Linux

A short guide to the popular free & open-source operating system

Contents

Articles

References

Article Licenses

Linux Guide

Getting started

- Introduction
- Distributions (more detail)
- Getting Linux

Installing Linux

- Basic
- Dual-booting
- Advanced

Using Linux

- How Linux Works
- Logging into Linux
- Using the shell (console)
- More Linux commands
- Desktop environments
- Getting Help
- Getting out of Freezes
- Installing software

Managing Linux

- Tools
- Runlevels
- Operating modes
- System administration
- Network administration
- Modules
- Security
- Monitoring

Programming on Linux

- Programming tools
- Book about X Window Programming

Applications of Linux

- Use Audacity to make MP3s
- Burn a CD or DVD
- LDAP authentication in Linux
- Installing SugarCRM On Linux
- Using SSH
- PlayStation 3

- Linux and Bluetooth

Ideas for fun stuff

Specific tips/fun stuff

- Virtual Network Computing (VNC) can be used to access your computer from anywhere if it's connected to the Internet and you know its IP address (use `ifconfig`). vnc-java can automatically create a webpage with a Java applet that eliminates the need for a VNC client. Try using x11-vnc
- for remote X you can also use simply X with xhost as X is a network protcol, or you can do X forwarding with ssh.
- WINE will often run your old Windows programs very well. It's hit and miss, but often more complex, graphics-heavy applications will work better than specifically integrated Windows apps.
- There are a number of unique, nicely-made open source games that run under Linux:
  - Battle for Wesnoth (turn-based strategy game with elements from RPGs) http://www.wesnoth.org
  - Blob Wars (platform shoot-em-up)
  - Celestia (not really a game, but a very detailed astronomical program that can take you anywhere in the solar system and beyond) http://www.shatters.net/celestia/
  - Enigma (marble-rolling puzzle game) http://www.nongnu.org/enigma
  - GnuGo (one of the better Go AIs available, use ccGo or JaGo or cGoban for a user interface) http://www.gnu.org/software/gnugo/gnugo.html
  - Frozen Bubble (similar to Bust-a-Move) http://www.frozen-bubble.org/
  - Kobo Deluxe (4-way scrolling shoot-em-up) http://olofson.net/kobodl/links.html
  - Orbit (simple 3D space combat simulator) http://www.head-crash.com/orbit/
  - Rocks'n'Diamonds (Boulderdash clone plus a couple other games, too) http://www.artsoft.org/rocksndiamonds/
  - Tux: A Quest For Herring http://tuxaqfh.sourceforge.net/
  - Extreme Tux Racer (sliding down the mountains and catching herring at over 100 kph) http://www.extremetuxracer.com/
  - Nexuiz (A Unreal Tournament like first person shooter) http://www.nexuiz.com
  - Warsow (A competitive quake3 like first person shooter) http://www.warsow.net
- If you're using KDE, check out these addons:
  - Kuake, a drop-down console, *a la* Quake's in-game console.
  - Karamba and Superkaramba, environments to support eye-candy interface features.

Linux Guide/Introduction

Introduction

Welcome to the world of free and open source software!

Welcome to Linux! GNU/Linux is descended from the UNIX operating system, but is open source software, which means that you can view its source code and change it to suit your needs. Of course since this book is geared to those new to Linux, we stay away from very technical issues that are more suited to Linux veterans. This book is going to try to be geared toward the person who has heard about Linux and might be considering trying it out or perhaps the person who has already "taken the plunge" and is looking for more information or wondering where to start now that they have Linux installed. But first, a little history lesson.

The name "Linux" technically refers to an operating system "kernel", a single but key component of a complete operating system. In everyday use, the term "Linux" is frequently used to refer to a complete operating system which consists of the kernel and some of the thousands of other programs required to make an operating system useful. Much of the important system software that is typically installed on a Linux system comes from The GNU Project [1], a project to build an operating system made entirely of free software.

The first Linux kernel was created by Linus Torvalds. It was first released on 5 Oct 1991. It was started as an x86-only, single-processor operating system, but grew to become one of the most ported pieces of software. Other parts of a complete GNU/Linux system come from other projects such as the GNU project, and are integrated into a complete GNU/Linux OS by your supplier. Usually your supplier will assign their own version number to the integrated whole.

The GNU Project is overseen by the Free Software Foundation, founded by Richard Stallman, who believes that the people should use the term "GNU/Linux" to refer to such an operating system, because so many of the required programs were in fact, written as part of the GNU Project.

References

[1] http://www.gnu.org/gnu/thegnuproject.html

Linux Guide/Distributions

A **distribution** is a type of Linux. Linux comes in a large number of distributions, some of which are designed for everyday use, and others designed with a specific task or device in mind. We'll discuss some of those differences below.

Most Linux distributions have a special type of CD, called a **live CD**. If you insert this CD and then restart your computer, the live CD will run Linux on the computer while avoiding changing anything on your computer as much as possible. For example it won't normally install any files on your PC, but run only from the CD. You can give the operating system a try to see if you like it without the risk of installing anything on your hard drive. You should remember that linux typically runs very fast - if the system seems slow, it is because it is running off your CD drive, not your hard drive.

Choosing A Distro

There are dozens of different Linux distributions. Here are some ways to help you narrow down the options to a short list.

How do you intend to use the system?

: Desktop or server? This distinction is probably the most important. Distributions for the desktop will have a graphical user interface, while server distributions won't.

Specific hardware requirements

: Try out a few LiveCDs of different distributions. Does it recognize and work properly with your hardware?

: If you intend to install Linux on a low-end specification computer, or you have other peculiar hardware compatibility problems or requirements, your choice might be influenced by this need. Most linux distributions should run fine on all but the lowest end of the spectrum.

Application support

: Which applications or desktop environment are important to you?

: Does a given distribution install those programs by default or is it easy to install and integrate them with the rest or your system?

: Does the distribution have a good package management system, and suitable software repositories?

Support

: What options will be available for getting support? Is commercial (paid) support available? Is there free community support? If the distribution has a small user base, you will have a harder time getting distribution-specific support, as compared to a more widely-used distribution.

Desktop environment

: For desktop systems, you'll need to feel at home. Check out ../Desktop environments for information on some common ones. GNOME and KDE are the two most popular.

Try a distro chooser, like http://www.zegeniestudios.net/ldc/. Often several different distributions will meet all your requirements. Your final choice from the short list may be based on whim or personal taste.

External links

- GNU/Linux pre-installation checklist [1]
- Picking a distribution [2]

Selected distributions

Main page: ../Distros in detail

Arch [3]

A minimalistic and lightweight distribution that caters to the experienced Linux user.

Debian [4]

Contains a *lot* of packages

Easy to use and update

Very stable

Fedora [5]

Free version of Red Hat Linux.

Gentoo [6]

Offers four levels of installation, very time consuming, but easy to upgrade.

Used mainly by advanced users.

Linux From Scratch [7]

This page details how to build a Linux system without relying on any particular distribution. **For experts only.**

Linux Mint [8]

Makes Ubuntu easier with improved hardware detection, scripts, and multimedia integration.

Mandriva [9]

Formerly known as Mandrake.

Very easy to install and use - targets newcomers to linux on the desktop.

MEPIS [10]

Debian-based distribution with superb hardware support and a rich Live CD environment.

Comes in two flavors: SimplyMEPIS for the regular desktop users and ProMEPIS for use in a more commercial environment.

openSUSE [11]

Distribution designed mostly for business, but very user friendly and easy to pick up and use.

PCLinuxOS [12]

Very popular for beginners.

Red Hat [13]

Commercial distribution (but see centOS [14]) designed mostly for business, yet also usable for individuals.

The free version is Fedora (above)

Sabayon [15]

Gentoo Linux based distribution that focuses on newest technologies.

Slackware Linux [16]

One of the first Linux distributions.

Easy installation, but a bit of knowledge would be useful.

TaraElla Computer System [17]

The TaraElla Computer System (TECS) is a system demonstrating a possible way to use the computer in accordance with the TaraElla Computer Lifestyle principles. It is made by pop singer songwriter TaraElla. As it is made by a member of the entertainment industry, it is considered very non-geeky.

Ubuntu [18]

The most successful and popular desktop distro based on Debian.

Uses the Unity user interface and has frequent updates.

Very easy to use and install.

Ubuntu Linux has different variations:

- Kubuntu: Uses KDE instead of Unity.
- Xubuntu: A light-weight version of Ubuntu that uses Xfce instead of Unity. This version is useful for less powerful PCs and for people who want a simple interface.
- Lubuntu: A very light-weight but still full featured that uses LXDE instead of Unity. This version is usefull for legacy PCs and for people that want a to get the most out of their hardware. Lighter than even Xubuntu.
- Edubuntu: Distribution designed for educational purposes.
- Fluxbuntu: Uses the Fluxbox Desktop Environment.
- Studio: Distribution designed for multimedia production.

Linux Mint (above) is a remastered version of Ubuntu

Baltix [19]

Very popular in Baltic countries Ubuntu based Linux distribution.

Optimized for Baltic sea region users.

Xandros [20]

Commercial distribution for anyone from businesses, to individuals.

Easy to install, and use.

External links

- DistroWatch.org [21] - Comprehensive listing of distributions, including popularity rankings.
- Live CDs [22] - Comprehensive list of 100+ live Linux distributions.
- Linux distributions [23] (Spanish)
- List of Linux Distributions [24]
- LiveDistro [25] - Free Operating System Resource

References

[1] http://tldp.org/HOWTO/Pre-Installation-Checklist/index.html
[2] http://tldp.org/LDP/intro-linux/html/sect_01_05.html#sect_01_05_03
[3] http://www.archlinux.org
[4] http://www.debian.org
[5] http://fedora.redhat.com
[6] http://www.gentoo.org/
[7] http://www.linuxfromscratch.org
[8] http://linuxmint.com/
[9] http://www.mandrivalinux.com
[10] http://www.mepis.org
[11] http://www.opensuse.org
[12] http://www.pclinuxos.com
[13] http://www.redhat.com/

[14] http://www.centos.org
[15] http://www.sabayonlinux.org
[16] http://www.slackware.org
[17] http://taraellacs.blogspot.com.au/p/blog-page.html
[18] http://www.ubuntulinux.org
[19] http://baltix.lt/
[20] http://www.xandros.com
[21] http://distrowatch.org
[22] http://www.frozentech.com/content/livecd.php
[23] http://www.argentilinux.com.ar/doku.php/linux_distribuciones:informacion
[24] http://en.wikipedia.org/wiki/Linux_distribution
[25] http://www.livedistro.org/

Linux Guide/Distros in detail

Arch

Arch Linux is a relatively young distro which offers more comfort than Slackware in that most things have reasonable defaults, the software does already work together within reasonable limits (so it's not "over-configured" like the big distros) and you can choose between binary package installation (with dependency support) and source compilations.

Debian

Debian was considered one of the harder distros to install, although with the release of 'Sarge' this has now improved due to a new installer. The installation is not too hard for the prepared user and what difficulties it may still cause is made up for by apt-get: a package management system that automatically downloads and installs or upgrades programs. It can even upgrade the entirety of Debian when it becomes necessary, so a computer running Debian will never need to be totally reinstalled for an upgrade.

Debian is popular enough that a very large number of packages are available for it — virtually every common open source program (and many libre closed-source ones) has a package made for it. However, the "stable" distribution is usually very outdated (they come out about every two years), so many home users will opt for the "testing" or "unstable" versions which come with more up-to-date software. Unstable is updated more quickly, but the official policy on it is "if it breaks, you get to keep both parts." Nonetheless, it is much more stable than the name implies.

Debian is entirely non-commercial, and only software that meets the Debian Free Software Guidelines(direct [1]) makes it to the main distribution. However, it is made easy to install non-free software with the same tools if you need it.

Debian is among the most widely ported distributions. Unlike other distrobutions which usually only run on x86 and x86_64 hardware. Debian has official ports to SPARC, alpha, powerpc, arm (big and little endian), mips (big and little endian), PA-RISC, IA-64, s390 (IBM mainframe), x86, and x86_64. There is also a community port or Motorola m68k. It can take full advantage of almost any hardware you have. Unlike other distributions, Debian also releases optimized kernels for more specific releases of processors.

Debian is known for extremely high quality releases, which are very stable.

Because of all these strengths Debian is also one of the most forked distributions. Prominent examples are Ubuntu, MEPIS [2], and Knoppix.

Some people have hardware that isn't recognized by the "easy" installers from any distribution. If you are one of those people, check out non-official Debian installers that are designed to run on the widest possible variety of computers here [3].

Debian's wiki is at http://wiki.debian.org/ [4]

Debian's main site is http://debian.org/ [5]

Gentoo

Gentoo is a source-based distro. It is far more time-consuming and difficult to install than any other major Linux distribution. This is primarily due to it lacking an installer; it merely provides "stage tarballs" and boot CDs. The CDs provide a basic environment for bootstrapping the system; they include packages necessary for install, drivers, a shell, and not much else. There is no graphical installer, nor even a text based one, despite several plans for creating one; the command line is used for the whole install.

Gentoo is not a distribution for any but the most technically oriented - not those necessarily who are already familiar with GNU/Linux (also called Linux), but for those who want to learn more about how a Linux system works. It is a highly flexible system, which can be tuned to every user's individual needs; this comes at the price of being somewhat more hands-on. There is far less autodetection than in other major distributions. Furthermore, most programs must be compiled from their source code. Fortunately, Gentoo's documentation site [6] contains comprehensive documentation on an expanding number of topics. It has a very good installation guide, which is now the "Gentoo Handbook", which teaches new users several things about the structure of Gentoo and how it works. There is also a Gentoo Wiki [7]

The installation process is long, especially if source packages are used. Binary packages are possible, and can be provided either from GRP or custom-made, but are not frequently used. Installation is an involved process, but not especially difficult to those who carefully follow the installation guide; help is provided in numerous ways to those who have trouble, whether the cause is unfamiliarity with the system, custom needs, or shortcomings in the install process. The installation guide provides highly detailed examples, complete with sample commands which often work unchanged, and include directions of how and what to change for highly variable details such as which hard disk partition to use.

Gentoo has a steeper learning curve than other distributions of Linux. This is both a cost of the flexibility, combined with the fact that Gentoo does not have a long history (so sometimes suffers from not entirely automating what should be automated). This is mitigated somewhat by the documentation and huge amount of community support, including very large forums. [8] Many questions on the forum get answered very rapidly. The reason for the large forums is that just about everyone who uses it is in love with it, largely due to its advanced package system, "portage".

Portage is source-based, and also supports quarterly binaries for those who would rather not do compilations. The package system is inspired by BSD's "ports" system, and shares some similarities to Debian's package system, such as automatic dependency resolution (which means, in short, that anything you need to compile/install a package will be automatically installed when you ask the system, with "emerge name-of-package", to do so; this contrasts with the historical frustrations presented to users of package systems where each dependency must be manually found and installed first. How to install each package is described by a simple machine-readable text file, called an "ebuild". Ebuilds are conceptually simple, but can become quite complex, and provide Gentoo-specific configuration for packages; they also frequently apply patches which increase functionality or fix bugs. Major packages (which have the best-maintained ebuilds) are consequently easy to upgrade and most start with a working configuration; further details on what a user may want to do are shown after a package is installed. There are around 7000 packages available in portage, and on average 50 new packages (including upgrades) are added daily. The newest updates can be seen on the "Fresh Ebuilds" [9] site.

Gentoo is most likely to be appreciated by people with fast computers and Internet connections, due to the bandwidth and processor-intensive nature of its package system. Those with unusual needs may also appreciate Gentoo; as Linux changes rapidly, and Gentoo is highly configurable, it can make some tasks easier than other distributions do. A typical Gentoo user wants to know exactly what makes up their system and appreciates major configurability.

Gentoo systems are easy to keep very up to date; unfortunately, some stability is often sacrificed, especially in the "~arch" [similar to Debian's "unstable"] branches.

Gentoo is least likely to be appreciated by those with slower computers and Internet connections, unless they are highly patient; if an advanced package manager is wanted, these users may prefer BSD or Debian. Users who want a Linux system to just work with no manual configuration are more likely to at least initially appreciate a distribution such as Lycoris, Xandros, or perhaps Mandrake or Fedora Core.

Red Hat & Fedora

Red Hat gears the majority of its production toward corporate clients. However, a frequently updated consumer edition, Fedora, is offered. It is a popular distribution built by a community under the central direction of Red Hat, and its code is the basis from which Red Hat's enterprise software is developed. While Fedora is easy to use and has very good graphical utilities (including an installer), it's still a flexible and powerful distribution, capable of everything from laptop computing to a full-fledged multi-use server to a thousand-node cluster, and beyond. The package manager, RPM, does not automatically handle dependencies, however it is quite popular and many Linux applications are available as pre-compiled RPM packages; also, users may opt to use **yum** or **PackageKit**, both included with Fedora, to install packages instead. They may install the alternative installers **apt-rpm** and **synaptic**. Ten versions of this distro have been developed among which Fedora 11 is the latest available at the official website. Previous versions were Fedora Core 1, 2, 3, 4, 5, and 6, and Fedora 7, 8, 9 and 10.

official website [10]

Mandriva

Mandriva was originally known as Mandrake. It was originally based on Red Hat; it is now a distribution in its own right and is designed for beginners and experts. Mandrake's installation program has the ability to resize NTFS partitions, meaning it is a good choice if you want to keep your existing Windows XP or Windows 2000 installation but still install Linux (however, Knoppix can be used to resize NTFS partitions before an installation no matter what distro is being installed).

Mandriva uses the **urpmi** package manager, an equivalent of apt-get for rpm packages. Urpmi has both graphical (see screenshot here: [11]) and text-based front-ends. With the graphical front-end it is very easy to manage all installed and available software.

The live-version of Mandriva is MandrivaOne, which runs KDE or GNOME, has a one-click installation method. The installed Linux is the Official Mandriva Linux 2007, which makes for easy update/upgrade/program installations.

Puppy linux

"Live-CD", installable, Recordable DVD

Q: Is it true that this can run on a PC that doesn't even have a hard drive?

A: Yes - true. It will run on older hardware. It will run from USB keydrive. It works like Windows. You can run it from Windows [12] You could even remove the CD ROM and floppy and have a machine with no hard disk no CD ROM even. It also can be run from a CD-RW. When you finish a session it writes any files to the CD-RW or DVD-RW

Puppy Linux can also be installed to a hard drive in the traditional way and the "install it to files on a vfat partition" way.

Sabayon

Sabayon is a binary Gentoo distro. This is the recommended distribution for those who want to learn Linux or use bleeding edge software. A typical DVD installation 1hr with an 8x DVD drive. Gentoo's the distribution with the sharpest learning curve because it requires knowledge of the hardware and compiling the kernel first thing. Sabayon is just the opposite of Gentoo except that it's still Gentoo, which is designed for optimizing your applications by compiling them from source. Sabayon comes with a kernel configured for all kinds of hardware along with precompiled binaries. The Gentoo Portage package manager is still there for optimizing your favourite applications and giving them a boost.

Slackware

Slackware tends to be favored by the more hardcore Unix fans. The entire operating system is based around tarballs and source installs. It definitely is a Linux distribution which requires the user to have a solid knowledge of the filesystem and its operations. It is not typically recommended for first time users, and should be installed by people with a desire to know the deeper and more complicated problems that arise within the Linux world. The install set typically is 4 cd's with 2 of them being a store for source tarballs and other packages. The opinion of some is that Slackware tends to be a more rock solid Distro, with less security vulnerabilities. Some first-timers consider it to be easy to install, but this is usually because they have worked with other UNIX-like OS's before. Installing it is quite simple, however configuring it to your liking is much more difficult. Slackware likes to have manual controls, so the user may find oneself using the terminal more often, needing to manually mount devices, etc. A large and growing number of Linux users have abandoned Slackware for newer, more advanced and overall less needy distributions.

Slackware does have support for other package modules, mainly RPMs. However, this support is limited, and without support generally. It's better suited to install applications by source compiling. Slackware is a good choice, but a user should probably be well versed in Linux prior to giving it a shot, it's not a bad learning distro, just a hard distro to learn.

SuSE

SuSE is mostly popular in Europe and was acquired by Novell in November of 2003. SuSE focuses, like Mandriva, on an easy installation procedure and graphical administration tools. A graphical installation program guides you through the individual steps necessary to install Linux.

The SuSE Linux distribution is updated regularly and was chosen by many cities in Germany like Munich, to convert their desktop and server computers to Linux.

SuSE has possibly the best installer software of any Linux distribution, called Yast [13] , which they recently (mid 2005) released as GPL (open source).

SuSE is a stable and easy to install Linux distro. And is by default LSB(Linux standard base) compliant.

Also SuSE like most other distributions is available on the form of downloadable ISO files (which you can write to CDs or a DVD). For this purpose you can use this link http://www.opensuse.com

It is usually available the same day as it is released in the stores. If you want installation support you must pay a one time fee for 90 days support. Earlier, before the openSuSE project you would have to wait for 3 months before the ISO of the latest version was available.

Ubuntu & derivatives

Ubuntu is a desktop Linux distribution. It is based on Debian and copies over many Debian packages.(albeit many outdated) Ubuntu is sponsored by Canonical Ltd, however all releases are free of charge. The name of the distribution means "humanity towards others". New versions are released every 6 months, and support (bug fixes) is provided for 18 months after the release. Ubuntu is developed for x86 and x86_64 systems.

Ubuntu uses the GNOME desktop environment as default. A version using the KDE desktop environment is available under the name Kubuntu.

Even more

- A list of over 300 Linux distributions: [14]
- DistroWatch.org [21]

References

[1] http://www.debian.org/social_contract#guidelines
[2] http://mepis.org/
[3] http://www.linuxmafia.com/faq/Debian/installers.html
[4] http://wiki.debian.org/
[5] http://debian.org/
[6] http://gentoo.org/doc/en
[7] http://gentoo-wiki.com/Main_Page
[8] http://forums.gentoo.org
[9] http://packages.gentoo.org/
[10] http://fedoraproject.org/
[11] http://images.mandriva.com/mdksoft/92/discovery/rpmdrake-discovery.png
[12] http://www.freeveda.org/linux/puppy/index.html
[13] http://forge.novell.com/modules/xfmod/project/?yast
[14] http://lwn.net/Distributions/

Linux Guide/Getting Linux

Use an Existing System

The easiest way to begin using a Linux system may be to use an already running system. For example, some systems may be available in various educational or work environments. In this case you only need to contact the appropriate administrator and obtain a user ID and password. You will only be able to explore the system to a certain extent, but the first steps can be taken without having go through the installation process.

Another trouble-free way of obtaining a Linux system is to buy a computer with Linux pre-installed. The number of vendors selling such systems is constantly increasing.

Download linux iso

After choosing a distribution, you should download it. Normally this will be an ISO image. An iso is all the data on a CD - after downloading the iso file, you will burn it to disk so you can use it. Whatever distribution you chose will likely have specific instructions for how to burn the CD.

Boot from Live CD

Live CD distributions allow you to "get your feet wet" by running Linux on your own PC at home without worrying about installing it or losing data.

Live CDs don't need to be installed to your hard drive for you to use them. Instead, you simply load the Live CD into your CD-ROM drive, restart your computer and a complete running Linux system should boot up with little, if any, intervention. There are some difficulties for some distributions working with certain sets of hardware, but most systems will boot with little to no problems.

A system running a Live CD often tends to be a little slow. This is because information must be fetched from the CD-ROM (which is much slower than a hard drive) and because the Live CD must store a lot of information in memory that would normally stay on the hard disk. Don't let this fool you into thinking that Linux is a slow operating system though. Linux systems are normally very fast and reliable. If you try a Live CD, it is recommended (unless you are trying a MiniLinux [1]) that the computer you use be a relatively recent one with a generous amount of memory (256MB or more). For most people, this should provide a trouble free way to begin to get to know Linux. Sometimes the entire Live CD can be loaded into RAM if a sufficient amount is available (say around 1GB) and this will offer excellent responsiveness.

Live CDs are a great way to test whether certain hardware is likely to be compatible with Linux. Just pop the disk into the CD-ROM drive of the system in question and reboot as described above. Most hardware problems (if there are any) should make themselves obvious during normal use.

If the Live CD didn't work

If your Live CD is ignored and you find yourself booting into your normal operating system, you may need to alter some settings in your BIOS. While this may sound daunting for some users, it's actually a lot easier than it sounds.

Begin by restarting your PC and when the boot process begins again you will need to strike a particular key on your keyboard. The key you need to strike will usually be displayed on the computer screen and is often the 'Del' key. Common alternatives are 'F1', 'F2', 'F10' and 'Esc'. The proper key will vary with your computers manufacturer. Once the proper key is struck your computer screen will display the 'BIOS' or 'Setup' editor. There are usually some simple instructions on the bottom of the screen, or off to one side, telling you how to navigate around the various choices and make changes. Be sure to read these before continuing, then look for something labeled 'Boot' or 'Boot order'.

Using the instructions you read a moment ago, change this so that your CD-ROM is the first item in the boot order. If you think you've made a mistake, there is usually an option to 'Exit without saving changes'. If you don't see that option, holding down the 'Ctrl' and 'Alt' keys and pressing the 'Del' key should save the day. If all goes well 'Save and Exit' and your computer should boot up, checking the CD-ROM for your Live CD as it does so. If you have run a Linux Live CD and have rebooted to use your regular operating system and have arrived at a notice declaring "Missing Operating System" or "Error on System Disk", or any scary notice stopping you from loading up as normal, it is likely that the computer has "forgotten" that you normally load up from the hard drive and has started to look for your operating system in the CD Drive. To fix this, follow the instructions above accordingly to your system but instead of selecting your CD drive as the boot device, select your hard drive.

Install Linux

After testing the system, you'll want to install it to your hard drive permanently. This does not require you to get rid of the current operating system, even if you have only a single hard disk. We will explore this topic in more detail in the next chapter, Installation Walkthrough.

References

[1] http://en.wikipedia.org/wiki/Minilinux

Linux Guide/Installing (basic)

Here, we will cover installing an Ubuntu system to your hard drive. Most other distributions use a very similar installation process. If the differences between this documentation and your distro are too great, check the documentation provided for your distro.

Ubuntu Linux has quickly become the golden boy of open source and is considered by some as being the first truly easy-to-use Linux operating system. This is largely because of the installation process which allows the user to run the operating system prior to installation.

Boot the LiveCD

To start an install, place the Ubuntu CD ROM (available for free via postage from shipit.ubuntu.com [1]) in your computer and reboot.

When the computer starts again, it should display a screen with the Ubuntu logo displayed on it and a list of options. If your computer does not do so, you may need set the BIOS to boot from CD.

Select the first option in the menu to boot into an Ubuntu desktop session. If you wish, you can get to know the Ubuntu operating system at this point. Using it in roughly the same manner that you would when fully installed (though it is advised not to log out).

Once you feel prepared to start installation, click on the "install" icon on the desktop.

Begin installation

Answer the questions on the first few screens of the installation window (language, location, etc.).

(Re-)Partitioning

For dual-boot systems, see ../Installing (dual-boot).

For advanced partitioning schemes, see ../Installing (advanced).

When you get to a screen asking which hard disk you wish to install to, select automatic partitioning unless you have planned in advance not to use this one. **This will erase everything on your hard drive!**

If you want to use one hard drive for multiple operating systems (for example, Windows and Linux), see ../Installing (dual-boot).

If you want to use multiple partitions for your linux operating system (it is recommended to use separate partitions for `/ and /home at least), see ../Installing (advanced).`

Set up your user account

Continue to the next screen and then fill in your name and desired password. This will become your administrative account. If you do not want to use such on a daily basis, fill in your name as master user or such and place an appropriate account name in its text box. Using this account, you can create further accounts on the computer (note that the name Administrator is not available).

Once you have done this, click on Next, check the details are correct and click Next again to start the actual installation process which will likely take around 20 to 30 minutes at the most.

After this a prompt will appear saying that you should reboot. After doing so, Ubuntu is ready to be used.

External links

- Linux Installation and Getting Started [2]
- The Linux Installation HOWTO [3]

References

[1] https://shipit.ubuntu.com/
[2] http://tldp.org/LDP/gs/gs.html
[3] http://www.tldp.org/HOWTO/Installation-HOWTO/

Linux Guide/Installing (dual-boot)

Dual-boot systems - Run Linux and MS Windows

You need to run both MS Windows and Linux on your PC and you have only one hard disk. Follow these steps:

Windows needs the first partition on systems where grub is being used as a bootloader. (Note that this requirement does not apply to systems that use lilo). It is strongly advised to put windows in the first partition (/dev/hda1 or /dev/sda1) otherwise you will have to do hide and unhide of partitions during grub configuration. If you have Linux on first partition then you have to move it. Golden rule: Put windows in first partition and Linux can run easily off any other partition

Notation: If using SCSI or SATA drives use 'sda' and for IDE drives use 'hda'

1. Download the partition tool LiveGParted from http://gparted.sourceforge.net/livecd.php and boot the LiveGparted CD-ROM
2. Resize the MS Windows partition and then create new partition(s) for Linux.
3. If Windows is NOT in first partition and if Linux is already existing on first partition then move the Linux from partition 1 to another partition like 2.

```
Open X-terminal console and
      mkdir /hda1
      mkdir /new
      mount /dev/hda1 /hda1
      mount /dev/hda2 /new
      cd /hda1
   Use tar and not cp -r.
      tar cpf - . | tar xpvf - -C /new
   Edit /new/etc/fstab and /new/grub/menu.lst as appropriate
```

1. Create primary partition 1 FAT32 for MS WindowsXP, Windows wants the first disk.
2. If you do not have Windows already in partition 1, install Windows XP on partition 1 (format to NTFS)
3. Boot Linux CD-ROM and install the grub and in menu.lst you do not need to do unhide and hide since MS-Windows is in first partition.

Troubleshooting: Grub not finding file or disk or partition. If the disk is not recognised inside the grub, then that partition is marked as "hidden". Also when you do 'find /boot/grub/stage1' it does not find the file. So you do this inside grub: Note: hd0 is /dev/hda1 and partition 1 is 0 in grub commands.

```
grub> unhide (hd0,0)
grub> find /boot/grub/stage1
(Now this will succeed and show some output)
```

Dual-boot references:

- GParted LiveCD http://gparted.sourceforge.net/livecd.php
- GParted Wiki http://en.wikipedia.org/wiki/GParted
- GRUB page: http://www.gnu.org/software/grub/manual/
- Grub Howto: http://tldp.org/HOWTO/Linux+Win9x+Grub-HOWTO/index.html
- Google Grub: Grub Topics [1]
- Wiki Grub: http://en.wikipedia.org/wiki/GNU_GRUB
- Wiki Lilo: http://en.wikipedia.org/wiki/LILO_(boot_loader)
- Virtualization: http://en.wikipedia.org/wiki/Virtualization

References

[1] http://www.google.com/search?hl=en&q=grub+howto&btnG=Google+Search

Linux Guide/Installing (advanced)

There are 4 ways of "installing" Linux:

1. "no-install": a "Run-from-CD" LiveCD distribution. Everything runs from the CD and RAM, leaving the hard drive untouched -- completely avoiding the trickiest part of setting up Linux, "hard drive partitioning". After you take out the CD and reboot, everything is restored to the way it was before. (Many of these work just as well with USB flash drive as with CD)
2. Linux on top of Windows: it's possible to keep windows applications running, and run Linux like any other application side-by-side with the Windows applications (this is an advantage compared to dual-booting, which shows either only Linux applications, or only Windows applications, and you have to reboot to get to the other one). andLinux, Portable Ubuntu, coLinux, Topologilinux, etc.
3. There's a lot of information on setting up "Dual-boot" systems. This is not for beginners. (Experts generally don't do this either -- if they need 2 operating systems, they keep them on 2 physically separate hard drives, or better yet 2 physically separate computers).
4. If you are certain there is nothing you want to keep on a computer's hard drive, you can completely erase the HD and install Linux only. This is the most fool-proof system -- Once the hard drive has been erased, the worst that can happen is erasing the hard drive again and having to start all over.
5. This is rare now, but some distributions allow installation to any FAT32 partition even if it contains an installation of Windows. This just copies files containing the Linux and uses a Floppy or a Boot Loader to load. Wubi, etc.

Dual-booting or saving data

Preliminary note

You most likely do not need to do this as it is integrated into the installers

Generally installers have excellent partitioning tools so this is all part of the install. If not, or if you are not installing it is easiest to use a live CD distro that has GParted on it, such as ubuntu. GParted does not support LVM volumes used on more complicated partitioning setups, however fedoras installer for example and debians can install and partition LVM volumes.

Hard way

To install Linux without erasing the contents of the hard drive requires a spare partition. One solution is to install an extra hard drive. However, one can also non-destructively resize an existing partition. A FAT-type volume can be resized with FIPS and an NTFS volume with ntfsresize [1]. If using FIPS, the hard drive will need to be defragmented before the resizing, but defragmentation is a good idea no matter what the file system.

What size?

You would be hard-pressed to fit a normal desktop Linux system in under 2GB. It's advisable to give Linux more space than that, however, because many, many applications are available for free for Linux, and especially with a high-speed Internet connection, one is likely to install quite a lot of them.

You should plan on leaving at least 20% of each of your hard drive partitions free at all times -- modern file systems (such as NTFS, ext3 and ReiserFS) try to keep fragmentation low on their own, but they need extra space to do it

with.

Manual resizing

Easiest way

(NOTE: Most of this only applies if there is already a version of Microsoft Windows on the computer) The volume resizing is a safe process, but afterwards the hard drive must be repartitioned. An error here can destroy the data on the hard drive, so double-check all commands. A typical session with ntfsresize ("/dev/hda1" is the most likely name for the NTFS partition. In this case, it is 10 GB in size.):

```
paul@faust:/$ su
Password:
```

*You must be root to run ntfsresize. Under Knoppix {*what about other live CDs? *Under Gentoo's LiveCD, you are already root, so the* su *will not prompt for a password. Anyone know about others? I'd assume no-password or already-root would cover most of them...}, you will not be asked for a password unless you had already set one.*

```
faust:/# umount /dev/hda1
```

This step is only needed if hda1 is already mounted, which is unlikely. However, if it is not needed, it will only give an error message and not do anything.

```
faust:/# ntfsresize -i /dev/hda1
ntfsresize v1.9.0
NTFS volume version: 3.1
Cluster size       : 4096 bytes
Current volume size: 10999992832 bytes (11000 MB)
Current device size: 11013617664 bytes (11014 MB)
Checking filesystem consistency ...
100.00 percent completed
Accounting clusters ...
Space in use       : 4197 MB (38.2%)
Estimating smallest shrunken size supported ...
File feature         Last used at      By inode
$MFT               :      8223 MB             0
Multi-Record       :      3160 MB         14852
You might resize at 4196970496 bytes or 4197 MB (freeing 6803 MB).
Please make a test run using both the -n and -s options before real resizing!
```

ntfsresize, version 1.90 and later, will automatically move files (including ones that the Windows defragmenter can't) in order to resize the partition, so defragmentation is not necessary before resizing (it might be easier to defragment while the partition is larger, though, so its a good opportunity).

The next step does a test run of the resizing process. Here, the user decided to leave Windows with about 6GB of space.

```
faust:/# ntfsresize -n -s 6000M /dev/hda1
ntfsresize v1.9.0
NTFS volume version: 3.1
Cluster size       : 4096 bytes
Current volume size: 10999992832 bytes (11000 MB)
Current device size: 11013617664 bytes (11014 MB)
```

```
New volume size    : 5999993344 bytes (6000 MB)
Checking filesystem consistency ...
100.00 percent completed
Accounting clusters ...
Space in use       : 4197 MB (38.2%)
Needed relocations : 251614 (1031 MB)
Schedule chkdsk for NTFS consistency check at Windows boot time ...
Resetting $LogFile ... (this might take a while)
Relocating needed data ...
100.00 percent completed
Updating $BadClust file ...
Updating $Bitmap file ...
Updating Boot record ...
The read-only test run ended successfully.
```

Now, write down the exact number of megabytes passed on the command line here (in this case, 6000). (The file system will probably not be resized to the exact size you specify, but ignore that — use what you entered.) You will need to resize the partition to the same size later.

Run ntfsresize with the same parameters, except leaving out the `-n`. This resizes the ntfs filesystem, but not the partition.

Splitting the partition is the only remaining task. Be sure to give it the hard drive (like "/dev/hda") as a parameter, rather than the partition ("/dev/hda1").

```
faust:/# cfdisk /dev/hda
```

If you don't have cfdisk, use fdisk instead. It has a less friendly interface, but it has exactly the same commands. Either way, no changes are made to the disk until you tell the program to (w)rite out the new partition table. If you accidentally write out the wrong partition structure, you should be able to save your data by replacing it with the correct one before doing anything to the affected partitions.

You need to

- delete the current NTFS partition (this does not destroy the data on it)
- (N.B. In the highly unlikely event that there was unpartitioned space before the NTFS partition, you need to create a partition filling it up. *This is false -- you can create a partition beginning anywhere on the disk. You just need to write down what the starting cylinder was.*)
- create a new partition with the same size you passed to ntfsresize earlier at the beginning of the unpartitioned space that was created.
- leave the remainder as free space. It will be filled with Linux's partitions, but some decisions need to be made first.

Partition recommendations

- `/`:
  - 5 GB
- `/boot`:
  - Minimum 3 MB per kernel, recommended 100 MB.
  - It's recommended if you have multiple HDDs, most BIOSes don't like loading kernels from disk different than the one on which bootloader resides on.
- swap partition or file = twice RAM size (Rule of thumb)
  - Using old linux kernels, swap partitions were faster than swap files; that's no longer the case.
- `/home`
  - Whatever remains
  - This is where the bulk of your data will reside, you will want *lots* of room
  - It is a very good idea to avoid having `/` and `/home` on one partition, if you're into that sort of thing, you know.

Note: If you're an advanced user and want to have few Linux systems installed side-by-side, or different file systems for experimenting, configuring Logical Volume Manager may be a good idea.

More:

- The Linux Installation HOWTO: Creating partitions for Linux [2]
- http://en.tldp.org/HOWTO/Partition/index.html
- http://www.gnu.org/software/parted/
- All About Linux Installation [3]

References

[1] http://mlf.linux.rulez.org/mlf/ezaz/ntfsresize.html
[2] http://www.tldp.org/HOWTO/Installation-HOWTO/details.html#AEN365
[3] http://www.editorial.co.in/linux/installing_linux.php

Linux Guide/Logging In

Linux supports multiple users on a machine, even at the same time. Each user must log in with his or her user name and password. Every user belongs to one or more groups. Users and groups are important in determining permissions. All the users are listed in /etc/passwd and all the groups are listed in /etc/groups. To find out your user name and groups you belong to, run the id command

```
$ id
uid=501(myname) gid=501(mygroup) groups=501(mygroup),502(anothergroup)
```

The numbers are user and group *id*s, which are numbers that are associated with the user and group in /etc/passwd and /etc/group, respectively. These numbers are what the system uses internally, so you don't have to look at them, but you can use them in place of user and group names in certain commands (like chown). The human-readable names are shown in parentheses.

You can change your group to any group in *groups* or to a group that has a group password by using the newgrp command. You can also change to a different user entirely without logging out and in again by using the su command. Changing your user and/or group not only changes what files you can access, but the user and group of any files you create.

Both of these commands put you in a new shell, and you have to exit the shell or logout to stop using the new group or user. If you just need to be user root to run one command (which is rather common on a UNIX machine, for doing administrative tasks), the sudo command is what you want.

Users and groups can be managed from an administrative GUI or from a shell with the commands

```
useradd
usermod
userdel
groupadd
groupmod
groupdel
```

which add, modify, or delete a user or group. You will have to be user *root* to run these commands.

Linux Guide/How Linux Works

The Linux Philosophy

Linux is built with a certain set of unifying principles in mind. Understanding these principles is very helpful in understanding how the system works as a whole. They are known as the "Linux Way", which is derived from the philosophy behind the UNIX system.

The Linux Way can be summarized as:

- Use programs that do only one task, but do it well.
- To accomplish complex tasks, use several programs linked together.
- Store information in human-readable plain text files whenever it is possible.
- There is no "one true way" to do anything.
- Prefer commandline tools over graphical tools.

Most traits of Linux are a consequence of these principles. In accordance with them, a Linux system is built out of small, replaceable components. We will examine the most important of them in more detail. Those are: the boot loader, the kernel, the shell, the X window server, the window manager and the desktop environment. After that, we will have a look at the file system in Linux. Finally, we will discuss the security of a computer running Linux.

Core components of a Linux system

Boot loader

This is the part of the system that is executed first. When you have only one operating system installed, it simply loads the kernel. If you happen to have multiple operating systems or multiple versions of the Linux kernel installed, it allows you to choose which one you want to start. The most popular bootloaders are GRUB (GRand Unified Bootloader) and LILO (LInux LOader). Most users don't need to care about the boot loader, because it is installed and configured automatically.

Kernel

The kernel is the central component of the system that communicates directly with the hardware. In fact, the name "Linux" properly refers to a particular kind of this piece of software. It allows programs to ignore the differences between various computers. The kernel allocates system resources like memory, processor time, hard disk space and external devices to the programs running on the computer. It separates each program from the others, so that when one of them encounters an error, others are not affected. Most users don't need to worry about the kernel in day-to-day use, but certain software or hardware will require or perform better with certain kernel versions.

Shell

The shell is a textual interface that allows you to run programs and control the system by entering commands from the keyboard. Without a shell (or something that can replace it, like a desktop environment) it would be hard to make your system actually do something. The shell is just a program - there are several different shells for Linux, each of them offering different features. Most Linux systems use the Bourne Again Shell (Bash). Linux shells support multitasking (running several programs at once).

X Window Server

The X window server is a graphical replacement of the command shell. It is responsible for drawing graphics and processing input from the keyboard, mouse, tablets and other devices. The X server is network transparent, that is, it allows you to work in a graphical environment both on your own computer as well as on a remote computer that you connect to across a network. The X server that is most used today is X.Org. Most graphical programs will only need the X server to run, so you can use them under any window manager and desktop environment.

Window Manager

The window manager is a program that communicates with the X server. Its task is managing windows. It is responsible for drawing the window borders, bringing a window to the front when you click it, moving it on the screen and hiding them when you minimize a program. Examples of popular window managers are:

- Metacity - GNOME Desktop Environment window manager
- KWin - KDE window manager
- Xfwm - Xfce window manager, a lightweight manager designed to consume as little resources as possible without compromising usability
- Compiz Fusion - an advanced window manager with lots of eye candy like customizable window animations, multiple desktops placed on a cube that you can rotate with your mouse, transparent window borders, wobbling windows while dragging them, etc.

Desktop Environment

Desktop environments, like GNOME Desktop Environment, KDE and Xfce, are collections of programs designed to present a consistent user interface for most common tasks. This is the thing most people think about when they say "operating system" even though it is only a piece of the larger operating system. Replacing the desktop environment is a rather complex task, so most new users should stay with the default environment offered by their distribution.

File System

There are several file systems that Linux-based distributions use. They are BTRFS, EXT3/4, VFS, NILFS, and SquashFS.

The hard drive of your computer has a rather simple interface. It only accepts commands like "read block no. 550923 and put it in memory address 0x0021A400". Let's assume you are editing a piece of text and want to save it on the disk. Using block numbers (addresses) to identify pieces of data, like your text, is rather awkward. You would have to tell the program to save the file i.e. in blocks from 239012 to 239088. You would also have to remember what data you have stored where, so that you won't overwrite your family photo, your music collection or even your system's kernel. To solve this problem, the concept of a file was introduced - that is, a named area of disk storing some data. To organise files, they are collected in directories. Directories can contain other directories, which gives us a tree-like structure. Each file can be uniquely identified by a path, which describes its place in the directory hierarchy. For the remainder of this section, it will be assumed that you are familiar with files, directories and paths.

In Linux, the top-level directory is called the root directory and denoted with a forward slash: "/". Every file and directory in the system must be a descendant of the root directory (it is common to talk about directories using the terminology of family relations, like "parent", "child", "descendant", "ancestor", "sibling", etc.). Directories are separated with "/". Names of directories and files can contain all characters except "/" and the null character (you don't need to care about it, because it is impossible to enter from the keyboard). An example path would be:

/var/logs/apache/error.log

This path refers to a file called "error.log" which is found in a directory called "apache", which is a subdirectory of a directory called "logs", which is subdirectory of a directory called "var", which is a subdirectory of the root directory

(*/*).

The root directory usually contains only a small number of subdirectories. The most important are:

- **bin** - programs needed to perform basic tasks, i.e. change a directory or copy a file
- **dev** - special files that represent hardware devices
- **etc** - configuration files
- **home** - contains private directories of users
- **media** or **mnt** - Mount point [1] for external drives connected to this computer, i.e. CDs or USB keys
- **tmp** - temporary files
- **usr** - programs installed on the computer
- **var** - variable data produced by programs, like error logs

Devices as files

Just as files can be written to and read, devices in the computer system may send and receive data. Because of this, Linux represents the devices connected to the system as files in the **/dev** directory. These files can not be renamed or moved (they are not stored on any disk). This approach greatly simplifies application programming. If you want to send something to another computer through a serial port, you don't even need another program - you simply write to the file **/dev/ttyS0**, which represents a serial port. In the same manner the file representing the sound card (**/dev/dsp**) can be read to capture the sound from an attached microphone, or written to in order to produce sound through the speakers.

Where are the drive letters?

If you have used Windows, you might be surprised that there are no drive letters in Linux. The root directory represents the drive on which the system is installed (C: in Windows). Other drives can be "mounted" or "unmounted" in any directory (preferably, an empty one) in the file system. By mounting a disk, you attach the root directory of this disk to a directory in the file system. After that, you can access the disk like it were a part of your system disk. For example: if you have a disk that contains a directory **text**, which in turn contains a file called **linux-intro.tex** and you mount this drive in the directory **/media/usbkey**, you can access the file **linux-intro.tex** through the path **/media/usbkey/text/linux-intro.tex**.

In most Linux distributions, USB keys and CDs are automatically mounted when they are inserted or attached, and the default mount directory is a subdirectory of **/media** or **/mnt**. For example, your first CD-ROM drive might be mounted at **/media/cdrom0**, while the contents of a USB key might be accessible through **/media/usb0**. You may manually change the mount directory, but you will have to learn two shell commands and know the device file that represents your drive to do that (the one we talked about in the preceding section - disks also get their file representation in the **/dev** directory). We will cover this subject later.

Users

The user is a metaphor for somebody or something interacting with the system. Users are identified by a user name and a password. Internally, each user has a unique number assigned, which is called a user ID, or UID for short. You only need to know your UID in some rare situations. Users can additionally be organized in groups. There is one special user in all Linux systems, which has the user name "root" and UID 0. It is also called the superuser. The superuser can do anything and is not controlled in any way by the security mechanisms. Having such a user account is very useful for administrative tasks and configuring the system. In some distributions (like Ubuntu) direct access to the root account is disabled and other mechanisms are used instead.

If you have more than one user account on a Linux system, you do not need to log out and back again to switch impersonations. There are special shell commands that allow you to access files and execute programs as other users,

provided you know their user names and passwords. Thanks to this mechanism, you can spend most of the time as a user with low-privileges and switch to a higher-privileged account only if you need to.

The advantage of running as a non-privileged user is that any mistakes you happen to make are very unlikely to damage the system. System-critical components can only be altered by the root user.

File permissions

Users exist to control the extent to which people and programs using the system can control it. This is accomplished by a system of file permissions. Each file belongs to one of the users - that is, each file has an owner. Additionally, a file can be assigned to a group of users, but the owner must be a member of that group. Each file has three kinds of permissions: **read**, **write** and **execute**. These permissions can be assigned to three kinds of owner relations: **owner**, **group** and **other**. **Other** includes all users who are not the owner of the file and do not belong to the group which owns the file. Only the file owner or the superuser (root) can change the permissions or ownership of a file.

This system allows precise control over who can do what on a given computer. Users can be prevented from modifying system files by removing the "write" permission from them, or from executing certain commands by removing the "execute" permission. Notice that users may be allowed to execute programs but not alter them. This is very important, since most Linux systems include a compiler that allows you to create your own programs.

File permissions are usually given as three octal digits (each from 0 to 7). The digits represent the permissions for, respectively, owner, group and other users. Each digit is the sum of permission codes: 1 for execute, 2 for write and 4 for read. For example, "755" allows everyone to read or execute the file, but only its owner can write it. "400" allows the owner to read the file, and no one else is allowed to do anything. "540" allows the owner to read or execute the file, group members to only read the file and other users to do nothing.

chmod (change mode), chown (change owner), and chgrp (change group) are used to change file permissions.

References

[1] http://en.wikipedia.org/wiki/Volume_Mount_Point

Linux Guide/Using the shell

Introduction

The command line interface (CLI or terminal) may seem intimidating at first, but it's important to remember that the command line is really, truly your friend. An army of tools are at your disposal that can take what would be a tedious and lengthy job (like removing the last four characters from every line of a lengthy file) and turn it into a two minute job.

For every Linux distribution the command line prompt will look a little different. For example, on one system you might see your username, the '@' symbol, the machine name, your current directory and the prompt.

```
user@linux ~/$
```

This is a very common prompt. You may also see your username, a space, the fully qualified domain name of the computer, the full path to your present working directory followed by the prompt. In the example above 'user' is the username and 'linux' is the name of the host(computer) you are logged on.

```
user linux.box.com /home/user$
```

The prompt varies from system to system based on a number of things. For example, it may be the default configuration set by the creators of your particular Linux distribution. It could also have been configured by the person who administors the computer. Maybe you, yourself, configured how you wanted your prompt to look.

The way you configure the look of your command prompt depends on what shell you use and the shell is the piece that most people commonly refer to as "the command line" when, in reality, it is simply a piece of software that provides an interface to the services of a kernel. The distinction between a 'shell' and the 'command line' is simply that a **shell** refers to a *specific piece of software* (e.g BASH, Tch, ksh, etc) that provides a **command line interface**. Most modern Linux systems use BASH (Bourne Again SHell) as the default shell.

A Little History

The commands used at the command line may seem a little cryptic due to their tendency to be very short. This is because the roots of the Linux command line are from systems where a single letter entry could take a significant amount of time to travel from a terminal, to a central server and back to the terminal where it was printed onto a roll of paper. In those old systems, the shorter the input was, the better as it meant less time waiting to issue your command and receive output. The best thing you can do to remember what commands stand for is to find out what word the command is an abbreviation for. This can go a long way to remembering the command later.

Some Useful Terminal Commands

Summary of Common Commands

- ls - This command **l**ist**s** the contents of your present working directory.
- pwd - Shows you what your **p**resent **w**orking **d**irectory is.
- cd - Lets you **c**hange **d**irectories.
- rm - **R**e**m**oves one or more files.
- rmdir - **R**e**m**ove an empty **dir**ectory.
- mkdir - **M**a**k**e a **dir**ectory.
- ps - Provides a list of currently running **p**rocesse**s**.
- cp - **C**o**p**y a file.
- mv - **M**o**v**e a file (this is also used to rename a file, "moving" it from one file name to another.)

- grep - The **g**lobal **r**egular **e**xpression **p**rint program lets you search through a file or output of another program.
- find - Find a file on the filesystem
- man - Displays the **man**ual for most commands (including 'man').

TIP

For help about a command, try *man <command>* which will bring up the manual for it. Note that some commands are built into your shell and do not have a man page, use your interpreter internal command (should be *help <command>*).

Using Common Commands

Using ls

```
user@linux ~/$ ls
Afilez  Report.doc  aFilea  finances.xls  myPic.jpg  report1.doc  vacation pictures
```

You may notice from the example, default output of 'ls' that the files are not in strict alphabetical order. This is because the output is sorted on the numeric ASCII value of each letter. Because upper case letters have a lower ASCII value than lowercase letters, the returned result will always be files starting with upper case letters first. For example, if we add the file AZTRIP to the directory we will get 'AZTRIP' as the file listed first followed by 'Afilez' because the ASCII value of 'Z' is numerically lower than the ASCII value of 'f'.

If the ASCII values of letters has you confused, the important thing to remember here is that uppercase letters trump lowercase letters and will appear first in listings. The result is that files are first sorted by case and sorted alphabetically second.

The default output of ls isn't as useful as it could be. For example, given the output in the example, we don't know what are files and what are directories. We can likely guess that "Afilez", "aFilea" and "vacation pictures" are all directories as they don't have an extension after the file name. To confirm this, we can pass the 'l' parameter to 'ls' to get "long listing format" output.

```
user@linux ~/$ ls -l
-rw-r--r-- 1 adrong mkgroup-l-d 163328 Jul 24 11:42 AZTRIP
-rw-r--r-- 1 adrong mkgroup-l-d  11264 Jul 24 11:42 Afilez
-rw-r--r-- 1 adrong mkgroup-l-d  22528 Jul 24 11:43 Report.doc
-rw-r--r-- 1 adrong mkgroup-l-d  24576 Jul 24 11:42 aFilea
-rw-r--r-- 1 adrong mkgroup-l-d 300032 Jul 24 11:43 finances.xls
-rw-r--r-- 1 adrong mkgroup-l-d  47104 Jul 24 11:41 myPic.jpg
-rw-r--r-- 1 adrong mkgroup-l-d  17920 Jul 24 11:43 report1.doc
drwxr-xr-x 1 adrong mkgroup-l-d      0 Jul 24 11:24 vacation pictures
```

The first character in the output above indicates the file type; where a dash indicates a file and the letter 'd' indicates a directory. From this output we see that the only directory is "vacation pictures"

NOTE

Most systems have an alias on the ls command to include the '--color' flag so that the command highlights different file types with different colors. Directories are commonly shown in blue. If the output of your 'ls' command is always the same color, try "ls --color".

TIP

Another common parameter (also known as an "argument") to the ls command is 'a' which shows all files including any hidden files and folders that start with a period. Arguments may also be combined together in a single block of letters. For example, "ls -l -a" can also be expressed as "ls -la".

Using cd

The cd command lets you navigate from directory to directory on your system. The first thing you may need to know when using the cd command is that, unlike windows, Linux does not use the concept of drive letters. Instead, everything is somewhere within the root directory, '/', including other disk drives.

There's a couple ways to use cd to navigate to the directory you want. These are known as "full path" and "relative path".

Full path always begins with the root directory '/'. For example, if we're in our home directory '/home/user' and we want to navigate to the 'vacation pictures' directory, we would issue the following command.

```
user@linux ~/$ cd /home/user/vacation\ pictures
user@linux vacation pictures/$
```

You may note that our prompt changed a little to indicate the directory we're currently in. This will be the case on most systems, but as previously discussed, this depends on how your shell is configured. You may also note the backslash between "vacation" and " pictures". This backslash is known as the "escape character" and tells the cd command that the space is part of the directory name. If we omitted the backslash, the cd command would error and tell us that it could not find the directory "vacation". If you don't like typing backslash or have a directory name with many spaces in it, you can use quotation marks around the path.

Relative path can start a number of ways, but always describes how to get to a specific directory from the directory you're currently in. To do this, we need to know how to move to directories that immediately surround the directory. These directories are parent directories and sub-directories

A sub-directory is any directory that is contained within the directory we're currently. Given the output below of "ls -l", we see that chicago, florida and "new york" are all sub-directories to the "vacation pictures" directory.

```
user@linux vacation pictures/$ ls -l
drwxr-xr-x 1 adrong mkgroup-l-d 0 Jul 24 12:44 chicago
drwxr-xr-x 1 adrong mkgroup-l-d 0 Jul 24 12:44 florida
drwxr-xr-x 1 adrong mkgroup-l-d 0 Jul 24 12:44 new york
```

A parent directory is the directory that contains the folder that we are in. This makes the "vacation pictures" folder the parent of the "chicago", "florida" and "new york" directories.

To cd to a sub-directory, you can start the path name with "./" which indicates the directory we're already in, or you can immediately start with the name of the sub-directory. For example, both of the following commands will change you to the "florida" directory.

```
user@linux vacation pictures/$ cd ./florida
user@linux florida/$
```

```
user@linux vacation pictures/$ cd florida
user@linux florida/$
```

To cd to a parent directory, you always specify "../". The following command will return us to the "vacation pictures" directory.

```
user@linux florida/$ cd ../
user@linux vacation pictures/$
```

If we are already in the "chicago" directory and want to cd to "new york", we must first tell the system we want to go up one level to the parent directory and then to the "new york" directory. We can do this by saying "cd ../" and then "cd new\ york" or we can do it as one step using the command below.

```
user@linux chicago/$ cd "../new york"
user@linux new york/$
```

NOTE

Notice how we used quotation marks around the path to cd to the "new york" directory. We could have, instead used the backslash before the space and omitted the quotes to get the same result.

Similar to "./" and "../", there's also "~/" which indicates the home directory of the user we're logged in as. If we wanted to cd to the "vacation pictures" folder while in the "/usr/lib" directory, we could do this with any of the following commands.

```
user@linux lib/$ cd ~/vacation\ pictures
user@linux lib/$ cd "/home/user/vacation pictures"
user@linux lib/$ cd ../../home/user/vacation\ pictures
```

TIP

Using the cd command without specifying a directory will return you to your home directory.

File manipulation

Copy directory options:

```
$cp -frvp /source /destination
 ^   ^^^^    ^        ^
 |   ||||    |        \----------- Path from root is best
 |   ||||    \-------------------- Path to copy from root best
 |   ||||
 |   ||||                         OPTION FLAGS----------------
 |   |||\------------------------- Keep permissions
 |   ||\-------------------------- Verbose
 |   |\--------------------------- Recursive files and folders
 |   \---------------------------- Force
 |                                ----------------------------
 \------------------------------- Copy command
```

```
cp exam test
-copy file named "exam" to directory named "test"
```

```
cp exam .
-copy file named "exam" to current directory (note period at end of command line)
```

```
 cp prod/p* reports
   -copy all files beginning with "p" to reports directory
   -can reverse p* to *p for files ending with "p"
```

Copy img to floppy:

```
dd bs=2x80x18b if=/dev/fd0 of=/tmp/floppy.img
```

Copy source to make img:

```
dd if=/dev/cdrom of=filename.iso
                  ^------^_________ replace with file name
```

DELETING

```
Remove directory:
       $rm -frv dir/
        ^   ^^^    ^
        |   ||| \---------------------- Remove directory
        |   |||
        |   |||                          OPTIONS--------------------
        |   ||\-------------------------- Verbose
        |   |\--------------------------- Recursively through all dir
        |   \---------------------------- Force
        |                                 ---------------------------
        |
        \-------------------------------- Remove command
```

```
rm *     <- Delete all files in directory
rm -r    <- Delete directory and contents
```

```
Command options for permissions
r = read
w = write
x = execute
X = execute only if user already has execute
s = set user or group id on execution
t = sticky +
u = permission granted to user who owns file
g = permissions granted to group members
o = permissions granted to users other than "u" or "g"
```

PERMISSIONS

```
          format:
                   drwxrwxrwx
                   ^^^^^^^^^^
                   ||||||||||
                   |||||||\\\----------- other users
                   |||||||
                   ||||\\\-------------- group
                   ||||
                   |\\\----------------- owner
                   |
                   \-------------------- directory = "d" (possible: -, b, c, q)
```

```
                Octal Method: chmod
                 permissions
    4 = read                  examples: 7 = 4+2+1  will show "rwx"
    2 = write                           6 = 4+2    will show "rw-"
    1 = execute                         5 = 4+1    will show "r-x"

  Usage:  chmod 755 FileName          <--- change file permissions
```

```
 Usage:  chmod -R 755 DirectoryName  <--- change directory permissions recursively
 will show permissions "-rwxr-xr-x"
       Note: first character in permissions show what kind of file
           - = file
           d = directory
   b, c, or q = device files
```

Usage format:

```
chmod -R ugoa =+- rwx filename
       ^ ^^^^ ^^^ ^^^     ^
       | |||| ||| |||     |
       | |||| ||| |||     \---------- file or directory
       | |||| ||| |||                 ----------------------------------
       | |||| ||| ||\---------------- Execute
       | |||| ||| |\----------------- Write
       | |||| ||| \------------------ Read
       | |||| |||                     ----------------------------------
       | |||| ||\------------------- Remove
       | |||| |\-------------------- Add
       | |||| \--------------------- assign
       | ||||                        ----------------------------------
       | |||\------------------------ All
       | ||\------------------------- Others
       | |\-------------------------- Group
       | \--------------------------- User
       |                             ----------------------------------
       \----------------------------- Recursively through all directories
                                     ----------------------------------
       Usage: chmod g+w filename
```

In addition, three extra permission may be assigned:

```
 For files:

      u +-= s  : Set UserID, run as if file-owner

      g +-= s  : Set GroupID, run as if file-group

      o +-= t  : Set "sticky bit", keep executable in swap-space - can only be set by root (no longer supported by Linux-kernel)

 For directories:

      u +-= s  : Set UserID, files in directory inherits directory's ownership

      g +-= s  : Set GroupID, files in directory inherits directory's group

      o +-= t  : Set "sticky bit", limits the ability to remove a file in a world-writeable directory (like /tmp) to the file's owner.
```

VI / VIM

VI/M uses 2 modes in which to process text. To process the various commands available use the "<esc>" key. To insert text simply type "i" to return to text insert mode.

For Starters/Basics----------

```
 Start Vi     vi     [From command line]
 Insert mode  i
 Command mode <esc>
```

```
  Save doc     :w
  Quit app     :q

        Usage:  :wq  [will save and exit document]
```

```
  Navigation-----------------------------
  Right        l
  Left         h
  up           k
  Down         j
  ---------------------------------------
```

Sort all

```
   :1,$ !sort
```

Replace text

```
   :1,$s /and/AND/g
   ^^^^^   ^   ^  ^
   |||||   |   |  |
   |||||   |   |  \----------- Globally
   |||||   |   \-------------- Replace with
   |||||   \------------------ Find
   |||||
   ||||\---------------------- substitute for text
   |||\----------------------- To last line
   ||\------------------------ separator
   |\------------------------- From line "1"
   \-------------------------- ":" operator to start command processing
Useful tid bit to remove EOL characters from DOS file
[use key sequence "<ctrl v><ctrl m>" together for EOL char]
 :%s/^M$//

Select Ranges
 :1,$   = All lines in file
 :1,.   = All lines from start of file to current (included)
 :.,$   = All lines from current to end of file (inclusive)
 :.-2   = Single line 2 lines before current line
 :10,20 w abc <enter>    = Write lines 10,20 to new file abc
 :r abc       <enter>    = Reads contents of file abc into working file
 :10,20d      <enter>    = Delete block lines 10 ~ 20
```

See Line Numbers

```
 :set number             to disable  :set nonumber
```

Write lines to new file

```
 :1,10w abc     [puts lines 1~10 into file abc]
```

Change working file from current to file abc [be sure to save first]

```
 :e abc
```

Read abc file into working file after cursor

```
 :r abc
```

Execute command from prompt

```
 :!cmd    [where cmd is could be a bash command]
```

```
Example ->    :r !date   [inserts time and date at prompt]
```

Enable syntax option for programming source text highlighting

```
 :syntax enable      To disable = :syntax off
```

```
VI Navigation  [escape key = <esc>]
0  [zero] = Move to beginning of line
G         = Go to end of file
5G        = Move to beginning of 5th line
u         = Undo
.         = Repeat last command
x         = Delete character
dd        = Delete line
dw        = Delete word
d$        = Delete from cursor to end of line
$ then J  = Join/append following line with current
$ then a  = Append input at end of line
A         = Append input at end of line [same as above, but one less key stroke]
i         = Insert text at cursor
/         = Bottom of screen type search string and <enter> will move
                cursor to first occurrence
```

Printing

Work from Bash Prompt:

```
        lpr -P //PrintServer/PrinterName "file name"

        Other options:
        -#  Number of copies [example "-2" for 2 copies]
        -c  Make copy before print
```

```
  lpq -P PrinterName : Displays printer status report
```

```
  lprm 4      : Removes job #4 from printer
```

```
  pr          : Formats, paginates, prints header
                OPTIONS
                  -d : forces double spacing
                  -h : Customize header
                  -l : Change number of lines per page | default=66

          example: pr customers | lpr  <format and print file "customers">
```

```
          example: lprm 1006           <canceled job 1006>
```

```
  lpstat -s
    ex: lp -n 2 all files
    ex: lp -n 3 all files -d a3printer
    ex: lpstat -d a3printer
```

See also: CUPS

Disk System

Disk space usage

du -h <dir> : lists an estimate of total space usage for each subdirectory found in <dir> and the estimated grand total disk space used

File System (mounted) Disks space usage

df : lists all mounted disks and blocks total, used, free, and logical disk name

df .

lists only current working mounted disk, displaying total, used, free in blocks.

df -h

lists all mounted disks total, used, free, and logical disk name in human terms (KB,MB,GB,TB).

df -kh

lists all mounted disks total, used, free and logical disk name in human terms (KB, MB, GB, TB) and space used in % (percentages)[-k] option.

List partitions

fdisk -l <disk> : list all partitions on a hard drive e.g. ***fdisk -l /dev/hda***

Linux Guide/Linux commands

An A-Z Index of the Linux BASH command line

Note: The majority of commands listed here belong to coreutils [1].

Command	Description
alias	Create an alias
apropos	Search manual for keyword
at	Schedule a job to run in the future.
awk	Find and Replace text within file(s)
break	Exit from a loop
builtin	Run a shell builtin
bunzip2	Decompress file from bzip2 format
bzip2	Compress file to bzip2 format
cal	Display a calendar
case	Conditionally perform a command
cat	Concatenate files to standard output
cd	Change Directory
cfdisk	Partition table manipulator for Linux
chgrp	Change group ownership
chmod	Change access permissions
chown	Change file owner and group
chroot	Run a command with a different root directory
chvt	Change the virtual Terminal
cksum	Print CRC checksum and byte counts
clear	Clear terminal screen
cmp	Compare two files
comm	Compare two sorted files line by line
command	Run a command - ignoring shell functions
compress	Compress file(s) to old Unix compress format
continue	Resume the next iteration of a loop
convmv	A perl script that converts filenames from one encoding to another
cp	Copy one or more files to another location
cron	Daemon to execute scheduled commands at predefined time
crontab	Schedule a command to run at a later time
csplit	Split a file into context-determined pieces
cut	Divide a file into several parts
date	Display or change the date & time
dc	Desk Calculator
dd [2]	Data Dump - Convert and copy a file

declare	Declare variables and give them attributes
df	Display free disk space
diff	Display the differences between two files
diff3	Show differences among three files
dir	Briefly list directory contents
dircolors	Colour setup for `ls'
dirname	Convert a full pathname to just a path
dirs	Display list of remembered directories
du	Estimate file space usage
echo	Display message on screen
ed	A line-oriented text editor (edlin)
egrep	Search file(s) for lines that match an extended expression
eject	Eject CD-ROM
enable	Enable and disable builtin shell commands
env	Disp, set, or remove environment variables
eval	Evaluate several commands/arguments
exec	Execute a command
exit	Exit the shell
expand	Convert tabs to spaces
export	Set an environment variable
expr	Evaluate expressions
factor	Print prime factors
false	Do nothing, unsuccessfully
fdformat	Low-level format a floppy disk
fdisk	Partition table manipulator for Linux
fgrep	Search file(s) for lines that match a fixed string
file	Determine type of file
find	Search for files that meet a desired criteria
fmt	Reformat paragraph text
fold	Wrap text to fit a specified width.
for	Expand words, and execute commands
format	Format disks or tapes
free	Disp, s memory usage
fsck	Filesystem consistency check and repair.
fstat	List open files
function	Define Function Macros
fuser	Identify process using file
gawk	Find and Replace text within file(s)
getopts	Parse positional parameters

grep	Search file(s) for lines that match a given pattern
groups	Print group names a user is in
gunzip	Decompress file(s) from GNU zip format
gzcat	Show contents of compressed file(s)
gzip	Compress file(s) to GNU zip format
hash	Remember the full pathname of a name argument
head	Output the first part of file(s)
history	Command History
hostname	Print or set system name
iconv	Converts the encoding of characters from one code page encoding scheme to another.
id	Print user and group id's
if	Conditionally perform a command
import	Capture an X server screen and save the image to file
info	Help info
install	Copy files and set attributes
join	Join lines on a common field
kill	Stop a process from running
less	Display output one screen at a time
let	Perform arithmetic on shell variables
ln	Make links between files
local	Create variables
locate	Find files
logname	Print current login name
logout	Exit a login shell
lpc	Line printer control program
lpr	Off line print
lprint	Print a file
lprintd	Abort a print job
lprintq	List the print queue
lprm	Remove jobs from the print queue
ls	List information about file(s)
ll	#ls -l List information about file(s)
lsof	List open files
m4	Macro processor
makewhatis	Rebuild whatis database
man	Print manual pages
mkdir	Create new folder(s)
mkfifo	Make FIFOs (named pipes)
mknod	Make block or character special files

more	Display output one screen at a time
mount	Mount a file system
mtools	Manipulate MS-DOS files
mv	Move or rename files or directories
netconfig	Configure your network
nice	Set the priority of a command or job
nl	Number lines and write files
nohup	Run a command immune to hangup
od	View binary files
passwd	Modify a user password
paste	Merge lines of files
pathchk	Check file name portability
popd	Restore the previous value of the current directory
pr	Convert text files for printing
printcap	Printer capability database
printenv	Print environment variables
printf	Format and print data
ps	Process status
pushd	Save and then change the current directory
pwd	Print Working Directory
quota	Display disk usage and limits
quotacheck	Scan a file system for disk usage
quotactl	Set disk quotas
pax	Archive file(s)
ram	ram disk device
rcp	Copy files between two machines.
read	read a line from standard input
readonly	Mark variables/functions as readonly
remsync	Synchronize remote files via email
return	Exit a shell function
rm	Remove (delete) files
rmdir	Remove folder(s)
rpm	RPM Package Manager (was RedHat Package Manager)
rsync	Remote file copy (Synchronize file trees)
screen	Terminal window manager
sdiff	Merge two files interactively
sed	Stream Editor
select	Accept keyboard input
seq	Print numeric sequences

set	Manipulate shell variables and functions
shift	Shift positional parameters
shopt	Shell Options
shutdown	Shutdown or restart linux
sleep	Delay for a specified time
sort	Sort text files
source	Run commands from a file `.'
split	Split a file into fixed-size pieces
strings	print the strings of printable characters in (binary) files.
su	Substitute user identity
sum	Print a checksum for a file
symlink	Make a new name for a file
sync	Synchronize data on disk with memory
tac	Print files out in reverse line order
tail	Output the last part of files
tar	Tape ARchiver
tee	Redirect output to multiple files
test	Evaluate a conditional expression
time	Measure Program Resource Use
times	User and system times
timidity	Play midi files and set up software synth to play midi files with other commands.
touch	Change file timestamps
top	List processes running on the system
traceroute	Trace Route to Host
trap	Run a command when a signal is set(bourne)
tr	Translate, squeeze, and/or delete characters
true	Do nothing, successfully
tsort	Topological sort
tty	Print filename of terminal on stdin
type	Describe a command
ulimit	Limit user resources
umask	Users file creation mask
umount	Unmount a filesystem
unalias	Remove an alias
uname	Print system information
unexpand	Convert spaces to tabs
uniq	Uniquify files (remove all duplicate lines)
units	Convert units from one scale to another
unset	Remove variable or function names

unshar	Unpack shell archive scripts
until	Execute commands (until error)
useradd	Create new user account
usermod	Modify user account
users	List users currently logged in
uuencode	Encode a binary file into 7-bit ASCII characters
uudecode	Decode a file created by uuencode
v	Verbosely list directory contents (`ls -l -b')
vdir	Verbosely list directory contents (`ls -l -b')
watch	Execute/display a program periodically
whatis	List manual pages by name
wc	Print byte, word, and line counts of a file
whereis	Report all known instances of a command
which	Locate a program file in the user's path.
while	Execute commands
who	Print all usernames currently logged in
whoami	Print the current user id and name (`id -un')
xargs	Execute utility, passing constructed argument list(s)
yes	Print a string until interrupted
zcat	Show contents of compressed file(s)
zip	Compress and archive file(s) to zip format
.period	Run commands from a file
###	Comment / Remark

*The next list is from Transwiki:Useful unix command. Commands should be merged into the table above, with longer explanations placed somewhere in Guide to Unix/Commands. X11 commands go in the Guide to X11 book. Some of these are applications (like **acroread**); someone needs to find a book or place for them.*

- a2p - translation utility from awk to Perl
- a2ps - translation utility from ``any'' to Postscript
- ac - print statistics concerning user connect time
- access - determine whether a file can be accessed
- acroread - Adobe utility for viewing pdf files
- adduser - used by root to add user to system
  - usage: adduser *userid*
- afio - utility to copy or restore files to an archive file. This utility is not part of the basic internal and external programs under Red Hat Linux. It is available as an rpm on the rpmfind.net site.
- agetty - enables login on terminals. See getty, mgetty, and uugetty.
- ali - list mail aliases
- alias - assign name to specified command list. This is actually a shell builtin. On my Red Hat system, I have global alias commands in my /etc/bashrc file and in my /.bashrc file. On my Slackware box, I have made the rm command a little bit safer with
  - alias rm='rm -i'

so that you can't recursively delete your /dev directory without telling the system you're sure you want to do it.

- alien - utility to convert to and from different Linux package formats. Can handle Debian (deb), Stampede (stp), Red Hat (rpm) and Slackware (tgz) packages.
- apropos - display command names based on keyword search
  - usage: apropos *keyword*
- apsfilter - printer filter called by lpd to deal with printing different types of files. This is a fairly sophisticated print filter. It is not set up by default in Slackware. It used to be available on the second cd of a Slackware distribution. Since 4.0, it is available as one of the main packages in the ap set. Read the mail that the installation program sends to the root user. I believe that Red Hat uses its own printer filters.
- ar - create, modify and extract from archives
- arch - print machine architecture type
- as - the portable GNU assembler
- asapm - AfterStep laptop advanced power management utility
- ash - a shell, a very simple shell program sometimes used on boot diskettes since it takes up much less space than bash, tcsh, zsh, etc.
- asload - AfterStep cpu load monitor
- asmail - AfterStep mail checking utility
- asmodem - AfterStep utility to monitor modem status
- aspell - a spell checking program along the lines of ispell
- aspostit - X Window postit note utility
- at - executes a shell script at specified time. Use atq to show pending jobs, and atrm to remove jobs from the queue.
  - usage: at *time*
  - or: at -f file *time*
- atq - shows pending jobs queued by at. If run by root, shows everybody's pending jobs.
- atrm - removes pending jobs queued by at. Use atq to determine the identities of various jobs.
  - usage: atrm *job*
- awk - searches for and process patterns in a file
- banner - print banner to standard output. Syntax is
  - banner [option] [characters]
- bash - Bourne again shell. This is the default shell in the Red Hat installation.
- batch - queue, examine, or delete jobs for later execution. See at.
- bc - a language (compiler) similar to C, with unlimited precision arithmetic
- bg PID - send process with pid ``PID'' to the background. This is the same as executing `[Ctrl]z` while interacting with the running process. This is a shell builtin.
- bh - puts a job in the background. This is a shell builtin.
- biff - mail notification utility. Notifies user of mail arrival and sender's name.
- bind - displays or redefines key bindings. This is a shell builtin.
- bison - parser generator similar to yacc
- bru - a powerful backup utility program. Commercial. Demonstration versions are often included with Linux distributions such as Red Hat.
- bsh - equivalent to ash
- bunzip2 - used to uncompress files compressed with bzip2
- byacc - parser generator
- bzip2 - compresses with algorithm different from gzip
- bzless - view bzipped files

- c++ - invokes GNU C and C++ compiler
- cal - displays a 12-month calendar for the given year or a one-month calendar of the given month
  - usage: cal *month year*
- cat - combine, copy standard input to standard output. Used to join or display files.
- cd - change working directory. This is a shell builtin in bash, tcsh and zsh.
- cdplay - command line utility for playing audio cds
- cfdisk - similar to fdisk, but menu-driven
- chat - used to interact with a modem via a chat script
- chgrp - changes group associated with file. Can be used to change the group associated with subdirectories and files of a directory.
  - usage: chgrp *group files*
  - or: chgrp -R *group files*
- chkconfig - Query or update system services/daemons for different runlevels. Manipulates the various symbolic links in /etc/rc.d. This utility is included with many rpm-based distributions such as RedHat and Mandrake. It is designed to work with System V initialization scripts. Graphical tools for configuring system services include ntsysv, tksysv and ksysv (the latter is a KDE utility).
- chmod - set permissions (modes) of files or directories. A value of 4 is used for read permission. A value of 2 is used for write permission. A value of 1 is used for execute permission. See umask for default file permissions upon file creation. Chmod can also be used to change the suid bit on files. The syntax for the symbolic version is
  - chmod [options] who operation permission file-list

The syntax for the absolute version is

  - chmod [options] mode file-list

To set the uid to the owner's permissions, use

  - chmod u +s file-name

To set the uid to the group's permissions, use

  - chmod g +s file-name

There are lots of security issues related to allowing a program to have root's permissions when run by an ordinary user. I don't pretend to understand all of these issues.

- chown - changes ownership of a file. Can be used recursively.
  - usage: chown *userid files*
  - or: chown -R *userid files*
- chsh - change default shell
- ci - creates or records changes in an RCS file
- clear - clear screen command
- cmp - compares two files for differences
  - usage: cmp *file1 file2*
- co - retrieves an unencoded version of an RCS file
- comm - compares sorted files
- configure - automatically configures software source code
- color-xterm - color xterm program. Under Red Hat, this is just a link to xterm-color.
- control-panel - graphical system configuration tool under Red Hat.
- cp - copies on or more files. Recursive copying is one simple way of archiving part of a directory structure. Use the command as follows:
  - cp -r /sourcedirectory /targetdirectory

- cpio - direct copy of files to an output device. Allows creation of archive file spanning multiple diskettes. Allows one directory structure to be mirrored elsewhere on the partition or on another partition. In order to back up an entire directory structure on diskettes, cd to the directory and use the following command:
  - find . -depth -print | cpio -ov > /dev/fd0

To restore from diskettes, use:

  - cpio -iv < /dev/fd0

The cpio command will prompt the user to insert more diskettes as they are needed. The command for mirroring a directory structure is the following:

  - find . -depth -print | cpio -pv /destinationdirectory

This copies the working directory and its contents, including subdirectories, into /destinationdirectory. In order to copy an individual file which is larger than a floppy, use:

  - find . -name nameoffile -print | cpio -iv > /dev/fd0
- cpkgtool - Slackware tool to install, uninstall and query packages. Front end to installpkg, removepkg, makepkg. This is the graphical version that uses ncurses.
- cpp - GNU C-compatible compiler preprocessor
- crontab - schedules command to run at regularly specified time
- csh - run C shell
- csplit - separate files into sections. See also split.
- cvs - manages concurrent acces to files in a hierarchy. Stands for concurrent version system. Is built on RCS. It stores successive revisions of files efficiently and ensures that access to files by multiple developers is done in a controlled manner. Useful when many developers are working on the same project.
- cut - selects characters or TAB-separated fields from lines of input and writes them to standard output
- date - displays or sets date and time
  - usage: date
  - or: date *date*
- dd - direct copy of file from one device to another. Can be used to make copies of boot or root diskettes for installing Linux. It can be used, for example, to make and exact copy of a floppy disk, as follows. First, place the diskette to be copied in the floppy drive. Then,
  - dd if=/dev/fd0 ibs=512 > floppy.copy
  - Replace the diskette with a fresh diskette.
  - dd if=floppy.copy bs=512 of=/dev/fd0

The ibs and bs options specify the block sizes for input and for both input and output. A boot disk image can be directly copied to a floppy using the second of the two dd commands above.

- declare - declares attributes for a variable (same as typeset). This is a shell builtin.
- df - displays capacity and free capacity on different physical devices such as hard drive partitions and floppy drives that are mounted on the file system. Gives free space in blocks. With the (undocumented) option -h, the program gives free space in Mb or Gb. This is useful for those accustomed to thinking of the capacity of a high-density 3.5 inch diskette as 1440k.
- diff - displays differences between two files
  - usage: diff *file1 file2*
- diff3 - compares three files and reports on differences
- dip - used to set up a SLIP or PPP connection. It can be used to set up an outgoing SLIP connection or an incoming connection.
- diplogin - used for setting up incoming dip connections. See the man page for dip.
- dir - a variation of the GNU ls command that defaults to printing file names in columns

- dircolors - set colors for GNU ls command. In Slackware, this command is run by the /etc/profile script. Then, whenever xterm is run with the -ls (login shell) option, ls displays different colors for different types of files. Typical usage is eval `dircolors -b`. In Red Hat and Mandrake, I get color directories by aliasing the ls command (see below).
- display - set display for output of programs under X Window. Can be used to run a program on a remote machine while displaying the output on a local machine. The remote machine must have permission to send output to the local machine. This is actually an environment variable. See the more detailed discussion in connection with the xhost command below.
- dmesg - displays messages from /var/log relative to the most recent boot
- dos - invoke the DOSEMU DOS emulator
- du - displays information on disk usage. The command
  - du / -bh | less

will display detailed disk usage for each subdirectory starting at root, giving files sizes in bytes.

- dumpkeys - print information about the keyboard driver's translation tables to standard output
- dvilj - send a dvi file to a Laserjet printer. There are specialized versions for individual models of Laserjet printer.
- dvilj2p - specialized version of dvilj for the IIp series of printers. See above.
- dvips - send a dvi file to a Postscript printer, to a Postscript capable Laserjet printer, or to a file (with the -o option). There is a switch to print only a subset of the pages, and another switch to print in landscape mode. Use -t landscape, which is one of the arguments to the paper type switch. If you have one page of a document that is a wide table, and you wish to print this in landscape mode, use
  - dvips *filename -pp pagenumber -t landscape*
- e2fsck - check an ext2 filesystem. The syntax is
  - e2fsck */dev/devicename*

where the filesystem is on /dev/devicename. The device should not be mounted, and this program must be run as root.

- echo - write arguments to standard output. One use is to print out information about environment variables, as in
  - echo $PATH - list paths to search
  - echo $HOME or echo ~ - list name of home directory

This is a shell builtin.

- editres - a dynamic resource editor for X Toolkit applications. Allows the user to change X resources for individual applications.
- efax - fax program
- efix - convert between fax, text, bit-map and gray-scale formats
- egrep - search files for lines that match regular expressions. Runs faster than grep and fgrep.
- elm - an interactive mail system
- elvis - a version of the vi text editor
- emacs - screen oriented text editor
- env - desplay the current environment or set a variable equal to a new value
- eval - scans and evaluates the command line. See dircolors command. This is a shell builtin.
- ex - interactive command-based editor. The man page lists it as being the same as vim, an improved version of vi.
- exec - system call which creates a subshell to execute a binary or a script. This is a shell builtin.
- execve - a variation of the exec command.
- exit - exit a shell. This is a shell builtin.
- expand - convert tabs in files to spaces and write to standard output

- expect - a program that ``talks'' to other interactive programs according to a script. Following the script, Expect knows what can be expected from a program and what the correct response should be. An interpreted language provides branching and high-level control structures to direct the dialogue. In addition, the user can take control and interact directly when desired, afterward returning control to the script.
- export - place the value of a variable in the calling environment (makes it global). This is a shell builtin.
- expr - utility evaluates an expression and displays the result
- f2c - FORTRAN to C translator
- f77 - FORTRAN 77 compiler
- false - null command that returns an unsuccessful exit status
- fax - simple user interface to efax and efix programs
- fc - views, edits, and executes commands for the history list. This is a shell builtin.
- fdformat - low level format of a floppy device
- fetchmail - retrieve mail from a remote mail server and pass it to local SMTP agents on the local machine
- fdisk - used to partition hard drives
  - usage: fdisk *device*
- fg PID - bring a background or stopped process with pid ``PID'' to the foreground. This is a shell builtin. If only one process is running in background mode, fg with no argument is sufficient to bring it to the foreground
- fgrep - search for patterns in files
- file - displays classification of a file or files according to the type of data they contain
- find - find files according to a large variety of search criteria. The find command that I use the most is
  - find . -name filename -print

in order to find files matching a particular name on the working directory and all subdirectories. Find can be incredibly powerful, but it is incredibly obscure.

- finger - display information about a specified userid or userids
- fmt - simple text formatting utility. Tries to make all nonblank lines nearly the same length.
- fold - break lines of specified files so they are no wider than a specified lengths
- fortune - available in the bsdgames package in Slackware and other distributions. Put a call to fortune in /etc/profile and get something inspirational or amusing every time you fire up an xterm as a login shell.
- free - gives used and free memory on system along with other useful information
- fromdos - takes a DOS text file from stdin and sends a Unix file to stdout.
- fsck - file system check and repair
- ftp - file transfer over network
- g++ - C++ compiler
- g77 - GNU Fortran 77 compiler
- gawk - GNU awk, mostly for processing delimited text files
- gcc - invoke C, C++ compiler
- getipts - parses arguments to a shell script. This is a shell builtin.
- getkeycodes - print kernel's scancode-to-keycode mapping table
- ghostscript - set of printing utilities. It seems to be obligatory to have this if a T_EX installation such as teT_EX is installed. How they communicate with one another is somewhat obscure.
- ghostview - Aladdin ghostscript interpreter/previewer
- gimp - image manipulation and paint program
- glint - Red Hat graphical front end for the rpm package installer and manager.
- grep - used to find a string within a file. The -i option returns matches without regard to case. The -n option means that each line of output is preceded by file name and line number. The -v option causes non-matched lines to be printed.

  - usage: grep *pattern files*
  - or: grep -i *pattern files*
  - or: grep -n *pattern files*
  - or: grep -v *pattern files*
- groupadd - create a new group on the system
- groups - shows which groups you are in
- grub - Gnu grand unified bootloader. Can be used instead of lilo to boot multiple operating systems. I encountered a couple of snafus trying to install grub on my home machine after installing Mandrake 8.0 and choosing the lilo bootloader during the initial install. The Mandrake installation program set up /boot/vmlinuz as a symlink to the actual kernel, vmlinuz-2.4.3-20mdk. It took me a while to figure out that grub doesn't understand symbolic links. The documentation suggests installing grub on a diskette using the ``dd'' command. This refused to work, but
  - grub-install '(fd0)'

  did work. The single quotes are necessary. The files necessary to run grub are normally located in /boot/grub. Once the file menu.lst has been edited and appropriated entries added to boot the different operating systems on one's hard disk(s), the following sequence of commands can be used to install grub in the master boot record (MBR) sector of the hard disk:
  - root (hd0,*x*)
  - setup (hd0)

  Here, the *x* should be replaced by the partition where the /boot/grub directory is located, which is probably the root partition of the Linux system. Note that grub has its own conventions for naming devices and numbering partitions, so that for example a partition which is called hda6 under Linux will be called (hd0,5) by grub.
- grub-install - command to install grub on the hard drive (or floppy drive).
- gunzip - used to uncompress files compressed with gzip
- gv - PostScript and PDF previewer, based on ghostview
- gvim - see vi
- gzexe - compresses executables
- gzip - used to compress or decompress files
- halt - shut down system as root, without reboot, immediately
- hash - remembers the location of commands in the search path. This is a shell builtin.
- head - displays first part of a file
- history - command for viewing and manipulating the shell command history list
- host - look up host names using domain server
- hostname - used to get or set hostname. Typically, the host name is stored in the file /etc/HOSTNAME.
- hwclock - used to query and set the hardware clock
- hylafax - commercial fax program
- id - display userid and groupid
- inetd - daemon which starts up other daemons on demand. Configured in /etc/inetd.conf.
- ifconfig - display (as root) information on network interfaces that are currently active. First ethernet interface should be listed as eth0, second as eth1, etc. First modem ppp connection should be listed as ppp0, etc. The ``lo'' connection is ``loopback'' only.
- ifdown - shut down the network interface
- ifup [interface_name] - start up the interface
- info - display system information. This is the GNU hypertext reader.
- init - the mother of all processes, run at bootup, executes commands in /etc/inittab. Can be used (with root privileges) to change the system run level.
  - usage: init *run_level*

- insmod - used (by root) to install modular device drivers
- installpkg - Slackware command to install one of the packages from the program sets
- intr - interrupt key, usually [Ctrl-C]
- ispell - checks files for spelling errors
  - usage: ispell *files*
- jed - programmer's file editor. Behaves like emacs. Has modes for T_EX, FORTRAN, C, etc.
- jobs - displays list of current jobs in the background. This is a shell builtin.
- joe - simple WordStar-like text editor. It can be invoked in emacs emulation mode with jemacs and in WordStar emulation mode with jstar.
- jove - Joseph's Own Version of Emacs. A simple emacs clone.
- kbd_mode - print current keyboard mode
- kernelcfg - GUI to add/remove kernel modules (as root in X terminal).
- kerneld - kernel daemon, a process that stays in memory and does all sorts of useful stuff, like automatic loading of device driver modules
- kikbd - a utility program that comes with KDE that allows users to switch on the fly among different international keyboards. It can be used under different window managers than kfm.
- kill - sends a signal to (especially to terminate) a job or process. This is a shell builtin in bash, tcsh and zsh.
- killall - kill processes by name. Kill all processes which are instances of the speciffied program. Also used to send signals to processes or restart them.
- killall5 - kill all processes except the ones on which it depends
- last - generate a listing of user logins
- lastlog - prints the last login times of all users
- latex - compile a L^AT_EX file
- ldconfig - creates the necessary links and cache (for use by the run-time linker, ld.so) to the most recent shared libraries found in the directories specified on the command line, in the file /etc/ld.so.conf, and in the trusted directories (/usr/lib and /lib). Ldconfig checks the header and file names of the libraries it encounters when determining which versions should have their links updated. Ldconfig ignores symbolic links when scanning for libraries.
- ldd - list the shared libraries on which a given executable depends, and where they are located
- leave - display reminder at specified time
- less - Linux alternative to ``more'' command. Displays text files, one screenful at a time. When less pauses, there is a large number of available commands to tell it what to do next. One can scroll both forwards and backwards.
- let - evaluates a numeric expression. This is a shell builtin.
- lilo - installs boot loader on the boot sector of a hard drive, of a diskette, or in another location. My 486 has a hard drive that is too large for the machine's BIOS, so I have to boot from a floppy. To create a boot diskette, I do the following (as root):
  - /sbin/fdformat /dev/fd0H1440
  - /sbin/mkfs.ext2 /dev/fd0
  - mount -t ext2 /dev/fd0 /mnt/floppy
  - cp -dp /boot/* /mnt/floppy
  - /sbin/lilo -C /etc/lilo.flop

  The -C option to lilo has lilo use the lilo.flop file instead of the default lilo.conf.
- linuxconf - interactive tool for configuring Linux system. Uses X if loaded. This is a Gnome tool. It comes with my Red Hat distribution, and is not included with Slackware. It would seem to be the easiest way to configure Linux under Slackware. Version 1.15 is available for Slackware. There is a pretty good introduction to the use of linuxconf in the Red Hat 5.2 installation manual, which is available online at their web site.

- ln - creates a link to a file. Used to create hard links and, with the -s option, symbolic links which can link files on different disk partitions. The syntax is
  - ln [options] source [dest]
- locate filename - find the file name which contains the string ``filename''. The syntax is easier than the find command.
- lock - temporarily lock terminal
- lockfile - create semaphore file(s), used to limit access to a file
- - log in to system
- logname - consult /etc/utmp for user's login name
- logout - execute logout as individual user and bring up login: prompt
- look - look for strings in files
- lpq - show print jobs that are waiting
- lpr - send file to be printed
- lprm - cancel a job from print queue
- ls - list directory contents. To get colored directory listings under Red Hat, Mandrake, etc., use
  - ls -color

To get this all the time, add

- alias ls='ls -color=auto'

to .bashrc. The following command

- alias ls='ls -Fskb -color=auto'

will give directory listings in color, with file sizes in kilobytes, and append a character to the file to indicate its type.

- lsattr - list attributes of files in ext2 file system
- lsmod - used (by root) to show kernel modules currently loaded
- lspci - utility to display information on pci buses and hardware devices attached to them. Part of the pciutils package that comes with many Linux distributions.
- lspnp - utility to display information about pnp devices. Part of the pcmcia or kernel-pcmcia package, depending on the distribution.
- m4 - an implementation of the traditional Unix macro processor. It can be used with the sendmail configuration package in Red Hat (and Slackware) to generate a sendmail.conf configuration file without having to edit the configuration file directly.
- magicfilter - general purpose printer filter. See apsfilter above. apsfilter is the printer filter that comes with the Red Hat and Slackware distributions.
- mail - sends or reads electronic mail
- make - keeps a set of programs current. This is a utility that helps when developing a set of programs. It works by executing a script called makefile, Makefile or GNUmakefile in the working directory. It is very often used in combination with configure when compiling and installing noncompiled software packages.
- makebootdisk - command in Slackware to do just what the name says
- MAKEDEV - executable script to make device files on /dev
- makeswap - configures swap space
- man - displays information from online Unix reference manual
- manpath - attempt to determine path to manual pages
- mc - Midnight Commander file manager and visual shell
- mesg - enables/disables reception of messages
- minicom - terminal program
- mkdir - create a directory

- mkfs - create a file system (format) on a device or partition. Should be invoked after lowlevel formatting of the disk using fdformat. It has several versions which are all links to the basic program, such as mkfs.ext2 and mkfs.msdos.
- mkswap - creates a Linux swap space on the specified hard disk parition (root privileges neede)
  - usage: mkswap *device*
- more - list file contents, stopping after each full screen
- mount -t [fstype] [device] [mountpoint] - mount device using filesystem of type [fstype] with device name [device] at the location [mountpoint] in the filesystem directory tree
- mount -a - mount all filesystems according to the specifications in /etc/fstab
- mouseconfig - mouse configuration utility under Red Hat. Located in /usr/sbin.
- mpage - print multiple pages per sheet on a Postscript printer. Can also be used to print a page in landscape mode.
- Mtools - package of MS-DOS utilities. Includes the following commands.
  - mcd - changes working directory on DOS disk
  - mcopy - copies DOS files from one directory to another
  - mdel - deletes DOS files
  - mdir - lists contents of DOS directories
  - mformat - adds DOS formatting information to a disk
  - mtype - displays contents of a DOS file

The default device for execution of these commands is /dev/fd0 and can be referred to as ``a:''.

- mv - moves (renames) files
- netconf - used (as root) to set up network
- newaliases - rebuilds the /etc/aliases database used by sendmail. Must be rerun every time /etc/aliases is modified for the changes to take effect.
- newgrp - similar to login. Changes user's identification
- nice program_name - sets the priority of the program ``program_name''.
- nm - lists the symbols from object files objfile. If no object files are given as arguments, nm assumes `a.out'.
- nohup - runs a command that keeps running after logout. The command is in principle immune to hangups, and must have output to a non tty. According to *Linux in a Nutshell*, this is necessary only in the Bourne shell, since modern shells preserve background processes by default.
- ntsysv - run level editor under Red Hat. This is the equivalent of tksysv, but does not require a graphical interface.
- nxterm - color xterm program. The man page for nxterm under Red Hat brings up the same page as xterm.
- od - dumps contents of a file
- passwd - change login password
- paste - joins corresponding lines from files
- patch - updates source code. Attempts to update a file from a file of change information, or pathces, created by diff.
- pathchk - determine validity and portability of filenames
- pdflatex - part of the pdftex program suite. Produces pdf output from a $\mathrm{L^AT_EX}$ file.
- pdftex - produces pdf output from a TeX file. See also pdflatex. This program is part of the tetex 0.9 distribution that is included with Red Hat 5.2 and above, and with Slackware 4.0 and above. It is also available as a separate program.
- perl - practical extraction and report language
- pg - display data one screenful at a time
- pico - simple screen oriented text editor. It is included as part of the Pine program.
- ping - check if Internet computer is responding. Can also measure the time it takes the queried computer to respond.

- pkgtool - Slackware tool to install, uninstall and query packages. Front end to installpkg, removepkg, makepkg. The cpkgtool is the ncurses graphical version of this program.
- popclient - retrieve mail via the Post Office Protocol. Supports POP2 and POP3.
- popd - pops the top directory of the directory stack and uses cd to change to that directory. This is a shell builtin.
- pr - paginates files for printing
- printenv - display list of environment variables
- printtool - run (as root) in an X terminal to configure your printer(s)
- ps - displays status of processes. Use the -a option for processes for all users. Use the -x option to include processes not attached to a terminal.
- pstree - display processes in the form of a tree structure. Killing a parent process will also kill all the children and their descendants.
- pushd - pushes the argument onto the top of the directory stack and uses cd to change to that directory. This is a shell builtin.
- pwd - print absolute path of working directory. This is a shell builtin.
- pwchk - checks the integrity of password and shadow files
- pwconv - converts passwords to the shadow password format
- pwunconv - unconverts passwords from the shadow password format. Generates a standard Unix password file.
- python - interpreted, interactive, object-oriented programming language
- rcp - copy one or more files to or from remote computer. The syntax is poorly explained in the documentation that I have, including the man pages. Usage is:
  - rcp filename username@remotehost:path

The user's home directory on the remote system must contain the file .rhosts with a list of users (preceded by the full domain name or exact IP address of their machine) with access privileges.

  - localhostname username
- rcs - creates or changes the attributes of an RCS file. Stands for Revision Control System.
- rdev - query/set image root device, swap device, RAM disk size, or video mode in kernel
- read - reads line from standard input. This is a shell builtin.
- readonly - declares a variable to be read only. This is a shell builtin.
- reboot - in Slackware, reboots the system. Seems to be equivalent to shutdown -r now in generic Linux.
- renice program_name - resets the priority of process ``program_name''.
- reset - used to reset the screen characteristics. This is useful if the screen gets messed up from, for example, trying to display a binary file in an xterm.
- return - exits from a function. This is a shell builtin.
- rlog - prints a summary of the history of an RCS file
- rlogin - log in to remote computer. The general syntax is as follows, using the UQAM Nobel machine as an example:
  - rlogin -l userid nobel.si.uqam.ca

The remote computer must recognize the local user and the local machine. See the rcp command for how to set up the .rhosts file on the remote machine.

- rm - remove files or directories. With the -r (recursive) option (very dangerous!), can be used to remove the contents of a specified directory including all subdirectories.
- rmail - interpret and handle remote mail received via uucp
- rmdir - remove empty directories
- rmmod - used to remove modular device drivers
- route -n - show routing table. The n option returns numerical addresses rather than names.

- rpm - invokes the Red Hat package manager in command line mode. I often use this command in query mode to query packages about what files they contain and to find out which package owns a particular file. Examples are
  - rpm -qil foo. Gives package information and a file list for the package foo.
  - rpm -qfil foo. Gives package information and a file list for the package that owns the file foo. Foo must be in the working directory, or the full path to foo must be specified.

We need to find out about installing the rpm package on a Slackware box. It's probably better to use a package converter such as alien.

- rpm2tgz - an extremely useful utility on Slackware systems that converts rpm packages to tgz format. They can then be installed using the installpkg command (or pkgtool).
- rsh - execute shell command on a remote computer. See rcp and rlogin.
- rstat - summarize host's status: uptime, load averages, and current time
- ruptime - show host status of local machines
- rusers - list who is logged on local machines
- rwall - write to all users over a network
- rwho - show who is logged in on a LAN. The rwho service must be enabled for this command to run. If it isn't, run ``setup'' as root. I don't understand this last remark, which comes from ``Linux Newbie Administrator Guide''.
- rxvt - a terminal program similar to xterm, but which has less features and uses less memory
- sed - edits a file (not interactively). Also a tool for processing text files.
- set - set or display value of shell variables. This is a shell builtin. The command
  - set | less

prints the current user environment, giving the values of currently defined variables.

- setenv - set or display value of environment variables
- setserial - used by root to configure a serial port
- setterm - set terminal attributes for a virtual console
- setuid - set the id of a program when it is run. Used, for example, to give root privileges to a program run by an ordinary user. This is actually done by running the chmod program as root. See the chmod command for the syntax.
- setup - Slackware program to set up program sets and configure system. Setup devices and file systems, mount root file system
- sh - standard Unix shell. On Linux, just another name for bash.
- shift - promotes each command-line argument. This is a shell builtin.
- showmount - show information about an nfs server
- shutdown - reboot or shut down system as root, after specified amount of time. With the -r option, reboot. With the -h option, halt the system.
  - usage: shutdown -r *minutes*
- sleep - creates process that sleeps for specified interval
- sliplogin - attaches a SLIP interface to standard input. Used to allow dialin SLIP connections.
- sort - sorts and/or merge files
- split - split file into specified number of segments
- ssh - secure shell. Apparently has many of the same functionalities as rlogin, telnet, ftp, rsh, etc., with better security and encryption features. We may want to learn how to set this up and use it.
- startx - front end to xinit in Linux. This is a script which starts up X clients and shuts down the X server on exit from the window manager.
- startx -- :1 - start the next X window session on the display 1 (the default is opened on display 0). One can switch between different graphical displays using `[Ctrl][Alt][F7]`, `[Ctrl][Alt][F8]`, etc.
- stty - sets or displays operating options for terminal

- su - log in as another user, including root
- sudo - allows individual users to have root permission to perform specified tasks
- swapoff - disables swap disk
- swapon - enables swap disk
- symlinks - provide list of and information about symbolic links
- sync - writes memory buffers to physical devices
- systat - query host for system information
- tac - print file in reverse
- tail - displays the last part of a file
- talk - visual communication program that copies lines from one terminal to that of another user
- tcl - scripting language
- tcsh - extended version of the C shell
- tee - copy standard input to standard output and one or more files
- telinit - used to change run level. Exact run level that corresponds to single-user, multi-user, and X levels depends on distribution.
- telnet - remote login over network (to login to serverX, use the command telnet serverX)
- test - evaluates an expression or compares arguments. This is a shell builtin in bash, tcsh and zsh.
- tftp - user interface to TFTP protocol
- time - displays times for the current shell and its children. This is a shell builtin. Strange, because there is also a /usr/bin/time program on my Red Hat system.
- tin - Netnews reader
- tkdesk - graphical desktop file manager for X
- tksysv - graphical runlevel editor under Red Hat. Allows root to configure the services that are started at each run level.
- tload - display system load average in graph format
- top - dynamically displays process status
- touch - update access and modification times of a file. If the file does not exist on disk, an empty file is created.
- tr - translation utility that can be used, for example, to replace specified characters in a text file
- trap - traps a signal. This is a shell builtin.
- true - null command that returns a successful exit status
- tset - initializes terminal
- tty - shows special file that represents your terminal. Displays the terminal pathname.
- type - displays how each argument would be interpreted as a command. This is a shell builtin.
- typeset - declares attributes for a variaable (same as declare). This is a shell builtin.
- ul - translate underscores to underlining
- umask - establishes the file-creation permissions mask. Usage is
  - umask xyz

The system subtracts x, y and z from the owner, group and other file permissions that it would otherwise assign to new files. This is a shell builtin.

- umount [device] - finish writing to the device and remove it from the active filesystem. The command umount -a will (re)mount all file systems listed in /etc/fstab.
- unalias - remove name previously defined by alias. This is a shell builtin.
- uname - displays information about the system. With no arguments, it displays the name of the operating system. With the -a option, it displays information about the operating system, the host name, and hardware.
- uniq - displays lines of a file that are unique
- unset - removes a variable or function. This is a shell builtin.
- unzip - uncompress files compressed with the zip utility, compatible with DOS PKzip

- updatedb - update file database used by locate command
- uptime - shows the time, how long the system has been up, the number of users, and average load.
- useradd - same as adduser
- userdel - remove an account (as root). The user's home directory and undelivered mail must be dealt with separately.
- users - prints list of users on the system
- vdir - variant of the GNU version of the ls command. Defaults to printing out the long version of directory entries.
- vi - standard screen oriented Unix editor
- view - vi in read-only mode
- vim - improved vi editor
- vrfy - query remote host to verify the accuracy of an email address
- w - display info about userids and active processes
- wait - waits for a background process to terminate. This is a shell builtin.
- wc - displays number of lines, characters and words in a file
- Wharf - the AfterStep application dock module
- whatis - display one-line summary of specified command
- whereis - use to find utilities in standard locations
- which - used to find utilities in search path. Will return the absolute directory path of the named utility program.
- who - display information about currently logged in userids
- whoami - display information about userid that is currently logged in
- wish - front end to tk, an X window extension of tcl
- workbone - console based cd player
- workman - graphical cd player program
- write - send messages to another local user
- X - the X server
- xadm - display advanced power management BIOS information
- xargs - converts standard output of one command into arguments for another. This is one of those powerful but obscure commands. Xargs reads arguments from the standard input, delimited by blanks (which can be protected with double or single quotes or a backslash) or newlines, and executes the command (default is /bin/echo) one or more times with any initial-arguments followed by arguments read from standard input. Blank lines on the standard input are ignored.
- xbiff - graphical mail delivery notification utility
- xcalc - simple calculator program
- xclipboard - name says it all
- Xconfigurator - Red Hat utility for configuring settings for X
- xdm - used to start an X login session. This can be used to start a login session on a remote system. See the discussion on the following site:
  - http://www.menet.umn.edu/~kaszeta/unix/xterminal/index.html [3]

See the man pages for X, xdm, and Xserver. As usual, the man pages are pretty obscure. The best single source seems to be the Xserver man pages. After X is configured, X needs to be started at bootup with the command (in /etc/rc.d/init.d/xterm):

- X -quiet -query remotemachineaddress

If the address of a nameserver is not configured, then the numeric address of the remote machine rather than its name should be entered. If the machines are connected through ethernet cards and the net, then obviously basic networking has to be set up. Gnome and KDE come with their own versions of X display/login managers, called respectively gdm and kdm.

- xdvi - view a dvi file compiled under L^AT_EX
- xedit - a simple text editor for X
- xf86config - graphical configuration tool for X
- XF86Setup - graphical configuration tool for X
- xfd - display an available font in X. Creates a grid in an x-term with one character per rectangle.
- xfig - utility for interactive generation of figures
- xfm - graphical file manager for X
- xhost - tell X server that remote computer has access to your machine and that you will use the remote computer. This can be used to set up remote X sessions. To set up a remote X session on the UQAM Nobel machine, run the following command on the local machine (one doesn't have to be root to do this)
  - xhost +nobel.si.uqam.ca

Then, log onto the remote machine using rlogin (see above) or telnet. Once logged in, use the following command to get the remote X server to open an X terminal on the local machine:

- setenv DISPLAY localhostname:0 ; xterm &

This is valid for csh, which is the default login shell on Nobel. For ksh, (and I think bash) replace with

- DISPLAY=localhostname:0
- export DISPLAY ; xterm

Other X-based programs such as Netscape or Gauss (graphical version) can also be run on a remote machine with display on the local machine with little trouble. The local X server is the program that has all of the information concerning the properties of the graphics card and terminal, so it must be necessary to have X running on the local machine. The following should also work. After using xhost to give permission to the remote machine to display on the local machine, use

- netscape -display localhostname:0.0

Question: can one start the X session on the local machine and then run a remote copy of a window manager?

- xinit - start X Window. The command startx is a front end to xinit in Linux, including Slackware.
- xload - displays a graphic of the system load
- xlpq - graphical interface to print manager. This is included on one of the XFCE menus, but does not seem to be a part of the base Red Hat distribution.
- xlsfonts - list fonts available under the X Window system.
- xman - browsable command reference. Displays manual pages under X.
- xmh - graphical front end under X to the nmh mail handling system. This program is part of the XFree86 package in Red Hat.
- xmodmap - utility for modifying keymaps and pointer button mappings in X. Can be used to install a French Canadian keyboard. Download the Xmodmap.cf file from www.linux-quebec.org, and insert the command
  - xmodmap /etc/X11/Xmodmap.cf &

into the .xession (with xdm) or the .xinitrc (with startx) file.

- xosview - displays bar graphs of system load, load average, memory usage, and swap usage
- xpaint - simple paint program for X
- xpdf - GPL'd utility for previewing dvi files. Doesn't seem to work too well on texts with a lot of math.
- xplaycd - X Window audio cd player utility
- xsetroot - utility to configure root window of an X terminal
- xsysinfo - graphical display of load and memory usage
- xterm - start an X Window terminal session
- xterm-color - color version of xterm
- xv - utility for viewing and manipulating many types of image files. This is a shareware program.

- xvidtune - utility for fine tuning of monitor settings under X
- xwd - screen capture command
- xwud - view images captured with xwd
- yacc - parser generator
- ytalk - multi-user program similar to talk
- zcat - read one or more files that have been compressed with gzip or compress and write to standard output
- zcmp - read compressed files and pass them to cmp
- zdiff - read compressed files and pass them to diff
- zgrep - read compressed files and pass them to grep
- Zharf - AfterStep button panel module
- zip - zip utility compatible with DOS PKzip
- zless - view zipped files
- zmore - print contents of compressed files one screen at a time
- znew - uncompress Z files and recompress in .gz format

References

[1] http://www.gnu.org/software/coreutils/
[2] http://en.wikipedia.org/wiki/Dd_(Unix)
[3] http://www.menet.umn.edu/~kaszeta/unix/xterminal/index.html

Linux Guide/Desktop environments

Linux GUIs

Desktop environments and window managers are not the same. A desktop environment includes all the software needed for a desktop, including a window manager, but a window manager only manages the windows and does not include any other software. Sometimes window managers provide extra features, such as a pager.

Desktop environments

GNOMEUbuntu 9.04 uses the GNOME desktop environment

> GNOME (pronounced /ɡəˈnoʊm/) is a desktop environment composed entirely of free and open source software. It is an international project that includes creating software development frameworks, selecting application software for the desktop, and working on the programs which manage application launching, file handling, and window and task management.
>
> GNOME is part of the GNU Project and can be used with various Unix-like operating systems, most notably those built on top of the Linux kernel and the GNU userland, and as part of the Java Desktop System in Solaris.
>
> See Using GNOME

KDEThe K Desktop Environment

> KDE (pronounced /ˌkeɪdiːˈiː/) is a free software project based around its flagship product, a cross-platform desktop environment designed to run on Linux, Windows and Mac OS X systems. The goal of the project is to provide basic desktop functions and applications for daily needs as well as tools and documentation for developers to write stand-alone applications for the system. In this regard, the KDE project serves as an umbrella project for many standalone applications and smaller projects that are based on KDE technology. KDE software is based on the Qt toolkit. The original GPL version of this toolkit only existed for the X11

platform, but with the release of Qt 4, GPL versions are available for all platforms. This allows KDE software based on Qt 4 to also be distributed to Microsoft Windows and Mac OS X.

See Using KDE

XFCEThe lightweight XFCE desktop environment

Xfce (pronounced as four individual letters) is a free software desktop environment for Unix and other Unix-like platforms, such as Linux, Solaris and BSD. It aims to be fast and lightweight, while still being visually appealing and easy to use.

The current version, 4.6, is modular and reusable. It consists of separately packaged components that together provide the full functionality of the desktop environment, but which can be selected in subsets to create the user's preferred personal working environment. Xfce is mainly used for its ability to run a modern desktop environment on relatively modest hardware.

It is based on the GTK+ 2 toolkit (the same as GNOME). It uses the Xfwm window manager. Its configuration is entirely mouse-driven, and the configuration files are hidden from the casual user.

Note: Both GNOME and KDE use comparatively higher system resources than XFCE and Fluxbox. So if you are on a old PC (eg., Pentium 200 MHz, or have less than 128 MB RAM) you should seriously consider XFCE and Fluxbox over the heavyweights. Same apply if you find KDE or GNOME somehow slow on your high end workstation. IceWM, though minimalist, is fast, reliable, themable, and less resource intensive than the others.

Window managers

CompizCompiz uses compositing and transparency to achieve attractive and useful effect, like window-switching, shown here.Compiz's famous rotating desktop cube.

Compiz is one of the first compositing window managers for the X Window System that uses 3D graphics hardware to create fast compositing desktop effects for window management. The effects, such as a minimization effect and a cube workspace are implemented as loadable plugins. Because it conforms to the Inter-Client Communication Conventions Manual standard, Compiz can substitute for the default Metacity in GNOME or KWin in KDE.

Compiz plugins include the famous cube effect (example to the right), Alt-Tab application-switching with live previews or icons, and a feature similar to Exposé.

Enlightenment [5]

Enlightenment, also known simply as E, is a stacking window manager for the X Window System which can be used alone or in conjunction with a desktop environment such as GNOME or KDE. Enlightenment is often used as a substitute for a full desktop environment.

FLUXBOX

Fluxbox is a stacking window manager for the X Window System with the aim to be lightweight. Its user interface has only a taskbar, a pop-up menu accessible by right-clicking on the desktop, and minimal support for graphical icons. All basic configurations are controlled by text files, including the construction of menus and the mapping of keybindings. Fluxbox has high compliance to the Extended Window Manager Hints specification.

Fluxbox can show some eye candy: colors, gradients, borders, and several other basic appearance attributes can be specified. Recent versions support rounded corners and graphical elements.

Because of its small memory footprint and quick loading time, Fluxbox is popular in many Live CDs such as Knoppix STD and GParted.

FVWM

The F Virtual Window Manager is a virtual window manager for the X Window system. Originally a twm derivative, FVWM has evolved into a powerful and highly configurable environment for UNIX systems.

IceWM

IceWM is a stacking window manager for the X Window System graphical infrastructure. It is relatively lightweight in terms of memory and CPU usage, and comes with themes that allow it to imitate the GUI of Windows 95, OS/2, Motif, and other graphical user interfaces. IceWM is meant to excel in look and feel while being lightweight and customizable.

IceWM can be configured from plain text files stored in a user's home directory, making it easy to customize and copy settings. IceWM has an optional, built-in taskbar with menu, tasks display, network and CPU meters, mail check and configurable clock. Official support for GNOME and KDE menus was previously available as a separate package. In recent IceWM versions, support for them is embedded. External graphical programs for editing configuration and menu exist.

References

[1] http://www.gnome.org
[2] http://www.kde.org
[3] http://www.xfce.org/
[4] http://www.compiz.org/
[5] http://www.enlightenment.org
[6] http://fluxbox.sf.net/
[7] http://fvwm.org/
[8] http://www.icewm.org/

Linux Guide/Getting Help

How to get help properly

This guide is going to give you a good start in the world of Linux, but I'm sure you will have questions along the way. Don't be afraid to ask questions but be sure to ask them right.

1. Try searching around for the answer first.

   You can imagine that help forums get asked the same questions many times over so see if someone else has run into the same problem as you. You'll save time and effort for both yourself, and whoever you might ask.

2. Go ahead and ask your question.

   Especially on IRC, you should not ask to ask your question ("Does anyone here know about linux?") -- just ask.

3. Try to be concise and comprehensive with background information.

   You're more likely to get an answer if people don't have to read large chunks of possibly irrelevant information they could have deduced from hearing your problem or asked at a later date.

If you're getting help on IRC, read a tutorial like [1] or [2] if you aren't familiar with getting help on IRC.

Built-In

help

Help for built-in commands are available from the **help** command. For example

```
help logout
```

Provides help on the logout command. Help just prints a few lines of text to the console. You may need to be in a shell for this to work.

Note that even common commands like cp and mv are not built-in. In addition, most built-in commands have man and info pages (see below) - so it is generally more useful to look in man or info first.

If you have no idea about what command you should use, but you have a vague idea about some keywords that may be related to the topic you want (e.g., you want to print something), you can use the apropos command, like this:

```
 apropos print
```

that will display a lot of things related to print.

man

See also: Guide to Unix

Man pages are a flexible format of help page: they can be viewed in the console, and can be viewed as HTML in a web browser. For example, to view the man page for cp, type

```
man cp
```

If you don't know the name of the command, you can search by using -k

```
man -k "copy"
```

Man is split into sections

1. Executable programs or shell commands
2. System calls (functions provided by the kernel)
3. Library calls (functions within program libraries)
4. Special files (usually found in /dev)
5. File formats and conventions eg /etc/passwd
6. Games
7. Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8. System administration commands (usually only for root)
9. Kernel routines [Non standard]

If a page with the same name is in more than one section, the correct one can be accessed by specifying the section number:

```
man 1 cp
```

alternatively, specify -a to walk through all the sections

```
man -a cp
```

Man pages can also be viewed in web browsers. Some web browsers, such as Konqueror allow you to use man: like a protocol - for example, typing

```
man:cp
```

into the location bar shows the man page for cp. More generally, however, the man command can generate an HTML file, and then display it in a browser by using the -H command with the browser of your choice - in this example, firefox:

```
man cp -Hfirefox
```

info

See also: Guide to Unix

Info pages are another type of help page. Pages here are sometimes duplicates of man pages - and sometimes are more complete. To view an info page, type

```
info cp
```

Yelp

Yelp is a graphical program to display help documentation on GNOME systems. Run the command 'yelp' to begin.

Online

Web pages

- Generic:
  - linuxhelp.net [3]
  - linuxquestions.org [4] & their wiki [5]
  - The Ultimate Linux Newbie Guide [6]
  - commandlinefu.com [7]
- Ubuntu-specific:
  - help.ubuntu.com [8]
  - ubuntuforums.org [9]
  - debuntu.org [10]
  - embraceubuntu.com [11]

Mailing lists

Most distributions will provide several public mailing lists for support issues. For example, lists.ubuntu.com [12] shows a plethora of mailing lists for specific types of support. ubuntu-users [13] is probably your best option if you use Ubuntu.

IRC

- ##linux on freenode
- #linux [14] on oftc.net
- irc:#ubuntu on freenode

Find more [15]

LUGs

Linux Users' Groups are a great resource for getting help. They're typically local organizations, or perhaps specific to a university. They may have a website with helpful information, or a mailing list or IRC channel where you can get personal assistance. Depending on the area served by the LUG, you might be able to meet face-to-face with fellow users who may offer to help you.

References

[1] http://workaround.org/getting-help-on-irc
[2] http://www.reactor-core.org/irc-help.html
[3] http://www.linuxhelp.net/
[4] http://www.linuxquestions.org/
[5] http://wiki.linuxquestions.org/
[6] http://www.linuxnewbieguide.org/
[7] http://commandlinefu.com
[8] https://help.ubuntu.com/
[9] http://ubuntuforums.org
[10] http://www.debuntu.org
[11] http://embraceubuntu.com
[12] https://lists.ubuntu.com/
[13] http://lists.ubuntu.com/mailman/listinfo/ubuntu-users
[14] irc://irc.oftc.net/linux
[15] http://www.google.com/search?q=linux%20help

Linux Guide/Freezes

This page describes how to get out of problems when programs go wrong. The most common way is by killing the process - an individual program or one of its parts. If you kill a process, you'll probably lose any unsaved data in that program, and possibly from other programs it has spawned or is communicating with..

This page also describes what to do if the computer is completely frozen.

These suggestions have more adverse consequences the further down the list you go. For example killing a window in KDE will lose the unsaved data in that program, whereas pressing the reset button could corrupt all data on a hard drive (in extreme cases). Don't resort to extreme measures unless you have no other choice.

On the Desktop

In X-Window (general)

The default easy-to-use application for killing frozen pieces of software when using an X environment is **xkill**. When executed, the mouse cursor will change to an icon of a skull and crossbones, and the next window you click will be killed.

```
xkill
```

Sometimes this can be done by doing Control+Alt+Escape, or alternatively by bringing up the "Run" dialog box with ALT+F2 and typing in "xkill" followed by RETURN.

In KDE

KDE System Guard [1] (installed by default) can be used to kill troublesome processes. **Ctrl+Esc** is the default key combination for starting it, or run *ksysguard*.

In GNOME

The **gnome-system-monitor** GUI program is one way to view active processes, end processes, and kill them too. There is also a "Force Quit" widget which can be added to a panel which replicates the function of **xkill**.

Killing at the Console

The easiest way of stopping a program running on the terminal you are using is pressing **Ctrl+C**, which asks a program to stop (sends SIGINT) - but the program can ignore this. Ctrl+C also works on programs like XTerm or Konsole. See also Alt+SysRq+K below.

In addition, there are several commands to kill processes from any terminal window, not just the one a program is running from. You can use a program like XTerm or Konsole to provide a fake terminal, or you can switch to a terminal by pressing ctrl+alt+function key; ie **Ctrl+Alt+F2** to Ctrl+Alt+F6 (depending on your system). Ctrl+Alt+F7 gets you back to the desktop (again, depending on your system).

The Guide to UNIX/Commands/Process Management page has more information on process management commands, but a few popular ones are below:

- **pkill** and **killall** are the most useful program-killing commands, and they do similar things. Just type

```
pkill -TERM program_name
```

  or

```
killall program_name
```

  In addition, **pgrep** can be used to list running processes.

Forcing the X Window System to Quit

- To kill and restart the X Window System, press **Ctrl+Alt+Backspace**. Lots of programs need X-Window, so using this will cause most running programs to quit or crash.
- You should probably try to reset your computer after this: type

```
shutdown -r now
```

  to do this. You may need to be root.

- If you have access to the command line, you can also type

```
sudo /etc/init.d/(gdm or kdm or xdm) shutdown
```

  to stop X.

- Note that the ability to restart X by pressing Ctrl+Alt+Backspace can be disabled from X's configuration file. In the X.Org implementation, a "DontZap" option can be added to xorg.conf to do this. If Ctrl+Alt+Backspace is not working, this may be the reason why.

If you do **Ctrl+Alt+Backspace** and you have a login manager like XDM enabled usually you will be thrown back to that after x RESTARTS. If you started X by typing startx this will usually throw you back to the command line

Ctrl+Alt+Backspace is useful for restarting X after changing the resolution, for example, too.

Alt+SysRq

If all else fails, you may still be able to communicate with the Kernel.

To use, just press the Alt and SysRq (PrintScreen) keys and one of the keys listed below, all together. You should try switch into a terminal window, if possible, first by pressing **Ctrl+Alt+F2** *(use Ctrl+Alt+F7 to return to the desktop)*.

If your computer is not responding you can use the following steps to perform a reboot with less chance of fscking your files. This is a suggested method:

1. Press Ctrl+Alt+F2 to switch to a terminal window. Sometimes, this is not possible.
2. Press Alt+SysRq+**R** to get the keyboard
3. If pressing Ctrl+Alt+F2 before failed, try it again now.
4. Press Alt+SysRq+**E** to term all processes.
5. Press Alt+SysRq+**I** to kill all processes.
6. Press Alt+SysRq+**S** to sync your disks.
7. **Wait** for OK or Done message. If you don't see a message, look at your HDD light to see if Sync made a difference.
8. Press Alt+SysRq+**U** to umount all disk drives.
9. **Wait** for OK or Done message. If you don't see a message, in 15-30 seconds, assume disks are unmounted (or that an unmount is not posssible) and proceed.
10. Press Alt+SysRq+**B** to reboot.

The Letters used spell REISUB - use the mnemonic *Reboot Even If System Utterly Broken*. Another popular one is *Raising Elephants Is So Utterly Boring*, or just spell BUSIER backwards.

The Full list of available commands is shown below. It's largely taken from Documentation/sysrq.txt [2] in the Linux Kernel Sources.

Alt+SysRq+	Action	Uses
R	Un**R**aw	Turns off keyboard raw mode. This allows input from the keyboard even if X-Window has crashed.
K	SA**K** - Kill all on console	Secure Access Key - Kills all programs on the current virtual console. This is useful if you want to make sure there are no programs waiting on the console to grab your password, or if a process won't let you switch consoles.
S	**S**ync	Attempts to sync all filesystems. This lessens the chance of data loss and fscking. Syncing is complete when "done" or "OK" is printed.
U	**U**mount	Attempts to remount all filesystems as read-only. Umounting is complete when "done" or "OK" is printed.
B	Re**b**oot	Will immediately reboot without syncing or unmounting any disks. Before using this use Alt+SysRq+S and Alt+SysRq+U to avoid data loss.
C	**C**rashdump	Will perform a kexec reboot, in order to take a crashdump. Before using this use Alt+SysRq+S and Alt+SysRq+U to avoid data loss.
O	Power **O**ff	Turns off the computer without syncing or unmounting disks. Before using this use Alt+SysRq+S and Alt+SysRq+U to avoid data loss.
P	Show **P**c	Attempts to dump all registers and pointers to console.
T	Show **T**asks	Attempts to dump a list of all tasks to console.
M	Show **M**emory Info	Displays memory info to console
V	**V**oyager processor info	Dumps Voyager SMP processor info to your console.
0-8	Kernel Error Verbosity	Set the console log level for kernel messages. Setting to 0 only shows messages like PANIC and OOPS

F	OOM Kill	Calls oom_kill to kill a memory hog process
E	T**e**rm	Sends SIGTERM signal to all processes.
I	K**i**ll	Kills (sends SIGKILL signal to) all processes.
L	Kill + Kill Init	Kills (sends SIGKILL signal to) all processes, including init. You will not be able to do anything else after using this!
N	**N**ice	Make real-time processes nice-able.
H	**H**elp	Prints some help

Certain systems (systems administrated by a department or group other than yourself, e.g. a corporate or government location) may remove the ability to use the SysRq key combinations in order to secure their system from low level commands. If you try a few commands and notice nothing is happening, or error messages occur, this may be the case.

The Big Red Button

The final option is, of course, the power button on your computer's physical exterior. Modern PCs have ACPI buttons, which just send the shutdown request to the kernel. If the ACPI daemon is listening and correctly configured, it can signal `init` and perform a clean shutdown.

If ACPI doesn't work, you will need to cut off electrical power. This is done through a "hard switch" (generally labeled "reset"), possibly using the same button as before but holding it pressed for 5 or so seconds. Taking out the battery (on laptops) and physically disconnecting all power cables is, of course, the only method guaranteed to work. (Make sure no other method listed above works. Even as a last resort, unplugging your computer while it is running is still not recommended.)

Resetting without shutting down can cause problems with the file system. To try and fix this problem, fsck will be run when you next boot up, and journaling filesystems will attempt to complete or rollback files which were changing.

References

[1] http://docs.kde.org/stable/en/kdebase/ksysguard/usingtheksysguard.html
[2] http://www.kernel.org/doc/Documentation/sysrq.txt

Linux Guide/Installing software

Obtaining and Installing Applications

There are many thousands of programs available for a Linux system and for several of the major distributions many thousands of these will be available in "pre-built" (ready to run) form through the appropriate *package management system*.

Package Managers

Package managers simplify the task of organizing the software installed on the system. For example, the Synaptic package manager for Debian.

apt

On a Debian, or Debian-based system, it might be possible to install a program by simply typing a command such as:

```
apt-get install mtools
```

In this case, the Debian package manager "apt-get", will check to see if such a program as "mtools" is available, and if so, it will check whether there are any other pre-requisite programs (dependencies) required for this program to run. If there are any dependencies, the package manager will check to see if these are available and in turn whether each of these has any dependencies, and so on, thus building a list of the required software to successfully run the required program. The package manager checks to see which of these are already installed, and for those not installed, it automatically downloads and installs each one. Ubuntu Linux (which is based on Debian Linux) can use this method, or it can use the method:

```
sudo apt-get install "whatever"
```

rpm

Or, on a Red Hat-based system, such as Fedora Core, you can use the 'Redhat Package Manager' (rpm). To install something, download an rpm package, and type a command like:

```
rpm -ihv mtools.rpm
```

To upgrade a package, use:

```
rpm -Uhv mtools.rpm
```

And to search your packages to find out which version of something you've got, do something like:

```
rpm -qa | grep mtools
```

Building Programs from Source

Main page: ../Building software

Sometimes, pre-built binaries are not available or suitable, and in this case it may be required to build the program from its source code. Instructions for how to do this are normally included with the program, but most programs can be compiled using the commands:

```
./configure
make
```

Normally you can then install it to the computer by using the following command as the root user:

```
make install
```

Common software

Main page: ../Common software

See also: ../Software equivalents

Linux Guide/Runlevels

Starting Linux (Booting)

When a computer is turned on, it normally begins the process of "booting" by reading software from the system's hard disk (or other non-volatile storage medium), and loading it into memory and then executing it. In this process the kernel is loaded and then various other system programs may be loaded.

A Linux system can be configured to boot up to one of several available "run levels":

```
0     Halt the system.
1     Single-user mode (for special administration).
2     Single User Mode with Networking
3     Multi-User Mode - boot up in text mode
4     Not yet Defined
5     Multi-User Mode - boot up in X Windows
6     Reboot.
```

Debian

The following quote from *The Debian GNU/Linux FAQ* explains how to add a custom procedure to a Debian or Debian based system boot:

Suppose a system needs to execute script foo on start-up, or on entry to a particular (System V) runlevel. Then the system administrator should:

- *Enter the script foo into the directory /etc/init.d/.*
- *Run the Debian command update-rc.d with appropriate arguments, to set up links between the (command-line-specified) directories rc?.d and /etc/init.d/foo. Here, '?' is a number from 0 through 6 and corresponds to each of the System V runlevels.*
- *Reboot the system.*

The command update-rc.d will set up links between files in the directories rc?.d and the script in /etc/init.d/. Each link will begin with a 'S' or a 'K', followed by a number, followed by the name of the script. Scripts beginning with 'S'

in /etc/rcN.d/ are executed when runlevel N is entered. Scripts beginning with a 'K' are executed when leaving runlevel N.

External links

- The Debian GNU/Linux FAQ Chapter 11 - Customizing your installation of Debian GNU/Linux [1]

References

[1] http://www.debian.org/doc/FAQ/ch-customizing.en.html

Linux Guide/Operating modes

Text Console Mode

When running Linux in a text console mode, the user is presented with a "command shell" and is able to type commands and have the system respond appropriately. In this example text console interaction, the user has typed the command "date" and the date and time was then displayed:

```
root@3[root]# date
Sun Nov 14 16:29:43 EST 2004
root@3[root]#
```

In many cases, accessing a system in text console mode may be more appropriate than running a GUI system. Often, such systems are operating as "servers", examples of which include web servers or database servers. Servers are not normally accessed directly by people, except those that tend to their operation, instead their primary function is to operate in a network context.

GUI Mode

When running a Graphical User Interface (GUI) mode, a range of options are presented to the user through a visual interface consisting of menus, icons and other visual and textual representations.

When a system is used as a general purpose desktop environment, a GUI is normally considered desirable as many feel it provides a more intuitive way to perform many tasks and provides a pleasant, colorful environment enabling rich graphical representations of various documents. Even so, advanced users who operate a GUI enabled system will often spend a significant time entering commands in text console mode, often through a window in their GUI environment known as an xterm in the X Window System.

Boot Messages

As a system boots it typically displays a series of messages indicating the status (success or failure) of the various attempts to load software, and communicate with the various peripheral devices that may be attached to the machine. These messages can be retrieved at a later time by using the **dmesg** command at a console prompt.

Once the boot process is complete, most systems will display, in one of various forms, a *login prompt* and challenge the user to provide the correct password for that user to access the system.

Linux Guide/System administration

Many Linux distributions come with the popular Apache web server program pre-installed. It is quite common to use this in conjunction with MySQL and PHP (or Perl or Python) to build websites. This is known as the LAMP stack. Instructions for installing and running Apache can be found at the documentation page [1] of the Apache website [2].

External links

- Guides to setting up Apache, PHP, and MySQL on Ubuntu Linux: 1 [3] 2 [4]
- Information about Apache on Fedora Core Linux: 1 [5] 2 [6]

References

[1] http://httpd.apache.org/docs-project/
[2] http://httpd.apache.org/
[3] https://help.ubuntu.com/community/ApacheMySQLPHP
[4] http://strdoc.net/ubuntu-apache-php-mysql-server/
[5] http://www.flmnh.ufl.edu/linux/install_apache.htm
[6] http://docs.fedoraproject.org/selinux-apache-fc3/

Linux Guide/Security

Linux, compared to some other popular systems, is very secure, thanks to its UNIX heritage. However, while infecting one's system is difficult, it is not impossible. A mindset of paranoia is usually the most appropriate one with regard to security. In addition, viruses are far from the only part of security. Proper configuration and special care in multi-user systems with many clients are also to be considered.

Keeping Up-to-Date

No software is perfect. One will almost always find security holes in any sufficiently complicated piece of software. Thus, it is important to stay on top of all security updates provided by your distribution. Many distributions provide update tools to automatically check for updates. Your system may be able to update many packages automatically. Setting up automatic updates is normally a very good idea. Check if your distribution provides a security announcement mailing list, which will inform you of important updates.

Anti-Virus

Viruses are very uncommon on Linux and in general, you should not have to worry much about your system being compromised by one. However, your Linux system may act as a file-server for Windows clients or as an email-server that delivers email to Windows clients. In that case, one should ensure that files and emails are clean to prevent infection to your clients.

Samba has ability to plug into the popular free Linux anti-virus system, ClamAV. Your distribution likely has tools to configure this for you. Consult your distribution's documentation for more information.

Simple Firewall

At minimum you should employ a simple firewall. Below you will find a sample script to provide a *simple* firewall. I have used this script as a base for more complex firewall schemes on slackware and debian systems. It is a *very* simple script, perhaps others will feel compelled to contribute more advanced and specific use scripts as well.

```
#!/bin/bash
# Basic script to keep the server secured

# Flush the current tables.
iptables -F

## Default policy to drop 'everything'uyuy
iptables -P INPUT DROP
iptables -P FORWARD DROP
#iptables -P OUTPUT DROP #BLOCKS ALL 'OUTBOUND' TRAFFIC

## Allow local loopback communications (localhost).
iptables -A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT

## Allow established connections with stateful connection tracking
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

## Drop packets of unknown origin which are not TCP/SYN related.
iptables -A INPUT -m conntrack --ctstate INVALID -j DROP

## Allow offered service clients to connect to ethernet interface.
## You may need to change your interface from eth1.
# HTTPd & SSL
#iptables -A INPUT -p tcp --dport 80 -i eth1 -j ACCEPT
#iptables -A INPUT -p tcp --dport 443 -i eth1 -j ACCEPT

#end script
```

First we flush the current tables, which basically means that your rule-set is about to change. We then establish 2 rules to drop (*not reject*) all traffic coming into or being forwarded through our computer. We would like to access the Internet so we create 2 more rules the first which allows traffic into our computer which is *distinctly associated* with our *self initiated* outbound requests. The second rule takes care of our localhost... our computer can access our computer and can make use of the network loopback interface. The final 2 rules allow port 80 http and 443 https, they are commented so the ports are not opened by default.

If you decide to utilize this simple script as a base reference please contribute to this page and topic. I have housed this script in several locations each has it's organizational pros and cons. I have placed this file in /root/ and in /root/bin/ and /etc/firewall/ .

Internet Technologies/SSH

SSH is a secure replacement for Telnet and rsh. All communications between the client and server are encrypted. To access an SSH client (usually OpenSSH) in most Unix OSs, type `ssh user@host.com` in a terminal window. If you don't specify the username, the user that entered the command (`$USER`) will be used. In Windows, you will need to download a 3rd-party utility such as PuTTY [1] or Cygwin. Find more information in the ssh(1) man page. On other Operating Systems (smart phones for example, you will have to use a webbased client [2]) There are several SSH apps for Android, including ConnectBot, Dropbear, ServerAssistant, and the Telnet / SSH Simple Client.[3]

Uses

SSH is actually so much more than just a way to access a remote shell securely. It can be used for lots of other ways to transfer information securely. It includes a neat utility "scp", which stands for secure copy, which is a great way to copy files between machines. It works almost exactly like your default unix cp command. scp also allows you to copy a file from a remote host to a remote host. An example of scp:

`scp user@host.com:~/files/ .` which means copy files from user's home directory on the host.com machine from his files/ directory (it will copy ALL files from the files/ directory to the CWD (current working directory)).

Another great use is to use it to encrypt the transport of any data from one machine to another. As an extreme example, you can use SSH to remotely move a disk from one machine to another (akin to ghost, but securely). This may not be the best use of SSH, or the fastest way to transfer data from one machine to another over a network, but it shows you how powerful SSH can be.

Another great feature is port forwarding. This allows you 'redirect' communication to and from a local application through SSH to another host. So, with SSH you can secure otherwise insecure communications over an encrypted 'tunnel'.

Using SSH

The secure shell client is conveniently called `ssh`.

```
ssh user@host
```

Using SFTP

SFTP has *nothing* to do with FTP. SFTP merely works like FTP, meaning you use it as you would FTP. Using SFTP requires only the SSH server. The FTP server is irrelevant to SFTP. Files are transferred as binary by default.

```
sftp user@host
```

Using SCP

`scp`, aka Secure Copy, works just like `rcp`. Don't know what `rcp` is, then don't fret.

- Copy to a remote host - You must use the colon. REMOTE_PATH is not necessary and all REMOTE_PATHs are relative to the user's home directory.

  ```
  scp FILE_PATH user@host:REMOTE_PATH
  ```

- Copy from a remote host

  ```
  scp user@host:REMOTE_PATH LOCAL_PATH
  ```

Note : If your filename contains spaces then, use scp like this

- file name is */media/sda6/Tutorials/Linux Unix/linux_book.pdf* then destination directory is *home/narendra/data*

  - `$scp user@host:"/media/sda6/Tutorials/Linux\\ Unix/linux_book.pdf" /home/narendra/data`
- file name is */home/narendra/linux_book.pdf* then destination directory is */media/Tutorials/Linux Unix/*
  - $ `scp /home/narendra/linux_book.pdf user@host:"/media/Tutorials/Linux\\ Unix/"`

Note : If you want to copy the whole directory then use

- `scp -r user@host:"<source_dirname>" <destination_dirname>`

Creating SSH Keys

Although SSH can be used with passwords, doing so is not recommended, and many servers will not allow password logins. Instead, use a key - this is more secure, and more convenient.

To create an SSH key...

Most modern Unix systems include the OpenSSH client. To generate a key, run:

```
$ ssh-keygen -t rsa
```

This will store your private key in `$HOME/.ssh/id_rsa`, and your public key in `$HOME/.ssh/id_rsa.pub`. You can use different filenames, but these are the default filenames, so it's easiest to not change them.

Permissions

Because the security of your private key is so important, SSH will not work if file permissions are insecure. SSH will create files and directories with the appropriate permissions, but sometimes things will go wrong. To fix permission issues:

```
$ chmod 600 ~/.ssh/KEY ~/.ssh/KEY.pub
$ chmod 700 ~/.ssh
```

Establish Trust

To log into a remote server, you'll need to put the public key on that server.

ssh-copy-id

The easiest way to do that is using `ssh-copy-id`. This requires some alternate form of authentication, usually password (since you haven't got a key on the server you cannot use key authentication yet).

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub user@host.net
```

Manual

- Assume the directory is not on the destination server

  ```
  ssh user@host "mkdir ~/.ssh && chmod 700 ~/.ssh"
  ```

- Upload your PUBLIC key only (not your private key)

  ```
  scp ~/.ssh/KEY.pub user@host:.ssh/
  ssh user@host
  cd ~/.ssh
  cat KEY.pub >> ~/.ssh/authorized_keys
  ```

- Command for advance *nix users only!

  ```
  cat ~/.ssh/KEY.pub | ssh user@host "cat >> ~/.ssh/authorized_keys"
  ```

SSH Personal Configuration

You don't need to set up a `~/.ssh/config` file, but it makes authentication easier. The important part is to specify your user name and your private key - if this is specified in the config file, you needn't provide it on the command line. Using HostName, you can shorten the ssh command to:

```
$ ssh servername
```

Example

```
#Specific configuration applied to one host
#This configuration applies specifically to a host which uses Windows Domain login
Host Short_Name
        HostName some_host.com
        User domain\username
        Protocol 2
        UseRsh no
        IdentityFile ~/.ssh/KEY

# Generic configuration that I apply to all hosts, especially on my private LAN
# Of note, the options to forward X11 and the SSH Agent.  X11 forwarding lets you
# tunnel and X session or programs via SSH.
Host *
        User USERNAME
        Protocol 2
        ForwardX11 yes
        ForwardAgent yes
        UseRsh no
        IdentityFile ~/.ssh/key_37_rsa
        FallBackToRsh no
        # In a pesky lab environment, add the following to your config
        # CheckHostIP no
```

You can now ssh into `some_host.com` with just `ssh Short_Name`.

Using an SSH Agent

This part assumes that you are not using a ssh client configuration file and that your keys are protected with a passphrase. An excellent BASH utility script called Keychain [4] automates and simplifies the tedious use of ssh-agents. If your host does not have Keychain installed, ask your administrator. Alternatively you can download and unpack the script into your home directory from the Keychain [4] website.

Using Keychain

- Start your agent on your local host

  `keychain` - honestly you don't need to type this, simply loading your keys causes this to happen
- Access your forwarded agent from a remote host

  `keychain --inherit any-once`
- Load your key

  `keychain ~/.ssh/KEY`

This will prompt you for a password (if you gave your key one!).

- Unload your key

  ```
  keychain --clear
  ```

- Stop the agent

  ```
  keychain --stop
  ```

BASH configuration Change

add the following lines to `~/.bash_login` and `~/.bashrc`

```
keychain
source ~/.keychain/${HOSTNAME}-sh
```

Public-key cryptography

The most significant difference between SSH and Telnet & rsh is in the realm of security. SSH uses RSA or DSA for public-key cryptography.

- The server or domain to which you are trying to connect generates 2 keys (public and private) for a client.
- The public key is given to the client the first time it tries to connect. The corresponding private key is a secret and kept with the server.
- The client sends the packets of data by encrypting it through the public key and this data is decrypted by using the corresponding private key stored there.

Communication from the server to the client is also possible in the same way—the server encrypts using the client's public key and the client decrypts using it's private key.

Setting up OpenSSH with public key cryptography

1. With your distro's package manager, install sshd (or openssh-server) on the server, and on the client install ssh (or openssh-clients). It's likely that they're already installed since they're probably part of the distro's default installation.
2. Make sure the following is there and uncommented (there's no # in front of them) in /etc/ssh/sshd_config on the server:

```
PubkeyAuthentication yes
PasswordAuthentication no
```

1. On the client, `ssh-keygen -t rsa`.
2. Copy where you saved your generated keys/id_dsa.pub to portable storage.
3. Bring the portable storage to the server and mount it as the user you will be remotely logging in as. Don't log out yet.
4. cat portable storage mount point/id_rsa.pub>>~/.ssh/authorized_keys
5. Add either sshd:ALL or sshd:IP of client to /etc/hosts.allowed.
6. Open TCP port 22. This varies depending on your firewall. For Fedora Core, RHEL, and derivatives, this can be done with system-config-securitylevel. For other GNU/Linux systems, echo '-A INPUT -p tcp -m tcp --dport 80 --syn -j ACCEPT'>>/etc/sysconfig/iptables and restart the iptables service. You may wish to run sshd on a non-standard port.
7. If the server's behind a router:
   1. Stop using DHCP and assign a static IP address to your server. See the Gentoo Handbook for instructions if you don't know how.

2. Forward TCP port 22 to your server.

8. (Re)start the sshd service.
9. Test the setup by running ssh user to login as on the server@IP or domain of the server. Tip: If the username that you're logging in as on the server is the same as the one you're currently using on the client, you don't need to specify the user to log in as on the server.

SSH as a Proxy

If you can make an SSH connection, you can (most likely) use that connection as a SOCKS proxy, without any extra setup on the remote computer. Traffic is tunneled securely through the SSH connection. If you are on an unsecured wireless connection, you can use this to effectively secure all your traffic from snooping. You can also use this to bypass IP restrictions, because you will appear to be connecting from the remote computer. Note that DNS traffic is not tunneled.

Pick some big port number (bigger than 1024 so you can use it as non-root). Here I choose 1080, the standard SOCKS port. Use the `-D` option for dynamic port forwarding.

```
ssh -D 1080 user@host
```

That's it. Now as long as the SSH connection is open, your application can use a SOCKS proxy on port 1080 on your own computer (localhost). For example, in Firefox on Linux:

- go to Edit -> Preferences -> Advanced -> Network -> Connection -> Settings...
- check "Manual proxy configuration"
- make sure "Use this proxy server for all protocols" is cleared
- clear "HTTP Proxy", "SSL Proxy", "FTP Proxy", and "Gopher Proxy" fields
- enter "127.0.0.1" for "SOCKS Host", and "1080" (or whatever port you chose) for Port.

SSH from your webbrowser

You can also use ssh from a webbrowser with javascript support even when you don't have a secure shell client. In order to do this you have to install AnyTerm, AjaxTerm or WebShell on the system where the SSH server is running or use a third party service like WebSSH [2].

Further reading

[1] http://www.chiark.greenend.org.uk/~sgtatham/putty
[2] http://webssh.uni.me/
[3] "Top Free Android SSH Tools" (http://www.linuxlinks.com/article/20111022065038482/AndroidSSHTools.html)
[4] http://www.gentoo.org/proj/en/keychain/

- OpenSSH
- SSH, the Secure Shell
- LPI Linux Certification/Secure Shell (SSH)

Article Sources and Contributors

Linux Guide *Source*: http://en.wikibooks.org/w/index.php?oldid=2243509 *Contributors*: Adam majewski, Adrignola, Bryant, Conan, Daniel USURPED, DarkPhoenix, DavidCary, Davidbspalding, Derbeth, Earizon, Fan, Frontier, Greenman, H2g2bob, Hagindaz, Hashar, Herbythyme, Jfmantis, Jguk, Joshr, Loquetudigas, Lyellin, Mattb112885, Mike.lifeguard, Mkn, Pedros.lol, Query, QuiteUnusual, Rastilin, Raylton P. Sousa, Recent Runes, Remi, Richard, Robert Horning, Sameerkale, Seraphimblade, Sk8philsk8, Snarius, Sunny256, SvartMan, Techman224, Uncle G, Webaware, Whiteknight, Wikibooks is Communism, Wikimi-dhiann, Wm, 144 anonymous edits

Linux Guide/Introduction *Source*: http://en.wikibooks.org/w/index.php?oldid=2425320 *Contributors*: AllenZh, Jguk, Lavallen, Mike.lifeguard, Snarius, Webaware, Winston Chuen-Shih Yang, 5 anonymous edits

Linux Guide/Distributions *Source*: http://en.wikibooks.org/w/index.php?oldid=2548696 *Contributors*: Az1568, Darrelljon, H2g2bob, Jguk, Kaylathecomputer, Mike.lifeguard, Pplude92, Snarius, Tchakkazulu, Tweenk, Webaware, 28 anonymous edits

Linux Guide/Distros in detail *Source*: http://en.wikibooks.org/w/index.php?oldid=1690290 *Contributors*: Adrignola, Mike.lifeguard

Linux Guide/Getting Linux *Source*: http://en.wikibooks.org/w/index.php?oldid=1689915 *Contributors*: Atallcostsky, JanEnEm, Mike.lifeguard, Tweenk, Webaware, 2 anonymous edits

Linux Guide/Installing (basic) *Source*: http://en.wikibooks.org/w/index.php?oldid=1968138 *Contributors*: Iwasapenguin, Mike.lifeguard, Webaware, 5 anonymous edits

Linux Guide/Installing (dual-boot) *Source*: http://en.wikibooks.org/w/index.php?oldid=1791787 *Contributors*: Adrignola, Mike.lifeguard, Nat682, 1 anonymous edits

Linux Guide/Installing (advanced) *Source*: http://en.wikibooks.org/w/index.php?oldid=2422813 *Contributors*: AK7, Adrignola, Alavoor, Alex S, Antidrugue, Az1568, Conan, Cspurrier, Daniel USURPED, DarkPhoenix, DavidCary, Dhodges, Frazzydee, Geocachernemesis, Glenn Willen, Google Spider, Kernigh, Krik, Mac, Mike.lifeguard, Mkn, Neoptolemus, Paullusmagnus, Perl, RobbDogg, Rory096, SamE, Scientes, SinnerBOFH, Snarius, SonicChao, Srborlongan, Stevenyu, Taikumi, Tomato86, Tweenk, Webaware, IIIiIIiam Shakespeare, 135 anonymous edits

Linux Guide/Logging In *Source*: http://en.wikibooks.org/w/index.php?oldid=1908003 *Contributors*: Adrignola, Mike.lifeguard, Mikel, Webaware, 1 anonymous edits

Linux Guide/How Linux Works *Source*: http://en.wikibooks.org/w/index.php?oldid=2545462 *Contributors*: Ajhbooks, Fishpi, Jfmantis, Jomegat, Knownot, Komitsuki, Michchampagnie, Mike.lifeguard, Pi zero, Recent Runes, Tweenk, Webaware, Xania, 23 anonymous edits

Linux Guide/Using the shell *Source*: http://en.wikibooks.org/w/index.php?oldid=2402963 *Contributors*: Ajhbooks, Floriopietro, Jwesterham, Mike.lifeguard, PurplePieman, Snarius, Vinchi007, Webaware, Whiteknight, Zoig, 22 anonymous edits

Linux Guide/Linux commands *Source*: http://en.wikibooks.org/w/index.php?oldid=2231161 *Contributors*: Adam majewski, Adrignola, Albmont, Bijee, DavidCary, Djamund, Fazal.kharim, Frozen Wind, Kernigh, Krischik, Nisse, PurplePieman, QuiteUnusual, Reach Out to the Truth, Snarius, Spoon!, Webaware, 17 anonymous edits

Linux Guide/Desktop environments *Source*: http://en.wikibooks.org/w/index.php?oldid=2511174 *Contributors*: Chelseafan528, CommonsDelinker, Jguk, Mike.lifeguard, Snarius, Webaware, 3 anonymous edits

Linux Guide/Getting Help *Source*: http://en.wikibooks.org/w/index.php?oldid=2244482 *Contributors*: Adrignola, Albmont, H2g2bob, Jfmantis, Mike.lifeguard, Thermaland, Webaware

Linux Guide/Freezes *Source*: http://en.wikibooks.org/w/index.php?oldid=2253469 *Contributors*: Az1568, Beland, Doc Dish, Dwlegg, Fitzhugh, H2g2bob, Jguk, LFaraone, Mike.lifeguard, Robert Brockway, Spelemann, Tohd8BohaithuGh1, Utkal Ranjan Sahoo, Webaware, 35 anonymous edits

Linux Guide/Installing software *Source*: http://en.wikibooks.org/w/index.php?oldid=1895383 *Contributors*: Elwikipedista, Jomegat, Mike.lifeguard, 2 anonymous edits

Linux Guide/Runlevels *Source*: http://en.wikibooks.org/w/index.php?oldid=1909333 *Contributors*: Mike.lifeguard, Snarius, Webaware, Whiteknight, 5 anonymous edits

Linux Guide/Operating modes *Source*: http://en.wikibooks.org/w/index.php?oldid=1689988 *Contributors*: Elwikipedista, Mike.lifeguard

Linux Guide/System administration *Source*: http://en.wikibooks.org/w/index.php?oldid=1691654 *Contributors*: Adrignola, Mike.lifeguard, Webaware, WillWare

Linux Guide/Security *Source*: http://en.wikibooks.org/w/index.php?oldid=2525839 *Contributors*: Dnas, Hannes Röst, Mike.lifeguard, Webaware, Wikipediandrey, 12 anonymous edits

Internet Technologies/SSH *Source*: http://en.wikibooks.org/w/index.php?oldid=2459496 *Contributors*: *nix, Aya, Dan Polansky, DavidCary, Jguk, Jleedev, Mike.lifeguard, Narendra Sisodiya, PbS, Pingswept, Spoon!, Tuskentower, Uncle G, 29 anonymous edits

Image Sources, Licenses and Contributors

File:JauntyGUI.png *Source*: http://en.wikibooks.org/w/index.php?title=File:JauntyGUI.png *License*: GNU General Public License *Contributors*: Prince781.

File:KDE 4.png *Source*: http://en.wikibooks.org/w/index.php?title=File:KDE_4.png *License*: GNU General Public License *Contributors*: KDE

File:Xfce-4.2.2.png *Source*: http://en.wikibooks.org/w/index.php?title=File:Xfce-4.2.2.png *License*: GNU General Public License *Contributors*: AVRS, EugeneZelenko, Janizary, Kernigh, Nickel, Sven, Thuresson

File:Xgl window switching.png *Source*: http://en.wikibooks.org/w/index.php?title=File:Xgl_window_switching.png *License*: unknown *Contributors*: liquidat

File:Fedora-Core-6-AIGLX.png *Source*: http://en.wikibooks.org/w/index.php?title=File:Fedora-Core-6-AIGLX.png *License*: unknown *Contributors*: liquidat

File:Adesklets.jpg *Source*: http://en.wikibooks.org/w/index.php?title=File:Adesklets.jpg *License*: GNU Free Documentation License *Contributors*: user:jinxs

File:Fluxbox.png *Source*: http://en.wikibooks.org/w/index.php?title=File:Fluxbox.png *License*: logo *Contributors*: AVRS, Armin76, FSII, Jcb, JorgePeixoto, Luinfana, Mardus, Maxim, Pieter Kuiper, Sevela.p, Sven, Teofilo, 2 anonymous edits

File:Fvwm wiki screen.png *Source*: http://en.wikibooks.org/w/index.php?title=File:Fvwm_wiki_screen.png *License*: unknown *Contributors*: Filus

File:Icewm1220.png *Source*: http://en.wikibooks.org/w/index.php?title=File:Icewm1220.png *License*: GNU General Public License *Contributors*: Dangarcia, Shooke, 1 anonymous edits

License