wine

and all of those packages would download and install themselves.

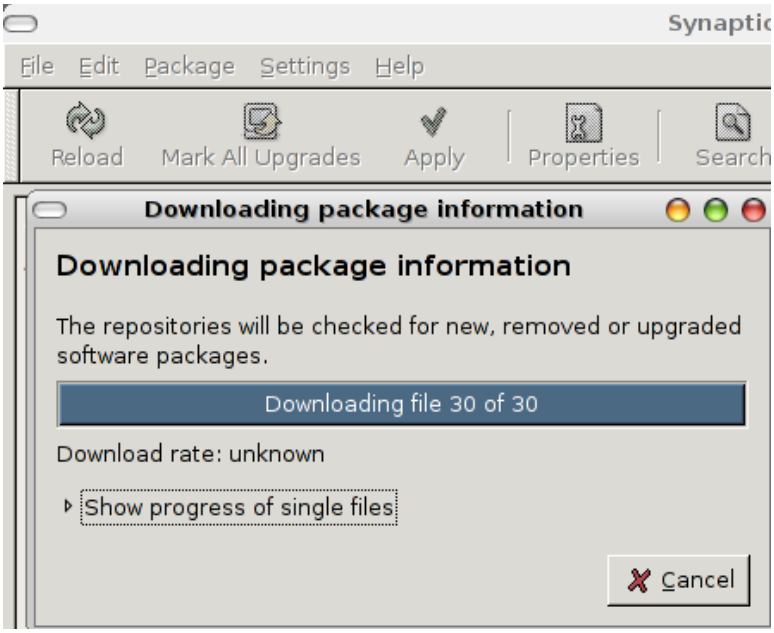
On the Ubuntu forums and on the Wiki and other guides, you'll often see instructions to *sudo apt-get install* some package or other. People will give you commands because it's easier than describing what to click in a graphical user interface, and you can just copy and paste the command.



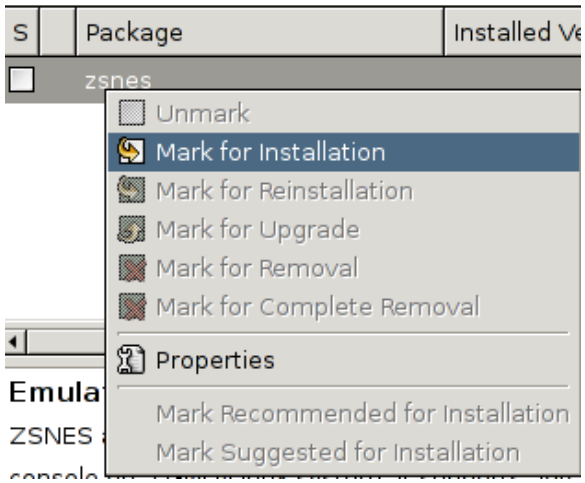
However, there is a graphical version of apt-get. For Ubuntu, it's Synaptic Package Manager. For Kubuntu, it's Adept. In both, you're essentially doing the same thing. There's also a nice "browsing" environment in which you can search for packages by

6 You can enable extra Ubuntu repositories by following the directions at http://www.psychocats.net/linux/sources.php–extra repositories means more software available to install. I also include a copy of these instructions in this manual.

name and/or description. You can browse by categories of software or look at what's installed versus what's not installed. It's a lot like an e-commerce model of "shopping" for software, except you don't have to pay when you "check out."



The equivalent of *sudo apt-get update* is clicking the *Reload button*. The *sudo apt-get install* command, however, is broken into different steps.



Instead of listing a bunch of applications you want to install, you mark each one for installation (or removal), and then click *Apply Changes* or *Commit Changes* and then everything's downloaded and installed (or uninstalled).

Manual installation of a .deb

While the Ubuntu repositories are quite extensive (especially if you add extra repositories), they don't cover everything. Sometimes (for the Opera web browser, for example), you have to install a separate file. If you must do so, then try to get

ahold of a .deb file. .deb is the native file format for Debian-based distributions like Ubuntu. In fact, if you actually visit the online repositories, you'll notice that the files there are mostly stored as .deb files.

The only difference between manually installing a .deb file and using apt-get to install a .deb file is that apt-get will resolve dependencies for you (if one package needs another to be installed, apt-get will install that "pre-requisite" package). If you manually install a .deb file, you will also have to manually install its dependencies. Don't worry, though—if you try to manually install a .deb file and it has dependencies, you'll soon find out what those dependencies are.

Here's what you should do.

Download the .deb file to your desktop. For this example, let's use Opera. Now, the Opera file that's currently available for Ubuntu has a quite ugly name: *opera_8.50-20050916.6-shared-qt_en_etch_i386.deb*. There are several ways to deal with this ugly name, seeing as how you have to type in the exact name of the .deb file in order to install it. You can rename it to something simpler (say, *opera.deb*), you can copy and paste the name, or you can just suck it up and retype it exactly as is. Let's assume, though, that you're going to do it the ugly way. You'd open up a terminal and type these commands:

```
cd Desktop
sudo dpkg -i opera_8.50-20050916.6-shared-qt_en_etch_i386.deb
```

That's it... well, as long as there are no dependencies.

Manual installation of a .rpm

Occasionally, for a program, you're just not able to find a .deb. There may seem to be, however, a plethora of .rpm files for the program. If you must use an .rpm (not native to Debian-based distros), then use an .rpm. It's a very similar procedure to the .deb one described above, just using a different command (one that converts the "alien" format of .rpm):

One-time deal, just to get alien:

```
sudo apt-get update
sudo apt-get install alien
```

Now you can actually *use* alien:

```
cd Desktop
sudo alien -i opera-8.50-20050916.5-shared-qt.i386-en.rpm
```

Again, no dependencies will be resolved.

Installing from source

Some people prefer to install from source, but I listed it last because it's what usually scares people off from Linux and makes them think "Why is it so difficult to install software in Linux?" However, it's still an option, and unfortunately it's sometimes the only option, depending on how obscure the software is you're trying to install.

The first thing you'll have to do in Ubuntu is install a meta-package called *build-essential* (a meta-package isn't a real package—it's a pointer that tells Synaptic/Adept/apt-get to install a bunch of other real packages):

```
sudo apt-get update
sudo apt-get install build-essential
```

I can't think of a program off the top of my head that I ever needed to install from source, so I'm just going to make something up—let's call it obscure-1.0. Most likely, it'll come as zipped file called *obscure-1.0.tar.gz*. Download this to your desktop. Then type this in a terminal:

```
tar -xvzf obscure-1.0.tar.gz
cd obscure-1.0
./configure
make
sudo make install
```

Installing from source, like the previous two methods, also does not resolve dependencies—you'll have to install those separately.

Note: in both Ubuntu and Kubuntu, you can "un-tar" (or unzip) a .tar.gz graphically. I've never had to use the tar terminal command. To un-tar a .tar.gz graphically, just open it (double-click usually does this), then click on *Extract*.

Those are the major ways to install software in Ubuntu. You may find on the Ubuntu forums (http://www.ubuntuforums.org), however, some great self-installer scripts and other helper programs.

Enabling Extra Repositories[7]
Open up a terminal and type in

If you're using **Kubuntu** (KDE):

sudo cp /etc/apt/sources.list /etc/apt/sources.list_backup
sudo kwrite /etc/apt/sources.list

If you're using **Ubuntu** (Gnome):

sudo cp /etc/apt/sources.list /etc/apt/sources.list_backup
sudo gedit /etc/apt/sources.list

Then replace your existing sources.list with the following if you're using **Hoary (5.04)**:

#deb cdrom:[Ubuntu 5.04 _Hoary Hedgehog_ - Release i386 (20050407)]/ hoary main restricted

Uncomment the following two lines to fetch updated software from the network
deb http://archive.ubuntu.com/ubuntu hoary main restricted
deb-src http://archive.ubuntu.com/ubuntu hoary main restricted

Uncomment the following two lines to fetch major bug fix updates produced
after the final release of the distribution.
deb http://archive.ubuntu.com/ubuntu hoary-updates main restricted
deb-src http://archive.ubuntu.com/ubuntu hoary-updates main restricted

Uncomment the following two lines to add software from the 'universe'
repository.
N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
team, and may not be under a free licence. Please satisfy yourself as to
your rights to use the software. Also, please note that software in
universe WILL NOT receive any review or updates from the Ubuntu security
team.
deb http://archive.ubuntu.com/ubuntu hoary universe
deb-src http://archive.ubuntu.com/ubuntu hoary universe
deb http://security.ubuntu.com/ubuntu hoary-security main restricted
deb-src http://security.ubuntu.com/ubuntu hoary-security main restricted

deb http://security.ubuntu.com/ubuntu hoary-security universe
deb-src http://security.ubuntu.com/ubuntu hoary-security universe

---

7 The most up-to-date version of this is http://www.psychocats.net/linux/sources.php. I highly recommend you use that instead of this PDF guide, as it is easier to copy and paste these updated sources than it is to type them in.

deb http://archive.ubuntu.com/ubuntu hoary multiverse
deb-src http://archive.ubuntu.com/ubuntu hoary multiverse

deb http://archive.ubuntu.com/ubuntu hoary-backports main restricted universe multiverse

Or replace your current sources.list with the following if you're using **Breezy (5.10)**:

#deb cdrom:[Ubuntu 5.10 _Breezy Badger_ - Release i386 (20051012)]/ breezy main restricted

Uncomment the following two lines to fetch updated software from the network
deb http://archive.ubuntu.com/ubuntu breezy main restricted
deb-src http://archive.ubuntu.com/ubuntu breezy main restricted

Uncomment the following two lines to fetch major bug fix updates produced
after the final release of the distribution.
deb http://archive.ubuntu.com/ubuntu breezy-updates main restricted
deb-src http://archive.ubuntu.com/ubuntu breezy-updates main restricted

Uncomment the following two lines to add software from the 'universe'
repository.
N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
team, and may not be under a free licence. Please satisfy yourself as to
your rights to use the software. Also, please note that software in
universe WILL NOT receive any review or updates from the Ubuntu security
team.
deb http://archive.ubuntu.com/ubuntu breezy universe
deb-src http://archive.ubuntu.com/ubuntu breezy universe

deb http://security.ubuntu.com/ubuntu breezy-security main restricted
deb-src http://security.ubuntu.com/ubuntu breezy-security main restricted

deb http://security.ubuntu.com/ubuntu breezy-security universe
deb-src http://security.ubuntu.com/ubuntu breezy-security universe

deb http://archive.ubuntu.com/ubuntu breezy multiverse
deb-src http://archive.ubuntu.com/ubuntu breezy multiverse

deb http://archive.ubuntu.com/ubuntu breezy-backports main restricted universe multiverse

Save your file and close either kwrite or gedit. Lastly, and most importantly, type this into the terminal

sudo apt-get update

Apparently, some people have been experiencing errors even after following this tutorial. There's some weird glitch where residual info hangs out and gives you a GPG signature error. If that's the case, do this:

Back up the sources.list you got from this tutorial with this command:
sudo cp /etc/apt/sources.list /etc/apt/sources.list_backup

Then edit the sources.list by typing sudo gedit /etc/apt/sources.list and delete everything.

sudo apt-get update with your empty repositories list.

Finally, *sudo cp /etc/apt/sources.list_backup /etc/apt/sources.list*
sudo apt-get update

Running Windows Programs

You have several options for running Windows-only programs in Ubuntu:

- Find Linux equivalents for those programs.
- Run the native Windows program using Wine.
- Run the native Windows program using Crossover Office[8] or Cedega.[9]
- Dual boot.
- Run an emulator.

For the first one, you can usually find some good guides if you do a Google search for the keywords *linux equivalents windows programs*.

The third you'll have to pay for, but I hear good things about these programs.

I don't know how to do the fifth, and I've already provided a link to the fourth.

A bit about Wine... I don't know how it works, but it does seem to work with a lot of simple Windows programs. I'll show you how I get Filezilla to work in Linux, as an example.

Assuming I've already enabled extra repositories, first, I install Wine:

sudo apt-get update
sudo apt-get install wine

Then, I download the *setup.exe* file for Filezilla. When I double-click on it, Wine will try to open the file. Then, the installer appears, just as if I were using Windows. Instead of installing Filezilla to C:\Program Files\Filezilla\, I'm going to override the default installation location and install it to z:\home\username\.wine\drive_c\Program Files\Filezilla. For some reason, z:\ is what Wine calls my Ubuntu partition.

Then, I set up a launcher (on the panel or in the menu) for the command *wine "z:\home\username\.wine\drive_c\Program Files\Filezilla\Filezilla.exe"*

That's it. Now when I click on that launcher, Filezilla will load up.

If a Windows program does not work with Wine, you may need Crossover Office. Cedega is a special version of Wine that's made for Windows-only games.

8 http://www.codeweavers.com/

9 http://www.transgaming.com/

XFCE: What to do if Gnome or KDE are too slow for your computer...

I've talked a bit about the Gnome version of Ubuntu (the original Ubuntu) and the KDE version of Ubuntu (Kubuntu). There's also something (which you cannot get prepackaged as a disk) called Xubuntu. You may be interested in Xubuntu if you have less than 256 MB of RAM on your computer. Or, you may be interested in it just to try something a bit different and a lot faster.

To install the XFCE version of Ubuntu, you'll need to enable extra repositories. Then:

sudo apt-get update
sudo apt-get install xubuntu-desktop

To use XFCE after you've installed it, log out. Then, select it under **Session**. Log back in again.

A couple XFCE tweaks worth noting (though, the best way to get to know it is to just play around a bit):

1. You can run certain Gnome programs in XFCE and still maintain the speed of XFCE. The two most popular to run are *gnome-volume-manager* (to automount external media) and *nautilus* (to have a “normal” desktop with icons).

2. The default XFCE behavior is to make you type in your password in order to shut down. To fix this, edit as root the /etc/sudoers file and add this line:

cooldanny ALL = NOPASSWD: /usr/sbin/xfsm-shutdown-helper

Substitute your actual username for *cooldanny*, of course.

Other Resources

This document is not intended to be comprehensive coverage of everything Ubuntu-related. It's a basic guide for the first-time user. You will have to explore a bit to figure out more. Here are some basic tips for best using your time with Ubuntu:

Learn how to search

Chances are, if you have a question or are experiencing a problem, someone else has had that question or experienced that problem before. There are two good Google searches to learn.

1. If you're encountering an error of some kind, Google the error. Just copy the error exactly as it appears in your terminal and paste it into Google.

2. If you're wondering how to do something—say, install Limewire—search for *site:ubuntuforums.org howto limewire*

Sign up for the Ubuntu Forums

I have to confess—I wasn't drawn to Ubuntu because it's such an amazing operating system. It's good, but it doesn't blow the competition out of the water by virtue of the software quality alone. The community and its helpfulness are what drew me to Ubuntu and kept me there.

You'll find all sorts of helpful tips and tricks there. Anything from restoring the original Firefox and Thunderbird icons to the Mozilla programs to getting your computer to boot more quickly.

Play Around

Click around. Enjoy your system! Find themes and icons at http://www.gnome-look.org or http://www.kde-look.org. Change the sizes, placement, and transparency of panels. Change your wallpaper. Change your screensaver. Install applications through Synaptic Package Manager or Adept. Uninstall them.