

Getting Started with **Linux: The Complete** **Guide**

By: Whitson Gordon & Kevin Purdy

Contents:



[Lesson 1: Why Install Linux?](#) - Everyone's heard about this thing called Linux, but you may still be wondering if it's really for you. Here's a look at the reasons you may want to install Linux.



[Lesson 2: Pick the Right Linux Flavor for You](#) - Now that you've decided to give Linux a try, you'll need to decide which flavor—or distribution, as they're called—is a good fit for you.



[Lesson 3: Installing Linux on Your Computer](#) - Once you've decided which flavor of Linux fits your needs, it's time to install it. Here, we'll go through burning a live CD or USB drive, partitioning your hard drive, and installing Linux on your PC.



[Lesson 4: Fine-Tuning Your Hardware Compatibility](#) - When you first install Linux, there's a good chance one or more pieces of hardware on your computer won't work perfectly. Here's how to get your Wi-Fi, 3D graphics, mouse, and more up and running.



[Lesson 5: Installing Apps and Going Further](#) - Now that you've got a working Linux installation, it's time to customize. Here, we'll go through how to install apps, customize the look and feel of your desktop, and get comfortable with the command line.

Why Install Linux?

Curious about Linux, but not ready to dive in head first without a little background? We're on it. As part of our our Night School series, we'll be detailing, troubleshooting, and taking a deeper swim into the open-source OS this week. Today, we're offering some encouragement for the hesitant.

Why, in this day of razor-thin and elegant Macs, a solid Windows 7 offering, and cloud-connected gadgets would anyone bother to carve up their hard drive and install Linux, the geekiest of the major operating systems? Linux will never be everyone's desktop, but here's why it might be just perfect for you, as a workhorse or a hobby.

In a general sense, the core operating system on your computer is becoming more and more abstract, as more services move onto the web, and as your browser and sites' own code are capable of doing much more of the traditional heavy lifting of daily work. So if you use a computer for browsing, email, IM, and some light app functions, Linux can work just as well for you as any other system.

There are many reasons to switch to Linux, as [explained by your fellow readers](#). Here are some of the strongest cases to consider.

It's Totally Free, for Any System You Have



Let's just come right out and say it—you, or at least a good number of people you know, have installed not-quite-squeaky-clean operating systems on your computers. Maybe it was testing out Boot Camp on a Mac, or upgrading a secondary machine, or just avoiding a sudden \$100-plus financial hit. You had to pull a few tricks to get it to work, then forever after had to keep a cautious eye on updates, service packs, and other potential slip-ups. The same dark alley feel often comes with hefty apps like Photoshop or Office.

Linux is free, for one or 8,000 computers, with all future upgrades and potential software included. If you've got hardware you want to use but not the software, Linux is where you can turn to make it usable. Generally, even a newcomer can get Wi-Fi access, a modern browser, and the essential desktop apps running on a system in short order. Beyond those basics, well—that's what this Lifehacker Night School Series will get into!

It Fits on Systems New or Old, Spacious or Small

If you've got an older computer, or maybe a newer netbook or laptop that's not quite running at turbo speed, Linux is where you can get both modern capabilities and swifter operation out of it.

The core of Linux is regularly worked on by efficiency-obsessed engineers who run thousands of expensive servers, and most of the desktop versions meant for personal use run lighter than their Windows and Mac counterparts. Linux can also take up far less space on a hard drive, and may support hardware and peripherals that you can't get working in the Big Two operating systems.

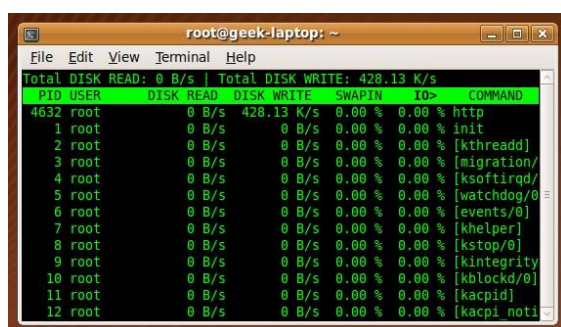
It's a Fine-Tuner's Dream



If you read this blog regularly, there's a good chance you like having your computer desktop Just So. You want pop-up notifications of certain important things. You like certain colors for your windows, taskbars, and buttons. Keyboard shortcuts? Oh, you like your keyboard shortcuts—all except for maybe a few of the defaults, which don't make sense for you.

When you're running Linux, everything can be modified. Right-clicking will get you a lot of places. If you really want to get down deep, you can edit a configuration file for nearly everything in the system. You can give certain apps more prioritized access to the system, and set up automatic tasks like nobody's business. You don't have to get in this deep, and there are many tools to make this tweaking easier, but if you're eager to tweak and customize, Linux is a dry canvas unto your imagination/obsession.

It's the Best Way to Learn Deeper Computer Knowledge



Getting good with a terminal, or command line commands, gives you the means to make your computer do what you actually, really want it to do, without having to click and guess. Linux is a great place to learn terminal commands, along with many other fundamentals of a modern operating system.

Some of that learning may come about because you need to fix something, but it's a rewarding trial by fire. You can learn, for example:

- How file permissions work
- How to spot runaway apps with the `top` command.
- How USB drives, hard drives, ISOs, and other storage are "mounted" and accessed.
- What `rsync` is, and how to use it as your own personal Time Machine backup.
- What the commands `grep`, `tail`, `whereis` and `find` do, and how they can make you feel like a Zen master of data.

There Are Some Really Impressive Linux Goodies to Play With



So, there's all that serious data/hardware/knowledge reasoning. But is Linux actually fun to mess around in? For sure.

You can make [brilliantly beautiful desktops](#) with real data and productivity at their core, or set the [whole earth as your wallpaper](#). Give yourself a great way to launch and manage apps with [Docky](#) or the uber-powerful [GNOME-Do](#).

You can fend off your fears of needing just that one Windows app by [running it inside Linux](#), and learn something about virtualization in the process. Need some productivity tools? [Linux has plenty of them](#).

Pick the Right Linux Flavor for You

If you've [decided to give Linux a go](#), maybe for the second time, it's time to pick out a system and at least load it onto a "live" USB stick. If you're new to the Linux world, here are the distributions we recommend trying out.

Note: We could publish our own Wikipedia about all the different distributions, or "flavors" in our headline speak, of Linux out there. Instead, we're going to simply make a pitch for four different systems Lifehacker editors have been known to use. You can try out all of them (except Arch, perhaps) from a live CD or USB image you can download from each of their sites—and we'll show you how to do that tomorrow.

Ubuntu: The Go-To Option For Most



Ubuntu is a lot younger, and different, than a lot of Linux distributions. It was founded by a man with a good chunk of money and a dream of making a kind of mainstream Linux: a distribution of, as the tag goes, "Linux for Humans." There are Linux flavors that are lighter on resources, or easier for beginners, or more robust in certain aspects. But Ubuntu is where you can find a distribution designed for

usability, supported by a huge community of enthusiasts, and moving forward at a pretty rapid clip.

Because of all that community support and attention, it's where we recommend newcomers start off with Linux. It's the best shot you've got at 100% hardware support, and it's honestly the most Google-able distribution when you want to learn or tweak something.

There are also a host of Ubuntu "variations" you can turn to. They're basically the same core system with a different or customized desktop environment (and [Here's what that actually means](#)). So try a KDE desktop ([Kubuntu](#)), the lower-spec-friendly XFCE ([Xubuntu](#)), a flavor specific to your tiny little computer ([Netbook Edition](#)), or the [many other options](#).

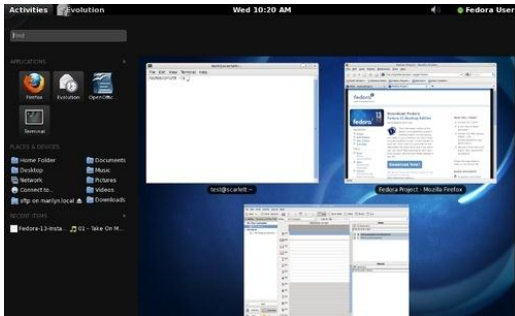
Linux Mint: Even More Beginner-Friendly



Linux Mint owes more than a little of its core software, and inspiration, to Ubuntu, but it's a successful branch into a more cohesive, and even more beginner-friendly realm. All the stuff you'd expect to find on an OS, like MP3, DVD, and Flash, are included by default, the menu

is more Start-like, and the system has a cohesive feel and a lot of smart choices made for newcomers. Blogger Jeff Hoogland makes a richer case for Mint over Ubuntu, and his points are good ones. Mint is a good pick for those tired of playing around with text files (though you're still *able* to do that, of course).

Fedora: The Solid Alternative



Fedora is the personal desktop offshoot of enterprise Linux firm Red Hat. The team puts out a robust operating system that updates regularly, they incorporate cutting-edge Linux developments at a rapid pace, and they support a variety of hardware, even extending an olive branch to PowerPC users. Using Fedora doesn't feel entirely different from Ubuntu, but one main difference comes in the package installer—the app you use to quickly

install apps from the distribution's own servers. Overall, it's a good pick, and really a matter of preference.

Arch Linux: The Starting-from-Scratch Project



Don't try to set up Arch Linux during a lunch break. Do dig into Arch Linux if you want to learn way more about Linux, get a system at just the right size and configuration for your needs, and want a crash course in how to tweak a Linux system for better performance.

Installing Linux on Your Computer

So you've decided to give Linux a shot, and you've found a distribution that suits you. But how do you actually get it installed? Here, we'll show you how to create a live CD or USB drive, boot into Linux, and install it on your system.

Whether you've played around with [live CDs](#) and [drive partitioning](#) before or this is your first time, this guide will get you up and running with Linux in just a little bit of time. It isn't the most straightforward process, but if you're reading this, you're probably experienced enough with a computer to pull this off—it just takes a bit of legwork.

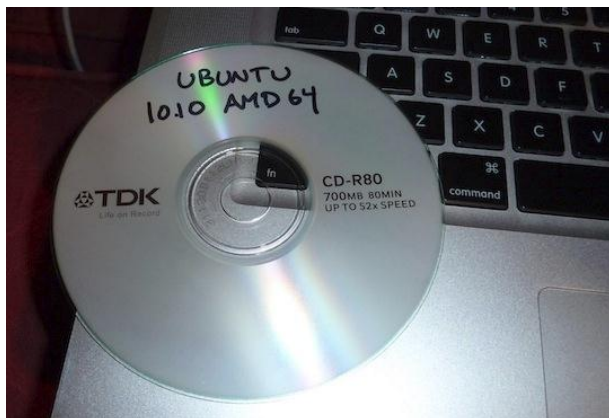
Note: If you're installing Linux on a Windows computer, the process shouldn't be too difficult (though you may want to [brush up with our previous installation guide](#) while you're here). If you're installing on a Mac or Hackintosh, however, you'll need to perform a few extra steps, so definitely check out our [triple booting guide for Macs](#) and [Hackintoshes](#), even if you aren't triple booting.

Burning a Live CD or USB Stick

There are a few different methods for installing Linux on your computer, but generally, the most popular way is by downloading and burning a live CD (which [has its own uses besides just installing Linux](#)). I'll show you how to do both here. The live CD method is probably easier, so go that route unless you're on a netbook, or otherwise can't burn a CD.

The Live CD Method

You'll have to get your live CD from the net, so head to your distribution's home page (like [Ubuntu's](#)) and look for a downloads section. Most will make it pretty easy to find. In addition, many sites will even host BitTorrent downloads of their distribution too, which will be a little faster—so if you see a link for "[alternative downloads](#)", check that out. Otherwise you can probably download it directly from the site.

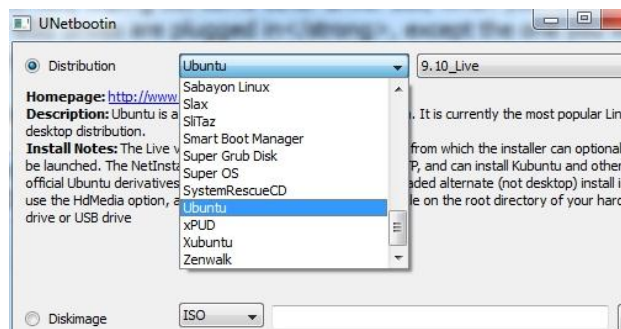


You'll probably have a few different choices when you download. For example, some distributions have netbook-optimized versions, while others (like Ubuntu) will offer different versions based on the [desktop environment](#) they come with. And, most will have 32- and 64-bit versions available as well (if you don't know which one's right for you, we've [written a](#)

[handy guide to help you out](#)). Generally, it shouldn't be too hard to figure out which one you want. When in doubt, just go with the 32-bit desktop version (sometimes labeled "i386").

Once your .ISO file is done downloading, open up your favorite burning program and burn that sucker to a blank CD. It's [pretty easy to do on Windows 7](#). Once it's done burning, restart your computer. Wait for the "press any key to boot from CD" prompt, and then press a key. Once the CD boots up, it'll give you the option to try out Linux or install it. Go ahead and click install. If you want to try it out, though, go for it—you'll be able to get a pretty good feel for what the desktop is like. When you're ready to install, you can usually launch the installer right from the desktop.

The Live USB Method



The Live USB method requires [previously mentioned Unetbootin](#) for Windows. All you need to do is download it, start it up, and you can manage everything from right inside the program. Pick your distribution from its list (remember to pick the right version, as described above), pick the drive letter that corresponds to your thumb drive

at the bottom, and hit OK. If Unetbootin doesn't list your distribution, you can still download an ISO as described in the live CD method and point Unetbootin to that file instead.

Booting from your USB drive will take a few extra steps over the live CD method. While your computer is probably set to check your CD drive for bootable discs, it probably *isn't* set to check your USB ports. So, with your newly created live USB stick plugged in, restart your computer and enter your BIOS setup (usually by hitting a key like Delete when you first start up—your computer's splash screen will let you know when you first turn on your computer). Head over to the "Boot" section of your BIOS and find the section for changing your PC's boot order. You'll want to move your USB hard drive to the top of the list. Save your settings and exit the BIOS. When your computer reboots, it should take you to the Unetbootin menu, from which you can boot into your Linux live session (as described in the live CD method).

The Installation Process

The installation process will be slightly different for every distro, but in general setup should guide you through the necessary steps pretty easily. Assuming you're installing Linux alongside another OS like Windows, though, there are a few things you'll want to pay attention to.

Partitioning Your Drive

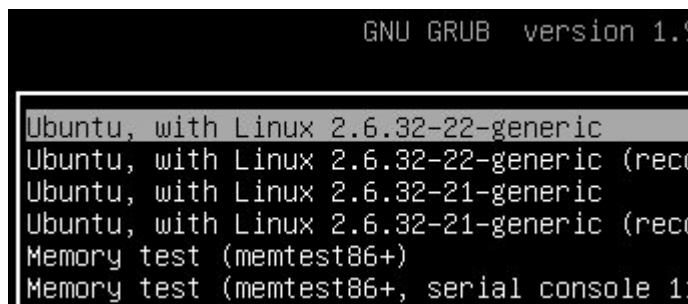
When the installer asks you where you want to install Linux, you'll have to partition your drive. We've [gone through how to do this](#) a [few times before](#), and it's usually pretty simple to add new partitions from the free space on your drive.

Ubuntu's installation should partition the drive for you automatically, and unless you have any special needs (like if you're on a Mac), you can breeze right through the installation with no problems. If not, you may be given a more advanced partitioning tool, and you'll have to create the partitions yourself. If this is the case, you'll actually want to create two new partitions. One is for the operating system itself, which I'd format as Ext4. Give it at least 10 GB of space, and set the mount point as `/`. You'll want to



create the second partition for what's called *swap space*. This essentially helps your computer manage memory more effectively and keeps it running fast. If you have a small amount of ram (one or two GB), you'll want your swap partition to be twice as large as the amount of RAM you have. If you have 3 GB or more, you can probably just make a swap partition that's the same size as the amount of RAM you have.

GRUB and Other Bootloaders



Linux is going to install a new bootloader for you called GRUB. It's going to replace your normal bootloader and give you the option to choose between Windows and Linux at startup. In general, you shouldn't have to do anything here—most distros will install GRUB by default,

and it should work correctly out of the box. Just note that Mac users will want to [install GRUB on the Linux partition itself](#), and Windows users will need to be careful since if you

reinstall Windows, you'll lose GRUB and have to [reinstall it yourself](#). Of course, if you prefer something a little prettier, you can mess around with [previously mentioned Burg](#), but we'd recommend doing that after you get your Linux installation up and running.

That's it! To boot into your newly installed Linux partition, you just need to restart your computer. When you do, it'll take you to the GRUB menu, which will let you choose whether to boot into Linux or Windows. From there, you can play around with your new Linux installation. It'll probably come with quite a few apps installed, some you'll recognize and some you won't, and you can poke around in the settings and see what's available to you. If you're lucky, most of your hardware will work. If you're unlucky, you'll have a few quirks with your hardware, and if you're really unlucky, your Wi-Fi won't work out of the box and you'll have to work just to get connected to the internet. Luckily, we'll be talking all about getting your hardware working in next chapter.

Fine-Tuning Your Hardware

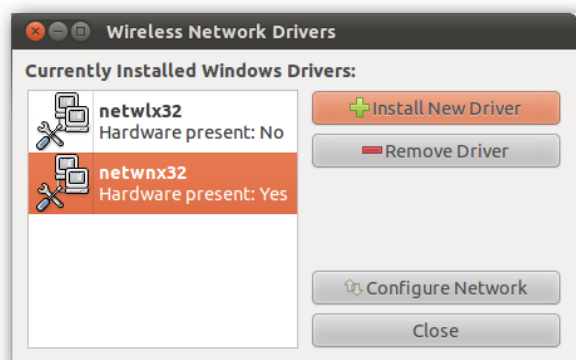
On some systems, Ubuntu and other Linux systems will install as if they had always been right there on your system. Other times, there's a piece missing. Here's how to patch up the last piece (or two) of your system if not everything's working right off.

***Note:** Like other sections of our [Getting Started on Linux series](#), this guide is written from the perspective of a new Ubuntu user. But the knowledge is univerally helpful, many of the tips can be applied in other systems, and the last bit on troubleshooting is handy for any Linux newcomer.*

The Biggie: Get Wi-Fi Working

You might load in an Ubuntu CD or USB drive and have your Wi-Fi working right away, as if it was built for this Linux stuff. Or you might spend an annoying bit of time searching out Windows drivers and re-jiggering them for Linux use. It's hard to tell, but it's very annoying to encounter Wi-Fi problems right off—since, you know, you can't go online to find a solution, or download the right fixes.

If you'd like to know where your wireless card stands, check out [HJ Heins' gigantic compendium](#). If your WI-Fi card is listed in the green, you're likely good to go, and most distributions can work with it. If not, the second place to check is Ubuntu's community documentation, where owners of certain finicky cards have [posted their solutions for all to follow](#). Beyond that, you can turn to this [basic troubleshooting guide](#), along with the help available through Google, the Ubuntu Forums, and other community resources (our tips on doing so are a bit further down).

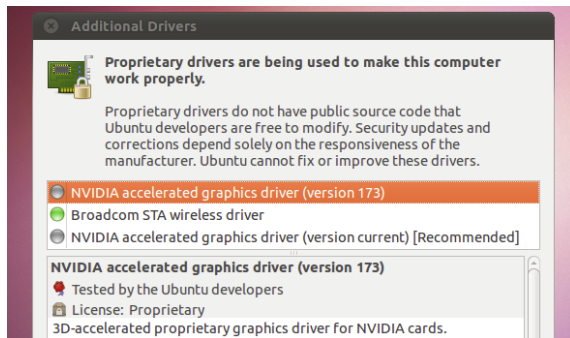


When all else fails, use your wireless card's Windows driver. Odd-sounding, sure, but you can use a tool to "wrap" the Windows driver for use in Linux. If you want to jump in, download the Windows driver for your WI-FI hardware, un-pack them, then install the `ndisgtk` package in Ubuntu (either with a temporary ethernet cable, or with [offline means](#)). Launch the app from the System->Administration menu ("Windows Wireless Drivers"), then load the INF files from your

Windows package with the "Install New Driver" button.

Luckily, the How-To Geek has [walked through the process, step by step, in Ubuntu](#), and it's a great resource.

Proprietary Drivers



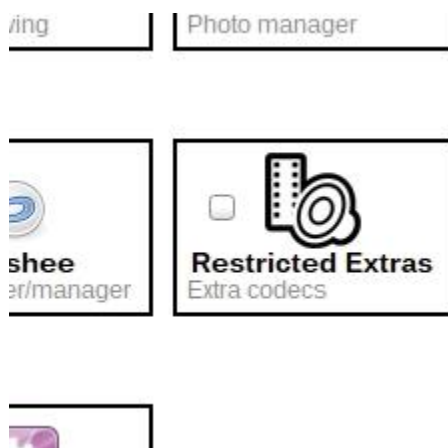
While many hardware manufacturers, like Intel, make their drivers free and open source, others (like Nvidia, ATI, Broadcom, and others) don't. This means that, as an open source operating system, most Linux distributions aren't allowed to package these closed source drivers with themselves—which means you have to install them manually.

For what it's worth, open source drivers for some of these peripherals do exist. Nouveau, for example, is an open source Nvidia driver that comes with many Linux distributions. It'll give you some pretty good support, but if you want the best 3D and high quality graphics your card has to offer, you'll have to install the proprietary drivers.

This is pretty different on every distro. Ubuntu has a very nice proprietary driver manager under System > Administration > Hardware Drivers, which will tell you the drivers available for your system and let you install them with the press of a button. Other distributions will require a bit more work. You'll have to search out the instructions for your specific distro on Google or that distro's documentation (for example, Mint has [their own instructions for installing Nvidia's proprietary drivers](#), as does [SUSE](#)).

This isn't just for graphics cards, either. If you have a particular Wi-Fi card or other piece of hardware that doesn't have open source drivers, you'll need to use this method for them too. Generally, this is the first thing I do whenever I install a fresh copy of Linux—check what proprietary drivers are available and enable them.

Proprietary Stuff: DVDs, MP3s, Video Formats, and More



Ubuntu is not the most puritanical of Linux distributions when it comes to open software, but it doesn't go so far as to build in access to MP3s, DVDs, licensed fonts, and other items that you probably take for granted on most computers, but which require tricky licensing.

But getting access to the good stuff has become remarkably easy. So easy, in fact, it's basically two steps now:

Step One: If you're in Ubuntu right now, [click here to install ubuntu-restricted-extras](#).

If you're not in Ubuntu at the moment, you could bookmark this page and click that link when you get into Firefox. You could also [grab an installer from Ninite](#), or head to the Ubuntu Software Center (found in the Applications menu in the upper-left desktop corner) and search for "ubuntu restricted," then install the `ubuntu-restricted-extras` result.

Step Two: Open up a Terminal (Applications->Accessories->Terminal), then enter this line and press Enter:

```
sudo /usr/share/doc/libdvdread4/install-css.sh
```

Now you've got support for MP3s, commercial DVDs, most of the video and audio formats you'll come across, Java, Windows-specific fonts, and everything else you might encounter. And it didn't hurt at all, did it?

Other distributions usually offer a similar kind of "restricted goods" package in their repositories, or app offerings. Search your installer package (YUM, Synaptic, or another offering from the main menu) for things like "mp3," "avi," "Windows media" and the like.

Extra Mouse or Keyboard Buttons

By default, your mouse probably works just fine—left click, right click, scrolling, and even sometimes forward and back buttons work out of the box in Linux. However, if you want to remap your fancy mouse's extra buttons, you'll need to do so manually. There are a ton of different methods, but I've always found `xbindkeys` to be the easiest program. It's quite intimidating but fear not—it's actually remarkably simple. For this demonstration, I'm going to map my mouse's hidden thumb button to Ctrl+T, so I can open new tabs in Firefox with ease.

To start, we want to figure out the names of the buttons on our mouse. Each button is numbered, but it's hard to tell which button is which just by looking at it. So, to find out, run the following command:

```
xev
```

This will open up a small white window with a box in it. Place your cursor in the white window, but outside of the box and press the button you want to remap. In this case, we're pressing the thumb button. Xev will give you a few lines of code in the terminal, like:

```
ButtonPress event, serial 33, synthetic NO, window 0x4e00001,  
root 0x142, subw 0x0, time 568329, (93,19), root:(96,714),  
state 0x0, button 10, same_screen YES
```

See on the last line, where it says `button 10`? That's what we're looking for. Looks like our thumb button is button [#10](#) on our mouse. Hit Ctrl+C to exit `xev`, make a mental note of your button's number and move on to the next step.

Next, we'll need to install two programs: `xbindkeys` and `xautomation`. In Ubuntu, that means running the following command:

```
sudo apt-get install xbindkeys xautomation
```

After that, you'll want to create a config file for `xbindkeys`. To do this, run:

```
xbindkeys --defaults > /home/your-user-name/.xbindkeysrc
```

Replacing `your-user-name` with your username. This command should be the same on all flavors of Linux. Next, we'll want to edit that file, so open up your file browser, head into your home folder, and go to View > Show Hidden Files. You should see the `.xbindkeysrc` file show up at the bottom of the list. Double-click on it, and add the following lines of text to the end of the file (but before the `# End of xbindkeys configuration #` line):

```
# Thumb Button = Ctrl+T
"xte 'keydown Control_L' 'key T' 'keyup Control_L'"
  b:10
```

Xte is a program that's part of the `xautomation` package that we downloaded earlier, and it simulates keypresses. In this case, we're simulating pressing the left Control key down, hitting T, and then releasing the control key. `b:10` corresponds to the button number we found earlier. See? Not that difficult. You can now go back to `xev` and figure out your other mouse button numbers, and remap them as you see fit. If you need to know the names of other keys on your keyboard, [xte's manual page](#) lists some useful ones.

You can do a whole lot of things with `xbindkeys`, and we won't dig into it here—this should get you started on at least getting a bit of functionality out of your mouse—but check out [Ubuntu's Keyboard Shortcuts guide](#) for more info. I'd also highly recommend googling `xbindkeys` and the name of your mouse, or what you want to do. You'll find some pretty neat tutorials out on the net. Good luck!

Where to Get More Help



[Ubuntu Documentation](#) > [Community Docs](#)

Obviously, this is just the tip of the iceberg. Because Linux is so endlessly configurable, there are a lot of things you can do—and a lot of hardware you can do it with. Luckily, there are also a lot of great places to get help, including:

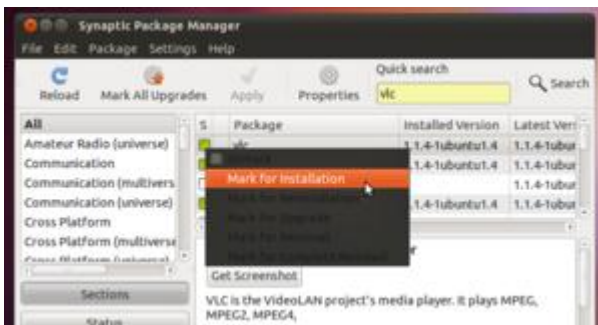
- [Ubuntu's Community Documentation](#): This is a great place for frequently sought-after info. You can find out about things like mapping your mouse buttons (which we discussed above), and even search for a specific laptop or netbook model to see what tweaks you'll have to make to get all your hardware working. In fact, if you're running Linux on a laptop or popular desktop computer, the first thing I'd recommend doing is searching the Community Documentation for your model—you'll find almost all the info you need to get your hardware up and running.

- **[The Arch Linux Wiki](#)**: While it's designed for Arch Linux users, it's still one of the best Linux wiki's I've ever seen. You'll need to keep in mind the distro you're using when you read it (for example, when they tell you to install a package with `pacman`, you'll need to remember to run the command using `apt-get`, or whatever your distro's package manager is), but other than that, it's a great resource, especially when it comes to command line tools and getting hardware working.
- **[The Ubuntu Forums](#)**: It's one of the biggest Linux forums out there, so if you have a problem, chances are someone's already asked about it on the Ubuntu Forums. Google the problem you're having (using the tips below), and if you don't find anything grab an account at the forums and ask it yourself—trust me, as a beginning Linux user, you'll be spending a lot of time here. You might as well get an account now.
- **Google**: Frankly, the best way to find what you're looking for is to search Google. The above resources are all great, but what I usually do is Google my problem, then look for a result from the top three sources—if they exist, it's likely they'll contain the solution to my problem. Make sure your search terms are good, too—[try using the word "solved" in your search terms](#), or if you see the name of a program on one page that you think does what you want it to do, try googling for tutorials on that program.
- **Lifehacker**: We've got quite a few Linux-savvy readers here at Lifehacker, so if you're looking for advice, our [weekly open thread](#) is a great place to find it. Your fellow readers may not be able to answer super-specific questions ("I'm trying to do X with program Y and it's giving me this error code. What do I do?"), but they'll be a wealth of knowledge when it comes to the general stuff ("What's the best way to remap my mouse's button to open a new tab in Firefox?").

Installing Apps and Going Further

So you've installed Linux, gotten all your hardware up and running, and now you're staring at a blank desktop. What next? Here are a few tips (and a few resources) that will help you build up the OS of your dreams.

Installing Applications



The first thing you'll probably want to do is install some of your favorite applications. Linux works a bit differently than Windows and OS X when it comes to installing apps. In Linux, you don't always have to download an installation package from a web site to install an app - very often, they're located in your distribution's *repositories*. Open up your distribution's package manager (in Ubuntu, you'll find it under System >

Administration > Synaptic Package Manager) and search for the package you're looking for—say, `vlc`. You'll be able to install it right from there without even opening your browser.

You can also do this from the terminal if you know the name of the package. In Ubuntu, the command for installing a package like `vlc` is:

```
sudo apt-get install vlc
```

You can, of course, replace `vlc` in the command above with whatever package you're installing. You can also install multiple packages at a time to save yourself keystrokes:

```
sudo apt-get install vlc chromium-browser
```

The great thing about this setup is that it allows Linux to update the OS and all your apps at once, with its update manager. No longer do you have to update apps individually—*everything* on your system will update through one channel.

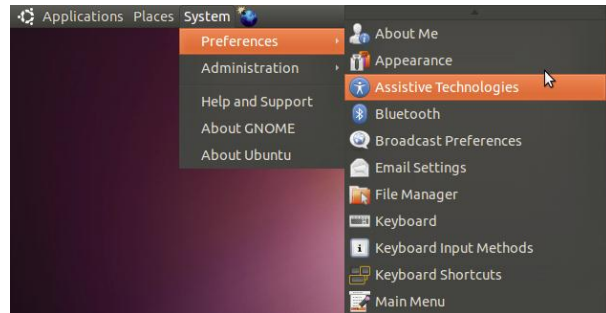
You'll have a lot of apps in your distribution's repositories, though some (like Dropbox) will need to add their own separate repositories. You'll usually find instructions for how to do this on an app's home page. In addition, distributions like Ubuntu and Mint allow you to download `.deb` packages for apps that aren't in the repositories. These work much like installation packages on Windows: just download them from the web site, double click on them, and they'll install the application *and* its repository, so it updates with the rest of your system.

If you're looking for good apps to install, our [Lifehacker Pack for Linux](#) is a great place to start. [Ninite also does bulk Linux installations now](#), so check out the apps they recommend installing on a new system to get your system up and moving.

Update: Many of you have mentioned the Ubuntu Software Center, which is a more beginner-friendly way to install apps. It's only in Ubuntu, but if you're just looking to install an app or two, you can go through it too. You can find it under the Applications menu; it works just like iTunes or other popular app stores.

Tweak Settings Galore

If you head to your distribution's settings menu, you'll find a *lot* of stuff in there. Honestly, one of the first things I do (on any system, really, Linux or not) is head into the settings panel and see what I can tweak. You can spend a good amount of time in there customizing your system to work like you want it to, so I'd recommend



just browsing around there. If you're running Ubuntu, I'd also highly recommend checking out [previously mentioned Ubuntu Tweak](#) for even more settings tweaks.

Get Comfortable with the Command Line

There comes a time, however, when you'll find a setting or app that you can't just tweak from the GUI. In those cases, you'll have to delve into the Terminal—but fear not! The command line isn't really that scary, and once you get comfortable with it, there really are no limits to how much you can customize your system. I'd recommend checking out our [command line primer for beginners](#) to get started. [Previously mentioned CLCompanion](#) is also a handy little tool if you're unfamiliar with the command line. And no joke, I wear [this shirt](#) when I know I'm going to have a night of deep Linux command line action.

Of course, the fact of the matter is that you can only learn so much Terminal from beginner's guides. I've found that almost everything I've learned about the command line I learned by doing. Whenever you find something that requires the command line to do, just look up a tutorial on how to do it—you'll probably find that it isn't that hard, and you'll have added a new command or trick or two to your repertoire. The best advice I can give is: don't be afraid!

Of course, there are a few things that you *won't* learn just by doing, and that's how to move through the Terminal at lightning speed with all its built-in shortcuts. Luckily, we've [written a guide on that too](#). Learn them and use them—you'll save years off your life typing out long commands.

Customize Your Desktop



Lastly, one of the things everyone loves to do on Linux is make their desktop look awesome. Often, this involves installing Compiz, a tool that will let you enable all sorts of cool desktop effects, including the [infamous 3D desktop cube](#). It also has some useful effects, like showing all the windows on your desktop (Exposé style). Check out [Ubuntu's guide to setting up Compiz](#), and if you've already

installed Ubuntu Tweak like we recommend above, be sure to check out its Compiz section, since it has some nice settings built-in.

As for the rest of your desktop, there are really a *lot* of different ways to customize it. I'd recommend checking out sites like [GNOME-Look.org](#) (or, if you're using KDE, [KDE-Look.org](#)) and search there for inspiration. You'll find cool window decorations, icon sets, wallpapers, and all sorts of other cool stuff. YouTube is also a cool place to search for ideas. And, of course, if you ever get lost, Google is your friend!

That should give you enough to get started customizing your installation to fit your tastes. Again, one of the best parts about Linux is that you can make it completely your own—so any time you find yourself saying "man, I wish I could set up my OS to do *this*", Google it! There's probably a solution out there, and as long as you're ready and willing to try anything, you can make it happen. If there's one thing I can't stress enough, it's don't be afraid of things that sound complicated. You'd be surprised how easy the command line, config files, and even [compiling apps from source code](#) can be if you just go for it! Good luck to you all, and be sure to share your experiences, issues, victories, and everything else here in the comments.