

# Welcome to the Ultimate Linux Newbie Guide!

---



## ***A Word from the Author***

Hi!

Thanks for visiting the Ultimate Linux Newbie Guide and downloading the offline PDF version. I've been writing the ULNG since 2001 in some way or another, and it's been online since then in various guises. I've had tremendous support since then but it's only been fairly recently that the whole thing seems to have taken off again. Maybe there is a newfound interest in finding out about Linux. Maybe more people are getting fed up Windows or Mac. Whatever the reason I'm very glad to have had so much support and continued comments and praise from the readers. Without you, there would be no reason to write or to keep it up to date. So if you like the guide, please spread the word! Tell your friends, put banners and links on your website and help out on the forums.

Over the years there has been many people that have offered their own time and help with the site – too many to mention here, but your names are on the site. Thank you very much for your kind help, it has been very much appreciated!

This PDF version of the site provides all of the main chapters and a little bit more, but please note that the latest stuff as well as the forums where you can ask newbie questions without fear of being flamed is always available online at [www.linuxnewbieguide.org](http://www.linuxnewbieguide.org).

Thanks again!

Ali Ross

Author, The Ultimate Linux Newbie Guide

## **CONTENTS**

- Chapter 1: What is linux?
- Chapter 2: Why Linux – What's the benefit?
- Chapter 3: Choosing a distribution
- Chapter 4: Preparing to install Linux
- Chapter 5: Installing Linux (Ubuntu 5.10)
- Chapter 6: What's this GNOME thing?
- Chapter 7: What's this KDE thing?
- Chapter 8: How do I get software?
- Chapter 9: How do I install software?
- Chapter 10: Using Linux every day
- Chapter 11: Managing my files
- Chapter 12: I don't know any commands!

## Chapter 1: What /s Linux?

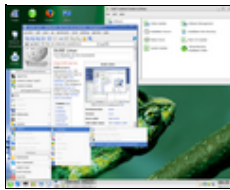
---

### Topics covered in this chapter:

- 1.1 What's An Operating system?
- 1.2 What's UNIX then?
- 1.3 How does Linux differ from UNIX?

### ***Linux, By Definition:***

**Linux is an open source UNIX-like operating system which is popular for it's robustness and availability.**



*Screenshot of OpenSuse 10,  
a popular Linux 'distribution'*

The above definition is probably not going to help you much if you don't know what an Operating system is, and what this UNIX thing is, so let's start at block one: What *is* an Operating system? I promise I won't make it boring!

### **1.1 What's An Operating System?**

Imagine you have a brand new computer. Imagine that nobody had put a disk of any kind into it, **ever**. That would mean that there was no **software installed** on the system. If you switched the computer on; It would beep a few times and then tell you that it couldn't start an operating system. In other words, you can't do anything with a modern computer unless you have some very important software on it — **the Operating System**.

An operating system is the layer between you, **the user**, and the hardware inside the computer (all the chips etc). If you click the mouse over an icon on your screen, the operating system interprets that you want to load the program that you are clicking on. For all of this to happen, The Operating system (or the OS or O/S) must know how to use a screen (to show you what's going on), to use a mouse (so you can move it around and click with it), to use your hard disk drive (to load up the data from it). It must also need to know pretty much everything else about the hardware installed inside your PC, ie: RAM, Floppy/CD drives, keyboards, joysticks, sound cards, scanners, etc.

So when you start typing a letter, for example, you have already loaded up a word processing piece of software. This software is called **application** software and is running 'on top' of the Operating System, but nonetheless, all of the time whilst the word processor application is running, it talks constantly to the O/S for vital information.

Okay, we've pretty much established that an O/S is vital, but what else does an O/S do:

Probably the most basic and yet essential tasks of an operating system is the job of managing our files and data. A basic O/S should be able to do the following with files and directories:

- ◆ Create them
- ◆ Move them to other directories

- ◆ Rename them
- ◆ Copy them
- ◆ Delete/remove them
- ◆ Send and receive files to/from other devices such as Printers/Scanners
- ◆ ...and a bit more.

Got the idea of what an Operating system is? Let's find out about a specific type of operating system called UNIX...

## 1.2 What's Unix Then?



UNIX began life in 1969, in Bell Labs, a division of the American Telephone firm, AT&T. There are now many different types of UNIX:

- Unix History Timeline

Linux is just one type of UNIX, it's a free (as in free speech) derivative. Most of UNIX's different flavours are still being updated and are still in use all over the world today. Here are some popular manufacturers and brands of UNIX, that you may or may not have heard of before:



Sun Microsystems: Solaris



Hewlett Packard: HP/UX



Berkley University: NetBSD and FreeBSD

With the exception of FreeBSD, there is a pretty grand fee to own one of the above versions of UNIX. Mainly large commercial organisations and universities have traditionally used these UNIX variants, however Linux appears to be replacing traditional UNIX on a lot of corporate systems due to it's proven track record, it's growing reputation as a contender to UNIX, and it's low price tag, which can often be free. UNIX is good because it is a **multi-tasking, multi-user** O/S. This means that it can do more than one thing at a time and it can provide all it's services to lots of users at the same time. Modern day workplaces rely on servers to provide a central resource of information and connectivity to users.

So, Unix is pretty clever, huh?

Well, yeah. It is. But then there has to be a down side, yeah? Well. Unix is pretty boring, and traditionally, you needed to learn lots of commands.

**Why don't we all use UNIX today if it's so good?**

**Microsoft**

In 1981, a small company based in Seattle called Microsoft released an operating system, which through chance (Digital Research were supposed to get the contract), were taken on by IBM to go on their new home/small office based computer: the IBM PC (or Personal Computer). This system was also not graphical. It required commands, in a similar format to UNIX or CP/M, but they were a little easier to use, at the cost of being simpler and less powerful. The main pitfall of MS-DOS – (Microsoft's PC Operating System) was, that it had no multi-user, multi-tasking or networking support as standard. By the early 1990's, this was really starting to wear on PC users. UNIX still had far more power than most operating systems of the time, it was just way too expensive, and legal issues between UNIX vendors licensing UNIX was causing headaches and therefore did not have much exposure outside of large organisations, educational establishments and government offices.



During the 80's, a firm called Apple had released another computer, which was separate from the PC, and did not run any PC software, because it relied on its own O/S, named MacOS. This time, Apple had decided to make an operating system that was graphical, and later, incorporated colour, pictures, icons and even sounds! Instead of typing everything into the keyboard as commands, the same actions could be made as clicks and movements with a mouse. A device that the PC had, but rarely used. In a similar sense, and time, the UNIX world, still very different to the market of the PC and the Mac, had a graphical front-end to its command-line console, it was called X, or 'The X Window System'.



An early version of MacOS

In 1990, Microsoft eventually released Windows 3.0 (versions 1 and 2 did not sell well) Windows at the time was a 16-bit, single-tasking, single user, graphical interface built on-top of MS-DOS. but UNIX still prevailed: it was multi-user, multi-tasking and it worked on 32 or 64-bit platforms. It took until 1995, with the advent of Microsoft Windows 95 for Windows to finally go 32 bit, multi tasking, and capable of being multi-user (although not best suited. Windows NT came along shortly after, to do that job).

### **1.3 How does Linux differ from UNIX?**

During the time from 1991 to 1995, many computing or engineering students were accustomed to the power of UNIX and X, at university. In Uni, most students had wonderful new things like E-Mail, The Internet and more. At home, they would have to make do with their 16 bit computers, waiting for all these powers to come to their homes one day.

Enter: Linus Torvalds

Linus was, in 1991, a student in Finland studying Computer programming at The University of Helsinki. Linus used UNIX at University on a daily basis. He got bored of his 386 PC running MS-DOS, and decided to start his own kernel, which is the name for the code at the heart of every operating system that talks to the hardware directly. He wanted to distribute the software freely, because it was a hobby, not a commercial product, and also to see what others thought of it. He finished the first Linux kernel in late 1991. Not only had he made a 32 bit kernel, in which programs could be run he made it do quite a lot to make it look and feel like Unix, but he didn't have any software to run on it.

- [Read Linus' very first post to the Internet in 1991, about the creation of what would become Linux](#)

Luckily, an ex-student in the USA, by the name of Richard Stallman had created a team of programmers devoted to free software, he called this the Free Software Foundation, who believed in making software free to distribute, and free to obtain the source code along with it so that others could make improvements to the software through the Internet. The GNU GPL (General Public License) that the Free Software Foundation made, also stated that the authors of the software could charge for the software, as long as they are willing for it to be freely distributed. By creating community-based software, that has open standards and is subject to peer review, the quality of the software would be good. The opportunities for profit would come from different avenues such as support and consultancies.

Stallman had been busy making a whole suite of software, for example: an editor called emacs, which is very popular today, and the bash (Bourne Again Shell), a command line interface based upon the original Bourne Shell, that comes with the BSD variant of UNIX. The FSF's software was entirely based upon the UNIX software suite, and essentially improved on it. In 1991, the only thing that the FSF were missing to make it a fully fledged operating system was **The Kernel**.

Linus altered the code to work on his platform so that the FSF's code would work with his new kernel, that he ended up calling Linux.



Tux says: Linux is pronounced 'Lih-nucks' **not** 'Ly-nucks'

[Click here to listen to Linus Torvalds pronounce Linux!](#)

In 2001, the year of the first writing of this site, 15 million people used linux, Today (2005), around 30 million people are estimated to be using it.

[Click here to see the Linux User Counter](#)

Now, visit [Chapter Two](#) to see why Linux may be the best thing you ever did with your computer!

## Chapter 2: Why Linux – What's the benefits?

---

### Topics covered in this chapter:

#### 2.1 So what does Linux actually offer me then?

#### **2.1 So what does Linux actually offer me then?**

So, you now know that Linux is a Unix-like operating system, and you know what all that means now, but that doesn't really tell you why you would prefer to use Linux, instead of Windows or Mac OS on your computer.

Linux is far more than a Unix-like operating system because of its unique Licensing system, enter 'Open Source':

#### **Linux is truly an Open Source Operating System:**

- ◆ Protected by the GNU General Public License.
- ◆ Trustworthy computing, as the source code for all software is distributed for free, often with the application.
- ◆ Open standards provoke less buggy software and compatibility with other platforms from vendors such as Microsoft and Apple.
- ◆ Open Source software means no vendor lock-in.
- ◆ No chance of Linux as a whole going out of business, as it is not owned by any one company.
- ◆ **Did we mention that Open Source software is generally (but not always) **FREE** in cost.**  
This means that software can be of high quality for everyone, and money can be made out of support, distribution, training or working with OSS.  
It's a new and revolutionary way to do business, but already, huge names like IBM, HP, Novell, Sun and even non IT firms such as Boeing, Glaxo Smithkline and thousands more are all using it and putting work back into it.

#### **It's a reliable platform for any sort of mission-critical work:**

- ◆ Linux does not have issues with crashing. It deals very well with rogue software, and will close programs down, rather than crashing the whole PC.
- ◆ There are presently no viruses for Linux. Viruses would need to change in their nature drastically for them to be plausible in the Linux arena.
- ◆ It's fast and works on computers that are pretty old, recycle that old PC today!
- ◆ Say goodbye to Spyware, Adware etc
- ◆ Security is built-in by default, not an add on

**Linux is home to some of the best new software, and best of all, most of it is free, have a look at just a few of these great titles:**



[OpenOffice.org](http://OpenOffice.org)

A fully featured Office suite including a Word Processor, Spreadsheet, Drawing Package, Database and Presentation suite. Compatible with Microsoft Office (DOC/XLS/PPT)



### Mozilla Firefox

The browser that reloaded the web – all thanks to Open Source!



### The GIMP

An excellent open source image editor, similar to Photoshop by Adobe. Used to edit the images on this website!



### Videolan

The ultimate media player, play your DVDs/VCDs/DiVX's on any computer connected to a network!



### Gaim

A universal Instant messenger that provides AOL, MSN, Yahoo, Gabber and IRC.



### Evolution

Novell Evolution is a fully-featured Groupware client which includes E-Mail, Calendars, Tasks, Address Books and the ability to connect to a Microsoft Exchange mail server if so desired.  
...and thousands more titles... check out sites like [www.freshmeat.net](http://www.freshmeat.net) for examples.

Now, visit [Chapter Three](#) to decide on which flavour of Linux you want!

Or, if you need more convincing, why not take a sneaky peek at [Chapter 10](#), where we detail how Linux helps out with your life every day.



## Chapter 3: Choosing a Linux Distribution

---

Topics covered in this chapter:

- [3.1 What exactly is a Linux Distribution?](#)
- [3.2 What Linux Distribution should I choose?](#)



As described in the [first chapter](#), we discovered that Linux was a flavour of the UNIX family of operating systems. This chapter talks about what types of Linux are available in the market today. These flavours are called **distributions** and all have their own merits and disadvantages. We will cover the most popular distributions in this chapter.

### **3.1 What Exactly is a Linux Distribution?**

If you ever watch the IT press, you will have probably heard of company names such as [Red Hat / Fedora Core](#), [Mandriva](#), [SUSE](#) (pronounced *Soo-zah*) and [Debian](#).

These are all companies or organisations that have created their own 'distributions' of Linux.

**In any distribution, the fundamentals stay the same:**

- ◆ There is always a Kernel (the core component of the Linux operating system)
- ◆ The default GNU software (tools like ls, rm, etc)
- ◆ General software to be expected of a Linux distribution (text editors, etc)

**What differs from distribution to distribution is usually:**

- ◆ Installation Software (for installing software, or the operating system)
- ◆ General software: (Office Apps, Prog. Languages, Games, Web Software etc)
- ◆ Documentation and Manuals (Quality of, Lack of, Quantity of)
- ◆ Cost – whether you pay nothing, a little, or a lot for a distribution depends on what you need from it.
- ◆ Quality of software (buggy or not buggy software, latest versions of software)
- ◆ Whether it is up to date or not
- ◆ Whether the distributor offers a good channel of support or not
- ◆ How easy it is to use overall.

As you can see, whatever distribution you choose. You get Linux, whatever distribution you choose. You may get a better range of options with distribution X over distribution Y, The choice is for you to decide, and because of Linux's excellent Copying/Licensing properties, you can often download a distribution from the net, or have a copy made for free by a friend without having to part with any cash.





### Why Buy Linux at all then?







Buying Linux often provides benefits that downloaded versions do not provide, such as:

- ◆ Physical manuals (SuSE is particularly good) to help you out when you need a 'covers-all' reference.
- ◆ Vendor support for a particular period of time
- ◆ Distributions like Red-Hat Enterprise give corporations a guaranteed Service Level Response
- ◆ More software than with other distributions
- ◆ Commercial software titles can be included (as it is non-free)

### 3.2 What Linux Distribution should I choose?

Choosing a Linux distribution is a personal thing. It greatly depends on what you want to do with it all.

If you like the look of a Linux Distribution and want more information, click on the appropriate distribution logo below to visit the distributors web site.

<u>Vendor</u>	<u>Logo</u>	<u>Latest Release</u>	<u>User Level</u>	<u>Good Points</u>	<u>Bad Points</u>
<a href="#"><u>Red Hat/Fedora Core</u></a>		FC5	Beginner to Advanced Server	Used to be very popular, easy to use, good installer	Has some annoying quirks, RPM software packaging can suffer from dependency problems
<a href="#"><u>SuSE</u></a>		10.1	Beginner to Advanced/Server	Good all rounder, good manuals & docs, masses of great software, brilliant support. Enterprise version great for corporate use	YAST Installer has had troubles
<a href="#"><u>Mandriva</u></a>		10.1	Beginner to Intermediate	Very easy to use and set up, nice control panel	Not as professional as some others, mainly home oriented
<a href="#"><u>Slackware</u></a>		10.1	Advanced to Server	A professional linux system, made for server market	Quite hard to install and use, Uses .tar.gz packages rather than more popular .deb or .rpm systems.
<a href="#"><u>Debian</u></a>		3.1 (Sarge)	Intermediate to Advanced	Very established Linux distro. DEB packages solve the tedium of the RPM software packaging in Redhat/Suse/Mandriva.	Traditionally known for being further behind than some other distros, but rock solid!
<a href="#"><u>Ubuntu</u></a>		5.10 (Breezy)	Beginner to Advanced/Server	This relative newcomer is top of the pops currently. It starts with a Debian core, but has a more regular release cycle, is more	It is a completely free distro, therefore copyrighted materials such as DVD playing

				polished, is easy to use and has major financial backing.	software do not come as standard with Ubuntu, you must download and install it seperately.
--	--	--	--	---	--

Note: If a distribution is at release 10 (ie: Slackware), but another distribution is only at 3.1 (ie: Debian), this does not indicate that Debian is an old version of Linux. The release numbers are only an indicator of how many releases that particular vendor has made. For example it is quite likely that Debian 3.1 and Red Hat FC4 share the same kernel version and many similar software titles.

Check out [The DistroWatch Linux distribution popularity list](#) for a good idea on what's hot in the world of Linux Distributions right now!

If you are ready to start preparing to install Linux on your system, then move forward to [Chapter 4](#)

If you need some more help, why not visit the [forum](#)?

## Chapter 4: Preparing to install Linux

---

### Topics covered in this chapter:

- 4.1 Different types of Distribution and installing software with them.
- 4.2 What sort of computer will I need for Linux?
- 4.3 Compatible CPU types
- 4.4 How much memory do I need?
- 4.5 How much hard drive space do I need?
- 4.6 What sort of graphics card is best?
- 4.7 Will my sound card work with Linux?
- 4.8 Will my network/wireless/modem work with Linux?

### **4.1 Different types of Distribution and installing software with them.**

As described in the previous chapter, there are lots of different types of Linux distributions floating around, and yes, this generally means that they all have different Installers to put them onto your PC, Mac or Alpha based computer.

This chapter focuses on installing a Debian based distribution, but gives a short insight to other distributions as well.

Debian is one of the oldest distributions out there. Slackware and Red Hat are pretty much the only other two that come close in age, Debian has stuck the test of time, and does not look like it's going anywhere soon. It's a free distribution (which will not change, because of it's license) and it has thousands of developers world wide.



The Debian Logo

A major difference between Debian-based distributions (such as Ubuntu and Linspire) is the fact that they use the **DEB** package management system to install software.

Installing software via binary packages, or if you need the source code, via source packages, is very convenient for most Linux users, because it means that you don't have to compile the source code of an application to get it working. In most cases, you can simply click on a package to install one via a tool in a GUI, and the software will be installed. Here we will discuss two major package systems, however, there are others, such as Slackware's .tar.gz based system, and BSD's ports system.

When the .DEB package format is combined with software such as APT or Synaptic, the .DEB system works very well at resolving things called *dependencies* (software that needs other software, in order to work).



The RPM Logo

Red Hat based distributions such as Fedora Core, SuSE and others use the **RPM Package Manager**, previously known as the Red Hat Package Manager. Packages have a .RPM extension (for example gimp-2.05.i386.rpm) are packaged binary applications (sometimes they package source code as well).

For software that is not bundled with your Distribution, you can usually obtain more titles from places like [www.rpmfind.net](http://www.rpmfind.net) and [www.freshmeat.net](http://www.freshmeat.net).

Distributions like Fedora Core seem to provide less packages for their distributions than the DEB based ones. This is mainly due to something called the *Debian Universe* which is a large repository of software which is available to the 'wild'. RPM based users often have to visit third party websites to download packages, as well as any dependent packages (**dependencies**). Both the use of the Universe, and third-party websites can have issues, because they contain software that is not guaranteed to work with your distribution, and may cause unexpected results – so be careful if you download software from other sources!

Distribution	Package manager
Red Hat/Fedora	RPM
SuSE	RPM
Mandriva	RPM
Linspire/Lindows	DEB (with Click N' Run)
Debian	DEB
Ubuntu	DEB

Figure 4.1: Popular distributions and their software packaging systems

Both RPM and DEB packages are very widely used in the Linux arena, most of which can be installed simply by using software, like the 'Add Applications' menu in Ubuntu.

#### 4.2 What sort of computer will I need for Linux?



*Only Kidding!*

This question has a lot of answers. The bottom line is:

*Depending upon what you want to do with Linux, the system requirements can range from an old Intel 386 to a state of the art Pentium 4.*

This section of the chapter will go through all the major parts of hardware attached to a typical PC and detail what is expected to run a typical modern Linux desktop, starting first, with the CPU.



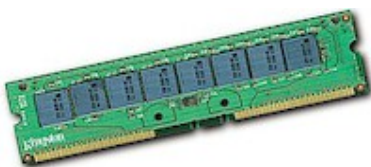
### CPU (Central Processor Unit)

Many types of CPU have been adapted to work with Linux, here are just a few:

Intel	Motorola/IBM	Alpha
i80386DX/33	PPC/Amiga/Apple	Compaq/HP servers
i80486DX/33	G3	Sun Sparcs
i80586/Pentium/AMD K5	G4	
i80686/Pentium II/III/4 AMD Athlon (and AMD64)	G5	

*Figure 4.2: Some types of CPUs that Linux will work with*

Linux was originally devised on an Intel 386, this however does not necessarily mean that Linux works better on a PC than a Mac. Today's Linux desktop is most popular on Intel, Mac (G3–G5) and AMD64 processors, therefore, most of the common software is actively developed for these platforms. If you have another platform such as a Sparc, Amiga, Atari or ARM based processor, Linux will no doubt be different in that many software titles may not exist for that given platform, or software is older than that of the most popular platforms.



### RAM (Memory)

Most Modern day Linux distributions will require a minimum of around 64–128MB to use it to a reasonable degree, but if you wish to use Linux for non-graphical based uses, such as running a web server, or firewall, you can run a basic installation of Linux from just 4 MB RAM. If you're going to be serious about Linux, and want optimal performance, then as with any software, the more RAM you point towards it, the better it runs. Ideally, if you reckon you're going to be a home user, at least 256MB RAM would be a good figure to live with. If you want to do demanding stuff like perform intricate artwork in the Gimp or edit lots of audio, then we're probably talking about 512MB+.

If you have the RAM, Linux *will* use it, and it will be used well, thanks to the superb memory and process management within the Linux kernel!



### Hard Disk Drive (HDD)

As with all things Linux, it's possible to do it in the smallest of setups. Using distributions such as ZipSlack, you can achieve a fully working Linux setup in around 100MB. However, if you want to install a standard workstation installation of any up-to-date distribution, you will probably want at least a 1GB hard disk drive, but in all honesty, a usable desktop system will take around 6GB of space as a typical user will store lots of stuff: Apps, MP3s, Movies, Documents, emails etc and over time, it uses up quite a lot of drive space. Modern Linux distributions easily support new drive technologies such as software RAID and SATA, so you will have no worries about your latest technology drives.

As you are just starting out, you may find it easiest to purchase a new hard drive to install Linux on, (or why not recycle an old 6 or 8 GB drive, it should be enough to begin with).

The reason for using a separate drive is because you are likely to be using another Operating System such as Microsoft Windows or Mac OS. If you wish to use both Linux and Windows/Mac OS (so you can see if Linux is for you), then by far the easiest way to set it all up, is if you have another drive to put Linux onto, so the two don't get mixed up:

Typically, Windows/Mac OS will allocate 100% of a hard drive to it's own use, meaning there is no space left for Linux.

If you don't want to buy a new hard drive for Linux, then you will somehow have to re-allocate some of the unused (free) space on your Windows/Mac OS drive for Linux.



Using something like PowerQuest's Partition Magic which will split partitions of Windows, Windows NT and Linux up any way you want it, and is very easy to use for the beginner.

Alternatives to Partition Magic are QTParted, which can be obtained free with a Live Linux CD (a distribution you can try out without installing it to a hard drive), Knoppix is a popular Live CD.

Finally, The Ranish Partition Manager is a free utility for resizing partitions, and also has an excellent guide on dual-booting Linux with Windows.



### Video Card (Graphics Adaptor)

Any bog standard graphics adaptor will do for linux. Optimally you will want to have an SVGA adaptor in your PC that has enough RAM to support resolutions of at least 1024x768. Graphics Accellerator cards of many types are supported by today's modern distributions for even faster graphics. If you're looking for really good graphics performance under Linux, the NVidia Geforce range are an excellent choice, because they are well supported under Linux by Nvidia, as are the Matrox G range. ATI cards are also popular, however their driver support for Linux does not appear to be as good as NVidia's at current (October 2005). If you don't know what card you have in your machine, visit your device manager in Windows, or System Preferences in Mac OS



### Sound Card

Most sound cards will work fine under Linux, however it appears that some work better than others. Most notably, some USB based external sound systems, and some older style ISA sound cards. Most standard PCI based sound cards work best, the Creative Sound Blaster range are popular and produce good results.



### Using Network Cards, Wireless Cards and Modems with Linux

Almost every network card available should be quite happy with Linux. Note that older ISA based cards require a bit of tweaking to get working. Modern PCI based cards such as the 3com and realtek range will automatically plug and play.

Wireless card support in Linux is generally very good, however, some vendors have made soft-pci, mini-pci or 'wintel' based cards, which do not work well or at all under Linux, because the vendors do not wish to provide drivers or any information for Linux developers to work with.

Cards known to work include the Orinoco chipset, Intersil Prism/Prism II and Cisco based cards. For more information on Wireless compatability under Linux, see [Jean Tourrilhes Wireless LAN resources for Linux page](#).

56K, ISDN, DSL and Cable modems in Linux are also well supported. Notable exceptions are with software based modems, specifically Windows-only based mini-pci 56K modems (rockwell etc). If you want guaranteed compatability with a dialup modem, go for an external Serial (RS-232) modem. The 3Com, Zoom and Diamond Supra series are all good, the same applies to ISDN modems.

ADSL modems (in the UK) are usually Alcatel based USB modems if it was supplied by an ISP. These are somewhat supported by Linux, but not by the vendor and they are sometimes buggy. If you are having troubles, try searching google for Linux support on your hardware. If you want guaranteed compatability, again, go for an Ethernet based external router. Good examples are Belkin, Netgear, Linksys, 3Com and Zoom external ethernet routers. These can be Wireless or Wired.

Cable modems (in the UK) are generally versions of the Motorola 'Surfboard'. If you have a Surfboard, then you are in luck, as it is very well supported, it appears to work best via the Ethernet port, but can also work via USB.

Now, armed with all the information you can get about the hardware in your computer, it's time to get Linux installed!

[Click here to visit Chapter 5...](#)



## Chapter 5: Installing Linux (Ubuntu 5.10)

---

### Topics covered in this chapter:

#### 5.1 Installing your Linux distribution: Ubuntu 5.10

##### 5.1.1. Network setup problems

#### 5.1.2. Installing to a hard drive that already has an Operating System on it

##### 5.2 Watch a video of a similar installation



### 5.1 Installing a Linux Distribution: Ubuntu 5.10

Although this example shows the installation of the Ubuntu Linux distribution, installing most other Linux distributions is a similar process. I have chosen Ubuntu as it is a friendly, free, highly compatible distribution of Linux and at the point of writing, it has been the most popular distribution for quite some time.

Please also note that this tutorial details the installation of Ubuntu Linux on a PC, if you have a Mac, the instructions are similar, but **not** the same. In particular, the tools you will use for partitioning your hard drive may be different.

#### **Step 1: Planning the Task ahead and readying your hard drives.**



If you remember back in Chapter 4 we discussed hard drives and partitioning. If you didn't read that part, skip back and read it now.

#### **WORDS OF CAUTION!**

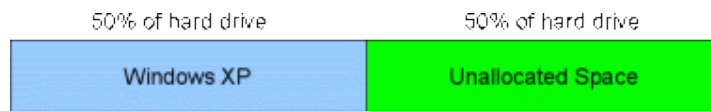
At this early point in the process of installation, you must be aware that you will be working with your hard drive in order to install Linux. If the hard drive contains any important information at all, you **MUST** make a backup of that data before starting. I cannot be held responsible if you delete your own data!

Okay. The first step is to use some software to 'slice' up your hard drive into partitions in order to put Linux onto it.

You will **not** have to perform this step, if you have chosen to use a new/recycled hard drive to install linux onto. Also, if you are brave and wish to simply scrub the operating system (Windows/Mac OS) clean off your computer, then this step is also not necessary, otherwise, proceed forward!.

As mentioned in Chapter 4, Using a tool like QTParted, PartitionMagic or the Ranish Partition Manager is the easiest way to 'slice' up the partitions.

Try and devote as much space to Linux as possible, if for example you have 4GB unused/free space on an 8GB drive, resize your windows partition down from 8GB to 4GB, leaving 4GB for linux. Your hard drive, if drawn as a sideways graph, would look like this:



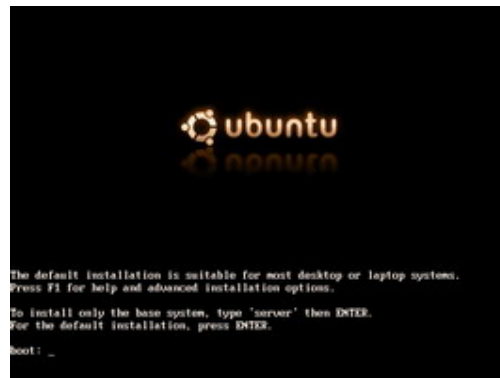
Believe it or not, this first step is actually the point at which most newbies have the most difficulties. The rest of the task is actually pretty easy!

### Installing Ubuntu 5.10: Initiating Countdown, Houston!

Okay. We have partitioned our drive, leaving free space at the end of the drive for Linux, it's time to put that CD in the drive and reboot the PC. If you don't already have an Ubuntu CD, visit [Ubuntu's Download site](#) and download the ISO image of Ubuntu 5.10 (Breezy Badger). It's a full CD big (700MB), so you will need a 700MB CD-R and a CD Burner then burn the image to the CD-R.

If you don't have a CD-R Burner or if you have a narrowband (dialup) connection then [Ubuntu Shipit](#). Ubuntu will ship you a CD free of charge, however postage takes some time. Alternatively, you can pay £4.99 at [Cheeplinux](#) to burn you a CD and to send it out to you.

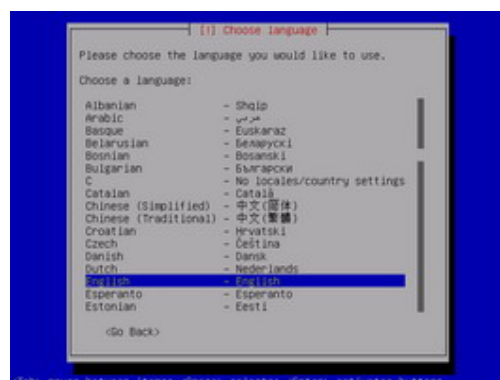
When The CD boots, you will be presented with a welcome screen, You will note that there are further options and compatibility switches available through keywords like 'server', or by using the Function keys. In a standard Ubuntu install, this will not be necessary. Simply hit return (ENTER).



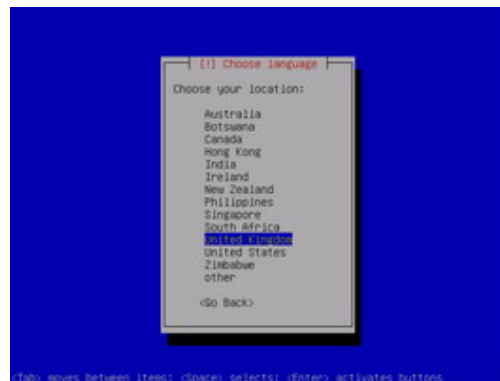
Click on the image to enlarge it

### Choosing the language

The next screen that you will see is a Language selector, That one's easy. Choose English, or if your native tongue is not English, choose your appropriate language by using the cursor keys. Press **RETURN** on the language you want to use.



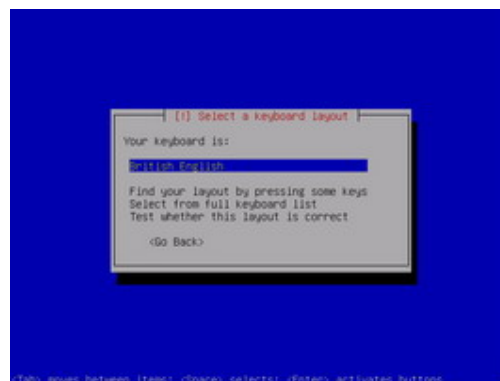
Click on the image to enlarge it



Click on the image to enlarge it

### Choosing the keyboard layout

On the next screen, you will be asked to choose a keyboard layout. Make sure that you choose the correct one for your country or you'll find yourself pressing a @ on your keyboard, when you really want a quote (you get the idea).



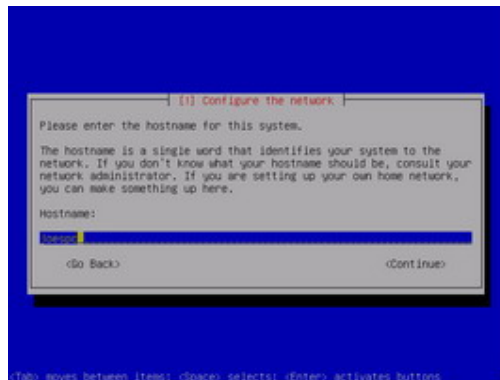
Click on the image to enlarge it

### Network Setup & Giving your machine a name

At this point, Ubuntu's installer will attempt to see if it can find out a few things about your network. Typically, this will be any network cards or cable modems connected to your machine, however, you may have a router set up, so it will try and figure out about all of that too. Ubuntu does this by the standard method, DHCP (Dynamic Host Configuration Protocol). If Ubuntu can't find your network hardware (eg Network Card/Wifi Card/Modem), it'll tell you that Network autoconfiguration has failed. Fear not! – As long as you have the hardware – The Ultimate Linux Newbie Guide will help you out!

Click [Here](#) if you can't get your network to autoconfigure.

If the autoconfiguration was successful, the installer will ask you for a hostname. A hostname is simply the name your computer will take on the Internet. I recommend that you do not take the default 'ubuntu', but personalise it with something you will recognise in the future.



Click on the image to enlarge it

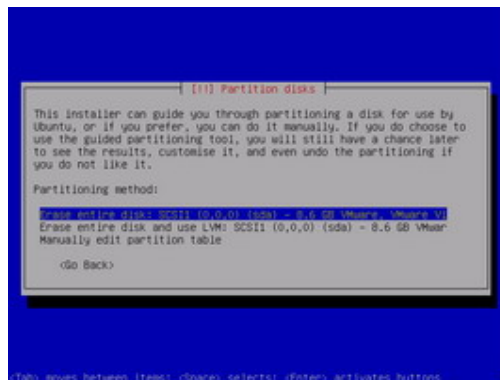
### Telling Linux to use the partitions on your hard drive.

The empty partition you set up at the start of this chapter will now be configured for Linux. There are a number of options. The easiest, but not always the best option, is the first option (Erase entire disk). If you have set aside a separate disk for Linux, then this will be an easy option, and you should give it some good consideration.

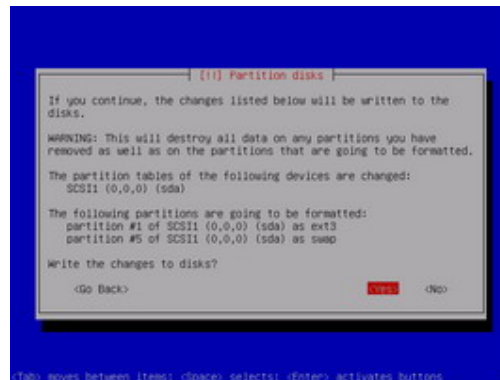


If you wish to manually set up a disk (this is the option you should choose if you are sharing the disk with another OS, for example Windows) then [Click HERE](#) to see how to manually configure the hard drive for Linux.

The next few screenshots detail the 'Erase entire disk' option.



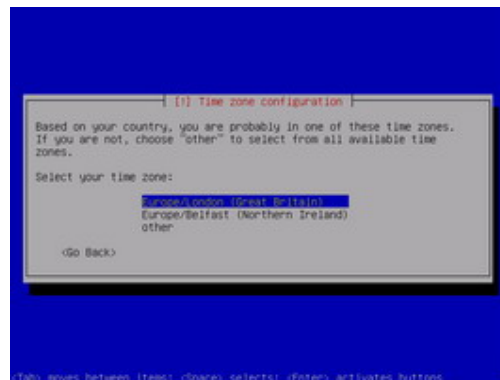
Click on the image to enlarge it



Click on the image to enlarge it

### Choosing a time zone

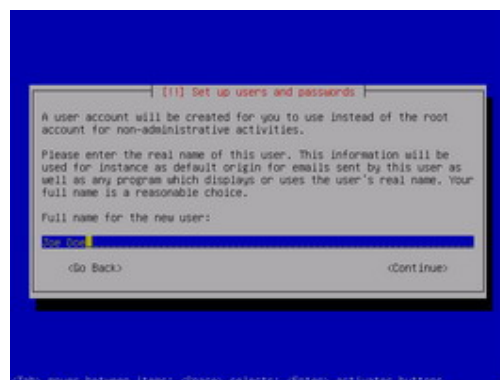
Simply choose the time zone for your part of the world. I live in Scotland, so I use London time. The installer automatically lists the most appropriate time zones for you, so this should be a no brainer!



Click on the image to enlarge it

### Getting Personal!

Believe it or not, but it is important to provide your system with your **real** name. This is because a lot of programs take this information from the system when they automate things, such as setting up email accounts.

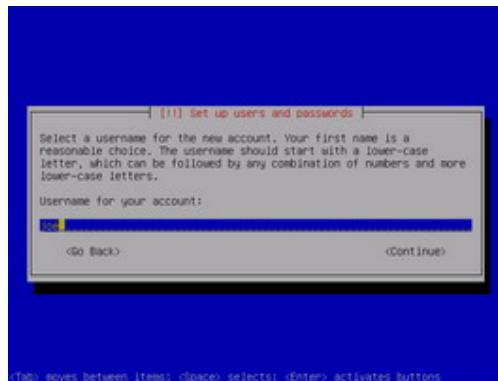


Click on the image to enlarge it

## Username

Now that you've told Ubuntu who you are, you must specify a username for this user. Make it memorable, sensible and usable.

Something to note about this first user on your machine: Ubuntu always sets the first user specified as an administrative user. In this context, this means that *you*, the first account on the machine (you can have as many as you like), will also get administrative privileges on the machine to do things like install software and deal directly with hardware and all things associated with it (eg the kernel). Take this information with a little caution. If you are asked again for your password when doing something in Ubuntu, it is asking you to escalate your own privileges into what Linux calls the **root user**. Root is simply the administrator, but as administrator you have free reign over the system at all times. Do not perform tasks as root unless you know what you are about to do, or unless you have good confidence in the task ahead!



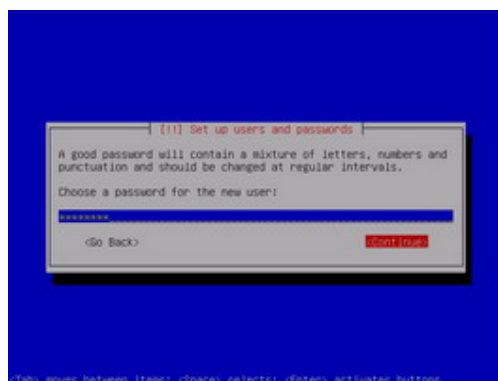
Click on the image to enlarge it

## Password

Next up is passwords. It's important that you follow this advice carefully, because in time to come, you may wish to open up services such as remote access onto your machine. It's simple but believe it or not, still a reasonably effective method of security, do not choose a simple word as a password. Also, don't think that your password is invulnerable to attack, simply because you use l33t-words (eg: h3ll0 for hello). Dictionary attacker bots/programs are wise to this these days. Don't reverse words either, choose random things, like for example, your favourite colour, and your first car, with a few numbers (maybe your year of birth) sprinkled in the middle for good measure. Here is a reasonable password:

blue77volvo740GLS

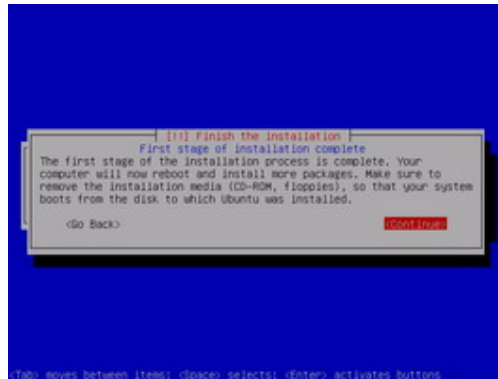
No, I wasn't born in 1977, my favourite colour isn't blue, and I've never had a Volvo 740GLS. However, you get the idea. The password is still important, especially if you ever run any **server** software on your machine in the future.



Click on the image to enlarge it

### Getting there!

By now, most of the software you will need will have been copied to your new partition on your hard drive, your user account will be set up and your regional settings are a distant memory. It's time to reboot the machine. Make sure the cd is removed from the drive and continue onwards!

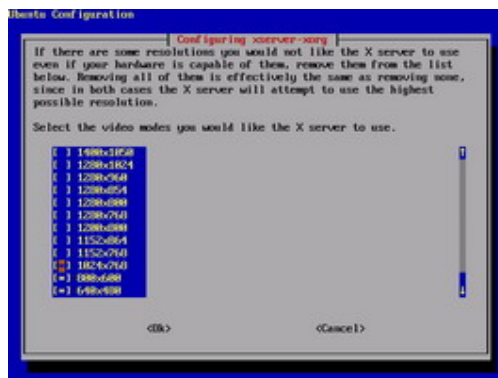


Click on the image to enlarge it

### Setting the screen up

One of the last things you need to set up is your screen. Make sure you know what resolution your monitor can handle (as a guide, a good 17" CRT monitor can handle up to around 1200x800 pixels). Ubuntu will try to guess the resolutions of your monitor, but it is not always great at doing this, so feel free to press space on any resolutions you wish Ubuntu to use. By default, the highest resolution you pick is the one Ubuntu will use.

You may be asked to provide screen refresh rates if your monitor cannot be probed for information correctly. If this is the case, go with the Simple setup unless you really know about the Horizontal and Vertical Refresh ranges of your monitor. You may choose Medium or Advanced, if you know this.



Click on the image to enlarge it

### Welcome home!

This is the login screen, and it's the screen you will see every time you start up Ubuntu. Type your username that you defined earlier, hit return, then enter your password. The system will then log you on.





Click on the image to enlarge it

### The Ubuntu Desktop

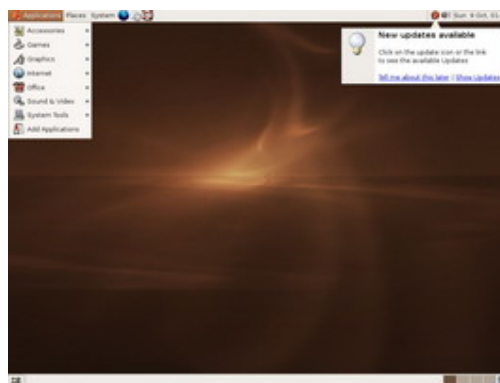
The Ubuntu desktop is a friendly place, which we will cover in later chapters, but briefly, you will see that a lot is very similar to either Windows or Mac OS. If you are used to Windows, your 'Start' menu can be located to the top left, as 'Applications'. All of your software can be located here in much the same way as it is in Windows. The Places menu contains a drop down list of all the major locations on your computer, much like the 'My Computer' dialogue in Windows does.

The System menu provides tools to change the setup of your Ubuntu system, like the background wallpaper, the screen resolution, and more.

The 'System Tray' is located to the top, far right, as well as a clock.

The 'Window list' is the long gray bar at the bottom of the screen. When you launch an application from the Applications menu, it will appear down here so you can switch between programs, just like Windows. Two other mentionable items on this panel include the 'Desktop selector', which allows you to have 4 desktops, each with separate programs running on them. This is really handy to stop cluttered desktops, as you can have for example, your E-Mail on one desktop, Web on another, and perhaps Music & Instant Messaging on another. Finally, you can see the Trash can, which provides a very similar task as to the 'Recycle Bin' in Windows.

Enjoy!



Click on the image to enlarge it

### Want More?

Having problems? Try asking some questions in the [Forum](#).

### 5.2: Video of a similar installation

Check out [this great video](#), showing how to install Ubuntu Linux and Windows on the same drive (please note that the ULNG are not responsible for content outside this site).

[Ready to move forward to Chapter 6?](#)



## Chapter 6: What's this Gnome Thing?

---

### Topics covered in this chapter:

What does Gnome mean?

So what *is* Gnome?

What are the strong points about Gnome?

### What does Gnome mean?

Gnome is an acronym for **GNU Network Object Model Environment**,  
But you didn't really want to know that....

### So What *is* Gnome then?

Here's what the Gnome Foundation say about GNOME at the [GNOME Website](#):

GNOME is a [free] Unix and Linux desktop suite and development platform.

Gnome started life in August 1997, and was brought to life, as most things Linux, through lots of contributors via newsgroups. Both KDE and Gnome have developed into wonderfully mature GUIs in this short period of time.



Tux says: Gnome is pronounced Guh-nome (no silent G)



Ubuntu 5.10 ships with GNOME 2.12 as it's default desktop interface



Gnome in 1999, using GTK+ Version 1

### What are the strong points about Gnome?

- ◆ It is a fully featured Graphical User Interface, which includes a Window Manager
- ◆ Gnome provides a very straightforward interface, which is powerful, yet straightforward. What it may lack in features compared to KDE, it makes up for in usability.
- ◆ The file manager for Gnome, Nautilus (which although still buggy is incredibly feature packed and is really cool!)
- ◆ Gnome is highly customizable and themeable.
- ◆ Unlike KDE, if you wish, you may change the default Window Manager, so that GNOME handles windows in a way you prefer. Typical Window managers could be Fvwm, Enlightenment, OpenBox amongst others.
- ◆ It has thousands of open source applications, games and utilities available.



Gnome 2.6 Today: With GTK+ 2

See an alternative to the GNOME Desktop in the next chapter..



## Chapter 7: What's this KDE Thing?

---

### Topics covered in this chapter:

What does KDE stand for?

So what *is* KDE?

What are the strong points about KDE?

### What does the acronym KDE stand for?

KDE stands for **The *K Desktop Environment***, and is made by the KDE organisation worldwide. KDE is open software in every sense, and the company is made up of the Unix community whom contribute to it. Anyone with the desire to can contribute to KDE, you needent just be a programmer either, you can be a bug reporter, a documentarist, a graphic design artist and many other imaginable posts. You make KDE what it is, and KDE provides back to you. The ideas behind the Gnome project are inherently similar.

### So What *is* KDE then?

KDE is a Windowing Manager and Graphical User Interface for the UNIX operating system, not just Linux (but was made on Linux machines). KDE has been with us since approximately 1997 and is the one of two most popular desktops for Linux. You can freely download and distribute KDE, and these days, there is no restriction on copying it.

At the heart of KDE, lies a graphical toolkit and fully featured programming language called Qt. Qt is found on many platforms including Mobile Phones/PDAs, Windows and Mac OS, but is still most popular on the UNIX platform. Qt is made by a commercial company called Trolltech. Before early 2000,

KDEs main aims are to make a highly usable, user friendly, functional and highly powerful graphical user interface. They definitely have fulfilled that goal.

Users coming from either a Windows or a Mac OS background will feel quite at home with KDE as it has a very similar look and feel to both. For example, the Windows style 'Start' menu is alternated by the 'K' menu. Like MacOS X, it allows for skinning, transparencies and other eye candies, and can incorporate many 'Themes'.



If you wish to switch to KDE, and you use Ubuntu, you can add it to your system by simply installing the package **kubuntu-desktop** in Synaptic.

### What are the strong points about KDE?

- ◆ It is a fully featured Graphical User Interface, much like the Mac OS or Windows Desktop
- ◆ It has an excellent combination File Manager / Web Browser called Konqueror
- ◆ It is very user friendly, offering extensive help almost all of the time
- ◆ It has an excellent control panel for simplistically configuring your Linux system in one, easy to find place.
- ◆ It is highly themeable and customizable, allowing you to make KDE look whatever way you wish
- ◆ It has an integrated sound system, Arts, which is very powerful and allows you to do your own sound synthesis
- ◆ It has thousands of open source applications, games and utilities available.
- ◆ By default, it comes with loads of great applications such as an award winning Email client, Kmail, a complete Office suite, KOffice, editors, games and graphical programs

- ◆ It handles files very well: knowing what to do with even files that are foreign to the operating system
- ◆ It's multimedia capabilities are extensively versatile, allowing for easy movie, sound and tv/capture playback and recording



[Click to view the image of KDE fullsize](#)

Image: Konqueror Web Browser, with amarok multimedia music player 'plugged-in' to the side, so you can play your tunes and surf the web at the same time!. To the left, the Kontact all-in-one Email, Contact Manager, Calendar, Journal, RSS feed viewer and PDA synchronisation tool.

Note: desktop is a dual-screen example.

[Click to move on to Chapter 8...](#)

## Chapter 8: How do I get software for Linux?

---

### Topics covered in this chapter:

How software is packaged up in Linux  
Locating the software that came with your distribution  
Using rpmfind.net to get software  
Using Freshmeat and Sourceforge to get software  
The (tar)ball game, getting software that works for your distro  
Still can't find it? Use google!  
Okay, I got the software. Now how do I install it?!

### **How software is packaged up in Linux**

If you are a Windows or Mac user, you will be used to software being delivered to you in binary format, that is a program that is ready to run by double-clicking on it. A binary has already been compiled (a process done by the programmers of the software, to make it ready for your Operating System).

Linux also uses binary format for most of the software you obtain, but because Linux software is made up of so much Open Source software, it can also come in source code (programming language) format. If you wanted to, you could go in and edit the code to make it do anything you wanted it to, however most people are content with simply compiling it on their own PC and using the binaries that the compilation makes.

### **Why would you want to compile source code applications?**

- You can see what the software is doing – make sure there are no nasties in the software (if you have the programming know-how to understand the code)
- Compiling it will most likely optimise it for your PC, so that the software will run at its utmost performance
- If the software you want has not been specifically released for your platform, eg software was written for PC, but you have a Mac, you can often compile source code (or port it) to your platform

Linux is all about choice, so more often than not, you will be able to choose whether you want to install a **source** version of a given piece of software, or whether you wish to install a binary version of software.

### **Why use binary at all if source code is so great?**

- Installing binary applications is usually done at the click of a mouse, no nasty code anywhere in sight – no keyboard commands etc.
- Some source may not work correctly with your distribution, as your linux vendor may already provide a binary equivalent
- Most Linux distributions now provide most of the software you would ever need within the distribution itself, or on an Internet repository, so the need to go out and download source or binaries is far lesser today than ever before

Software in Linux today is packaged up for easy installation in most cases. The most common formats are as follows:

### **Package Name    Example Distributions that use this type**

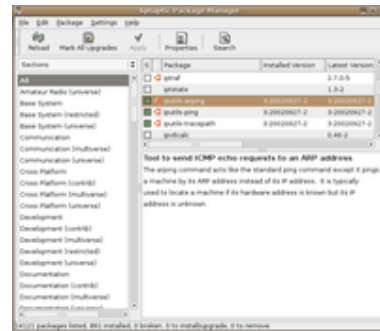
.tar.gz / .tar.bz2	Gentoo, Slackware, Vector
.deb	Debian, Ubuntu, Linspire
.rpm	Red Hat, Fedora Core, SuSE, Mandriva

The next section goes on to demonstrate where you can obtain software for your Linux system, in respect to each packaging system shown above.



Linux has come a long way in recent years to make locating and installing software an easy process. Historically, Linux suffered from two things:

- Each distribution has its own way of installing software, but thankfully they are generally straightforward and are centralised either around the RPM or DEB packaging system:



**Ubuntu 5.10 (DEB): The Synaptic Package Manager.**  
*Lets you easily find and install software for your system.*  
*Click the thumbnail for a larger image*

The great thing about Linux today, is the stark contrast to the way users obtain software between operating systems like Mac OS or Windows.

With Mac OS and Windows you have two choices when obtaining software:

- 1) Purchase it in a store, *or*,
  - 2) Download it from the net
- Either way, you will need to go through a separate installation process for each piece of software you purchase. This is not the case with most Linux distributions today. With specific regards to distributions such as Ubuntu and Linspire (Click-N-Run), you simply start up the Package Manager (eg: Synaptic, as shown above). Click on the title you wish to obtain, and it is downloaded from the net, installed and configured for you automatically. Just think – No going to the store, no InstallShield Wizards, No crazy config files, just click, and go!

This almost sounds too good to be true, but it is almost a perfect system in a modern Linux distribution. Only a few exceptions exist; mainly with RPM-based distributions:

For example, Fedora Core is a well organised distribution, notorious for it's long history in the Linux arena, as well as generally well-thought-out approach. However, Fedora and other similar RPM distributions suffer from the fact that if the software title you require is not provided on the CDs that came with the distribution, then you will have to obtain it from elsewhere. This can sometimes (but to a lesser extent these days) lead to the nasty situation of *"Dependency Hell"*.

Debian based distributions (Debian, Ubuntu, etc) generally have access to the Debian Universe & Multiverse, which put simply is a vast Internet repository of almost all of the Linux apps you could ever want. The best thing about this repository is the fact that it plugs in directly to the APT package system (which automatically downloads software and any other dependent software), and ultimately to your Package Manager, eg: Synaptic.

It is for the reason that you may find yourself one day in a situation where you need to find software from an alternative source than from your distribution, that the rest of the chapter exists. The rest of this chapter discusses where you might obtain software in the .tar.gz and .rpm format, that should (but with no guarantees) work with your distribution.

One final note which is very worthy if you are using Fedora Core, or similar is **YUM**. Yum is the Fedora equivalent of APT. Whilst it does not contain such a large software library as the Debian Universe/Multiverse, it will resolve problems with dependencies. Check out [This Link](#) for more information on YUM.

### **Using rpmfind and RPMForge to get RPM packaged software**

RPMfind and RPMForge are just two of a number of good sites for all non-distribution specific RPM packages. If you use Red Hat Linux/Fedora Core, SuSE, Mandriva, Caldera, YellowDog, ASP Linux, Falsehope, Kondara Linux, Trustix and many many more RPM based linux distros, then these are great places for you to find software, however, be warned that you should use this software at your own risk, as it will likely not have been tested with your specific Linux distribution.

As soon as you go to the RPMFind, you are presented with a search box. Type in the name of the rpm that you want and it generally returns a large amount of results.

Pick an RPM from the list based for your distribution (if there is one), and download it.

There is also a search by category, index, vendor and group search (amongst others). Some of these indexes are handy, but are quite big and can take quite a bit of going over.

### **Obtaining software from Freshmeat.net and Sourceforge.net**

Both of these sites, part of the OSTG network, are open source havens for the Linux user, with hundreds (possibly thousands) of software titles being added or updated every single week.



freshmeat.net in action. Click to enlarge.

If you know the particular software that you desire, you can type in the software's name, or perhaps it's purpose in the search box in either Freshmeat or Sourceforge.

However, if you don't quite know what you are searching for, you can often find what you need by browsing by category.

Pressing the browse link at the top middle of a Freshmeat page will take you to the Freshmeat appindex where literally thousands of applications, utilities, games, multimedia tools and well, everything you could possibly think of is neatly categorised. When you find the relevant category or the relevant search result you need, click on it's title for a page on that program. It'll give you links to the authors homepage, quick links to download it in (usually) various formats, including tarballs, debbs and rpms, however, these sites are most useful when searching for source code.



Tux Tip

After using freshmeat and rpmfind for a while, you'll find it's easiest to *find* what you're looking for on **freshmeat**, and once you know what it's called, go to **rpmfind** and download it. For example. I could go to freshmeat, look up web servers, find out that I needed a program called **apache** and then go to rpmfind.net and key in apache in the search box. Et Voila!

### **The (tar)ball game of getting software**

Most of the time, you will find that the .tar or .tar.gz format is used to zip up source code format software, a .tar.gz file is often referred to as a **tarball**.

Tarballs are usually double-zipped files. This means, that they're first archived with a tool called **tar** (the standard unix archiving tool), and then for best compressions sake, the tar file is then zipped up with gzip (the GNU Zip tool, which is akin to PKZIP).



Tux Tip

You can still unzip/zip WinZip/PKZIP compressed files with Linux, using the **unzip** and **zip** tools, rather than gunzip and gzip. Furthermore, KDE and GNOME both have a GUI tool built in to handle .gz and .zip files effortlessly!

### **Still can't find it? Use google!**

It's happened to the best of us, and in time, it may happen to you. If you are looking for an RPM package, but you can't find one on the authors web page, nor on rpmfind.net. You can only find the tarball and you just can't find it anywhere.

Well, you would be surprised at who actually makes up these RPM packages from tar files. Mainly you'll find that distribution vendors package them, but you'll find other people out there doing it for the hell of it, to help you and the rest of us out. This generally happens if the program is a dog to compile as a tarball.

A word to the wise though: If you are downloading an RPM or DEB package from a third-party, it could have been tampered with, or worse, wreak havoc with your Linux installation – use third party binaries as a last resort and with caution!



**Google** searching for a GNU napster client, Gnapster. Click to enlarge.

### **Okay, I got the software, Now how do I install it?!**

A– ha! For that, you'll need to move on to **Chapter 9: How do I install software?**



## Chapter 9: How do I install software in Linux?

### Topics covered in this chapter:

Installing .deb files (debian/ubuntu/inspire packages)  
Installing .rpm files (red hat/fedora/suse/mandriva packages) with RPM and YUM.  
Installing .tar.gz files (tarballs)

### Installing .deb (Debian packages) and the APT System



Debian's package management system is a very well made packaging system. Much like the one used by the RPM system, however, when DEB packages are combined with the APT System, both the act of obtaining software, and issues with dependent software are almost completely removed.

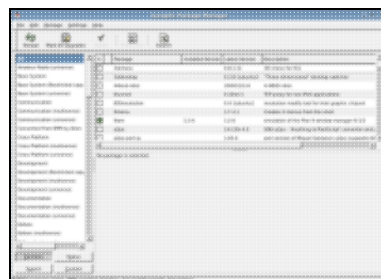
The DEB/APT system is now highly popular, and is found on many other systems other than Debian, including the Ubuntu system we installed in Chapter 5, as well as many others.

These are the main parts to the DEB system:

- ◆ **dpkg** – the main package management program, used mainly to install DEB packages
- ◆ **apt-get** – offers a simple way to install packages, and unlike dpkg, does not work with the .deb package, but uses a file found in /etc/apt/sources.list to obtain the relevant .deb file from the net.
- ◆ **dselect** – A menu driven interface that acts as more than just a frontend to dpkg. Allows for installation and removal of packages
- ◆ **Aptitude** – An ncurses terminal based front end to APT. It's popular for it's user friendly interface and highly descriptive nature.
- ◆ **Synaptic or Adept** – The GUI tools for GNOME and KDE respectively that provide an easy to use graphical front-end to installing software, which most users will feel comfortable with.

Now that you know what the main Debian DEB tools are, we'll step through installing a Debian package first using the Synaptic package manager, which is the default tool for Ubuntu Linux, and then we will briefly cover installation at the command-line shell:

### Installing software with the Synaptic Package Manager:



Click on image for a larger view

To start up Synaptic, click on the System menu at the top left of the desktop menu. Then Click on Administration.

Finally, Click 'Synaptic Package Manager' Once you type your administrator password (which is the same password as the main user of the machine), you will shortly be presented with a screen similar to the one on the left.

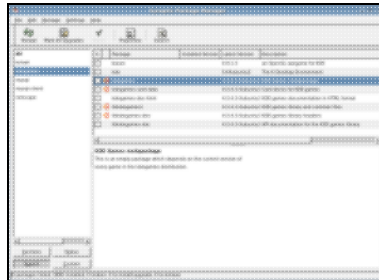
Amongst the many options you see, there is a large list on the left hand side of the window.

This list contains all sorts of different categories of software. If you don't know what you are looking for, start here!

By clicking on the Search button at the right hand side of the toolbar, it is possible to type in the name of a program you know the name of. Note I am typing 'kdegames' here, because I know that I want the software package entitled kdegames.

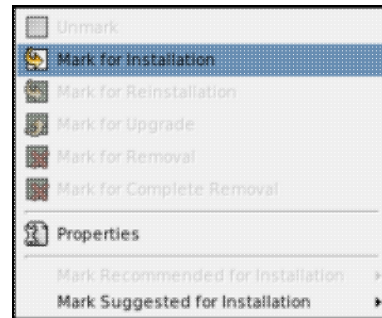


Click on image for a larger view

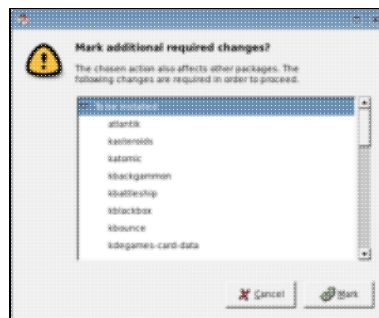


Click on image for a larger view

If you are sure you want to install this title, then right click on it. The menu on the right will appear. By choosing 'Mark for Installation', the software 'kdegames' will be chosen for installation, when you apply these changes.



Click on image for a larger view



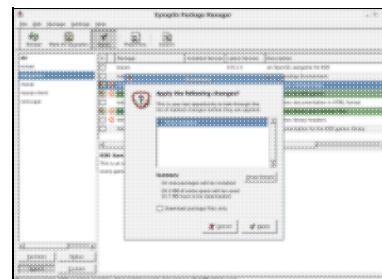
Click on image for a larger view

One you click 'Mark', Synaptic will take you back to the main screen so that you may choose more software to install, if you wish.

If you are ready to install the software you chose earlier, click the 'Apply' button.

Synaptic gives you a warning message, telling you that new packages are going to be installed. By clicking on the Arrow beside 'To be Installed', you can see which ones.

If you are happy with this, click

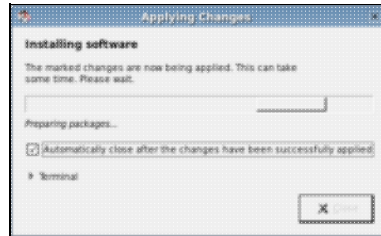


Click on image for a larger view

The example of 'kdegames' was specifically chosen because it is an example of software that has dependencies on other software. Synaptic/APT is showing you this fact and is conveniently telling you that it will also download the extra software shown in the menu to the left to 'satisfy' these dependencies.

You should click 'Mark', in order to continue.

Apply.



Click on image for a larger view

Finally, the Synaptic Package Manager goes out to the Net, or perhaps requests a CD (if the data is to be found on the installation CD of your Linux distro), and then installs the software onto your system. Once the software is installed, you can generally use it directly from the 'Applications' menu. If you can't see it, some older applications may need manually entered into the menu – to do this, right click your Applications menu, and click 'Edit Menu'. You can find the location for most software, by typing **which** followed by the software's program name, at the command line shell. If that doesn't work, try using Synaptic again, locate the software you just installed and right click it, view the Properties, and click on 'Installed Files'.



[Watch a Video of How to install software using Synaptic, and also how to install .tar.gz source!](#)

#### Using APT-GET to Install software at the command-line shell:

Installing a deb package through apt-get is as simple as running the following command at the root Linux prompt:

**\$ sudo apt-get install firefox**

*you will need to input your administrator password* That example would install the most recent version of the program 'firefox'. Apt would also tell you that it needs to download some other software (dependencies) in order for 'firefox' to run.

A few other commands for Apt you'll need are:

**apt-get update**

Updates the APT source information, to tell it about any new software in the APT repositories.

**apt-get upgrade**

Upgrades any old software on your machine to the latest versions automatically.

**apt-get remove [software]**

Removes [software] from your system, and any non-required dependencies.

#### Installing .DEB packages at the command-line shell:

**\$su**

**Password:**

**#dpkg --install package-name.deb**

That should be the package installed, although again, there are dependencies to think about, and as with RPMs, make a note of any dependency errors, download the appropriate DEB package(s) to meet the dependenc(ies) and try again.

#### **Installing RPM files (red hat packages)**





Red Hat, one of Linux's first ever distributors came up with a neat solution to the problems .tar.gz files and compiling has for the normal user. They pre-package the tar file, zip it up and make it do all the hard installation work for you. This system is called RPM and it's the standard software installation method for a lot of Linux distributions today, such as Red Hat, Fedora Core, SuSE and Mandriva.

If you are unsure if you have an RPM system or you just want to check what version of RPM you are using, then try typing the following at the Linux terminal:

```
$ rpm -q rpm
```

You should get a similar answer to this:

```
rpm-4.0.3-5
```

If you get something like 'command not found', then it sounds like you don't have RPM installed, you may be using a Linux distribution that does not use RPM natively, for example, Debian, Ubuntu, Gentoo, Slackware, Mepis or Xandros have a look at the DEB or TGZ sections of this page.



It's important that if you go to download an RPM from the net, always try to get one that was packaged by the vendor of your distribution.

For example, if you go to rpmfind.net and type in **netscape**, and you get 3 RPMs back: One from Red Hat, one from Mandriva and one from SuSE. If you have a Mandriva Linux distribution on your PC, make sure you use the Mandriva one. The reason for this is all down to fitting into your system configuration structure and things called libraries, which vary from distro to distro.

### How to install the package it's self

Okay, let's presume that you have an RPM file ready to install called netscape-4.76-3.i386.rpm

You can install it in the following ways:

- ♦ Open up a terminal/console:

```
su - (to become the root superuser)
```

```
Password: <your root password>
```

```
# rpm -Uvh netscape-4.76-3.i386.rpm
```

```
Preparing [#####] 100%
```

```
Installing [#####] 100%
```

*The options -Uvh stand for the following:*

**U** – Upgrade package if already installed, or install if not installed

**v** – Be verbose about the installation

**h** – show hash symbols to indicate progress of installation

### **Don't use RPM – YUM is easier and better:**

YUM (mentioned in the previous chapter) is a system much like Debian's APT, but for Fedora Core and other RPM based distributions. It makes Dependency problems far less likely for Fedora Users.

Installing an RPM package through YUM can be done by the following steps (again, as root):

```
$su -
```

```
Password: <root password>
```

```
#yum install netscape
```



Note that you do not need to specify the version of software you are installing. YUM goes out to the Internet and automatically pulls down the latest version it can find of 'netscape', and installs it for you, along with any other software you may need, in order to run 'netscape'.

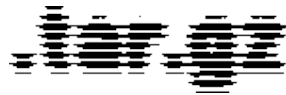
What if I don't want to type commands in to install software via YUM?  
Then use the graphical program, Yum Extender!  
How do you install it, I hear you ask?

### **\$su – Password: #yum install yumex**

Now you can access Yum Extender in the "red hat" menu, under "System Tools."

For further information on YUM, see [This page](#)

### **Installing .tar.gz. files (tarballs)**



.Tar.gz files, or, *Tarballs* are the Unix equivalent of zip files for the Windows world. They can contain any files, but are often used to package up source code for programs.

Tarballs come packaged in five main flavours:

<b>.tar</b>	<b>.tar.gz</b>	<b>.tgz</b>	<b>.tar.bz2</b>	<b>.bin.tar or .bin.tar.gz or .bin.tgz</b>
standard tar archive with no compression.	standard gzip compressed tar archive.	same as .tar.gz.	tar archive compressed with extra compression tool, bzip2.	rarely seen tar archive that contains binary files rather than source.

Most of the time, you will be dealing with **.tar.gz** files.

Here's how to extract (unzip) a .tar.gz file in two different ways:

- ◆ At the [Console](#):

```
$ cd /directory_that_the_tar_file_is_in  
$ tar zxvf tarfile.tar.gz
```

- ◆ In Gnome or KDE (Graphical Desktop):

Right click on the icon for the appropriate tar file in your file manager (nautilus/konqueror).  
Choose Extract (or in KDE, choose Open with Archiver).  
Extract with the tar file with the relevant archiver program for your WM.

Okay, now you've extracted it you have to either:

- a) Compile the source code you just extracted
- b) Run the installer script which is part of the files you extracted

So, how do you distinguish whether you have just extracted a tar file with source code in it, or whether it's a binary, with an installer in it?

Usually, the contents of the .tar.gz file will help you out here – A file containing source code, will often contain a file called 'Makefile' somewhere in the first folder

within the extracted volume. This file is used to compile, or, *make*, the software. A tar file which does not contain source code mainly holds a binary installer file in it, the filename of the installer usually ends in .sh or .pl.

For example, the program VMWare, contains a program called **vmware-installer.pl** in the extracted root folder.

To run the file, you usually need to give yourself 'permission' to run it:

**\$su root (or sudo bash, depending upon distribution)**

**Password:**

**#chmod 755 vmware-installer.pl (change permission)**

**#./install-vmware.pl (runs the installer)**

If you found a 'Makefile', then you need to compile the source code. Here's how to do it:

Most of the time, you will need to use the terminal to compile source, so use an xterm/console/terminal and go into the directory that has been made by the package, eg:

**\$ cd /directory\_that\_the\_tar\_file\_is\_in**

**\$ ls -l**

**Total 302**

**-rwxr--r-- 1 user group 2907 May 21 17:15 mytarfile.tar**

**-rwxr--r-- &nbsp;1 user group 0015 May 21 17:15 newdir/**

**\$ cd newdir/**

at this point, make sure you read the INSTALL file. You'll find that almost every tarball that you download (especially GNU software) has at least a file called INSTALL, COPYING, README and CHANGES

Most of the time the INSTALL file says the same thing, it's a generic process for installing tarballs, but if a program requires to be compiled in a special way, you'll find out in either INSTALL or README.

If you were installing a generic program, extracted from a tarball and presuming that we just changed into our directory, as above, we could do the following to compile the program:

**\$ ./configure**

*(Take good notes here for any configure errors)*

**\$ make**

*(Take good notes here for any compile errors)*

**\$ make install**

*(Take good notes here for any compile errors)*

**\$ make clean**

*(this cleans up after a sucessfull compile)*

**Why compiling is a pain in the ass (for most people), and the problems you might have.**

The above procedure doesn't sound too difficult, and in theory, it shouldn't be. But it doesn't always work.

Most of the time this is because of dependencies on other programs, you need other software (usually programming libraries) to be installed first, in order to compile this software.

Picture this scenario: You attempt to install tar.gzipped game called xtux.

The ./configure bombs, and you noticed on the web site of xtux, and also from the output of the ./configure something about SDL. You're not quite sure what it is, but you go onto freshmeat.net anyways and type in SDL.

You find out that SDL is infact a popular graphics library for X and that it's necessary for xtux.

You download SDL-1.2.3.tar.gz from the SDL website and install that tarball. It installs fine, so you try installing xtux once more. It still bombs out, but this time it gives you a different message: could not find Qt equal or greater than 1.3 on a ./configure.

You check your system for QT version 1.3 or greater. You have 1.3.4 so you should be fine. Why is this error coming up? Well, it's probably because Qt (which is another graphics / programming toolkit) is installed, but is not in the folder that ./configure is looking in. You can edit ./configure yourself to see if there is anything you can do to amend the situation yourself, or try removing Qt, and installing another instance of it from another source.

I find that RPM based distros often put stuff *like* Qt in places that a normal tarball dosen't, so that's often the reason for these compile problems, make sure that if you have installed the normal version of an RPM (binary version), that you also install it's accompanying -dev RPM if you wish to compile .tgz based source against it. By this I mean:

Make sure you have installed qt.i386.rpm and qt-dev.i386.rpm if you are compiling something that relies on QT, as the -dev package will provide the qt source code to the source program you are installing.

### **Dealing with your files and programs**

If you've installed your tarball, RPM or DEB package and you want to run it, or perhaps you are interested in knowing more about the Linux file system including dealing with permissions, move over to [chapter 11 – Managing my files](#).

Alternatively, If you want to know more about using your desktop every day, for office use, for multimedia and more, continue on to [Chapter 10](#).

## Chapter 10: Using Linux Every Day (or, Using The Linux Desktop)

---

### Topics covered in this chapter:

Using Linux for all your office needs

Using Linux for artwork

Playing and editing music/audio

Watching and editing videos

Burning CDs and DVDs

Watching and recording TV

Playing games on Linux

What would be the use in Linux if it didn't have kick-ass software? I'd rather put up with a poorer Operating System than have no software, wouldn't you? Fortunately, Linux is teeming with great software for all sorts of purposes, and most of it is free of charge. This chapter discusses the most popular uses of a modern-day PC, and where Linux fits in with all of this.

### Using Linux for all your office needs

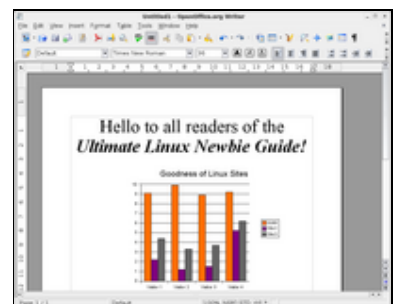
Historically one of the downfalls of Linux was that good office software was lacking, making it an impractical system for Business users. From the late nineties onwards, this is no longer the case. Software like OpenOffice and KOffice are excellent, complete packages, and standalone software like AbiWord and MrProject add further to the great selection of tools.

With respect to Groupware tools, either Novell Evolution or Kontact provide all the features of popular groupware offerings from Lotus, Microsoft or Novell.

### Office Package Software

**OpenOffice.org:** Microsoft Office users will be pleased to know that the most popular Office suite in Linux, OpenOffice.org, supports Microsoft Word Documents, Excel Spreadsheets and PowerPoint presentations.

The latest incarnation of OpenOffice.org (version 2.0) includes a Word Processor (with export to PDF feature), Spreadsheet, Formula Maker, Sophisticated Database, Vector Drawing Tool, Presentation/Slide Creator and more!



How much does it cost? – Nothing! check out [www.OpenOffice.org](http://www.OpenOffice.org) for more information.

Although OpenOffice.org is the most popular office suite for Linux today, it should be noted that KDE sports KOffice, which is similar to OpenOffice in many ways, and features much of it's functionality. GNOME also has various office components such as AbiWord and Gnumeric.

OpenOffice.org is developed in Java and is a fork of Sun Microsystem's StarOffice and is known to be a tad demanding upon machines with lower specifications and therefore you should have a look into these and other alternatives if you have a slower computer.

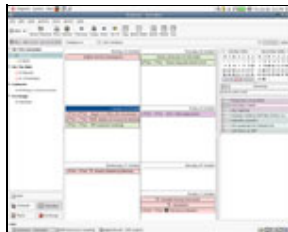
**What if I don't want to use another office or other windows app? – I like My**

### software just fine!

It is understandable that some users may not wish to move from what they know best. Even although OpenOffice is very similar in looks and features to Microsoft Office.

You may find solace in knowing that it is possible to run Microsoft Office as well as a great many other Windows apps on Linux! – Although it is not software written for Linux, it is possible to run it using software from a firm called CodeWeavers. The software, called Crossover Office supports Microsoft Office 95–2003, as well as many other popular Windows titles such as **Adobe Photoshop**, Microsoft Visio, Lotus Notes, Apple iTunes and **Macromedia Dreamweaver**.

### Groupware Suites



**Novell Evolution 2.0** is a groupware suite for performing all of your day to day E-Mail, Scheduling, Contact Management, Address Books and more.

Users can retrieve their E-Mail from a vast array of sources including IMAP, POP3, Microsoft Exchange and Novell GroupWise Servers.

It's overall look and feel is similar to that of Microsoft Outlook, but enhances on a number of features, especially where security is important. It also supports PDA devices and LDAP servers for great connectivity, wherever you are.

The Evolution Groupware software now comes pre-installed with most distributions of Linux.

Amongst Evolution's competitors are:

#### ◆ Kontact

KDE's contact management software that keeps on getting better and better.

Kontact features a Summary overview, a very capable E-Mail client, contact manager, to do list, journal, usenet news reader, note taker, RSS/Syndication client and a mobile device synchronisation tool. Support for Novell GroupWare and Microsoft Exchange are not as mature as Evolution, but are pretty much there in terms of usability.

#### ◆ Mozilla Thunderbird

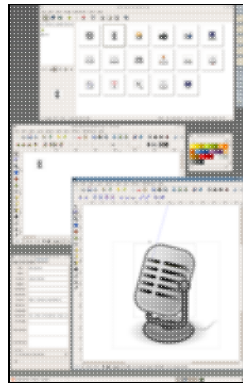
Thunderbird is part of the FireFox family and is as sharp as it's browser counterpart. It offers E-Mail with built in Junkmail filters, IMAP and POP mail server support, contact manager and address book.

### Using Linux for Artwork

Artwork has often been the lair of Mac users. This is mainly for historic reasons these days – (Apple used to have an exclusive contract with Adobe). However, times have changed, and Windows users enjoy the same fruits as the Mac users do. It's also fair to say that some excellent software exists for Linux for vector and bitmap artists alike.

**The GIMP** is the classic Photoshop-like image editor for Linux. It's now ten years old, and can do most of the features of all the popular off-the-shelf competitors. In case you were wondering, GIMP stands for GNU Image Manipulation Project.

GIMP was made with tasks like photo retouching, image composition and authoring in mind, and it should be fairly easy for an accomplished Photoshop user to convert to the GIMP. If you can't live without Photoshop, it is possible to use Photoshop in Linux, using Crossover.



**Inkscape** is a vector graphics editor, similar to Adobe Illustrator, Freehand, CorelDraw, or Xara X using the Scalable Vector Graphics format.

Features include shapes, paths, text, markers, clones, alpha blending, transforms, gradients, patterns, and grouping. Inkscape also supports Creative Commons meta-data, node editing, layers, complex path operations, bitmap tracing, text-on-path, flowed text, direct XML editing, and more.

It imports formats such as JPEG, PNG, TIFF, and others and exports PNG as well as multiple vector-based formats.

Don't forget that OpenOffice.org also has a component called Draw, which has a reasonably good vector graphics editor. Also Kivio (part of KOffice), and Dia are great alternatives to Microsoft Visio.

## Playing and editing Music or Audio

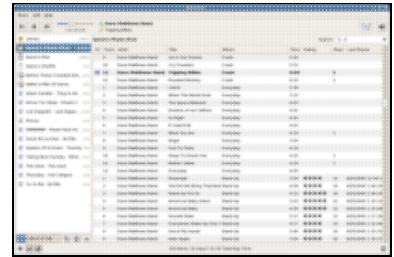
Linux has many applications for playing music and it's also really taken off as a platform for audiophiles. If you are a budding DJ, or play with MIDI instruments, Linux is a great operating system to use. There is a wealth of software available today which exceeds the requirements of many a Cubase user, and all for free!

### Audio Players

The first audio player for Linux was **XMMS**. It's very similar to the popular Winamp product for Windows. XMMS suits as a straightforward no-nonsense audio player, which accepts many plugins and Winamp 2.0 skins, however it's getting a tad dated, especially when it comes to competing with the likes of iTunes and similar software.

As usual, both KDE and GNOME desktops have excellent offerings, and whilst there is a wealth of audio player software available for both platforms, I will limit the following reviews to just two players, one from each desktop platform: Banshee (GNOME), and AmarOK (KDE).

**Banshee** is one of these new–super apps made in the new language framework called Mono, which is based on C#/.NET. As well as that, it's a damn fine audio program.



Similar in interface to iTunes, it allows you to download/upload music to your ipod or other music device, as well as listen and organise all your music on your PC. It'll also rip audio CDs at the touch of a button, and will also soon burn music CDs as well. Whilst it is not the default music player for most GNOME based Linux distributions, it could well be soon!

**IMPORTANT NOTE!: Why can't I play MP3s, and why don't I have Banshee?**



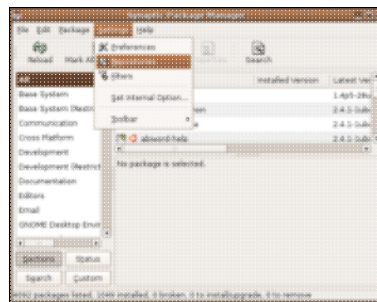
If you have just started using Ubuntu 5.10, you may notice that the default music player is called Rhythmbox, and Banshee is not installed.

Furthermore, if you try to play an MP3 file, you will find that can't do it!

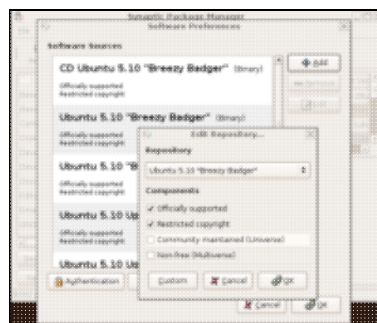
There is a good reason for this – Ubuntu, as with Fedora Core, Suse Open Edition and any other free Linux distribution cannot ship with software that is either not free, or contains a license which is considered commercial. It may not be illegal to distribute such software, but you may be required to accept a separate license, or use the software on different terms than the rest of the system. In order to keep the licensing system sane, this software is distributed separately from Ubuntu (as with other distributions).

In the case of MP3 support, the MPEG license is propriatery, but free to distribute, so adding support for MP3 is fairly easy to do. Here is how it's done in Ubuntu 5.10:

Firstly, open the Synaptic Package Manager (System Menu> Administration), you will need to enter your password to enable it. Next, add the Universe and Multiverse Repositories, by clicking on the Settings Menu and then clicking Repositories:



Click on the Settings Menu, then click 'Repositories'



Tick both of the bottom boxes, to add the Multiverse and Universe repositories

You will be asked if you wish to reload the package list from servers, Click Yes. Once the update has been completed, you will be returned to the main Synaptic screen, where you will be able to use the 'Search' button. Click Search and type **banshee** in the text box. Scroll down the lists of packages until you see banshee. Right click on Banshee and select 'Mark for Installation'. You may be asked to 'Mark Additional Changes', which is fine. Click on Mark. Repeat that same process for the following packages:

- ◆ gstreamer0.8-plugins
- ◆ gstreamer0.8-plugins-multiverse
- ◆ gstreamer0.8-ffmpeg

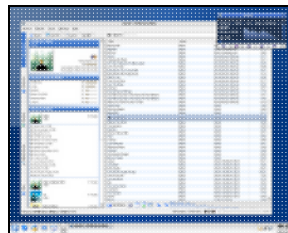
Once you have marked all the packages required, click on the Apply button (the green tick). You will be asked to confirm the changes, press the small apply button to continue and the packages will be downloaded and installed for you. Once the installation process has completed, you will be able to run Banshee from the Applications> Sound and Video menu.

If you need a refresher on installing software, refer to Chapter 9 for detailed instructions on installing software for your particular distribution

See the official Ubuntu documentation for more info on restricted formats.

**amaroK**, apart from having a stupid name is arguably the best music player out there.

Fans of iTunes will be immediately relieved to hear that it syncs with your iPod flawlessly.



It creates dynamic playlists too, like the party shuffle feature in iTunes, but better! amaroK features automatic CD cover finders for each album on your PC, so you know what the CD looks like when you play it. It also grabs the lyrics for every song you play on demand, as well as telling you pretty much all the info you would ever want to know about the band you are listening to, from Wikipedia. It also features a built in ID3 tag editor to sort out those rogue MP3s with invalid entries, and features MusicBrainz to take some of the guesswork out of it.

### Sound Editing

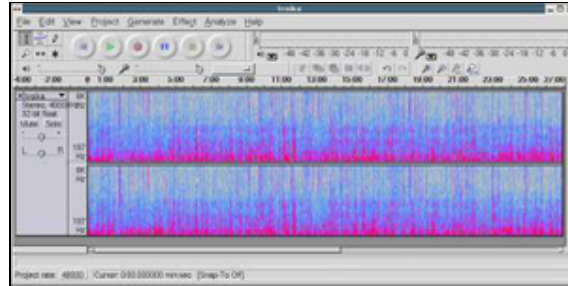
Again, there are so many good tools out there for sound editing, it is hard to name but a few here, but we will try to keep it to a good few!

**Ardour** is a digital audio workstation which can be used to record, edit and mix multi-track audio. It even goes as far as mixing video soundtracks. Ardour is similar to ProTools in it's ultimate quest, and uses the much acclaimed JACK audio system. Ardour has an excellent manual, and a quick and easy to use GUI. Ardour appears to be quick, snappy and doesn't use up too much CPU time, making it an excellent choice for a sound buff!





[Click here for a list of the applications using JACK technology](#)



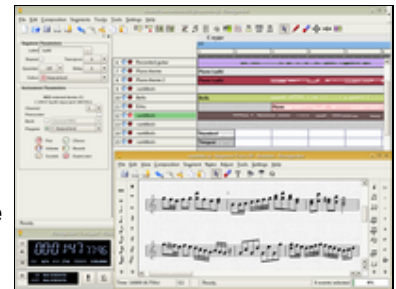
Audacity is well worth a mention because whilst it does not have nearly the same amount of features and power of Ardour, it does fill all of the needs of a sound-editing novice, whilst still beating any ~\$100 sound editor on the shelves.

Audacity's standard features can be picked up in a matter of minutes and allows for some really professional results. It offers unlimited tracks and many different effects in-built, so if all you need is straightforward multitrack or single-track editing, Audacity is a great tool.

Rosegarden is a professional audio and MIDI sequencer, score editor, and general-purpose music composition and editing environment.

Rosegarden is an easy-to-learn, attractive application and ideal for composers, musicians, music students, small studio or home recording environments.

Sound on Sound magazine called Rosegarden the closest native equivalent to Cubase for Linux, which is no mean feat, and Rosegarden has come to the acclaim of many other publications such as Linux Format's Top Stuff awards.



## Watching or Editing Video

A new breed of Movie players cropped up around the advent of a new framework called GStreamer, which made it easy to code video players. The most popular titles today are (in desktop order):

◆ GNOME based:

gXINE  
MPlayer  
VLC or VideoLan  
Totem

◆ KDE Based:

Kaffeine

## NoAtun

XINE it's self is a console based media client, but there are a large number of frontends for it, including gXINE and Kaffeine, mentioned above. XINE has a very robust architecture and is one of the oldest video players available for Linux. Mplayer is almost the same age. I personally have never fully managed to get Mplayer to work on any platform, however, many others will tell you that it has more features than XINE, and a better interface. The choice will come down to you in the end, but the good thing is that you can usually try both!

VideoLAN/VLC is a great piece of software and offers a wonderful array of playback possibilities – handling almost every format out there, it's very versatile, but it's real power comes from the fact that you can serve up media from any pc and get it to appear anywhere else – beam videos over the net, or just to another monitor in another room in your house. Support for resampling makes bandwidth problems with Internet broadcasting less of a problem.



## **Watching DVDs and other Rights Managed Media Types**



Just as with playing MP3s, DVDs and other formats such as DivX/Xvid, wmv and Quicktime all have their own propriatery plugins or *codecs*.

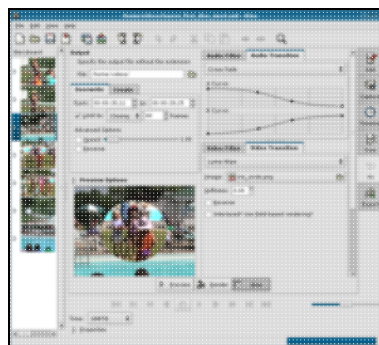
Unfortunately, each Linux distribution deals with this slightly differently, although here are the instructions to get DVD/DivX/WMA working under Ubuntu 5.10:

Download the debian-mallrat win32codecs file from [Here](#). Install the w32codecs deb file by using the terminal (Linux console/shell, and ensure that you have enabled the Ubuntu Multiverse and Universe):

```
cd /directory_where_you_downloaded_w32codecs_to
sudo dpkg -i w32codecs_20050412-0.0_i386.deb
sudo apt-get install libdvdread3
sudo /usr/share/doc/libdvdread3/examples/install-css.sh
```

See [Ubuntu's guide on restricted formats for more information](#).

## **Video Editing**



Currently, the two major contenders in the video editing arena for linux are [Kino](#) and [LIVES](#).

Kino is part of the KDE desktop apps, but will run with only the QT libraries. Kino features excellent integration with IEEE-1394 for capture, VTR control, and recording back to the camera. It captures video to disk in Raw DV and AVI format, in both type-1 DV and type-2 DV (separate audio stream) encodings.

You can load multiple video clips, cut and paste portions of video/audio, and save it to an edit decision list (SMIL XML format). Kino can export the composite movie in a number of formats: DV over IEEE 1394, Raw DV, DV AVI, still frames, WAV, MP3, Ogg Vorbis, MPEG-1, MPEG-2, and

## MPEG-4.

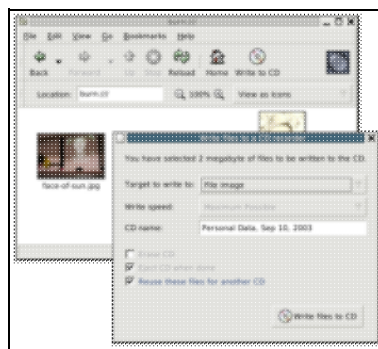
### Burning CDs and DVDs

Over the last two years, burning media in Linux has become a whole lot easier for the average user. If you are a GNOME user, [GnomeBaker](#), [Serpentine](#) and the GNOME file manager (Nautilus) will burn CDs for you.

KDE users can use the excellent [K3B](#) software, and if all of that choice isn't good enough for you, Nero software are now producing [Nero Burning ROM for Linux](#).

Burning data CDs or DVDs using GNOME is as simple as placing a blank CD in your burner. The gnome file explorer (Nautilus) pops up a file window, where you can drag and drop your files into. Once you are ready to burn, simply click the 'Write to CD' button.

If you have .iso images on your hard drive, you can burn these easily too, by simply right clicking on them in the file manager and selecting 'Write to CD'.



Ubuntu 5.10 ships with 'Serpentine', an audio CD burner, which will burn audio CDs, converting all your MP3s or OGGs into the CD Audio format for you automatically.

K3B is the best tool to use if you have KDE, and is similar in nature to Nero burning ROM. Creation of CDs, DVDs and Audio Discs can all be made from a simple menu.

### Watching and Recording TV in Linux

Recently, Linux has come to great acclaim in the TV arena, due to the excellent range of software available at no cost. You can literally turn an old PC into a Linux style Sky Plus/Tivo box, all you need is a TV card and a big hard drive!

[MythTV](#) is open source software that turns your PC into a PVR (Personal Video Recorder). It enables the user to pause live TV, Skip ads, use an electronic program guide, set recordings to record whole series of a particular program, edit recording schedules, organise and view your home photo and videocamera collections as well as listening to music and record content from the Internet, it excels at many points, making it a cut above the current offerings from Microsoft (Media Centre) and PowerCinema.



Also check out [KnoppMyth](#), which is a pre-packaged distribution of MythTV, making it easy to install on a PC.

If you only want to do the basics, you can record and watch live TV then you don't need to use something as feature-laden as MythTV. Popular titles for viewing and recording TV are [tvtime](#) and [xawtv](#).

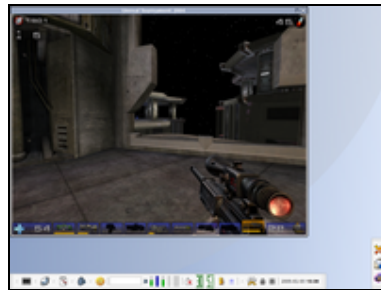
The Hauppauge series of TV cards seems to work best with the Linux TV software as it is the most popular range, and best supported through the video4linux driver.

### Playing games in Linux

Games have been a mixed bag with Linux, and still needs perfecting, however companies like [id software](#) and others are releasing their top titles for Linux as well as Windows these days, and is helping the overall popularity of Linux as a gaming platform no end.

Games such as the Quake, Doom and Wolfenstein series are available 'natively' for Linux (that is, the software is written to work in Linux). Other games which will also work with Linux such as the Soldier of Fortune series have been ported from their Windows base.

However, the majority of games software for the PC is still released for Windows only.



There is a solution, however – a company called [Transgaming Technologies](#) has a product called [Cedega](#) which allows the user to play most of the Windows games titles within Linux. This software is not free, but is fairly priced, and offers a good degree of ease of use.

There are a great number of games which are also freely available Linux software which you can see at places such as [linuxgames.com](#) or [happypenguin.org](#). [Tux Games](#) have done a great job of porting commercial games to Linux which you can buy for great prices.

[Click here to move forward to Chapter 11...](#)

## Chapter 11: Managing my files

---

### Topics covered in this chapter:

Tell me about /usr, /dev, /etc, /home and friends

Keeping your files nice and tidy

Making folders

Operating on files: Executing, Copying, moving, deleting and renaming files

Security and files: Permissions and Ownership

Using the **chmod** command to set permissions

Using the **chown** command to set the owner of files

Using the **chgrp** command to set the group owner of files

Even further in the depths of network file management: NFS

### Tell me about /usr, /dev, /etc, /home and friends

You've probably wondered what goes where on your Linux system, and what all of the directories in the / (root) folder on your system are doing. This lists a typical root partition on a standard Linux system. Yours may differ slightly, but the main ones (/bin, /usr, /lib, /var and /etc will always exist).

#### Folder Name Contents

**/bin** The standard set of GNU programs are held here. These programs are essential to the operation of your Linux system. It's like a system folder, except it only holds binary programs (as the name /bin suggests). Don't delete anything in here or your system will most likely become 'pooched'.

**/usr** /usr is a place where binary programs are installed into. It is also where your kernel source is held. There are various folders within /usr, such as **bin** (where installed software goes), **src** (your Linux Kernel source is here), **X11R6** or **X11**, (The X Window system is here), **share** (things like icons and pictures, wallpapers and fonts are held here — so that they are shared between your systems users), **doc** (where the programs on your system hold their documentation or manuals), **lib** (where all the libraries, such as Qt or an OpenGL driver would be held), **etc** (global configurations for installed software) and finally **local** (local binaries and programs to your system are located here, just another place for bin really).

**/dev** Large directory which holds files that link to real hardware devices attached to your system, ie: /dev/fd0 is the file for the first floppy disk drive (fd) on your PC. /dev/scd1 is the second SCSI cdrom drive attached to your PC and /dev/ttyS3 is the equivalent of COM4 under DOS. Due to this handy architecture, you can point the output of text files or running programs straight to a hardware device, without actually knowing how the hardware works. The kernel knows that. Ie, doing a `cat /home/my_modem_at_instructions.txt > /dev/ttyS0` would send the contents of the file my\_modem\_at\_instructions.txt to the modem on COM1 (or ttyS0).

**/etc** An important directory that you should try not too much to fool with (although some times, you've gotta go there to change things). This directory can be basically described as a configuration directory for system-wide applications and resources. For example, the file /etc/lilo.conf holds the configuration for the LinuxLOader program — it boots Linux from your hard drive for you. If you

edit this file, this will allow you to change the way that your system boots. The file `/etc/X11/XF86Config` holds the basic configuration for the X server. Don't delete this folder. Your system will crumble!

Note: individual users program profiles are stored in their home directories, ie: a single users profile for the bash shell is stored in `~/.bashrc` (a file with a dot infront of it is a hidden file btw).

<b>/home</b>	This is where all users home directories should be. For example, if your username is <code>jdoe</code> then you will probably have a home directory called <code>/home/jdoe</code> or <code>/home/users/jdoe</code> . All of <code>jdoe</code> 's personal files (excluding applications generally), profiles and directories will be stored in here.
<b>/root</b>	The root user's (system administrator) home directory. Root and ONLY root should be able to access this folder.
<b>/lib</b>	Where standard system libraries are stored, such as the <code>pam</code> (Pluggable Authentication Module) library.
<b>/mnt</b>	On a UNIX system, all drives are mounted before and after use, instead of just being <code>A:</code> , <code>C:</code> and <code>D:</code> . This means that a drive can just be part of your filesystem (as a directory). Most of the time, people like to put their mounted drives (such as a CDROM or Floppy) in the <code>/mnt</code> (short for mount) folder. Typically, you might have the following folders within a <code>/mnt</code> directory: <code>cdrom</code> , <code>floppy</code> , <code>cdrw</code> and possibly a windows drive: <code>win_c</code> . Whatever you have in the <code>/mnt</code> directory, you will find that the file <code>/etc/fstab</code> (file system table) has an entry relating the <code>/mnt</code> directory to the physical drive. ie: <code>/mnt/cdrom</code> usually points to <code>/dev/cdrom</code> in the <code>fstab</code> , and <code>/mnt/floppy</code> usually points to <code>/dev/fd0</code> .
<b>/proc</b>	If you have a good look at the <code>proc</code> folder, you will find that it's not actually a real folder at all, it's a virtual folder – it's 0 bytes big. Inside <code>/proc</code> , you will find handles for various devices and also, some informative files such as <code>meminfo</code> (if you type <code>cat /proc/meminfo</code> it will tell you how much memory you have and are using). <code>/proc/pci</code> holds all the information about your PCI bus and cards attached to it. Have a look, you can't edit any of these files.
<b>/tmp</b>	Temporary files usually created by running software are held here, sometimes these are called sockets.
<b>/sbin</b>	Where system programs are placed. Much like <code>/usr/local/bin</code> and <code>/usr/bin</code> , however, these binaries are usually ran as root, for system maintenance reasons. For example, RedHat/Fedora systems store the <code>ifconfig</code> tool here (which will allow you to alter network settings)
<b>/var</b>	Holds files that are generally server oriented, such as mail spools (where mail is held for a user until they pick it up). Lock files are held in <code>/var/run</code> . These files stop multiple instances of one program running by one user if necessary. Red Hat 7 and above also puts the web server in this directory, although Apache's default directory is <code>/usr/local/apache</code> .
<b>/opt</b>	Placed on your system generally when KDE is installed. Some KDE and Qt Based programs like to install themselves in the folder <code>/opt/kde/bin</code> for some reason.

## Keeping your files nice and tidy

It's important that you keep your files tidy on your Linux system, mainly because you'll find them easier, but also because you want to keep your system secure. Place all of your own personal files in

your own folder, and make sure that the permissions are set that only you can access them. For example, if you have lots of documents, consider making a folder in your home directory called docs. In docs you can put all of your documents.

## **Making folders**

To make a folder, you can usually enter your file manager in Gnome or KDE and right click on a white space in the folder that you want to create the folder, then select Create Directory or New Folder. However, to do it at the terminal, change to the directory that you want to make the folder in (cd /name\_of\_folder\_you\_want\_to/go\_to). Then type:

```
mkdir new_folder
```

## **Operating on files: Executing (running), copying, moving, renaming and deleting files**

### **Binary Execution**


To execute a binary program, at the terminal, you can do the following:

```
/folder_of_program/name_of_binary_file
```

So, if I wanted to execute the file myscript.pl in the current directory, I could issue:

```
./myscript.pl
```

Some programs such as LokiGames software ([www.lokigames.com](http://www.lokigames.com)), Sun's [StarOffice](#) and [Corel's WordPerfect Office](#) have their own dedicated binary installers. Most of the time, these sort of binary installers are initially executed from the terminal, by using the example above.

 Note that sometimes installer packages come without permission to execute (run), so you will need to provide it permission, try **chmod 755 [filename]** before you run the program.

### **Copying files**

To copy a file at the terminal, the cp command is used, with the following syntax:

```
cp source_dir/source.file destination_dir/destination.file
```

For example, I want to copy myfile.pl from the current directory to /home/bob/perl:

```
cp myfile.pl /home/bob/perl
```

### **Moving files**

To move files at the terminal, the format is very similar to the cp command. The move command is mv.

Here is the standard syntax:

```
mv /source_dir/source.file dest_dir/source.file
```

So, as a typical example:

```
mv myscript.pl /home/bob
```

The mv command has many other arguments (options you can pass to it at run time), such as -R, lets investigate the -R switch:

```
mv /home/jane/docs/* /home/bob/docs -R
```

This would copy the entire contents of the folder /home/jane/docs (because we gave it the wildcard \*), to the folder /home/bob/docs. Additionally, as we passed the -R argument, if there were any further directories inside /home/jane/docs (ie: folders called CV, letters and notes could exist inside the docs

folder), the `-R` argument moves the CV, letters and notes folders as well as the docs folder.

### Renaming files

Unlike DOS, which had a separate command to rename files with (`ren`), Unix simply uses the move (`mv`) command to do it. Take this example:

```
mv oldname newname
```

That would rename the file in the current directory called `oldname` to `newname`.

### Deleting files

Deleting files can be done simply with the `rm` command. Deletion of empty directories can be done with `rmdir` command. Here are some examples:

Deletion of a single file:

```
rm myfile.txt
```

Deletion of every file in current directory:

```
rm *
```

Deletion of every file in current directory and all directories within it (recursive deletion):

```
rm * -Rvf
```

Note: if you want to delete a directory and all of its contents in one swoop, instead of using `rm *`, then `cd ..`, then `rmdir dirname`, you can simply do a `rm dirname -Rvf`.

Removal of an empty directory:

```
rmdir mydir
```

### Security and files: Permissions and Ownership

In Unix, all files have permissions. Enabling files to have permissions allows security over and above simple password protection to a system. Imagine the situation: You have a Linux server which is used for 5 people from your company's Research and Development department. The five people are:

- bob
- jane
- mary
- mark
- simon

Consider for a moment that Bob wanted to keep his data (which is stored on the linux server over the network) private from jane, mary, mark and simon. Without the use of permissions, it would be impossible to do this. So for this reason we have the following permission system in Unix:

```
/home/bob          bob      users          drwxr-xr-x
```

What you see above is an edited example of a directory listing on a unix system. You can achieve the same result as what you see above by using the `ls` command with the long listing function (`ls -l`).

Consider that the above result was an abstract of the output of `ls -l /home` on the linux server.

The part that reads `rw-r-xr-x` is the permissions bit for the file (or directory).

The bit that says **d**, can also read **-**, **s**, **l** or **c**. **-** means normal file, **s** means sticky, **l** means symbolic link (like a shortcut, but but better) or **c** is a block device (you'll find these in the `/dev` folder). The **d** bit that is on the listing that we have is because the listing is of a directory.

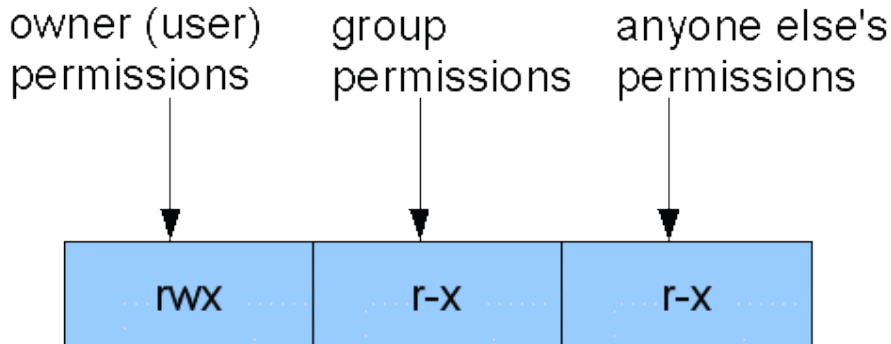
So, from the above listing already we know that it is a directory, it has some permissions set, and that



the directory is called bob and it is in the /home folder.

### **The Permissions for /home/bob**

But what do the permissions `rw-r--r--` actually mean, I hear you cry?! Have a look at this:



As you can see from the diagram, the first `rwx` is mapped to the owner of the file. This means that the owner (which is presumably bob), has read, write and execution access for that folder.

The second permission, `r-x` is mapped to the group of the directory. Not only can an individual user own a file, but a whole group (if desired) can access a file. This example illustrates that everyone in the group users can read the files in the directory, and that they can execute programs within it, or change into that directory.

The third permission, `r-x`, is for others (commonly referred to as world), which means everyone else. This is the most dangerous permission bit. Our example allows anyone on the server to be able (like the users group) to read anything within the bob directory, and execute files in the directory.

Ideally, what bob should really do, is set the following permissions, for total privacy:

```
/home/bob      bob      users      drwx-----
```

### **Using the chmod command to set permissions**

When you want to change the permissions of a file, you can use the ch commands: `chmod` (which changes the `rwx` bits of a file), `chown` (which changes the owner of a file or directory) and `chgrp` (which changes the group which accesses a file or directory).

This part deals with `chmod`.

Imagine a file called `cv.doc` with the following permissions User: read, write, execute. Group: read.

Others: none, ie:

```
cv.doc rwx/r--/---
```

Let's imagine that we wanted to change the permissions to: `rw-rwx---`, so user and group can read, write and execute the file `cv.doc`.

This command would let us change the desired permissions:

```
chmod ug=rwx,o= cv.doc
```

So, what we did there, is we ran the `chmod` command and told it to set the user and group permissions to read, write and execute. We set the group permissions to nothing. Finally, we specified the file that we wanted to change the permissions for, `cv.doc`. Take this other example, as with Unix, there is *always* more than one way to do something:

```
chmod 0770 cv.doc
```

You will notice the only difference to the above command is the `0770` against the `ugo` bit. The `0770` (or just `770` would do) means exactly the same thing as `ug=rwx,o=`. `0770` is the numerical representation for the permissions, octal is a base 8 number system, and the permissions of files are based on this. ie:

```
1 = execute only
2 = write only
4 = read only
```

You will notice, there are 8 (as in octet) combinations of numbers (0 through 7). If you add 1, 2 and 4 together, you get 7, which is read, write and execute. By taking any combination of all of the numbers 1, 2 or 4, you get your desired permission that you require. For example, 1+2 = 3, and `chmod 300` would give you write and execute permissions.

It really doesn't matter which way you choose to use `chmod` (either ie: `u=rwx,g=rwx,o=rx`) or `(775)`, just use what you feel most comfortable with, although it is handy to know how each way works.

### **Using the `chown` command to set the owner of files**

The syntax of the `chown` command is relatively simple:

```
chown bob mydoc.txt
```

changes the owner from whomever the current owner of `mydoc.txt` to `bob`. Note that only the person(s) with write permissions to this file can perform this action.

Quick Tip: `chown user.group` will change the owner and group of the file

### **Using the `chgrp` command to set the group owner of files**

As you saw in the previous hint, `user.group` will quickly change the group of a file or directory, and I personally prefer using `chown user.group`, but the original and proper way to change the group ownership of a file or a directory, can be done with the `chgrp` command, as follows:

```
chgrp users mydoc.txt
```

...which changes the group ownership of `mydoc.txt` to `users`.

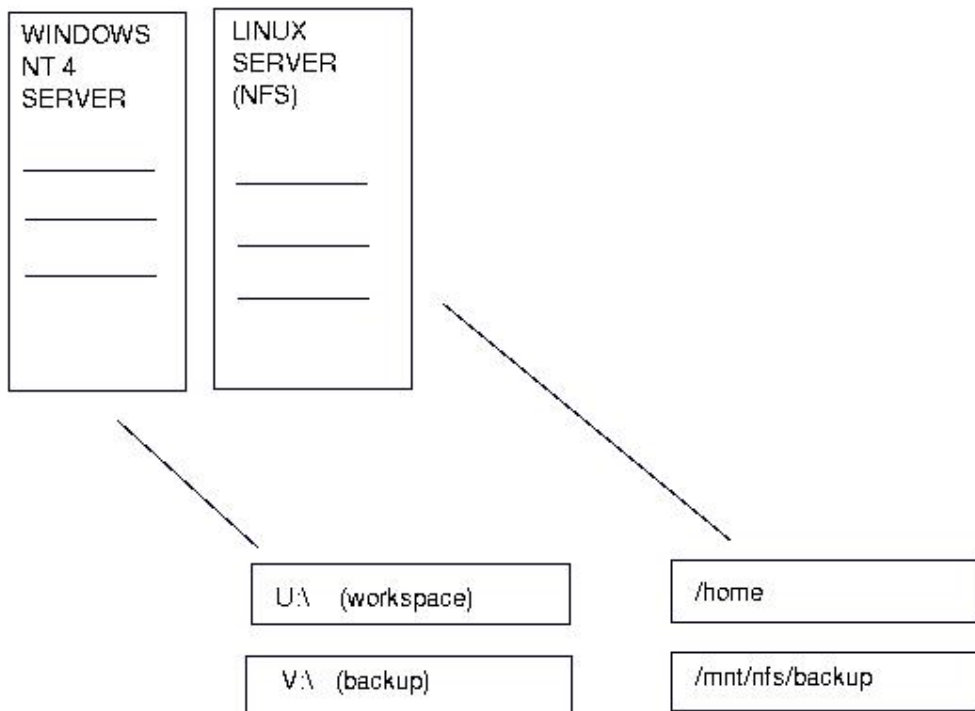
### **Even further in the depths of network file management: NFS**

Some of you will have a brief idea of what NFS is, and some of you will have never before heard of NFS, and because of this, I will describe NFS from the top. If you know completely what it is, and just want to know how to set it up, feel free to scroll down a bit until we get to the section 'Using NFS'.

#### **What Exactly *is* NFS?**

NFS is a method that you can easily and very efficiently share files accross a network in a Unix environment. Almost all types of Unix support NFS, and although not natively, Windows NT can be told to read NFS volumes.

A distinct advantage that NFS takes over the 'Windows Shares' mechanism (the system used to share files in Windows), is that NFS does not require to assign a seperate drive on your computer for the use of shared files. Imagine the following scenario:



The image shows that directories of files or drives themselves, under Windows, must be exported to a drive letter (of which only a possible 24 are available). With Linux and Unix, you can export any directory or drive, and it can be viewed under any name on the client.

Imagine the scenario: You have the directory `/shared/home` on your server alpha. You want your other client machines, bravo and charlie to be able to see this folder, and everything inside it. Your two client systems are to have their home directories inside this share. You use NFS for this purpose. Perceive the users mary (who uses bravo), and bob (who uses charlie). Mary's current login directory on bravo is `/home/mary`, and bob's login directory on charlie is `/home/bob`. Both of these directories are local to bravo or charlie and cannot be seen on other machines. You make two folders inside `/shared/home` on Alpha, one called mary and one called bob. You move everything inside mary on her computer (bravo) into her folder on alpha, and you do the same for bob.

You remove the `/home` directory and all its contents on bravo and charlie — as it's now redundant. You then export the directory `/shared/home` to bravo and charlie (and call it `/home`)

The net result of all of this is that mary and bob (to them anyway, nothing has changed), but to the administrator of the network on alpha, all of the files are stored on the alpha server, and any files they now request from their home directories is now taken from the folder `/shared/home` on alpha.

## Using NFS

Taking the three servers, alpha, beta and charlie a little further, this is a demonstration of how you can actually set it up:

NFS works on the basis of one of the rpc (Remote Procedure Call) tools called **nfsd**. All good Linux distributions should have rpc tools already with them (although they may be disabled in the sysv init startup), make sure this is installed and ready to go, and make sure that you have installed **nfsd** on the system that you wish to export your directories or drives (in our case, this is alpha).

Edit the file `/etc/exports` which is where all your exported volumes are listed. On alpha the format of

the file would look like this:

```
/shared/home      bravo  (rw,no_root_squash)      charlie
(rw,no_root_squash)
```

This is what the above line means to you and I:

- 1) Take the directory /shared/home
- 2) Allow the computer **bravo** to read and write (rw) to it, and allow root to be treated as root over exported filesystems (no\_root\_squash)
- 3) Allow the computer **charlie** to read and write (rw) to it, and allow root to be treated as root over exported filesystems (no\_root\_squash)

Obviously, alpha must know, either by the file /etc/hosts or by using DNS / NIS how to resolve the hostnames bravo or charlie. If you don't think you can manage this at your current stage of Linux networking skills, then I recommend simply replacing the hostname (ie: bravo) with the IP address of that particular host (ie: 10.0.0.2).

You may or may not know about the infamous /etc/fstab, but this file holds the key for configuring bravo and charlie with the information they need to see the 'share' that alpha has now been set up to provide.

If you investigate /etc/fstab, you will notice that it has all the points of where all your drives are mounted, and this file is read by Linux when it starts up, to mount each drive. NFS Volumes are no different from physical disks, they too get added to this list, assuming that you want the volume to be mounted every time you use the computer.

Adding this line to the /etc/fstab on bravo and charlie would then kick in the mounting of the nfs volume on boot each time:

```
alpha:/shared/home      /home      rsize=8192,wsiz=8192
```

- And in english:
- 1) Take the export or 'share' on the system alpha (/shared/home) (alpha:/shared/home).
  - 2) Mount it on this computer (either bravo or charlie) as /home (ps: an empty directory with the name /home must exist to put this in). (/home)
  - 3) Make the amount of retrieved bytes per block 8K / 8192 bytes (rsize=8192)
  - 4) Make the amount of sent bytes per block 8K / 8192 (wsiz=8192)

Parts 3 and 4 of the statement are not necessary, but they help to improve efficiency over the network.

Also note that again, if you haven't set up an /etc/hosts.txt or DNS to resolve the name alpha, use its IP address in the place of the '**alpha**:', to read something like '**10.0.0.1**'.

Once you are sure that nfsd is running correctly, you can do this by typing `ps ax | grep "nfsd"` at the shell. If you get a process called nfsd appearing (don't confuse this with `grep nfsd`, which is what you just typed), then you know that nfsd is running, and it sounds like it's configured correctly.

Now all you need to do is either start nfs manually on alpha, or reboot it, so that the startup script brings it up. You could reboot bravo and charlie, but preferably, just typing '`mount -a`' at the root prompt on both of the systems is enough for Linux to re-read its /etc/fstab file and cotton on to the fact that it has a new entry. If you get any errors when it re-reads this configuration, make sure to double check all of your settings.

## **Further Reading**

The Linux Documentation Project should have the definitive guide to using NFS, although it's possibly not too newbie-friendly.

Typing **man nfsd**, **man exports** and **man fstab** will all help you, and mainly man exports.

If you want to know about the Linux command line, click here to move forward.

## Chapter 12: I Don't know any commands!

---

### Topics covered in this chapter:

The whole Linux command thing: Why have commands?

Some simple commands, you may recognize

Some more commands, plus how to get help on almost any command

### The Whole Linux Command Thing: Why Have Commands?

Just as cavemen learned to develop their skills from grunting and groaning to writing meaningful representations on the walls of caves, The computer world started off with (well, if you wanna be accurate, punched cards), the keyboard and the CLI (Command Line Interface). All action that happened on any system would be inherently text. As talked about in one of our Hot Topics (X Windows), Xerox's PARC research center managed to invent the very first graphical user interface. When Linux first came to be in 1991, most things were still text. True, Windows 3.0 was out, and MacOS had been with us since 1984 but the PC world was still dominated with text. MS-DOS still ran the PC show. Anyone who wanted to use their PC efficiently back in 1991 with a Microsoft machine, would use Windows to run their graphical programs like Aldus PageMaker and Lotus 1-2-3, but they would open an MS-DOS prompt box, or close Windows and drop to DOS, so that they could quickly rename a file or move it from one place to another. Sure, users could even then, click into the File Manager, select the file(s) they wanted to rename or move and drag them to the relevant location, but a lot of people were finding that DOS was still quicker to use if you knew the commands. If you spent a few days learning the basic commands of DOS, you could really use your PC efficiently, coupled with the graphical usefulness of Windows (ahem), you could cut workload down by a good deal. Even today, when using modern GUI O/S's like Windows XP and Mac OS, I still find myself going to Start>Run and typing in **command** to access the DOS prompt to do something quickly. To the people who used the PC before the GUI ruled (circa 1995), typing a command even now is sometimes more efficient than using a drag and drop idea.

Imagine the following scenario:

You have 300 files in a folder (call the folder 'work'), and within that folder you have 2 other folders, one called 'old' and one called 'new'. Your task is to separate the files in work to the new and old folder. You must put the files that are older than 1 month into old, and the files less than 1 month into new. In a graphical file manager, you would need to right click each of the 300 files separately to find out their creation date, and then move each of the files one by one into their relevant folder. This operation is seriously time-consuming.

How's about the command-line option: type in one line at the shell, and the files are automatically sorted into each folder, determined by today's date. Sure, you would know that you would have to type

**'find /usr -ctime +30 -exec ls -ld {} \;'** to get it to happen, but hey, it's still faster than going through 300 files.

There are yet many other reasons that you may want to have a CLI, for example, a User Administrator has 3 dead (crashed) programs on 3 different workstations across a building. He has two choices: Go run around the building like a headless chicken and kill the programs manually at each workstation, or sit at his desk and telnet into each machine, using the CLI, killing each of the programs, finishing the job 20 times faster.



What about a remote GUI though? – Nowadays with faster networks it is possible to run X or even an MS Windows Desktop over a network. Remember however, all that graphics has a large bandwidth requirement, and means performance is often pretty poor (especially from afar). At the CLI, most of the time you will barely notice

whether you are local or remote.

If you start to use Linux a lot, you'll start to find it's a lot easier just to issue a direct command instead of clicking on a hundred different icons to do something you could do in one command at the CLI.

A common misconception (I don't know why this is) about Linux is that it is completely text based, and the story could not be further from the truth, but I expect Linux to always hold onto it's console, as it's a highly useful tool, and whilst I probably use it more than your average user will, it's still handy at some points for most, given a few hours to learn it's power.

### **Some simple commands, you may recognize**

GNU/Linux has a lot of commands, luckily though, you'll probably only ever want to know around 10–15 of them for anything at the CLI.

You can find the files for these commands in folders like **/bin** and **/usr/bin** on your Linux box.

If you have used computers since at least the late 90s, then you will probably remember DOS, you'll find that the syntax of DOS commands compared to UNIX commands are often similar, and even some of the commands themselves are the same. Have a look at this comparison table if you can remember any DOS commands:

<b><u>DOS Command</u></b>	<b><u>UNIX Counterpart</u></b>	<b><u>What it does</u></b>
dir	dir or preferably <b>ls</b>	Shows the files in the current folder
cls	clear	Clears the screen
type	cat	Shows the contents of a file, eg cat foo.txt
move	mv	Move a file from one place to another, eg: mv foo /home/
copy	cp	Copies a file, eg: cp /home/foo /users/foo
ren / rename	mv	The mv (move) command also doubles up for renaming
echo	echo	output (or echo) something to the screen

### **Some more commands, plus how to get help on almost any command**

Here are a few more advanced commands, along with their explanations and one example for each

<b><u>Command</u></b>	<b><u>Action</u></b>	<b><u>Example</u></b>
chmod	Changes permissions for files	chmod u=rwx myfile.txt
chown	change the owner for a file or folder	chown myuser /home/myfile.txt
chgrp	change the group owner for a file or folder	chgrp mygroup /home/myfile.txt
mv	Moves or renames a file	mv /home/me/myfile.txt /home/them/
grep	searches text for results	cat myfile.txt   grep "hello"
ls	list the files in a folder	ls -l
find	finds files or file designators	find /home -name "myfile.txt"

mount	mounts a drive/block device for use	mount /dev/cdrom /mnt/cdrom
ps	Show the list of live processes	ps ax   more
kill	kill a process (running software) by process ID	kill -9 12345
tar	de facto Unix archiver	tar zxvf myarchive.tar.gz
rpm	The Red Hat Package manager, used to install, remove, upgrade and query RPM packages.	rpm -Uvh mynewpackage.i386.rpm

**Important tip:**

If you ever need to know how to use a command, or just find out what a command does **man** (short for manual) is your friend. For example, by typing `man ls`, a comprehensive guide is given on using the `ls` command, what its purpose is and what its syntax is. For more information on using `man`, type **man man**.

There is not a better source for quick, easy to find information on almost every Linux command. The only times when `man` lets you down, is when you don't know that a command that does a specific purpose exists, thus, you don't know the name of the command, so you don't know what manual page you want to ask `man` for. For this reason, I have started the lengthy process of creating a commands guide, that is easy to understand, although somewhat less verbose than manual pages themselves.

If you wish to visit the Commands Guide, which *should* list almost every Linux standard shell command, but is currently unfinished. Click [HERE](#).

## Thanks for reading the Ultimate Linux Newbie Guide

I thought I'd throw in some other popular content from the site, including the extra bits in Chapter 5 on installation tricks and troubleshooting as well as some of the hot topics found on the site. Enjoy!

### The following pages are (in order of contents):

- Troubleshooting installation of Ubuntu 5.10: Networking Problems
- Troubleshooting installation of Ubuntu 5.10: Custom Hard Disk Partitioning
- Hot Topic: What's a Terminal / Console / Xterm?
- Hot Topic: What to do if you really can't get your WiFi working
- Hot Topic: Dual Booting Windows and Linux
- Hot Topic: What *is* X Windows?



## Chapter 5: Installation Network Problems

---

### EEK! What's all this about DHCP?



Click on the image to enlarge it

DHCP is the Dynamic Host Configuration Protocol. It's often found when you have a router or modem on a local network. Simply put, it provides addresses (much like telephone numbers), called IP addresses to your computer(s). In this case, DHCP is either not set up, or the hardware in your computer isn't able to hear the DHCP messages being sent around.

Possible reasons for this are:

- 1) You don't have a router or cable modem
- 2) Your network is incorrectly configured
- 3) You decided you didn't want automatic IP configuration on your router
- 4) Some cable or wireless configuration isn't set up somewhere
- 5) Your network could just be slow, try doing it again

If you suspect your problem might be case 4, Go make sure that the necessary cables are plugged into your machine and into your router and restart the auto-detection. If you think everything is plugged in correctly, then the best thing to do is configure your network settings manually.

First of all, make a note of any network settings you have, for example, if you have Windows set up somewhere, take a note of any of the following information you can get your hands on:

Your IP address (eg: 192.168.0.1)

Your Subnet mask (eg: 255.255.255.0)

Your DNS servers (eg: 60.50.40.30, often two IP addresses)

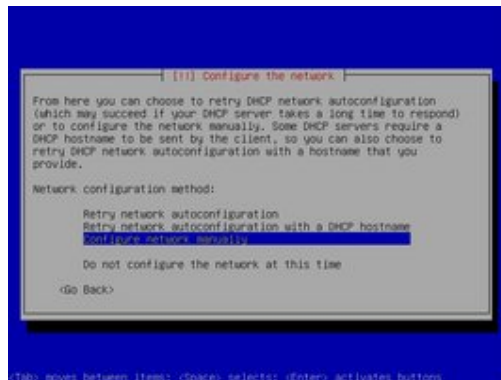
Default Gateway (eg: 192.168.0.254)

You can often get this information from your Control Panel > Networking in Windows or the System Preferences in Mac OS.

If you use windows, visit the DOS prompt and type **ipconfig /all**, which will often give you this information.

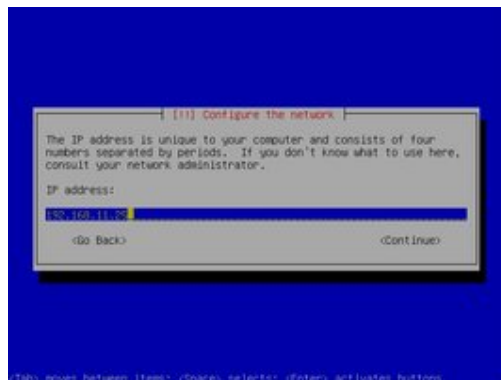
If you use Mac OS X, then visit the Terminal and type **ifconfig**

### Configuring the Network Manually...



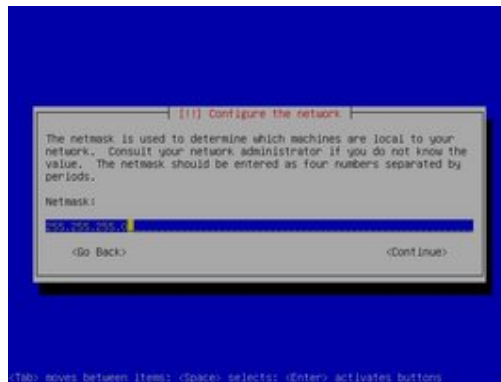
**Click on the image to enlarge it**

So, here we select 'Configure network manually' if we can't get things done the easy way.



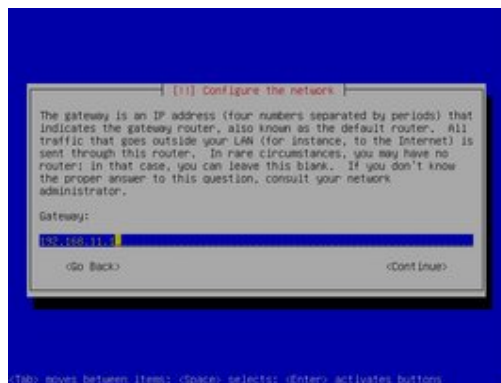
**Click on the image to enlarge it**

Hopefully you will have collected a valid IP for your computer. If you can't find one, try to find out from any avenues you have – support for your router or modem, or maybe your Internet provider. 192.168.0.1 is a popular IP for the first host on a network, but this is only a hint. IP's are always made in decimal blocks of four, with values not exceeding 255, eg: 1.2.3.254 (valid), 266.100.200.254. Note that 255 must not be at the end of the host address, as it is a 'broadcast' address.



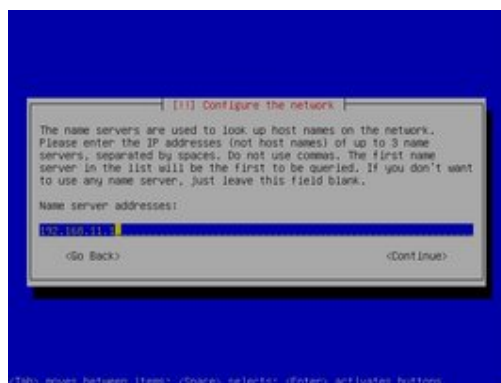
Click on the image to enlarge it

The subnet mask is vital to the correct operation of your network. If you have a simple small home network, it's almost always simply **255.255.255.0**. Be aware though, that any customised networks, or larger LANs, or perhaps ISP assigned IP addresses, will most likely **not** use this IP. Consult with your ISP or router information.



Click on the image to enlarge it

The Default Gateway is an IP address to tell your computer how to get 'out' to the internet. The IP is generally the IP address of your router, if you have one. If you don't know what your default gateway IP is, try contacting your ISP for help.



Click on the image to enlarge it

Finally, it's time to set up your name server or DNS server. The IP address here is used by your computer to translate names like 'yahoo.com' into IP addresses like '1.2.3.4', because the Internet really only works on IP addresses at it's lowest level, your computer needs access to a DNS server (sort of like an address book for the Internet).

Again, in a small home setup, it is likely that your router will have a DNS server on it, so as long as you know the IP address of your router, this IP address will be the same one you used for the default gateway. If however, you have a different setup, you will need to consult your ISP for this IP. You will only need one, but if you get given two, this is no bad thing.

[Back to Installing Ubuntu 5.10...](#)

## Chapter 5: Manually Partitioning A Hard Drive

If you want to install Linux on a hard drive that is occupied by another operating system such as Windows, or perhaps you prefer to set up your hard drive manually then you should follow this rough guide.

The procedure is a little different for most people, because no two hard drives are set up the same.

If you are using this guide to set up Linux and Windows on the same drive, then it is assumed that you have already resized Windows on the drive to allow space for Linux by the use of QTParted or similar tools. See [Chapter 4](#) for more information.



### Choose the drive you want to partition

Assuming you have chosen to manually partition your drive, use the cursor keys to navigate to the bottom section of the partitioning method dialogue – and select the drive you want to partition.



### Select free space

This particular drive does not have any other partitions on it, however if you had Windows on your drive, it would show it here.

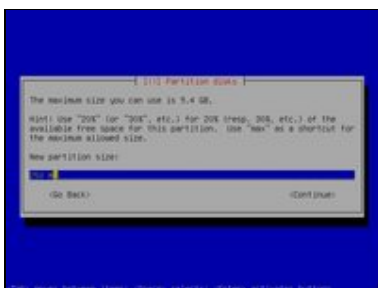
If you are installing Linux alongside Windows, make sure you select the partition on the hard drive that you have cleared out as 'Free space', rather than your Windows partition.



### Create a new partition for Linux

Linux needs to allocate the 'Free space' you selected in the previous screen so select the default option here, Create a new partition.

### Setting up the first partition (/boot)

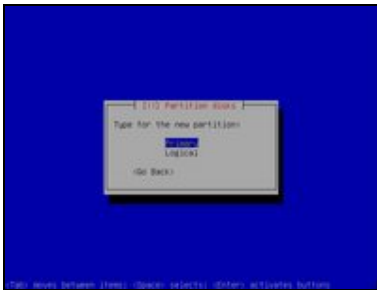


In this example, we are going to create four hard disk partitions to put Linux into. You can do it with less, but the way we will set up the partitions allow you to have one partition for booting, documents and settings, system programs and swap space, as follows:

/boot (booting the system)  
 /home (for your own documents and settings)  
 / (all the system data and programs)  
 swap (which helps out your memory).

This partition we are setting up is for booting Linux, and it contains very little data, so type **250 M** in the text box, to allocate 250 Mega bytes to /boot.

## Primary, or Logical?



If you are completely new to setting up disks, then you will need to know that for mainly historic reasons, hard drives can be split into Primary and Logical Partitions.

Two things to note about Primary partitions:

- ◆ Primary partitions should be used to boot your O/S.
- ◆ There is a limit of four primary partitions to any drive

Due to these restrictions, we will use Logical partitioning for our last partition, and Primary for our first three, if you have another O/S such as Windows installed, you should take this into account also. The only partition that **must** be Primary is the one that contains /boot.

Partition Number	Partition Type	Logical or Primary?	Partition Use (name)	% Taken up of Disk
1	Windows	Primary	C:	50%
2	Linux EXT3	Primary	/boot	1% (or approximately 200MB)
3	Linux EXT3	Logical	/	20%
4	Linux EXT3	Logical	/home	25%
5	Linux SWAP	Logical	Not Applicable	4% (or, typically double the size of your RAM)



## Beginning or End?

All of the partitions we are setting up in this exercise will be contiguous. Each partition will come one after another, from the beginning of the drive, to the end of it. As this is the case, all of our partitions will start from the Beginning of the free space.



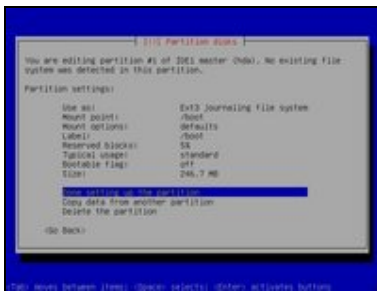
## Summarising so far...

So far, we have created a new 250MB partition on our free space. This screen summarises just this, however it shows a few things that are not to our liking – the mount point is set for '/', and we want to use '/boot', therefore, use the cursor keys to select 'Mount point' and hit return to change it.



## Mount Point Name

As we wish to delegate this 250MB partition to /boot, select /boot from the menu with the cursor keys and hit return.



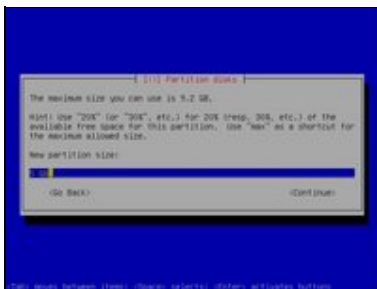
### A last check...

If everything looks good now, then select 'Done setting up the partition' and hit return. Note that the size of our 250MB partition is slightly smaller than 250MB (246.7MB to be precise), this is due to drive information occupying the remaining 3.3MB space.



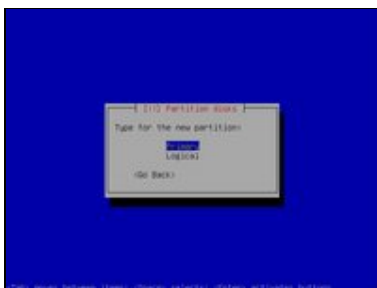
### Back at the main partition menu...

Once the partition is set up, we are returned to the main partition menu, where we can see that our newly made partition has been added to the list. If you are happy with this arrangement, proceed on to allocate the remaining free space by selecting 'FREE SPACE' with the cursor keys and pressing return.



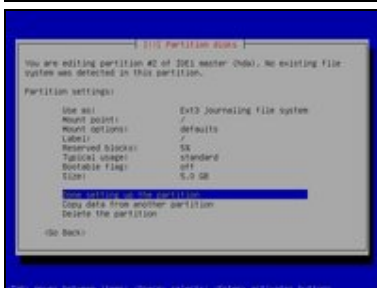
### Choose the size

The partition that we will be setting up now is called the / (or, root) partition. This is the main partition of any Linux system, as all system data will be held under here, such as your program files and system settings. If you have a large hard drive and lots of space to spare, give this a good amount of space, but be sure to leave space for your home and swap partitions.



### Just like before, the partition is at the beginning of the free space...

Select 'Beginning', and hit return.



### Summary of the root partition

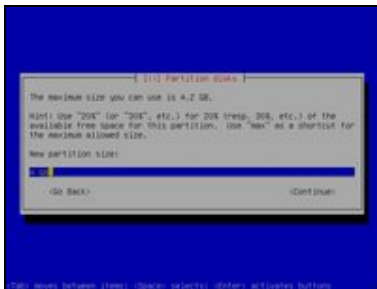
If you are happy with your size allocation for /, then Select 'Done'.



## Selecting Free Space for /home

In our example, we have a separate partition for home, this is where we will define how much space to provide your own files (documents, music etc), and your preference files.

Select 'FREE SPACE' and press return.



## Select the size for /home

Once again, give your own data area as much space as you can, whilst leaving space for the smaller swap partition.



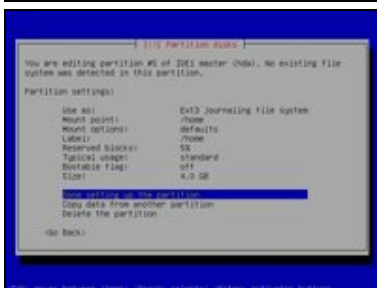
## Primary or Logical?

In our example, this will be our third partition, so we will make it a logical partition.



## Just like before, the partition is at the beginning of the free space...

Select 'Beginning', and hit return.



## Happy at home?

If you are ok with the settings for /home, then confirm by selecting 'Done setting up this partition'.



## Time to swap..

Before we go on to set up swap, you can see that the numbering of the partitions on this drive are as follows:

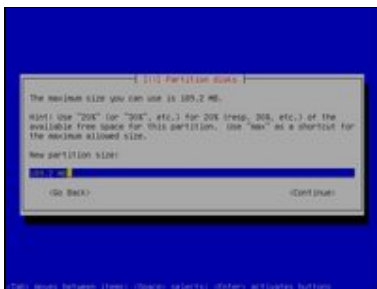
- 1 – Primary (/boot)
- 2 – Primary (/)
- 5 – Logical (/home)

Why does it go from 2 to 5? Has Linux lost the ability to count?

The answer is no! As the Primary partitions can occupy up to four partitions, the PC reserves these first four for the Primary allocation.

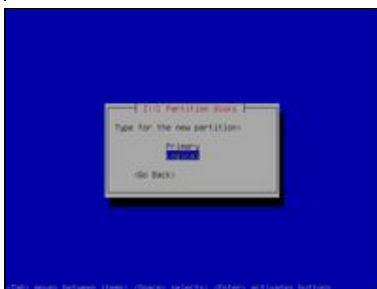
Ok, Where were we?...

If you are happy with your /home partition, select the remaining free space and hit return.



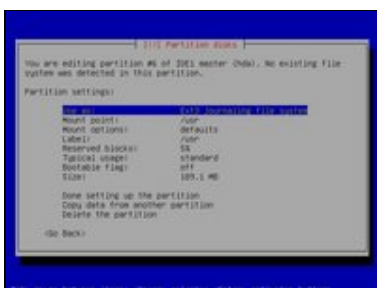
## Create another new partition

If you don't intend on using the drive for anything else, you can use up the rest of the drive for the swap, but a good guideline for the size of your swap partition should be around twice the size of your system RAM (memory), so if you have 512MB RAM, then your swap space would be 1024MB (1GB).



## Logical, again..

How did you guess?! Another logical partition.

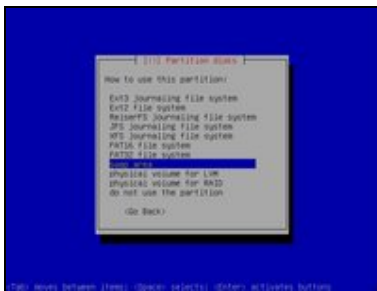


## /usr – but I want swap!

The Ubuntu installer tries to be clever again, allocating /usr to this partition type, however, /usr will actually appear under /, so we don't want a separate partition for it.

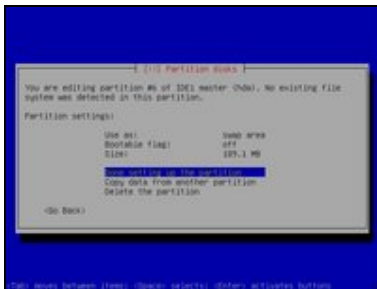
The steps for changing /usr into swap are different from the time before, so watch closely as we change the partition type from ext3 to swap.

Select the 'Use as' line and press return.



### What type?

As it's a swap partition we want, change the partition type to 'swap area' and hit return.



### Anything else?

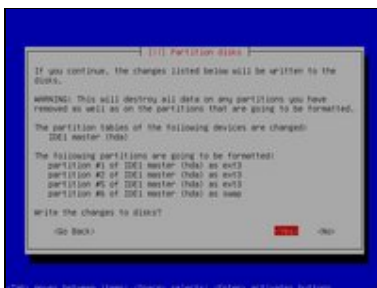
Next you will be asked if there are any amendments for this partition, and as the swap partition type is pretty simple, there is nothing to do. Select 'Done setting up the partition' and press return.



### A good job done well!

We've filled this hard drive up with Linux partitions and it's now time to commit these changes to the disk.

If you are happy with all of your changes, select 'Finish' and press return.



### Warning!

Anything you have done before this point is fully reversable, but once you Finish the partitioning here, there is no going back!

Select Yes, and your partitions will be formatted and readied for Linux.

**Note: only new partitions you have set up will be formatted, partitions which you have left will not be touched.**

[Back to Installing Ubuntu 5.10...](#)

## What is a terminal / console?, How do I open up a shell?

```

-rw-r--r-- 1 markos users 12413 2003-05-18 19:45 bookcontents.html
-rw-r--r-- 1 markos users 8648 2003-05-05 17:15 bookcontents.sxw
-rw-r--r-- 1 markos users 8592 2003-04-26 12:05 bookcontents.txt
-rw-r--r-- 1 markos users 23920 2003-07-27 20:04 chapter2.iso.txt
-rw-r--r-- 1 markos users 18202 2003-06-06 20:09 chapter2.sxw
-rw-r--r-- 1 markos users 39089 2003-06-06 19:46 chapter2.txt
-rw-r--r-- 1 markos users 20380 2003-06-02 09:48 chapter3.txt
-rw-r--r-- 1 markos users 17288 2003-06-04 10:58 chapter4.txt
-rw-r--r-- 1 markos users 39222 2003-06-06 23:14 chapter5.txt
-rw-r--r-- 1 markos users 738 2003-06-06 21:24 chapter6.txt
-rw-r--r-- 1 markos users 632209 2003-07-02 00:03 debianguide.html
-rw-r--r-- 1 markos users 1597565 2003-07-28 05:48 debianguide.pdf
-rw-r--r-- 1 markos users 130168 2003-07-01 23:43 debianguide.stw
-rw-r--r-- 1 markos users 254199 2003-07-29 19:45 debianguide.sxw
-rw-r--r-- 1 markos users 1226735 2003-07-15 15:17 debianguide.tar.bz2
-rw-r--r-- 1 markos users 700919 2003-07-21 07:29 debianguide.txt
-rw-r--r-- 1 markos users 0 2003-06-25 10:25 debianguide.xml
-rw-r--r-- 1 markos users 92028 2003-05-19 14:27 glossary.zip
-rw-r--r-- 1 markos users 257708 2003-02-10 16:00 nuke_pn_glossary.csv
-rw-r--r-- 1 markos users 164383 2003-07-22 11:05 nuke_pn_glossary.sxc
-rw-r--r-- 1 markos users 740352 2003-07-18 23:21 nuke_pn_glossary.xls
-rw-r--r-- 1 markos users 202352 2003-07-20 12:24 nuke_pn_glossary.zip
drwxr-sr-x 22 markos users 4096 2003-07-29 19:10 pics
markos@silmaril:~/Documents/My Own/Debian Book$

```

The 'Konsole'. The Terminal Application in KDE.

A "terminal" is a way of typing commands to the computer.

That may sound very strange, but it's actually really, really nice and easy once you get used to it.

You can use Linux as a totally graphical system, or you can use it totally with the terminal. Many people like to do both; I use whichever one is easier for the task I'm doing.

You can either open a terminal inside of your graphical environment, or you can switch your whole screen to a terminal.

To open a terminal in a graphical environment:

Open up your main menu.

This is usually the icon to the left of the bottom panel of the screen (where the Start button is in Microsoft Windows). The main menu is signified by a few icons depending upon which desktop you use, it is often a 'K' icon, a suse logo, a red hat, a foot symbol, a little penguin, or perhaps just the word 'Applications'.

The Terminal is usually found under System Tools, then by clicking on Terminal.

## I've tried everything, but my WiFi Connection refuses to work

---

The situation is this: You have tried out the GUI menus available to you by your standard distribution (eg, Fedora, Ubuntu, SuSE) and you've also had a look around on the [forum](#) here and maybe elsewhere as well but your Wireless Internet connection refuses to work.

This method is non distribution specific and as long as you can get your drivers to work, the rest of the guide should get you going nicely, note however it's non-standard, and as a result, your distribution may choose to delete or disable this configuration at a later date, especially after an update!

First things first, make sure that your hardware works. In any Linux PC I have ever used with wireless, I have always done one thing:

Gone with what I *know* works.

This means that a lot of the new fancy 802.11g and 802.11ng and MIMO stuff is probably not your best bet for compatability. If you know 100% that the hardware you have definitely works not only with Linux, but with your distribution, then go ahead, but my advice is go with one of two card types:

Cisco Aironet 3xx series (340/350)

Orinoco / Prism 2 / Intersil Chipset-based cards

These cards are 802.11b cards, they are old (they started coming out around 2002) and are probably only available on places like Ebay. Both PCI and PC Card (PCMCIA) versions should work fine for PCs or Laptops alike.

Because the cards are 802.11b, this limits their speed to a maximum of 11mbps. Fine for most people's broadband internet needs today (most people have a max of around 8mbps at time of writing), but not great for transferring files on your home network at speed.

For a more complete look at which wireless adapters work with linux, check out Jean Tourrilhes page on [Linux and Wireless LANs](#).

If you have been lucky and gotten yourself an old-trusty wireless card then you should be able to pop it into your PC and notice it working. Here is a possible (but not definitive) way to check whether Linux has loaded a driver module for your wireless card or not:

1. Turn on your PC without wifi card being installed.
2. At Linux prompt, type **lsmod > ~/before.txt**
3. Shutdown the system.
4. Insert your wireless card.
5. When the system starts, again at the prompt, type:  
**lsmod > ~/after.txt**
6. Now compare the two files:  
**cd ~**  
**diff after.txt before.txt**

The output I got on a laptop with an LG branded Orinoco-based chipset wireless card was as follows:

```
35,39c35,36
< orinoco_cs 8872 1
< orinoco 39756 1 orinoco_cs
< hermes 7264 2 orinoco_cs, orinoco
< pcmcia 26568 5 orinoco_cs
< yenta_socket 25292 3
---
> pcmcia 26568 4
> yenta_socket 25292 2
41c38
< pcmcia_core 49348 4 orinoco_cs, pcmcia, yenta_socket, rsrc_nonstatic
---
> pcmcia_core 49348 3 pcmcia, yenta_socket, rsrc_nonstatic
67c64
```

Yes, I appreciate that it looks like a load of rubbish, but what it shows is that there was a difference between the

two files, and all that should have changed is the fact that the Wireless card was present in the system. In the above example, it shows that the wireless card is being loaded with the driver 'orinoco\_cs'. It was all automatic. If you have the same luck, that's half the battle done with – the driver loads, you can get on with the next stage!

If you didn't fare this well, and generally there was no difference between the files above then no driver was found for your card. All may not be lost, but a bit of good hard googling is in order I'm afraid, and possibly –eek– building a new kernel. To start with, google for the **exact** version of your card (this is very important), along with the word 'linux' and see how you get on from there!

If you are running a modern version of Linux, anything from say, 2004 onwards should come pre-installed with the basic set of wifi-tools. These are basic command-line tools which we will use because they are included with every distribution of Linux, whereas GUI apps are often different between distros.

The main tools we are using are:

iwconfig  
ifconfig  
dhclient  
route

a text editor (to edit a few things)

Please ensure that you have these programs installed before we go forward. How do I check where these programs are? Try typing simply **iwconfig** at the prompt – you are likely to be presented with a screen of information about your wireless card. If not, try **/sbin/iwconfig** (redhat/fedora put iw/ifconfig in /sbin). Still no joy – try typing **locate iwconfig** or **which ifconfig**. If still no luck, then you will need to install it from your distro's CD or website.

You should be doing all of the following as the root user from now on. To become root, this depends upon your distro:

RedHat / Fedora / Suse / Mandriva:

```
$su – root
Password: [root password]
#
```

Ubuntu and some others:

```
$sudo bash
Password [root password]
#
```

Now that you are root, lets see what we get from iwconfig. Here's what my configuration shows:

```
eth0 No Wireless Extensions

eth1 IEEE 802.11-DS ESSID:"oddjob" Nickname:"Prism I"
Mode:Managed Frequency:2.442 GHz Access Point: 00:0D:0B:4A:9B:5F
Bit Rate:11 Mb/s Tx-Power=15 dBm Sensitivity:1/3
Retry min limit:8 RTS thr:off Fragment thr:off
Power Management:off
Link Quality=53/92 Signal level=-49 dBm Noise level=-139 dBm
Rx invalid nwid:0 Rx invalid crypt:3507 Rx invalid frag:0
Tx excessive retries:2588 Invalid misc:0 Missed beacon:0
```

The above output may differ to your setup, but here, it shows that the interface eth1 is my wireless card. It shows that it's currently associated to Wireless Access Point called 'oddjob'. This may or may not be correct, check the name of your wireless point in the configuration for your wireless router/access point to confirm that the card hasn't picked the wrong name.

If you are happy that the AP selected by the card is good, then let's manually set up the card (just once), so that we know we've got it all right. If you want more information about iwconfig, type iwconfig --help, but here is the basic stuff you want to use to configure your card:

`ifconfig eth1 up`  
(sets up the actual interface for use)

`iwconfig eth1 essid oddjob`  
(This sets the card to use the AP name oddjob, may not be necessary)

`iwconfig eth1 mode managed`  
(This tells the card that you have an Access Point/Wifi router, rather than a peer-to-peer setup, again, may not be necessary)

`iwconfig eth1 enc on`  
(If you use WEP then you want to switch WEP on with this command)

`iwconfig eth1 enc FE:ED:DE:AD:BE:EF:A7:DA:CA:FE:FE:ED:00`  
(This command sets the wireless encryption code you have set on your router)  
(note that this is a hex based number – rather than ascii text string, because it's easier to use hex numbers, eg 0–F rather than text. If you use text strings on your router, consider using just one hex line instead if you run into bothers with your setup.)

`dhclient eth1`  
(This line uses DHCP from your router to give your PC an IP address and a way out to the Internet)  
(If your router doesn't provide DHCP, or you use static IP addresses, then you will need to do as follows to set your IP etc:)

**`ipconfig eth1 [the IP you use] netmask [eg 255.255.255.0]`**  
**`route add default gw [IP address of internet router or default gateway]`**  
**`echo "nameserver [IP address of DNS server, eg routers IP]" > /etc/resolv.conf`**

As you can see, the whole process is a lot easier when you have DHCP enabled on your router/AP, so unless you have a good reason not to, enable DHCP on your router so that all the DNS, Routing and IP information is automatically configured on the card.

To see whether your configuration worked, try doing **ping google.com**. If it worked you'll see something like:

64 bytes from 64.233.167.99: icmp\_seq=1 ttl=240 time=244 ms

Press Ctrl+C to stop the ping.

If you are happy with the results, then you have correctly configured your card. If you are still having problems, make sure that a few things are set:

- ◆ The Card interface is up (eg `ifconfig CARD up`)
- ◆ If you don't use DHCP that your `/etc/resolv.conf` (dns) file is valid
- ◆ Check the output of the route command again if you don't use DHCP, check its right
- ◆ Check the IP address of your card is right by typing simply 'ifconfig' on it's own – most routers use 192.168.x.x

If things are all good, take your settings and put them into a text file with a text editor like so...

```
#!/bin/bash
```

```
echo "Starting wireless network..."
ifconfig eth1 up
iwconfig eth1 essid oddjob
iwconfig eth1 mode managed
iwconfig eth1 enc on
iwconfig eth1 enc FE:ED:DE:AD:BE:EF:A7:DA:CA:FE:FE:ED:00
dhclient eth1
```

Save the file in /etc/init.d, call it wifi-start or something like that.  
Give it permission to execute:

```
chmod 755 /etc/init.d/wifi-start
```

Make a link to the file in your init startup folder:

```
ln -s /etc/init.d/wifi-start /etc/rc2.d/S36wifi-start
```

```
ln -s /etc/init.d/wifi-start /etc/rc3.d/S36wifi-start
```

S36 is the order it starts up. If you look in your rcX.d folders you'll see other services starting with different numbers. If you have a problem with the order in which the wifi-start script starts, eg, you want pcmcia-cs to start up first (it should do on laptops), then change the S36 to SXX where XX is a number greater than the one for pcmcia-cs etc.

If you restart your machine then the wireless settings in that script should start up with your machine. If it doesn't, check the runlevel of your machine which you can find from /etc/inittab. My Ubuntu machine shows the following:

id:2:initdefault:

Meaning that /etc/rc2.d is the startup directory for Ubuntu. This is different for other distros. Make sure that the script is linked into the folder for whichever default your distro uses.

If you have problems doing this, the easiest thing to do is place the file in /etc/init.d and give it permission (chmod) to run, and then the KDE System Settings window > System Administration Menu > System Services menu can be put to use. Make sure you click 'Start during boot' for start-wifi.

If you have any comments, fixes or updates for this Hot Topic then [Contact Us!](#)

This Hot Topic was last tested on: Ubuntu 4.06

## How Do I install Linux on one hard drive, and use Windows on another?

---

The situation is this: You have two drives. One has been running Linux on it, and the other, you wish to run Windows upon, for booting back and perhaps playing the occasional game in real Windows.

How did I do it? Here it is:

I have SATA as my primary Linux drive, so it sits on a completely different interface than the drive I used to test this, which was an IDE drive. Regardless, this scenario should work just fine on an IDE only setup (assuming you can put one drive on IDE channel 1, and one drive on IDE channel 2 (not master/slave, two separate masters).

First off, I have Linux working away just fine on my main drive.

Next, I remove that drive from the PC completely, so that the computer would not know anything about it.

Next, I attach a blank IDE drive to my IDE connector and install Windows in a completely normal fashion.

Again, make sure that if you have an IDE only setup, install this drive on the secondary IDE channel, not the same one as the Linux drive.

Once Windows is set up on the drive, install your Linux drive back in. The PC should now boot into Linux as before.

Now, edit your bootloader file (grub) by editing the /boot/grub/menu.lst file as root:

\$sudo gedit /boot/grub/menu.lst go to the bottom of the file, and add an entry for Windows there. Here is what the end of my menu.lst looks like:

```
### END DEBIAN AUTOMAGIC KERNELS LIST
title Windows 2000
map (hd0) (hd1)
map (hd1) (hd0)
root (hd1,0)
chainloader +1
```

finally, edit, near the top of the file the part that says :

hiddenmenu

put a # in front of it, to comment it out.

Save the file and reboot your PC, you should now see Windows 2000/XP in your boot menu, along with Linux.

The various options covered perform the following tasks:

```
map (hd0) (hd1)
map (hd1) (hd0)
```

This part is the weird one: This makes the drives swap around for my SATA setup. For example, it makes the SATA drive (hd0) become the second drive in the system (hd1), by a bit of trickery. It also reverses (hd1), the second drive (windows), into being the primary drive on the system. If you don't have a SATA/IDE setup like this, you may not need this, so if it gives you problems, try removing these two lines.

```
root (hd1,0)
```

This line tells grub what partition to boot

chainloader +1 This line is necessary to boot a Windows system, it passes grubs boot handling sequence down to the Windows bootloader.

This Hot Topic was last tested on: Ubuntu 5.10 /Windows XP



## What's the X Window System?

---



### Topics covered in this hot topic:

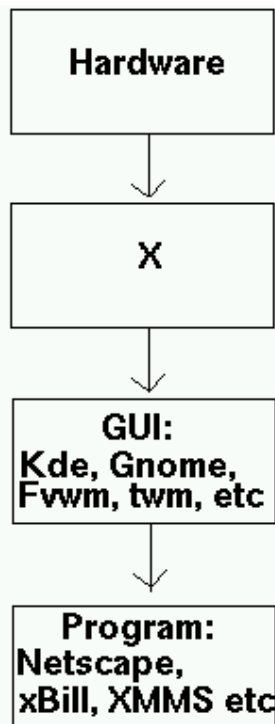
What X is not  
What X does  
Some X heritage

#### What X is Not:

- X is not a product by Microsoft with the letter X before it (yes, I've heard that before!).
- X does not affect the way graphical windows on your screen look
- X won't let you browse through files in a graphical manager
- X is NOT a GUI (Graphical User Interface)
- X doesn't sound like it does a lot, does it? **Well It DOES!**

#### What X does:

X is the layer between the hardware on your system (your graphics card, and so on) and the GUI that sits on top of X. Have a look at the following diagram to get the general idea:



*When a program is started up, it goes through the process of first talking to the GUI, about what to do with it's windows, ie: placement, focus and so on. The GUI applies it's thoughts to the process, applies the look, menubars (File, Help, Close, Minimise, etc), and all decorations to the window, then passes it to X. X has the final decision on where it places it on a screen. It then talks to the hardware, making it issue the process.*



Tux says: GUI stands for Graphical User Interface, and comprises of things like icons, menus, pointers and windows. It is usually pronounced 'Gooley'.

Without X in the equation, the GUI that you use, or properly termed: the Window Manager couldn't do anything. X is also a server. We usually refer to it as XFree86 or just as X, but indeed it's proper title is an X Window Server. When you run X on your own linux box, X has to determine whether you want to display it on a local machine and on a local monitor, or on a remote monitor. You can also have more than one X session running on your linux box at the same time. If you have 2 monitors, and a dual headed graphics card (ie: Matrox G400 DH), you can get X to display Jane's X session on monitor 1 and Bob's X Session on monitor 2, all running on one computer, at the same time. Show me Windows 98 do that, please :)

### Some X Heritage

If you're interested in knowing how X came to be, this chapter is for you.

X was an idea conceived by many: In 1984, Apple had released it's first graphical user interface. If you were either too young, or never saw it. I reckon you should definitely take a look at the 1984 TV advertisement of the first Macintosh. It's a wonderful piece of art. You can see this great piece in computing history [HERE](#), although be weary— it's around a 13MB quicktime movie— you'll need a fast connection if you want to see it in the next hour! Even prior to this, a group of real boffins with some way far-out ideas (circa 1978) had been stroking their beards, drinking lots of hi-caffeine coffee/jolt and creating the very \*VERY\* first GUI. It was made by the researchers in Xerox's PARC (Paulo Alto Research Center), where they designed a GUI called Elixir, and terminal systems to go with it (called the Star). Unfortunately, it was not a commercial success, mainly due to high pricing.



*The Xerox Star Computer, Running Elixir*

The last main demonstration of the Star and the Elixir Desktop os was around 1981, to see what I'm jabbering on about, check out [Bruce Damer's Personal Histories of the Desktop User Interface](#). Which is a compelling insight into the very first moments of the Graphical User Interface. There are some pretty zany pictures of bearded people in bell bottoms as well, so you have been warned!

Anyways... After Elixir, a few GUIs appeared (amongst others): GEM, TOS and the Mac System OS, in 1984. Less than two years later in 1986, a consortium of UNIX developers including Sun Microsystems, Silicon Graphics and

AT&T created X. Originally it shipped with the window manager called TWM. The source of this product and who created it, are still unknown.

TWM (or, as most know it today; "THE Window Manager"), was the interface that sat ontop of X, and allowed you to open up Xterms, resize windows and do basic window manipulation. It was pretty darn basic to say the least. Then came FVWM, which was a major improvement. For Linux geeks and for people who want a really light window manager, you'll find twm and fwm still in use today. It still ships with all the latest distros. FVWM2 came out years later, along with other window managers such as NeXt/NeXtStep, AfterStep, IceWM, QvWM (which is supposed to look identical to Windows 98 btw), KDE, Gnome and many others. In 1990, Microsoft Windows 3.0 was released (Versions 1.0 and 2.0/286 of Windows was actually text based -- essentially).

If you read the next 2 chapters, on Gnome and KDE, Linux's top 2 Window Managers, you'll soon see what all the fuss is about. Originally, like MS-DOS and UNIX et al, Linux was essentially a command line driven operating system and it excelled at that, it's now got multiple point and click GUIs, and well, obviously, it excels at that too!