

The Edubuntu Handbook

Copyright © 2004, 2005, 2006 Canonical Ltd. and members of the Edubuntu Documentation Project

Copyright © 2004, 2005, 2006 Canonical Ltd. and members of the Ubuntu Documentation Project

Credits and License

Maintainers for the Edubuntu Documentation Team:

- Jonathan Carter
- Will van der Leij

Contributors:

- Susan Stewart
- Robin Shephard
- Richard Johnson
- Scott Balneaves
- James McQuillan
- Jonathan Mueller
- Francis Giraldeau
- Carol Wiebe

Disclaimer

Every effort has been made to ensure that the information compiled in this publication is accurate and correct. However, this does not guarantee complete accuracy. Neither Canonical Ltd., the authors, nor translators shall be held liable for possible errors or the consequences thereof.

Some of the software and hardware descriptions cited in this publication may be registered trademarks and may thus fall under copyright restrictions and trade protection laws. In no way do the authors make claim to any such names.

THIS DOCUMENTATION IS PROVIDED BY THE AUTHORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

COLLABORATORS

	<i>TITLE :</i> The Edubuntu Handbook	<i>REFERENCE :</i>	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 2, 2008	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Introduction	1
1.1	About Edubuntu	1
1.2	About the Edubuntu Handbook	2
1.2.1	Aim	2
1.2.2	Intended Audience	2
1.3	Ubuntu and its Philosophy	3
1.3.1	Free and Open Source Software	3
1.3.2	Free software	3
1.3.3	Open Source	3
1.4	Why Edubuntu?	4
1.4.1	Compatible	4
1.4.2	Secure	4
1.4.3	Manageable	4
1.4.4	Cost Effective	4
1.4.5	Well Supported	4
1.4.6	Built for Education	5
1.5	Choosing The Correct Installation Type	5
1.6	Basic Concepts: Networks and Networking	5
1.6.1	Hardware	5
1.6.1.1	Wired	5
1.6.1.1.1	Hub	5
1.6.1.1.2	Switch	6
1.6.1.1.3	Router	6
1.6.1.2	Wireless	6
1.6.2	Software	6
1.6.2.1	DHCP (Dynamic Host Configuration Protocol)	6
1.6.2.2	DNS (Domain Name System)	6
1.6.2.3	NTP (Network Time Protocol)	6
1.6.2.4	Web Server	7
1.6.2.5	Web Proxy	7
1.6.2.6	Firewall	7

2	Edubuntu Workstation	8
2.1	Hardware requirements	8
2.2	Installing Edubuntu Standalone	8
2.2.1	Boot to the Edubuntu install CD	9
2.2.2	Make sure your Edubuntu install CD is complete and undamaged	9
2.2.3	You are now ready to install Edubuntu	10
2.2.4	Choose language	10
2.2.5	Select keyboard layout	10
2.2.6	Configure the network	10
2.2.7	Choose a mirror of the Ubuntu archive	11
2.2.8	Partition Disks	11
2.2.9	Configure time zone	11
2.2.10	Set up users and passwords	11
2.2.11	Finish the installation	11
2.3	The Edubuntu Addon CD	12
3	Edubuntu Server and Thin Client Computing	14
3.1	Hardware Requirements	14
3.1.1	LTSP Thin Client hardware requirements	15
3.1.1.1	Clients	15
3.1.1.1.1	Older hardware	15
3.1.1.1.2	CPU	15
3.1.1.1.3	Network	15
3.1.1.1.4	Video Card	15
3.1.2	Edubuntu Server requirements	15
3.1.2.1	Recommended specs	16
3.1.2.1.1	Memory	16
3.1.2.1.2	Processors	16
3.1.2.1.3	Disks	16
3.1.3	Network	16
3.2	Theory of operation	17
3.2.1	The LTSP chroot environment	17
3.2.2	The boot process of a thin client	17
3.2.3	Network booting the thin client	19
3.2.3.1	Boot ROM	19
3.2.3.1.1	Etherboot	19
3.2.3.1.2	PXE	19
3.2.3.2	Local media	19
3.2.3.2.1	Floppy disk	19

3.2.3.2.2	Hard disk	19
3.2.3.2.3	CD-ROM	19
3.2.3.2.4	USB Memory device	20
3.3	Customizing thin client behaviour	20
3.3.1	Format of the lts.conf file	20
3.3.1.1	Section headings	20
3.3.1.2	Variable Assignments	20
3.3.1.3	Location of the lts.conf filename	20
3.3.1.4	About using NBD instead of NFS	21
3.3.1.5	Sample lts.conf file	21
3.3.2	The LDM display manager	22
3.3.2.1	Theory of operation	22
3.3.2.2	Encrypted versus unencrypted sessions	23
3.3.2.3	Auto login features	23
3.3.2.4	Load balancing features	23
3.3.3	General thin client parameters	23
3.3.4	Screen Scripts	24
3.3.5	Modules and startup scripts	25
3.3.6	X-Windows parameters	25
3.3.7	Printer configuration parameters	28
3.3.8	Keyboard parameters	28
3.4	Multiple server setup	29
3.4.1	Infrastructure setup	29
3.4.1.1	Network topology	29
3.4.1.2	Common authentication	29
3.4.1.3	Shared home directories	29
3.4.1.4	Shared printers	30
3.4.1.5	Synchronization of packages	30
3.4.1.6	Managing the SSH known hosts file	30
3.4.1.7	Setting network forwarding	30
3.4.1.8	Replication of desktop profiles	30
3.4.2	Session dispatching	31
3.4.2.1	Define the server list	31
3.4.2.2	Dispatching method	31
3.5	Network Swap	31
3.6	Managing DHCP	32
3.6.1	Adding static entries to the dhcpd.conf	33
3.6.2	DHCP failover loadbalancing	33
3.7	Lockdown with Sabayon and Pessulus	33

3.8	Edubuntu Thin Client Manager	33
3.8.1	Using Thin Client Manager	33
3.8.2	Operations	34
3.8.2.1	Process Management	34
3.8.2.2	Logging Users Out	35
3.8.2.3	Sending A Message	35
3.8.2.4	Starting A Program	35
3.8.2.5	Lockdown Editor	36
3.8.2.6	Managing users with groups	36
3.8.2.7	Plugins	36
3.8.2.8	Screen Viewing	36
3.9	Keeping your Edubuntu server in shape	36
3.9.1	Edubuntu Server Management	36
3.9.2	Updating your LTSP chroot	39
3.9.3	Changing the IP of your LTSP server	40
3.10	Backing Up	40
3.10.1	What needs to be backed up?	41
3.10.1.1	Back up everything	41
3.10.1.2	Back up configs and data	41
4	Use, Configure, Support, and Contribute	42
4.1	Introduction to available software	42
4.1.1	Office Suite	42
4.1.2	Internet	42
4.1.3	Graphics	43
4.1.4	Sound and Video	43
4.1.5	Science	44
4.1.5.1	Explore the stars	44
4.1.5.2	Research the periodic table	44
4.1.5.3	Test your chemistry skills	44
4.1.6	Mathematics	45
4.1.6.1	Program in Logo	45
4.1.6.2	Create geometric constructions	45
4.1.6.3	Plot mathematical equations	45
4.1.6.4	Calculate percentages	45
4.1.6.5	Practice Fractions	45
4.1.6.6	Play at Math	45
4.1.7	Drawing	46
4.1.7.1	Tux Paint	46

4.1.7.2	Generate Fractals	46
4.1.8	Language	46
4.1.8.1	Practice Spanish verbs	46
4.1.8.2	Play Hangman	47
4.1.9	Basic Skills	47
4.1.9.1	Generate and give tests	47
4.1.9.2	Learn touch typing	47
4.1.9.3	Gcompris	47
4.2	Documentation and Support	48
4.2.1	Supplied Documentation	48
4.2.1.1	Accessing Documentation from the Command-line	48
4.2.2	Online Resources	48
4.2.2.1	Ubuntu Web Sites	48
4.2.2.2	Other Documentation Web Sites	49
4.2.3	Asking Questions Online	49
4.2.3.1	Using Web Search Engines	49
4.2.3.2	Community Support Facilities	49
4.2.4	Commercial Support and Training Providers	49
5	Contributing to Edubuntu	50
5.1	Are you a user?	50
5.2	Are you a teacher, parent, or student?	50
5.3	Are you multilingual?	50
5.4	Are you a writer, editor, or proofreader?	50
5.5	Are you an artist?	50
5.6	Are you a coder?	51

Abstract

A comprehensive guide to using Edubuntu in a thin-client (LTSP) and stand-alone environment.

Chapter 1

Introduction

Welcome to the Edubuntu Handbook. This text has been many release cycles (and many titles) in the making. We, the authors, hope to continue building on this work for the life of the Edubuntu distribution of Linux.

This version of the Handbook is written for Edubuntu 7.10, the *Gutsy Gibbon*, released in October 2007.

Thank you for allowing us to walk you through installing and using Edubuntu. We hope to make it an interesting journey, and we look forward to your feedback.

Kind regards,

The Edubuntu Documentation Team

1.1 About Edubuntu

Edubuntu is an operating system designed with education in mind. It is based on **Ubuntu**, a complete operating system that uses the Linux kernel and is freely available to anyone. As an education-driven operating system, Edubuntu provides a complete solution for:

- An organisation or Education Ministry that is wanting to take full advantage of the benefits that open source software offers to the educational environment.
- Educators and school personnel who would like to set up Edubuntu in a networked learning environment.
- Home users who would like to have a standalone computer system that focuses on education for the younger members of the household.

Edubuntu gathers together the best available free software and digital materials for education.

Since Edubuntu is based on Ubuntu, the team behind Edubuntu is part of the growing Ubuntu community. The Ubuntu community is built around the ideals enshrined in the **Ubuntu Philosophy**:

- that software should be available free of charge;
- that software tools should be usable by people in their local language and despite any disabilities;
- and that people should have the freedom to customize and alter their software in whatever way they see fit.

For these reasons, Edubuntu makes the following public commitment to its community of users worldwide:

- Edubuntu will always be free of charge, and there is no extra fee for the "enterprise edition;" we make our very best work available to everyone on the same Free terms.
-

- We include the very best in translations and accessibility infrastructure that the free software community has to offer, to make Edubuntu usable for as many people as possible.
- Releases are regular and predictable; a new one is made every six months. You can use the current stable or the development release. Each release is supported for at least 18 months and selected ones are supported for up to 3 to 5 years on a Long-Term Support (LTS) basis.
- We are entirely committed to the principles of open source software development; we encourage people to use open source software, improve it, and pass it on.

Edubuntu is meant for anyone who wants a computer laboratory that makes good use of free software and the widest variety of hardware. It is designed to be easy to administer and is well documented. It can run on inexpensive, 10-year-old IBM-compatible computers as well as brand new I64-based systems. The growing Edubuntu community is working together to share ideas and solutions for bringing computing and communications power to teachers, students and community members everywhere.

One of the key technologies included is the Linux Terminal Server Project (LTSP) which allows you to boot thin clients from an Edubuntu LTSP server. For educational environments, LTSP lowers hardware costs by enabling the use of older or less powerful machines as thin clients, as well as reduced administration overhead by having only to install and maintain the software on the server. When a workstation fails, it can simply be replaced without data loss or reinstallation of the operating system.

1.2 About the Edubuntu Handbook

The Edubuntu Handbook is a collaborative effort from many individuals within the community. It is intended as a guide and a reference to help the new Edubuntu user and administrator with their system whether it is a standalone system or in a thin-client configuration.

The community has been busy at work finding ways to make Edubuntu meet their practical needs, and the solutions they've provided may be just what you are looking for to meet your needs as well. This is the spirit of open collaboration upon which Edubuntu thrives.

1.2.1 Aim

The ultimate aim of Edubuntu is empowerment: to place state-of-the-art Information and Communication Technology (ICT) within the reach of everyone. However, computers and networks are complicated concepts and simply knowing how to ask the right questions can be very hard if you don't already have a lot of experience. Having the technology at your fingertips should go hand-in-hand with the resources that enable you to know how to use it effectively. This book is intended to provide you with the information necessary to be confident and self-sufficient in working with Edubuntu.

1.2.2 Intended Audience

This book will cater to different audiences:

- It will provide a step-by-step guide to get started with an Edubuntu computer lab. Anyone with enough enthusiasm, as well as access to the equipment necessary, can use this book as a blueprint for setting up local community computer labs.
 - It will serve to document what Edubuntu gives you in a computer lab and to help you understand it in order to keep it in good running order.
 - Your level of ICT skill and competency does not need to be high to come to grips with the contents of this book. Whether you are an educator with a basic understanding of ICT in general, an ICT administrator in a school without knowledge of open source systems or even an experienced IT professional, this book should empower you to get the most out of your Edubuntu system.
-

1.3 Ubuntu and its Philosophy

Our work on Ubuntu and Edubuntu is driven by a philosophy of software freedom that we hope will spread and bring the benefits of software technology to all people, everywhere.

1.3.1 Free and Open Source Software

Ubuntu is a community-driven project to create an operating system and a full set of applications using free and open source software. At the core of the Ubuntu Philosophy of Software Freedom are these core philosophical ideals:

1. Every computer user should have the freedom to run, copy, distribute, study, share, change and improve their software for any purpose, without paying licensing fees.
2. Every computer user should be able to use their software in the language of their choice.
3. Every computer user should be given every opportunity to use software, even if they work under a disability.

Our philosophy is reflected in the software we produce and included in our distribution. As a result, the licensing terms of the software we distribute are measured against our philosophy, using the Ubuntu License Policy.

When you install Ubuntu, almost all of the software installed already meets these ideals. We are working to ensure that every single piece of software you need is available under a license that gives you those freedoms. Currently, we make a specific exception for some "drivers" which are only available in binary form, without which many computers will not complete the Ubuntu installation. We place these in a restricted section of your system which makes them trivial to remove if you do not need them.

1.3.2 Free software

For Ubuntu, the "free" in "free software" is used primarily in reference to freedom and not to price, although we are committed to not charging for Ubuntu. The most important thing about Ubuntu is not that it is available free of charge, but that it confers rights of software freedom on the people who install and use it. It is those freedoms that enable the Ubuntu community to grow, sharing its collective experience and expertise to improve Ubuntu and make it suitable for use in new countries and new industries.

Quoting the Free Software Foundation's "What is Free Software", the principles at the core of free software are defined as:

- The freedom to run the program, for any purpose.
- The freedom to study how the program works, and adapt it to your needs.
- The freedom to redistribute copies so you can help others.
- The freedom to improve the program, and release your improvements to the public, so that everyone benefits.

Free software has been a coherent social movement for more than two decades. This movement has produced millions of lines of code, documentation, and a vibrant community of which Ubuntu is proud to be a part.

1.3.3 Open Source

Open source is a term coined in 1998 to remove the ambiguity in the English word "free". The Open Source Initiative described open source software in the Open Source Definition. Open source continues to enjoy growing success and wide recognition.

Ubuntu is happy to call itself open source. While some refer to free and open source as competing movements with different ends, we do not see free and open source software as either distinct or incompatible. Ubuntu proudly includes members who identify with both the free software and open source camps and many who identify with both.

Our work on Ubuntu is driven by a philosophy **based** on software freedom that we hope will spread, and bring the benefits of software technology to all parts of the globe.

1.4 Why Edubuntu?

1.4.1 Compatible

Most educational computer networks are heterogeneous, consisting of various operating systems, and it is sometimes difficult to configure them to work together. Edubuntu plays well with others.

Edubuntu servers can happily coexist on the same network as other operating systems. OpenOffice.org can open and save many commercial office suite formats. You can even install Edubuntu and another operating system on the same machine. Need to share files between Edubuntu and other operating system or connect to a remote server? Edubuntu does that too.

The Edubuntu developers work hard to make sure that Edubuntu is the most compatible operating system you can find in schools today.

1.4.2 Secure

Security has become a key challenge for educators and the team behind Edubuntu recognize this. Often schools lack the specialized IT staff or time to lock and clean up computers.

Edubuntu, being a Linux-based operating system, enjoys the security advantages of its Unix-like and open source heritages. This translates into higher quality code and less spyware and viruses than plague other operating systems. In addition, it has a strict, proactive security policy which means that many common problems, such as open ports or misconfigured software, never make it into the released product. Finally, it is a true multi-user operating system, making it easy to allow users to complete their tasks without having a level of access that could compromise the system.

1.4.3 Manageable

With teachers and school IT departments deploying and administering an increasing number of computers, it is difficult to find time to manage individual machines. Edubuntu, with the option of using LTSP thin client technology, makes deployment and management simple and easy. A single server is all that is needed to set up, manage and administrate an entire network. The Edubuntu team also recognizes that not every school's setup is the same, so they've made Edubuntu easy to customize for your unique needs.

1.4.4 Cost Effective

With ever-increasing demands on school budgets, expensive technology is often last on the list. Edubuntu can help you offer what your students increasingly require from computer technology, without breaking the bank. Edubuntu is and always will be free to acquire, use and modify.

Need to set up another machine? Or another 100? Just install them! With Edubuntu you'll have no more expensive OS upgrades and licenses, and having specialized programs on only some computers will become a thing of the past. When you build your network on Open Source software, you are freed to seek support for your computers from whomever you wish.

Edubuntu can also help you save hardware costs, by allowing you to redeploy older machines as thin clients using LTSP technology.

1.4.5 Well Supported

Edubuntu support is available from both the Edubuntu community and the larger Ubuntu community. Many of the authors of the software included in Edubuntu, including the Edubuntu developers themselves, can be contacted directly via mailing lists and IRC channels.

There are many forms of support available, including mailing lists, Wiki websites, IRC channels, and bug trackers. There is also a special support group for using Edubuntu in schools. Should you want paid support, Canonical, the company which funds Edubuntu and Ubuntu development, can offer assistance, or you can find a local company who offer similar services. In fact, some of the money that would have gone to purchasing software can instead be spent to hire local experts to help train you, and to help you support your network. Edubuntu can help you take more control over your network while also benefitting your local economy. With Edubuntu, these choices are yours.

1.4.6 Built for Education

Above all, Edubuntu is set apart from other operating systems by its unwavering focus on the educational needs of learners all over the world. Edubuntu's motto is "Linux for Young Human Beings" and every development decision and application has that goal in mind.

Edubuntu comes with translations for many languages and localization features that allow people from all over the world to enjoy their computing experience. Edubuntu's accessibility features strive to provide a pleasant, high-quality computing experience to disabled users. The Edubuntu LiveCD makes it easy for pupils to boot their home computers temporarily into Edubuntu and use exactly the same applications that they are using at school. The LTSP server software allows teachers and administrators to create a low cost computer lab so that students can have access to the educational opportunities that Edubuntu and the Internet can provide.

1.5 Choosing The Correct Installation Type

Edubuntu can be installed or configured in two primary forms:

- Workstation or Stand-Alone install, just like an ordinary desktop PC fine-tuned for educational purposes.
- Server install which has a configured LTSP environment by default and various added network and collaboration tools.

The choice of install method, therefore, really depends on the type of network configuration that is desired.

1.6 Basic Concepts: Networks and Networking

There are two components of a network: hardware and software. This section will give an introduction to both.

1.6.1 Hardware

Networking works by breaking files and other data into little packets of information. These packets are transferred over a network. The difference between various types of networks is how they transfer packets.

There are two types of networking hardware: wired and wireless.

An important fact to remember is that a network will be only as fast as the slowest part. So, if you have a faster wired network but a slower wireless network, the speed to the wireless clients will be restricted by the speed of the wireless network.

1.6.1.1 Wired

Wired networking transfers packets over a cable that resembles a telephone cord, but with more wires. Wired networks can transfer packets at one of three possible speeds: 10 Mbit/sec, 100 Mbit/sec, or (Gigabit) 1000 Mbit/sec.

A network is only useful if it can connect multiple computers. There are some pieces of hardware that allow multiple computers to be connected in a network. They look alike, but they function differently and, likewise, operate at different speeds.

1.6.1.1.1 Hub

A hub is the simplest way to connect multiple computers. A hub has a lot of ports in the front and usually has several small lights corresponding to each port. The hub takes a message it receives on one port and resends it to all the ports. As a result, only one port can talk at a time.

1.6.1.1.2 Switch

A switch looks a lot like a hub; it has a lot of ports in the front and usually has several small lights corresponding to each port. However, a switch is unlike a hub because it only makes a connection between the ports it needs to. A switch can have multiple connections at the same time. This allows a switch to be faster than a hub.

1.6.1.1.3 Router

A router is used to make a connection between two networks. Routers are also commonly used to connect a LAN (local area network) to the Internet.

1.6.1.2 Wireless

There are four types of wireless networking.

Three types use the 2.4Ghz band of the radio frequency spectrum:

- 802.11b - 11 Mbit/sec
- 802.11g - 56 Mbit/sec
- Pre-802.11n - ~200 Mbit/sec

One type uses the 5Ghz band:

- 802.11a

1.6.2 Software

The most common network infrastructure services include:

1.6.2.1 DHCP (Dynamic Host Configuration Protocol)

Each computer on a network needs a unique identifier called an IP address. The IP address allows packets to be directed to the computer, much like a street address allows mail to be delivered to the correct house. An IP address follows a specific form: four groups of digits forming a number from 0 to 255. For example, a local IP address might be 192.168.2.50.

For convenience, a computer's IP address can be given by a server running the Dynamic Host Configuration Protocol (DHCP) service. DHCP automatically provides network settings to the computers on the network. With DHCP, there is no need to keep track of each computer's IP address.

1.6.2.2 DNS (Domain Name System)

DNS is a service that runs on a server, and it is like a phone book for computers, except that it stores IP addresses instead of phone numbers. Your computer talks to a DNS server every time you refer to another computer system with a name instead of an IP address. For example: www.edubuntu.org, wikipedia.org, and google.com are all DNS hostnames.

1.6.2.3 NTP (Network Time Protocol)

NTP is a service that runs on a server and allows other computers to synchronize their clocks. The server synchronizes with an extremely accurate atomic clock, and then the clients synchronize with the server.

1.6.2.4 Web Server

A Web server answers queries using protocols such as HTTP, and sends content such as web pages back to clients. Your Web browser almost exclusively talks to Web servers.

1.6.2.5 Web Proxy

A Web proxy is a service that runs on a server and accesses Web sites on behalf of the clients. A proxy can cache some data to allow faster repeated access to commonly accessed pages. A proxy can also restrict access to content that you deem inappropriate.

1.6.2.6 Firewall

A firewall can be a service on the server or a separate piece of hardware that runs a similar service. A firewall can protect your server (and clients) by restricting computers on the Internet from initiating connections into your server or network.

Chapter 2

Edubuntu Workstation

A workstation installation of Edubuntu is ideal for:

- Home computers.
- Small labs that lack the server power (See Section 2.1) for thin client computing, but have workstations capable of running Edubuntu standalone.
- Laptops and other computers that will not always be connected to the same network.

A workstation installation of Edubuntu is just what it sounds like: an install that can boot and function on its own, with or without a network connection. (An internet connection is needed when downloading new or updated software.) The first user created on the system can perform administration functions such as change settings that affect all users, installing new software, etc. System files, applications, and user data are all stored locally on the computer's hard disk, as opposed to being stored on a central server as in other Edubuntu installations.

2.1 Hardware requirements

A workstation Edubuntu system needs at least:

- 500 Mhz processor
- 128 MB of RAM
- 2.1 GB of hard disk space

Exceeding the minimums is always okay. For example, if you have a lot of user data, such as a large music or digital photo collection, that you would like to keep on your hard disk, you will need a bigger hard disk.

2.2 Installing Edubuntu Standalone

To make a workstation installation of Edubuntu, you will need:

- A computer that meets the hardware requirements for Edubuntu workstation.
- A keyboard and monitor.
- An Edubuntu install CD for your type of computer (architecture). You can order free CDs using [Shipit](#), or [download](#) and burn your own.
- These instructions.

A mouse or trackball is optional, but extremely useful.

2.2.1 Boot to the Edubuntu install CD

Put the CD in your CD drive, shut down the computer, then turn it back on, you should see the Edubuntu logo and install options appear on your screen.

If this does not work, shut down the computer again and carefully watch the messages that are displayed when you turn your computer on. Some computers require you to tell them that you want to boot to a CD instead of to your hard drive. If this is the case, your computer's start-up messages will tell you what button(s) to press to boot to CD.

Option one: Look for a message that contains the words "CD", "boot", and/or "media" and a button or buttons to push. It will be similar (but not necessarily identical) to one of the following:

- Press **F12** to change boot order.
- Boot CD/DVD: **F2**
- Press and hold the **Del** key to boot removable media.
- Press Ctrl-B to choose boot media

Pressing the button(s) when asked will give you a list of things you can boot. Choose the drive your Edubuntu CD is in.

Option two: Look for a message that contains the words "BIOS", "options", "set-up", or "configure", as well as a button or buttons to press. It will look similar (but not necessarily identical) to one of the following:

- Press the **Del** key to enter set-up.
- Press **Esc** for options.
- Press **F4** now to enter BIOS configuration.

Pressing the correct button(s) when asked will bring up a configuration screen. Each computer is slightly different, so just read the displayed information and find the option to boot from your CD drive first. Make certain that the hard drive boots after the CD drive, or your new Edubuntu system will not boot properly after it is installed.

Note

Often, these start-up messages scroll by so quickly that by the time you've read which button to press, it is too late! Don't worry. Just shutdown the computer and turn it on again. When the message comes up this time, you will be ready.

2.2.2 Make sure your Edubuntu install CD is complete and undamaged

Sometimes a CD gets scratched or marked, or there is an error while burning it. To prevent installation problems, it is wise to give the CD a check-up before installing from it. Use your arrow keys, or the Tab key to highlight "**Check CD for defects**" and press **Enter**.

Your computer will load some tools it needs, and then you will see: "Warning: this check depends on your hardware and may take some time. Check CD-ROM integrity?"

Use your tab or arrow keys to highlight **Yes** and press **Enter**. This is usually a good time to go grab a snack.

If your CD passes its check-up, you will see: "Integrity test successful. The CD-ROM integrity test was successful. The CD-ROM is valid."

*******If you do not see the success message, do not use the CD! Get a new one. Damaged or faulty CDs will not work properly and may cause your computer not to function.*******

When you see the success message, press **Enter** to continue. You will end up back at the "Warning: this check..." message. You are done, so choose **No** to reboot.

If you had to press a button to boot your CD last time, you will need to do so again.

2.2.3 You are now ready to install Edubuntu

Choose "Install a workstation" to begin installing Edubuntu. Your computer will load some tools it needs.

2.2.4 Choose language

Use your arrow keys to select the language you would like the installer to use, and press **Enter**.

The next screen asks what country you are in. Highlight your country and press **Enter**, or choose "other" if your country is not listed.

2.2.5 Select keyboard layout

The Edubuntu installer will guess what kind of keyboard your computer has. If you are sure it is correct, just press **Enter**.

If the layout shown is incorrect, you may choose "Find your layout by pressing some keys" and follow the on-screen instructions to help Edubuntu learn what type of keyboard you have.

If you already know the name of your keyboard layout, you may choose "Select from full keyboard list" to do just that.

If you would like to double-check that you have chosen the correct keyboard layout, choose "Test whether this layout is correct" and type a few letters, numbers, and symbols in the space provided.

Use the tab or arrow keys to select **Continue** and press **Enter**.

If your typing displayed correctly during the test, you should now highlight the keyboard name on this screen and press **Enter**.

If the test looked wrong, you should choose "Find your layout by pressing some keys" and then test again when done.

Once your layout has been confirmed, your computer will take a closer look at your CD-ROM drive and prepare for the rest of the installation.

2.2.6 Configure the network

The Edubuntu installer will now ask you to choose the network connection to use for the installation. If you are not sure which to choose, try the interface that the installer has highlighted for you. (If this turns out to be incorrect, you can always go back and try again.)

A network connection is optional. If you do not have one, you will not be able to download updates to software packages at the end of the install process, but you can still install all of the great software found on your Edubuntu install CD.

If your only network connection is in a cafe or library, you may not be able to use this connection to install and update from. These connections often require that you visit a local web page from your computer and enter a password or agree to a usage policy before connecting to the internet. This is okay! Just perform the installation without a network connection, and then add the connection after you boot to your new Edubuntu system.

If you are using a wireless network, the installer will ask you for a WEP key. If you do not know what this is, you probably don't need one and can just choose **Continue**.

The installer will try to detect your network settings for you. However, in some cases this is not possible, and the installer will prompt you for an IP address, netmask, and gateway. The person who manages your network will know what the correct settings are. For a small home network, some good guesses are:

- IP: 192.168.0.x or 192.168.1.x, where "x" is a whole number between 2 and 254
- Netmask: 255.255.255.0
- Gateway: 192.168.0.1 or 192.168.1.1

Still not sure? If you have another computer on this network, look at its network settings and use the same netmask and gateway. Set the IP so that the first three numbers are identical but set the last number slightly higher than the other computer. For example, if the other computer's IP is 192.168.1.100 you might try 192.268.1.105.

2.2.7 Choose a mirror of the Ubuntu archive

This section asks for proxy information. If you are unsure, you probably don't need a proxy. In this case, just choose **Continue**.

2.2.8 Partition Disks

In this step, you tell the Edubuntu installer how to set up your computer's hard drive.

Most people will want to choose the second option, "Erase entire disk:..." "This will erase everything on your hard disk" and use the entire hard disk for your new Edubuntu system.

If you plan to dual-boot Edubuntu and another operating system, you probably want to choose the first option, "Resize..."

Do not choose the third or fourth options, "Erase entire disk and use LVM..." and "Manually edit..." unless you are sure you know what you are doing with them.

The next screen will tell you about the changes being made to your hard disk. Select **Yes** to continue.

If your hard drive is large, this is a good time to go look for a beverage to go with your snack.

2.2.9 Configure time zone

Select your time zone and press **Enter**.

2.2.10 Set up users and passwords

This section walks you through creating a log-in for your first user. This user is the one responsible for installing and removing software, managing system-wide settings, and other maintenance.

Note

If you have used other Linux distributions before, you may be expecting to create a root login password. The user you are creating now is a normal user, but will have sudo privileges for package management and other tasks normally performed by root. By default, logging in directly as root is disabled on Edubuntu. If you don't know what this note means, just ignore it. You won't think anything is weird.

Once this user has been created, the Edubuntu installer will load the Edubuntu operating system and a selection of great applications on to your hard disk.

Note

The "Select and install software" progress bar stays at 5% for what seems like a very long time. Don't worry, the install is still progressing normally. The progress bar will move again after a few minutes.

2.2.11 Finish the installation

Take out your Edubuntu install CD and choose **Continue** to restart your computer and begin using your newly installed Edubuntu system.

Congratulations!

You now have a fully-functional Edubuntu system!

Enter the user name and password you created during the install process to log in.

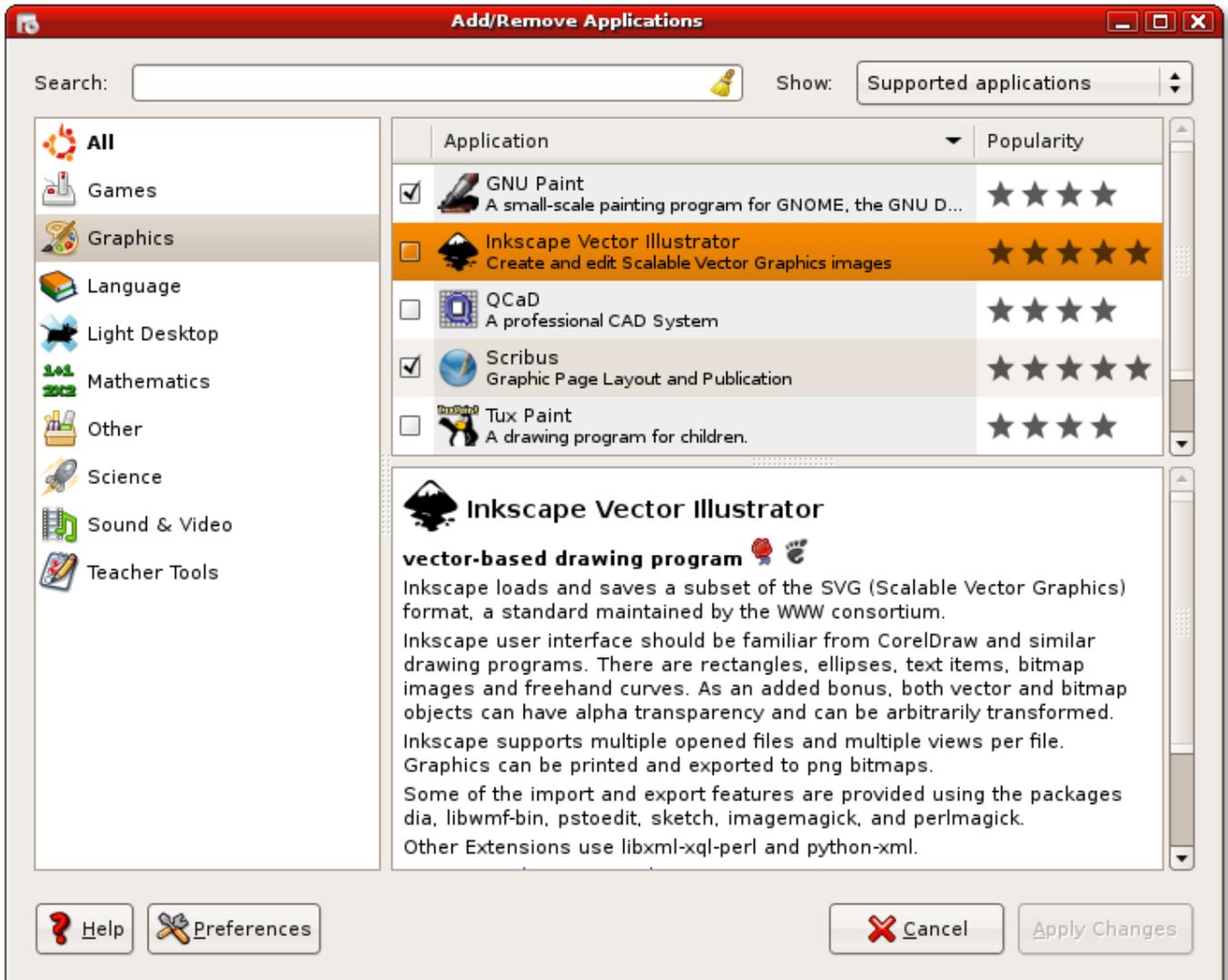
2.3 The Edubuntu Addon CD

Due to the amount of great software included in Edubuntu, the distribution doesn't all fit on one CD. We've split some of the educational software into a separate CD, and made it easy to install the software from there.

When you log in as an administrative user, and insert the Addon CD, you'll see the following dialogue box pop up:



Click on the **Start addon installer**, enter your password, and you'll see the following window pop up:



Simply check off the applications you'd like to add to your Edubuntu system, and press **OK**. They'll be installed from the CD.

Chapter 3

Edubuntu Server and Thin Client Computing

Edubuntu Server is almost the same thing as Edubuntu Workstation. It contains all the same great educational programs, and the same ease of use. The only difference is that it also installs, by default, a thin client environment.

The Edubuntu community has worked very closely with the Linux Terminal Server Project (LTSP), to come up with the next generation thin client solution for GNU/Linux. It comes automatically installed on an Edubuntu server, so you can begin setting up a thin client school lab right out of the box.

Thin client computing has been around for a long time in the UNIX world. Although the implementation has evolved quite a bit, the concept has remained the same:

- The thin client only takes care of the basic functions like display, keyboard, mouse and sound.
- The server does the heavy weightlifting. All the applications run on the server, and they simply display on the thin client.

Because the thin clients have a limited number of tasks to manage, the hardware for the thin client can be small and cheap. The thin clients themselves are basically maintenance free. They last longer because they have no storage with moving parts like hard disks. If they break no data is lost since nothing is stored on the client itself. Simply swap the client with another one and go back to work. If your thin client is stolen or put in the trash, no data ends up in the hands of unauthorized people.

The terminal server runs all applications and contains all the data. All the regular maintenance (software updates, administration) takes place on the terminal server. The number of thin clients that a terminal server can support is proportional to the power of the server. Because GNU/Linux in general, and Edubuntu in particular, make efficient use of resources, you can support a surprising number of thin clients from a machine which might only be considered a powerful single user system running other operating systems. Please see Section 3.1.2 for more details.

In a thin client computing environment, the stability of the server is important. It's important to make sure that your server has good power management, like installing a UPS, and depending on how much availability is required, redundant power supplies may be called for. As well, users who have the resources may decide to invest in multiple disks for RAID support, and other options which may be needed in a High Availability environment. However, you certainly don't need them in all environments, and Edubuntu's high quality means that in all but the most demanding environments, this won't be needed.

3.1 Hardware Requirements

A person setting up a LTSP thin client environment for the first time, typically asks two questions:

- Will my existing machines work as terminals, or, what should I buy to use as a terminal?
- How big a server do I need?

Chances are, hardware that you already have is more than sufficient for terminals. One of the great advantages of an Edubuntu Server is that you can set up a high quality lab of terminals for your students to use, by leveraging the machines you already have. As for servers, usually, it's very easy to turn any high-end single user desktop machine into a terminal server capable of handling many thin clients. We'll present some guidelines that should help in making the most of your resources.

3.1.1 LTSP Thin Client hardware requirements

A lot of Edubuntu deployments are in classroom environments, and usually, in these situations, the primary goal is to re-use existing hardware that the school already owns. However, specifically designed thin clients can be used also.

3.1.1.1 Clients

3.1.1.1.1 Older hardware

There are three things to consider when trying to re-use existing hardware:

- CPU
- Network
- Video Card

3.1.1.1.2 CPU

For using the default, secure mode of LTSP, you'll need to have a slightly faster CPU. Any 233 MHz or better CPU should provide acceptable performance.

If you have slower clients, in the range of 133 MHz to 233 MHz, you may be able to use them, if you're willing to reduce the security of your thin client network. More information on this is available in Section 3.3.

For the best possible performance, a 533 MHz processor will provide a first class thin client desktop experience.

3.1.1.1.3 Network

A thin client boots over the network, using a small program called a network boot loader. This network boot loader is sometimes located on the card itself, or, for older cards without one, the user can provide one on a floppy or CDROM which can be used to boot the thin client.

Three common network boot loaders which can be used are:

- **PXE:** This one is the most common, and many network cards and motherboards with built-in network cards support this. If you have one of these, you'll be able to boot without any problems.
- **Etherboot:** For older cards that don't have PXE included on them, you can use the Free Software equivalent, Etherboot. This excellent alternative to PXE can either be booted from a floppy, memory stick, or CDROM, or, if you're handy with electronics, be burned onto a EPROM if your card has a socket for one. More information on the project can be found at <http://www.etherboot.org>, and you can download ready-to-use Etherboot images at <http://www.rom-o-matic.org>.
- **Yaboot:** For Macintosh PowerPC machines (iMac's and later), you can use the built in Yaboot network boot.

3.1.1.1.4 Video Card

Typically, any video card that uses the PCI bus and has 2 MB or more of memory, should make a reasonable client.

3.1.2 Edubuntu Server requirements

An Edubuntu/LTSP thin client network is quite scalable; a moderately powerful machine can serve several thin clients, and if you need to add more thin clients, you can either expand the capabilities of the existing server, or, simply add more servers.

3.1.2.1 Recommended specs

Server sizing in an LTSP network is more art than science. Ask any LTSP administrator how big a server you need to use, and you'll likely be told "It depends". How big a server you need does depend largely on what it is you're planning on doing with your thin client network. The server requirements needed for a network where the only use will be a little light web-browsing, with no Java or Flash, will be greatly different from a network where you want to do heavy graphics, interactive games, and Flash animation. Here are some common guidelines that should fit most "average" cases.

3.1.2.1.1 Memory

Edubuntu, being a GNU/Linux based operating system, makes efficient use of memory. The usual formula that's used for adding memory to a thin client server is:

$256 + (128 * \text{users}) \text{ MB}$

So, if your target is to have a server with 20 terminals, you'll need:

$256 + (128 * 20) = 256 + 2560 = 2816 \text{ MB}$

Rounding up, you'll need 3 1 Gig simms. Making sure you've got enough memory is the single most important thing you can do to help the performance of an Edubuntu/LTSP thin client server. If you do not have enough memory in your server, you'll find your server will have to use the hard drive as an overflow "virtual" memory. Hard drives are much slower than memory, so you'll find things getting very slow if this happens.

If you intend to make heavy use of graphics work in your curriculum, you may want to add even more, perhaps doubling the previous estimate.

3.1.2.1.2 Processors

How fast a processor you need is entirely dependant on what programs you plan to use. Interactive games require a bit more than say, a word processor. If you plan to use Java and Flash plugins in your web browser, these can consume a lot of processing power. For a "mixed" model, i.e. some people playing TuxMath, a few people browsing the web, and a few people typing in OpenOffice.org, a 2GHz or better processor should be able to adequately handle 20 people with some minor delays. A 3GHz processor would be better.

For larger networks, moving to an SMP (Symetric Multi Processing), or multiple CPU server may be advantageous. If you plan to handle 30 or more clients, a newer dual-core Xenon server or dual-core Opteron will provide good results.

Remember, if you need to serve a large number of clients, it will be worth your while to configure multiple Edubuntu servers, each handling some of the terminals.

3.1.2.1.3 Disks

It's advisable to use some form of RAID in the terminal servers. Besides saving your data when a single disks fails, it improves the performance (especially read performance, which is the most common type of file access). For people on a budget, setting up software RAID 1, with 2 ATA or SATA disks with NCQ (Native Command Queueing) will provide good results. If you have a bit more money, you might consider using a good SCSI hardware raid controller, with SCSI disks. This will provide you with top notch performance and reliability.

3.1.3 Network

If you have more then 20 users, it is recommended to use gigabit ethernet connected to a gigabit port on a switch for your LTSP servers. Although normal usage ranges from 0.5 to 2mbit, clients can peak quite high (70mbit), especially when watching multimedia content.

3.2 Theory of operation

Booting a thin client involves several steps. Understanding what is happening along the way will make it much easier to solve problems, should they arise.

There are four basic services required to boot an LTSP thin client. They are:

- DHCP
- TFTP
- NFS or NBD
- SSH

3.2.1 The LTSP chroot environment

In order to turn a computer into a thin client, we need to run a mini version of GNU/Linux on the workstation. It needs to boot this mini version of GNU/Linux over the network, since it probably won't have a hard drive on its own. This mini GNU/Linux installation needs to live somewhere, and the best place for it is on the server.

This scaled-down GNU/Linux installation, customized so that it's efficient to boot over the network, is called a **chroot environment**. You can have several of them, based upon several different CPU architectures. They'll normally live under `/opt/ltsp` on the server, with subdirectories for each of the architectures. For instance, if you have a lab full of old Power PC Macs, and older PC's, you'll have an `/opt/ltsp/ppc` and an `/opt/ltsp/i386` directory on the server.

The reason why it is called a chroot environment is that to install it, the GNU/Linux command **chroot** is called to actually set the installation root to `/opt/ltsp/arch`. From there, a scaled-down version of the distribution is installed. What this means is that for you to manage the chroot, performing such things as updates, all you need to do is use the **chroot** command to change the root of your installation. Then you can use all your tools like you normally would.

3.2.2 The boot process of a thin client

1. Load the Linux kernel into the memory of the thin client. This can be done several different ways, including:
 - (a) Bootrom (Etherboot,PXE)
 - (b) Floppy
 - (c) Harddisk
 - (d) CD-ROM
 - (e) USB Memory Device

Each of the above booting methods will be explained later in this chapter.

2. Once the kernel has been loaded into memory, it will begin executing.
3. The kernel will initialize the entire system and all of the peripherals that it recognizes.
4. This is where the fun really begins. During the kernel loading process, an `initramfs` image will also be loaded into memory.
5. Normally, when the kernel is finished booting, it will launch the new task launcher **upstart**, which will handle starting up a server or workstation. But, in this case, we've instructed the kernel to load a small shell script instead. This shell script is called **/init**, and lives in the root of the `initramfs`.
6. The **/init** script begins by mounting `/proc` and `/sys`, starts **udev** to discover and initialize hardware, especially the network card, which is needed for every aspect of the boot from here on. As well, it creates a small ram disk, where any local storage that is needed (to configure the `xorg.conf` file, for instance) can be written to.
7. The **loopback** network interface is configured. This is the networking interface that has `127.0.0.1` as its IP address.

8. A small DHCP client called **ipconfig** will then be run, to make another query from the DHCP server. This separate user-space query gets information supplied in the `dhcpd.conf` file, like the `nfs` root server, default gateway, and other important parameters.
 9. When **ipconfig** gets a reply from the server, the information it receives is used to configure the ethernet interface, and determine the server to mount the root from.
 10. Up to this point, the root filesystem has been a ram disk. Now, the **/init** script will mount a new root filesystem via either NBD or NFS. In the case of NBD, the image that is normally loaded is `/opt/ltsp/images/i386.img`. If the root is mounted via NFS, then the directory that is exported from the server is typically `/opt/ltsp/i386`. It can't just mount the new filesystem as `/`. It must first mount it to a separate directory. Then, it will do a **run-init**, which will swap the current root filesystem for a new filesystem. When it completes, the filesystem will be mounted on `/`. At this point, any directories that need to be writable for regular startup to occur, like `/tmp`, or `/var`, are mounted at this time.
 11. Once the mounting of the new root filesystem is complete, we are done with the `/init` shell script and we need to invoke the real **/sbin/init** program.
 12. The **init** program will read the `/etc/event.d` directory and begin setting up the thin client environment. From there, `upstart` will begin reading the startup commands in `/etc/rcS.d`.
 13. It will execute the **S32ltsp-client-setup** command which will configure many aspects of the thin client environment, such as checking if local devices need starting, loading any specified modules, etc.
 14. Next, the **init** program will begin to execute commands in the `/etc/rc2.d` directory
 15. One of the items in the `/etc/rc2.d` directory is the **S20ltsp-client-core** command that will be run while the thin client is booting.
 16. The `lts.conf` file will be parsed, and all of the parameters in that file that pertain to this thin client will be set as environment variables for the **S20ltsp-client-core** script to use.
 17. If Sound is configured at this point, the **pulseaudio** daemon is started, to allow remote audio connections from the server to connect and play on the thin client.
 18. If the thin client has local device support enabled, the **ltsfsd** program is started to allow the server to read from devices such as memory sticks or CD-Roms attached to the thin client.
 19. At this point, any of the screen sessions you've defined in your `lts.conf` will be executed.

Screen sessions are what you want to launch on all of the virtual screens on your terminal. These are the standard virtual screens that all GNU/Linux distros have, i.e. alt-F1, through alt-F10.

By default, a standard character based `getty` will be run on screen 1 (`SCREEN_01` in the `lts.conf` file).

As well, if nothing else is specified in the `lts.conf` file, an **ldm** screen script is run on `SCREEN_07`. The LTSP Display Manager (**ldm**) is the default login manager for LTSP.
 20. If `SCREEN_07` is set to a value of **ldm**, or **startx**, then the X Windows System will be launched, giving you a graphical user interface.

By default, the Xorg server will auto-probe the card, create a default `/etc/X11/xorg.conf` file on the ramdisk in the terminal, and start up `xorg` with that custom config.
 21. The X server will either start an encrypted **ssh** tunnel to the server, in the case of **ldm**, or an XDMCP query to the LTSP server, in the case of **startx**. Either way, a login box will appear on the terminal.
 22. At this point, the user can log in. They'll get a session on the server.

This confuses a lot of people at first. They are sitting at a thin client, but they are running a session on the server. All commands they run will be run on the server, but the output will be displayed on the thin client.
-

3.2.3 Network booting the thin client

Getting the thin client to boot over the network can be accomplished in a variety of ways:

- Boot ROM
- Local media

3.2.3.1 Boot ROM

Depending on your network card, it may already contain a boot ROM, or you may be able to use an EPROM programmer to create your own. Check the hardware documentation for the network card in your thin client for details.

3.2.3.1.1 Etherboot

Etherboot is a very popular open-source bootrom project. It contains drivers for many common network cards, and works very well with LTSP.

ROM images suitable for booting from floppy, CD-ROM, etc., can be obtained from <http://www.rom-o-matic.org>

Linux kernels must be tagged with the **mknbi-linux**, which will prepare the kernel for network booting, by prefixing the kernel with some additional code, and appending the `initrd` to the end of the kernel.

The kernels that are supplied with Edubuntu/LTSP are already tagged, and ready to boot with Etherboot.

3.2.3.1.2 PXE

Part of the 'Wired for Management' specification from the late 1990's included a specification for a bootrom technology known as the *Pre-boot Execution Environment* commonly abbreviated as **PXE**.

A PXE bootrom can load at most a 32 kilo-byte file. A Linux kernel is quite a bit larger than that. Therefore, we setup PXE to load a 2nd stage boot loader called **pxelinux**, which is small enough to be loaded. It knows how to load much larger files, such as a Linux kernel.

3.2.3.2 Local media

If your network card in the thin client doesn't have a boot ROM built in, and you don't have access to an EPROM burner, have no fear! Chances are, that old machine has a floppy drive, or CD-ROM in it. If so, then you can use local media to boot the thin client.

3.2.3.2.1 Floppy disk

Booting Etherboot from a floppy is an excellent way of booting an LTSP thin client that doesn't have a boot ROM. Etherboot is loaded in the boot sector of the floppy. Then, it will act just like a bootrom. The boot code will be executed, the network card will be initialized, and the kernel will be loaded from the network server.

3.2.3.2.2 Hard disk

The hard disk can be used with LILO or GRUB, to load the Linux kernel and `initrd`. You can also load the Etherboot bootrom image from the hard disk, and it will act like a bootrom.

3.2.3.2.3 CD-ROM

A bootable CD-ROM can be loaded either with a Linux kernel, or an Etherboot image.

3.2.3.2.4 USB Memory device

Just like a CD-ROM, Floppy disk and Hard disk, you can use a USB Memory device to boot an Etherboot module.

3.3 Customizing thin client behaviour

By default, most thin clients will automatically configure themselves correctly, and just work when they're plugged in. However, sometimes you may wish to customize their behavior. You would do this by editing the `lts.conf` file.

3.3.1 Format of the lts.conf file

When LTSP was designed, one of the issues that needed to be dealt with was varying hardware configurations for the thin client. Certainly, whatever combination of processor, network card and video card available today would not be available in 3 months, when you want to add more thin clients to the network.

So, LTSP.org devised a way of specifying the configuration of each thin client. The configuration file is called `lts.conf` and it lives in the `/opt/ltsp/i386/etc` directory.

The format of the `lts.conf` allows for 'default' settings and individual thin client settings. If all of your thin clients are identical, you could specify all of the configuration settings in the '[Default]' section.

3.3.1.1 Section headings

Section headings begin with an identifier in square brackets. the identifier can be one of:

- a mac address for a workstation, in the form of `XX:XX:XX:XX:XX:XX`, where `X` is the digits 0-9, or A-F. You can usually read the mac address for a network card from a sticker on the card itself, or use some kind of network tool to discover it. The `ifconfig` can tell you the mac address of your network cards.
- an IP address. You'll need to statically assign host IP addresses for this to work, as by default, Edubuntu ships with a `dhcpcd.conf` that hands out dynamic addresses. This means there's no guarantee which host will get what IP address.
- a hostname. Same issue as an IP address, but additionally, you must have either defined the hostname in DNS, or in `/etc/hosts`.
- The special section heading [Default]. This section can set defaults that apply to all terminals.

3.3.1.2 Variable Assignments

After the section heading, you can then define variables. Variables are either boolean values, requiring a True/False or Y/N answer. Note that you can either use True or False, Yes or No, or Y or N. Whichever you prefer. Other variables may simply be strings, supplied after the `=` sign. The general format of an assignment looks like:

```
BOOLEAN_VARIABLE = True
STRING_VARIABLE = Information
```

Comments can be inserted into the file for your documentation purposes. Comments start with a `#` character, and everything after the `#` for the rest of the line is considered a comment.

3.3.1.3 Location of the lts.conf filename

In order to speed up LTSP, by default, we're using NBD (Network Block Devices) rather than NFS. The `/opt/ltsp/i386` still exists, but now, it's compressed into a squashfs image, so it's much smaller than simply exporting via NFS. This means that the client uses less network bandwidth than before. However, it would mean that every time you change the `lts.conf` file, you'd have to re-create this image using the command `ltsp-update-image`. This takes a while to do. So, in order to avoid this, we've moved the `lts.conf` file to the TFTP directory, in `/var/lib/tftpboot/ltsp/i386`. This means you can make changes to the file immediately, and simply reboot the terminal, without recompiling the image.

3.3.1.4 About using NBD instead of NFS

Using NBD instead of NFS has several advantages:

- Using a squashfs image we can now merge that together in a unionfs to get writeable access which is a lot faster during bootup.
- A squashed root filesystem uses less network bandwidth.
- Many users and administrators have asked us to eliminate NFS, for reasons of site policy. Since the squashed image is now served out by **nbd-server**, which is an entirely userspace program, and is started as the user nobody, this should help to eliminate concerns over NFS shares.

However, some people still want to use NFS. Fortunately, it's easy to switch back to NFS, if it's so desired:

- On the server, use the **chroot** command to maintain the LTSP chroot:

```
sudo chroot /opt/ltsp/i386
```

- Now edit `/etc/default/ltsp-client-setup` and change the value of the `root_write_method` variable to use bind mounts instead of unionfs, it should look like this afterwards:

```
root_write_method="bind_mounts"
```

- Next, create the file `/etc/initramfs-tools/conf.d/ltsp` and add the following line (set the value of the `BOOT` variable to `nfs`):

```
BOOT=nfs
```

- Regenerate the `initramfs`:

```
update-initramfs -u
```

- Hit **CTRL-D** to exit the `chroot` now. Make sure **LTSP** uses the new `initramfs` to boot:

```
sudo ltsp-update-kernels
```

3.3.1.5 Sample `lts.conf` file

Here is an example of the `lts.conf` file:

```
#####
# Global defaults for all clients
# if you refer to the local server, just use the
# "server" keyword as value
# see lts_parameters.txt for valid values
#####
[default]
    X_COLOR_DEPTH=16
    LOCALDEV=True
    SOUND=True
    NBD_SWAP=True
    SYSLOG_HOST=server
    XKBLAYOUT=de

#####
#[MAC ADDRESS]: Per thin client settings
#####
[00:11:25:84:CE:BA]
    XSERVER = vesa
```

```
X_MOUSE_DEVICE=/dev/ttyS0
X_MOUSE_PROTOCOL=intellimouse

#####
# A Thin Client Print server
# (switch off X by pointing tty7 to shell,
# to save resources)
#####
[00:11:25:93:CF:00]
    PRINTER_0_DEVICE=/dev/usbLp0
    SCREEN_07=shell

#####
# A workstation that executes a specific
# command after login
#####
[00:11:25:93:CF:02]
    LDM_REMOTECMD=/usr/bin/myloginscript
```

3.3.2 The LDM display manager

The LTSP Display Manager, or **ldm** is the display manager specifically written by the LTSP project to handle logins to a GNU/Linux server. It is the default display manager for LTSP thin clients running under Edubuntu, and has a lot of useful features:

- It is written in C, for speed and efficiency on low end clients.
- It supports logging in via either a greeter (a graphical login application) or autologin.
- It can be configured to encrypt X Windows traffic, for increased security, or leave it unencrypted, for better performance on slower clients.
- It contains a simple load-balancing system, to allow the system administrator to allow load balancing across several servers.

We'll go over the `lts.conf` entries you'll need to control these features below.

3.3.2.1 Theory of operation

To help understand the following sections, a bit of an explanation of how **ldm** does it's work is needed. Most thin client display managers tend to run up on the server. The **ldm** display manager is unique in that it runs on the thin client itself. This allows the thin client to have a lot of choice as to how it will set up the connection. A typical login session goes as follows:

- **ldm** launches and starts up the X Windows display on the thin client.
- **ldm** starts up the greeter, which is a graphical program which presents the user with a nice login display and allows them to select their session, language, and host they'd like to log into.
- **ldm** collects the information from the greeter, and starts an ssh session with the server. This ssh connection is used to create an ssh master socket, which is used by all subsequent operations.
- Now, the users selected session is started via the master socket. Depending on whether or not an encrypted connection has been requested, via the `LDM_DIRECTX` parameter, the session is either connected back to the local display via the ssh tunnel, or via a regular TCP/IP connection.
- During the session, any memory sticks, or other local devices that are plugged in, communicate their status to the server via the ssh control socket.
- When the user exits the session, the ssh connection is closed down, the X server is stopped, and **ldm** restarts itself, so everything starts with a clean slate.

3.3.2.2 Encrypted versus unencrypted sessions

By default, LTSP5 encrypts the X session between the server. This makes your session more secure, but at the cost of increased processing power required on the thin client and on the server. If processing power is a concern to you, it's very easy to specify that the connection for either an individual workstation, or the default setting should use an unencrypted connection. To do so, simply specify:

```
LDM_DIRECTX=True
```

in your `lts.conf` file in the appropriate stanza.

3.3.2.3 Auto login features

This new version of LDM supports auto login of accounts, if specified in the `lts.conf` file. Simply create a config stanza for each of the terminals you want to log in automatically (you can use either MAC address, IP address, or hostname) and specify the variable **LDM_USERNAME** and **LDM_PASSWORD**. Note that you must have created these accounts on the server, with the passwords specified. An example will serve to illustrate how to use this:

```
[00:E0:81:27:D6:AE]
  LDM_USERNAME=station1
  LDM_PASSWORD=sekrit1

[00:30:48:73:FC:A3]
  LDM_USERNAME=station2
  LDM_PASSWORD=sekrit2
```

3.3.2.4 Load balancing features

In this version of LTSP, there's a simple load-balancing solution implemented that allows administrators to have multiple Edubuntu servers on the network, and allow the thin client to pick which one of the servers it would like to log into.

The host selection system is simple and flexible enough to allow administrators to implement their own policy on how they want the load balancing to happen: either on a random, load-based, or round robin system. See Section 3.4 for details.

3.3.3 General thin client parameters

There are several variables that one can define in the `lts.conf` file which control how the thin client interacts with the server. These are:

SERVER This is the server that is used for the `XDM_SERVER`, `TELNET_HOST`, `XFS_SERVER` and `SYSLOG_HOST`, if any of those are not specified explicitly. If you have one machine that is acting as the server for everything, then you can just specify the address here and omit the other server parameters. If this value is not set, it will be auto detected.

SYSLOG_HOST If you want to send logging messages to a machine other than the default server, then you can specify the machine here. If this parameter is NOT specified, then it will use the 'SERVER' parameter described above.

NBD_SWAP Set this to **Y** if you want to turn on NBD swap. The default is **Y**.

SWAP_SERVER The NBD swap server can exist on any server on the network that is capable of handling it. You can specify the IP address of that server. The default is whatever the value of `SERVER` set to.

NBD_PORT The port on which NBD swapping will occur. This is set to 9572 by default.

USE_LOCAL_SWAP If you have a hard drive installed in the thin client, with a valid swap partition on it, this parameter will allow the thin client to swap to the local hard drive. The default is **N**.

DNS_SERVER Used to build the `resolv.conf` file. Not needed by default.

SEARCH_DOMAIN Used to build the resolv.conf file.

SOUND This parameter enables sound for the thin client. The default is **Y**.

LOCALDEV This parameter enables local devices support, like CD's and USB sticks. Users plugging them in should see them on the desktop, after they've been added to the fuse group on the server. You can do this by going to: System → Administration → Users and Groups selecting the user, clicking on "Properties", the going into the "User Privileges" tab, and making sure the "Allow use of FUSE filesystems..." box is checked. The default is: **Y**.

3.3.4 Screen Scripts

Screen scripts are how LTSP determines what type of login will run on what virtual screen. Most GNU/Linux machines have 12 virtual consoles, which you can access by pressing Control-Alt-F1, through Control-Alt-F12. There is a text based getty that is started on screen 1, but you normally can't log into it, as there are no local users on the thin client.

However, for debugging purposes, you may want to set up root to log in on the thin client. You may need to do this if you're debugging problems with local devices, for example. Fortunately, it's easy to do: on the server, just chroot into the LTSP chroot, and set the password with passwd.

```
sudo chroot /opt/ltsp/i386
passwd
```

By default, if there's nothing else mentioned in `lts.conf`, an LDM session will be started on screen 7.

PARAMETERS RELATING TO SCREEN SCRIPTS

SCREEN_01 thru SCREEN_12 Up to 12 screen scripts can be specified for a thin client. This will give you up to 12 sessions on the thin client, each accessible by pressing the Ctrl-Alt-F1 through Ctrl-Alt-F12 keys.

```
SCREEN_07 = ldm
SCREEN_02 = shell
```

Currently, possible values include:

- **ldm**: This is the default display manager. It collects a username and password, and then establishes a secure, encrypted tunnel to the server via **ssh**. This should be good for most environments. Edubuntu deployments with lower-powered clients or servers may find that the extra overhead involved in encrypting the X traffic might slow their sessions, and may wish to enable the **LDM_DIRECTX** parameter described in Section 3.3.2
- **sdm**: Similar in functionality to ldm, but a little less graphically intensive.
- **startx**: Older X connection requiring the use of XDMCP to connect to the server. Some legacy installations may want to use it, however, the introduction of the **LDM_DIRECTX** parameter has eliminated much of the need to run it. Enabling this will require you to turn on XDMCP for the **gdm** login manager. As an administrative user, go to System → Administration → Login Window and in the "Remote" tab, change the drop down to "Same as local". Additionally, you may wish to click on the "Configure XDMCP" button on the lower corner, and increase the "Maximum remote sessions" to something a little higher than the number of thin clients you have. **Please note that doing this means local devices will not work for you, as they rely on the ssh tunnel that ldm provides.**
- **telnet**: Text screen telnet into whatever host TELNET_HOST is set to. See below for an explanation.
- **shell**: spawns a shell on the thin client. Useful for testing.
- **rdesktop**: spawns an rdesktop session to a remote windows server. Note that you must have the **rdesktop** package installed in the chroot.

Look in the `/opt/ltsp/i386/usr/lib/ltsp/screen.d` directory for more scripts, or write your own, and put them there.

TELNET_HOST If the thin client is setup to have a character based interface, then the value of this parameter will be used as the host to telnet into. If this value is NOT set, then it will use the value of **SERVER** above.

3.3.5 Modules and startup scripts

For the most part, LTSP does a very good job of detecting what hardware's on your thin client. However, it's possible that you may want to manually specify a kernel module to load after boot. Alternatively, you may have a script you've written that you've put in the chroot, and want to make sure gets run at startup. LTSP provides some hooks to allow you to do this.

MODULE_01 thru MODULE_10 Up to 10 kernel modules can be loaded by using these configuration entries. The entire command line that you would use when running `insmod` can be specified here. For example:

```
MODULE_01 = uart401.o
MODULE_02 = "sb.o io=0x220 irq=5 dma=1"
MODULE_03 = opl3.o
```

If the value of this parameter is an absolute pathname, then **insmod** will be used to load the module. Otherwise, **modprobe** will be used.

In normal circumstances, you shouldn't need to specify anything here, as most hardware will be auto-detected.

RCFILE_01 thru RCFILE_10 Additional RC scripts can be executed by the **ltsp-client-setup** script. Just put the script in the `/opt/ltsp/i386/etc/init.d` directory, and specify the name of the script in one of these entries.

3.3.6 X-Windows parameters

Setting up X windows on the thin client's normally a pretty easy operation. The thin client uses X.org's own auto configuration mode to let X determine what it thinks is installed in the box. The thin client just runs the command **Xorg -configure**, and then uses that output, slightly modified, for the X config file.

However, sometimes, this doesn't always work. Either due to strange/buggy hardware, or buggy drivers in X.org, or because X detects default settings that you don't want. For instance, it may detect that your monitor is capable of doing 1280x1024, but you'd prefer it to come up in 1024x768 resolution. Fortunately, you can tweak individual X settings, or, alternatively, simply provide your own `xorg.conf` to use.

X.ORG CONFIGURATION

X_CONF If you want to create your own complete X.org config file, you can do so and place it in the `/opt/ltsp/i386/etc/X11` directory. Then, whatever you decide to call it needs to be entered as a value for this configuration variable. For example:

```
X_CONF = /etc/X11/my-custom-xorg.conf
```

Note that for the thin client, you reference it from `/etc/X11`.

X_RAMPERC Some programs allocate a large amount of ram in the X.org server running on your thin client. Programs like **Firefox** and **Evince** can use up so much ram, that they eventually exhaust all your physical ram, and NBD swap, causing your thin client to crash. If you find your clients being booted back to a login prompt, or freezing up when viewing certain PDF's or web pages, this may be the problem.

The `X_RAMPERC` variable stands for X RAM PERCent, and is a number between 0 and 100 that specifies how much of the free space on your thin client X.org is allowed to consume. You'll generally want to set it at something lower than 100 percent, if you're having problems. Experimentation has shown a value between 80 and 90 will usually keep the terminal alive. What will then happen is the program consuming the memory will die, as opposed to the thin client itself. If you're having unexplained terminal problems, specifying:

```
X_RAMPERC = 80
```

in your `lts.conf` file may improve things.

XDM_SERVER If you're using the older **startx** screen script, and need to specify a different server, then you can specify the server here. If this parameter is NOT specified, then it will use the 'SERVER' parameter described above.

XSERVER You can use this parameter to override which X server the thin client will run. For PCI and AGP video cards, this parameter should not be required. The thin client should normally be able to auto-detect the card.

If, for some reason you do need to manually set it, here are the valid values:

- ark
 - ati
 - atimisc
 - chips
 - cirrus_alpine
 - cirrus
 - cirrus_laguna
 - cyrix
 - dummy
 - fbdev
 - fglrx
 - glint
 - i128
 - i740
 - i810
 - imstt
 - mga
 - neomagic
 - newport
 - nsc
 - nv
 - r128
 - radeon
 - rendition
 - riva128
 - s3
 - s3virge
 - savage
 - siliconmotion
 - sis
 - sisusb
 - tdfx
 - tga
 - trident
 - tseng
 - v4l
 - vesa
 - vga
 - via
 - vmware
-

- `voodoo`

X_MOUSE_DEVICE This is the device node that the mouse is connected to. If it is a serial mouse, this would be a serial port, such as `/dev/ttyS0` or `/dev/ttyS1`. This is not needed for PS/2 or USB mice, as they are auto-detected.

X_MOUSE_PROTOCOL Should be auto-detected. However, valid entries include:

- `sunkbd`
- `lkkbd`
- `vsxxaa`
- `spaceorb`
- `spaceball`
- `magellan`
- `warrior`
- `stinger`
- `mousesystems`
- `sunmouse`
- `microsoft`
- `mshack`
- `mouseman`
- `intellimouse`
- `mmwheel`
- `iforce`
- `h3600ts`
- `stowawaykbd`
- `ps2serkbd`
- `twiddler`
- `twiddlerjoy`

X_MOUSE_EMULATE3BTN Normally unset, may need to be set to **Y** for certain 2 button mice.

X_COLOR_DEPTH This is the number of bits to use for the color depth. Possible values are **8**, **16**, **24** and **32**. 8 bits will give 256 colors, 16 will give 65536 colors, 24 will give 16 million colors and 32 bits will give 4.2 billion colors! Not all X servers support all of these values. The default value for this is **24**.

USE_XFS You have a choice of running the X Font Server (XFS) or reading the fonts through the file system. XFS has been pretty much superceded by the **RENDER** extension of X.org, but for special cases, you can specify it. The 2 values for this option are **Y** and **N**. The default value is **N**. If you do want to use a font server, then you can use the **XFS_SERVER** entry to specify which host will act as the font server.

XFS_SERVER If you are using an X Font Server to serve fonts, then you can use this entry to specify the IP address of the host that is acting as the font server. If this is not specified, it will use the default server, which is specified with the **SERVER** entry described above.

X_HORIZSYNC This sets the X.org **HorizSync** configuration parameter. This should be auto-detected for your monitor, however, if you want to force a lower resolution, use this parameter to do so.

X_VERTREFRESH This sets the X.org **VertRefresh** configuration parameter. This should be auto-detected for your monitor. If you need to force a lower resolution, use this parameter to do so.

X_VIDEORAM This sets the X.org **VideoRam** configuration parameter. This should be auto-detected for your monitor. If you need to force a different video ram setting, use this parameter to do so.

X_OPTION_01 through X_OPTION_12 This allows you to specify **Option** settings in the `xorg.conf` file, to add options to the video driver. A common use for this will be to test turning off acceleration in your driver, if you're having trouble. An example usage would be:

```
X_OPTION_01 = "\"NoAccel\""  
X_OPTION_02 = "\"AnotherOption\" \"True\""
```

. You probably won't need these except in special circumstances.

X_MODE_0, X_MODE_1, and X_MODE_2 These sets the X.org **ModeLine** configuration. For example, if your thin client comes up in a higher resolution than what you want, say, 1280x1024, specifying:

```
X_MODE_0 = 1024x786
```

should get your desired resolution on startup.

3.3.7 Printer configuration parameters

Sometimes, it's convenient to hang a printer off of a thin client in a lab, so that the computer lab has access to local printing resources. Fortunately, LTSP can accomodate printing on the workstation.

LTSP can connect up to 3 printers per workstation to the network via a small daemon called JetPipe. Both parallel and USB printers are supported. JetPipe makes the printer look like a standard HP Jet Direct printer interface. You can then create any cups printer on your server, and point it at the printer via a Jet Direct connection.

In your `dhcpd.conf` file that controls your thin client IP assignments, you'll want to assign a static IP for the terminal with the printers, to guarentee that it gets the same IP address every time it boots. Otherwise, your printing won't work if the terminal leases a different IP address.

PRINTING RELATED PARAMETERS

PRINTER_0_DEVICE The device name of the printer. Names such as `/dev/lp0`, or `/dev/usb/lp0` are allowed.

PRINTER_0_PORT The TCP/IP Port number to use. By default, it will use '9100', for `PRINTER_0_DEVICE`, '9101' for `PRINTER_1_DEVICE`, and '9102' for `PRINTER_2_DEVICE`.

3.3.8 Keyboard parameters

All of the keyboard support files are copied into the `/opt/ltsp/i386` hierarchy, so configuring international keyboard support is simply a matter of configuring X.org. There are several configuration parameters for this.

The values for the above parameters are from the X.org documentation. Whatever is valid for X.org is valid for these parameters.

We would like to add documentation to show what values are needed for each type of international keyboard. If you work with this and can configure your international keyboards, feedback to Edubuntu would be greatly appreciated.

CONSOLE_KEYMAP Allows you to specify a valid console keymap for `TELNET_HOST` sessions. Default is `en`.

XKBLAYOUT Consult the X.org documentation for valid settings.

XKBMODEL Consult the X.org documentation for valid settings.

XKBVARIANT Consult the X.org documentation for valid settings.

XKBRULES Consult the X.org documentation for valid settings.

XKBOPTIONS Consult the X.org documentation for valid settings.

3.4 Multiple server setup

A multiple server setup is useful for larger thin client networks. Instead of using one big server, it makes it possible to use smaller servers, and dispatch users on them. You can adjust computing resources as the demand grows simply by adding a new server. To make sure that every server behaves the same from the users point of view, new services and configurations that are required will be discussed. In addition, some configurations specific to thin clients will be presented.

3.4.1 Infrastructure setup

3.4.1.1 Network topology

The network topology is the same as a standalone server setup, except that there are more than one server on the thin client lan.

You will need to select one server to behave as the primary server. This server will be used to run additional services, hold users files, and network boot thin clients.

Secondary servers will be used only to run desktop sessions. They are simpler, and will be configured to use the central services from the primary server. Installing the package **ltsp-server** will install required software.

3.4.1.2 Common authentication

A user should be able to start a session with the same login and password, no matter which server it connects to. For this purpose, a central authentication mechanism must be used. There are many possibilities offered. Here are the major technologies:

- LDAP authentication: On the master server, setup an OpenLDAP server. Configure each servers to use this LDAP server as the authentication base, with the `pam_ldap` plugin.
- NIS authentication: On the master server, setup a NIS server. Configure each server to use this NIS server for the authentication.
- Winbind authentication: Useful if you already have an Active Directory server.
- The `smbldap-installer` offers system administrators an easy way to set up a Samba/LDAP server which can authenticate both Windows and non-Windows (Linux, Mac OS X, and others) clients. Home directories can be centrally located on one server and shared with other servers. For more information: <http://www.majen.net/smbldap>

For detailed instructions, see their respective manuals.

3.4.1.3 Shared home directories

Shared home directories are easy to setup using an NFS server on the primary server. To configure the share, install the package **nfs-kernel-server**. Then, edit the file `/etc/exports` and add the following line, by replacing **secondary_server_ip** by the real IP address of the server.

```
/home secondary_server_ip(rw,no_root_squash)
```

After this modification, the service needs to be restarted.

```
sudo invoke-rc.d nfs-kernel-server restart
```

You can define the mount point on each secondary servers, by editing the file `/etc/fstab` and adding the following line:

```
192.168.0.1:/home /home nfs hard,intr,rsize=8192,wsiz=8192,bg 0 0
```

With the default setup, the home directory of the user must exist before the first login. To create the home directory on the fly on the first login, you can use the `pam_mkhomedir` plugin. The NFS export option **no_root_squash** allows any secondary server to create directories on the primary server.

3.4.1.4 Shared printers

For printers to be accessible on each server, the cups server must be configured to share printers. Refer to the CUPS manual for detailed instructions.

3.4.1.5 Synchronization of packages

Once you install one desktop application on one server, then you must install it on all the other servers, as well. Otherwise, users may not be able to use the same set of applications. First, make sure that `/etc/apt/sources.list` file is the same on each server. Then, list packages of the reference server. Using the primary server for this purpose is not recommended, since it may install other server packages that are not necessary on secondary servers. To build the package list:

```
dpkg --get-selections > deblist
```

Then, copy this file on the target server you want to sync applications, and perform the following steps:

```
sudo dpkg --set-selections < deblist
sudo apt-get dselect-upgrade
```

Apt will install additional packages that are not already installed on the target machine.

3.4.1.6 Managing the SSH known hosts file

For security reasons, a thin client won't connect to an untrusted server. You must add the keys of secondary servers inside the client root on the primary server. To do this, first export the key file of the secondary server to add:

```
sudo ltsp-update-sshkeys --export ssh_known_hosts.myhostname
```

Then, copy the file on the primary server, in the directory `/etc/ltsp/`, and run `ltsp-update-sshkeys` on the primary server. Then, thin clients will trust the freshly added server, and will be able to connect to it.

If a secondary server changes its IP address, then this procedure must be repeated.

3.4.1.7 Setting network forwarding

Primary server will act as a network gateway for other servers. With this configuration, other workstations will be able to access the network behind the primary server. Here is an example of script that setup the network forwarding. We put it in `/etc/network/if-up.d/forward.sh`, and make it executable. The script will run at each network start. In this example, the primary server private IP is **192.168.0.1**. It must be adapted for the IP address used.

```
#!/bin/bash

echo 1 > /proc/sys/net/ipv4/ip_forward

echo Setting up the forwarding

LAN_IP_NET="192.168.0.1/24"
LAN_NIC="eth1"
OUT_NIC="eth0"

iptables -t nat -A POSTROUTING -s $LAN_IP_NET -o $OUT_NIC -j MASQUERADE
iptables -A FORWARD -j ACCEPT -i $LAN_NIC -s $LAN_IP_NET
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

3.4.1.8 Replication of desktop profiles

If you customize user's desktop, then custom desktop profiles should be copied to every server. Gnome desktop profiles created with Sabayon are located in `/etc/desktop-profiles`

3.4.2 Session dispatching

3.4.2.1 Define the server list

LDM is a login manager for thin clients. Users can select a server from the available ones in the host selection dialog box.

The displayed server list is defined by the **LDM_SERVER** parameter. This parameter accepts a list of server IP address or host names, separated by space. If you use host names, then your DNS resolution must work on the thin client. If defined in the `lts.conf` file, the list order will be static, and the first server in the list will be selected by default.

You can also compute a new order for the server list, by creating the script `/opt/ltsp/i386/usr/lib/ltsp/get_hosts`. The parameter **LDM_SERVER** overrides the script. In consequence, this parameter must not be defined if the `get_hosts` is going to be used. The `get_hosts` script writes on the standard output each server IP address or host names, in the chosen order.

3.4.2.2 Dispatching method

You can change this behavior by using a script to rearrange the list. The simplest way to do it is by randomizing the list. First, define a custom variable in the file `lts.conf`, for example **MY_SERVER_LIST**, that will contain the list of servers, the same way as **LDM_SERVER**. Then, put the following script in `/opt/ltsp/i386/usr/lib/ltsp/get_hosts`.

```
#!/bin/bash
# Randomize the server list contained in MY_SERVER_LIST parameter

TMP_LIST=""
SHUFFLED_LIST=""

for i in $MY_SERVER_LIST; do
rank=$RANDOM
let "rank %= 100"
TMP_LIST="$TMP_LIST\n${rank}_${i}"
done

TMP_LIST=$(echo -e $TMP_LIST | sort)
for i in $TMP_LIST; do
SHUFFLED_LIST="$SHUFFLED_LIST $(echo $i | cut -d_ -f2)"
done

echo $SHUFFLED_LIST
```

More advanced load balancing algorithms can be written. For example, load balancing can be done by querying `ldminfd` for the server rating. By querying `ldminfd`, you can get the current rating state of the server. This rating goes from 0 to 100, higher is better. Here is an example of such a query:

```
nc localhost 9571 | grep rating | cut -d: -f2
```

3.5 Network Swap

Helping thin clients with smaller amounts of ram Just like on a full fledged workstation, it helps to have swap defined for your thin client. "Swap" is an area of disk space set aside to allow you to transfer information out of ram, and temporarily store it on a hard drive until it's needed again. It makes the workstation look like it has more memory than it actually does. For instance, if your workstation has 64 Megabytes of ram and you configure 64 Megabytes of swap, it's theoretically possible to load a 128 Megabyte program.

I say, "theoretically", because in practice, you want to avoid swapping as much as possible. A hard drive is several orders of magnitude slower than ram, and, of course, on a thin client, you don't even have a hard drive! You have to first push the data through the network to the server's hard drive, thus making your swapping even slower. In practice, it's best to make sure you have enough ram in your thin client to handle all your average memory needs.

However, sometimes that's not possible. Sometimes, you're re-using old hardware, or you've simply got a program that isn't normally used, but does consume a lot of ram on the thin client when it does. Fortunately, LTSP supports swapping over the network via NBD, or Network Block Devices. We include a small shell script called `nbdswpd`, which is started via `inetd`. It handles creating the swap file, and setting up the swapping, and removing the swap file when it's no longer needed, after the terminal shuts down.

By default, swap files are 32 Megabytes in size. This was chosen to give your workstation a little extra ram, but not use up too much disk space. If you get some random odd behaviour, such as Firefox crashing when viewing web pages with a lot of large pictures, you may want to try increasing the size of the swap files. You can do so by creating a file in the directory `/etc/ltsp` on the Edubuntu server, called `nbdswpd.conf`. In it, you can set the `SIZE` variable to the number of Megabytes you wish the file to be sized to. For instance, to create 128 Megabyte files, you'll want:

```
SIZE=128
```

in the `nbdswpd.conf` file.

Please note that this is a global setting for all swap files. If your server has 40 thin clients, each using 128 Megs of memory, you'll need $128 * 40 = 5120$, or a little over 5 Gigabytes of space in your `/tmp` directory, where the swapfiles are stored.

3.6 Managing DHCP

DHCP stands for Dynamic Host Configuration Protocol and is the very first thing your thin client uses to obtain an IP address from the network, in order to allow it to start booting. In Edubuntu, the `dhcpd` file is located in `/etc/ltsp`. Any changes you want to make to booting behaviour should be made there.

By default, Edubuntu ships a `dhcpd.conf` that serves thin clients in a dynamic range (i.e. it will hand out ip addresses to anyone who asks for them) from 192.168.0.20 to 192.168.0.250. The default `dhcpd.conf` file looks like:

```
#
# Default LTSP dhcpd.conf config file.
#

authoritative;

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.20 192.168.0.250;
    option domain-name "example.com";
    option domain-name-servers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    option routers 192.168.0.1;
    # next-server 192.168.0.254;
    # get-lease-hostnames true;
    option subnet-mask 255.255.255.0;
    option root-path "/opt/ltsp/i386";
    if substring( option vendor-class-identifier, 0, 9 ) = "PXELinux" {
        filename "/ltsp/i386/pxelinux.0";
    } else {
        filename "/ltsp/i386/nbi.img";
    }
}
```

This `dhcpd.conf` should handle most situations.

By default, Edubuntu will detect an unused network interface and configure it to be 192.168.0.254. Edubuntu's recommended single server installation is to use a separate network interface for the thin clients. If, however, you're not using two network interfaces, or you already have an interface in the 192.168.0 range, then you might have to configure the thin client interface differently, which means you may have to adjust the `dhcpd.conf` accordingly.

If the network interface that you're going to connect the thin clients to has, say, a TCP/IP address of 10.0.20.254, you'll want to replace every occurrence of 192.168.0 with 10.0.20 in the `dhcpd.conf` file.

Always remember, you'll need to re-start the dhcp server if you make any changes. You can do this by issuing the command:

```
sudo invoke-rc.d dhcp3-server restart
```

at the command prompt.

3.6.1 Adding static entries to the dhcpd.conf

Sometimes, you may need to have a certain terminal boot with a guaranteed fixed TCP/IP address every time. Say, if you're connecting a printer to the terminal, and need to make sure the print server can find it at a fixed address. To create a fixed address, use a low number in the range of 2-19, or otherwise, if you change the range statement in the `dhcpd.conf`.

To create a static entry, simply add the following after the "option root-path" line:

```
host hostname {
    hardware ethernet    MA:CA:DD:RE:SS:00;
    fixed-address        192.168.0.2;
}
```

Substitute the mac address for the mac address of the thin client you wish to fix the address of. The fixed-address will be the TCP/IP address you want, and "hostname" is the name you wish to give the host.

3.6.2 DHCP failover loadbalancing

Another common method of loadbalancing is to use DHCP loadbalancing. There's an excellent writeup on the topic at: <https://wiki.edubuntu.org/EdubuntuDHCPload-balancingFailover>

3.7 Lockdown with Sabayon and Pessulus

A common requirement in both schools and businesses is having the ability to lock down the desktop and provide certain default configurations.

In Edubuntu, the applications you'll want to use are Sabayon and Pessulus. You'll want to add them from the **synaptic** package manager. After you've added them both, then you can start them from System → Administration → User Profile Editor

The Sabayon profile editor looks like a window that contains a smaller sized picture of your desktop. Within this window, you can create a default layout: add icons to panels and the desktop, lock down the panels so they can't be modified, remove access to the command line, etc.

Once you're done, you can save your profile. You have the option of applying your profile to either individual users, or all users on the system. Please consult the manual included with Sabayon for all the details.

3.8 Edubuntu Thin Client Manager

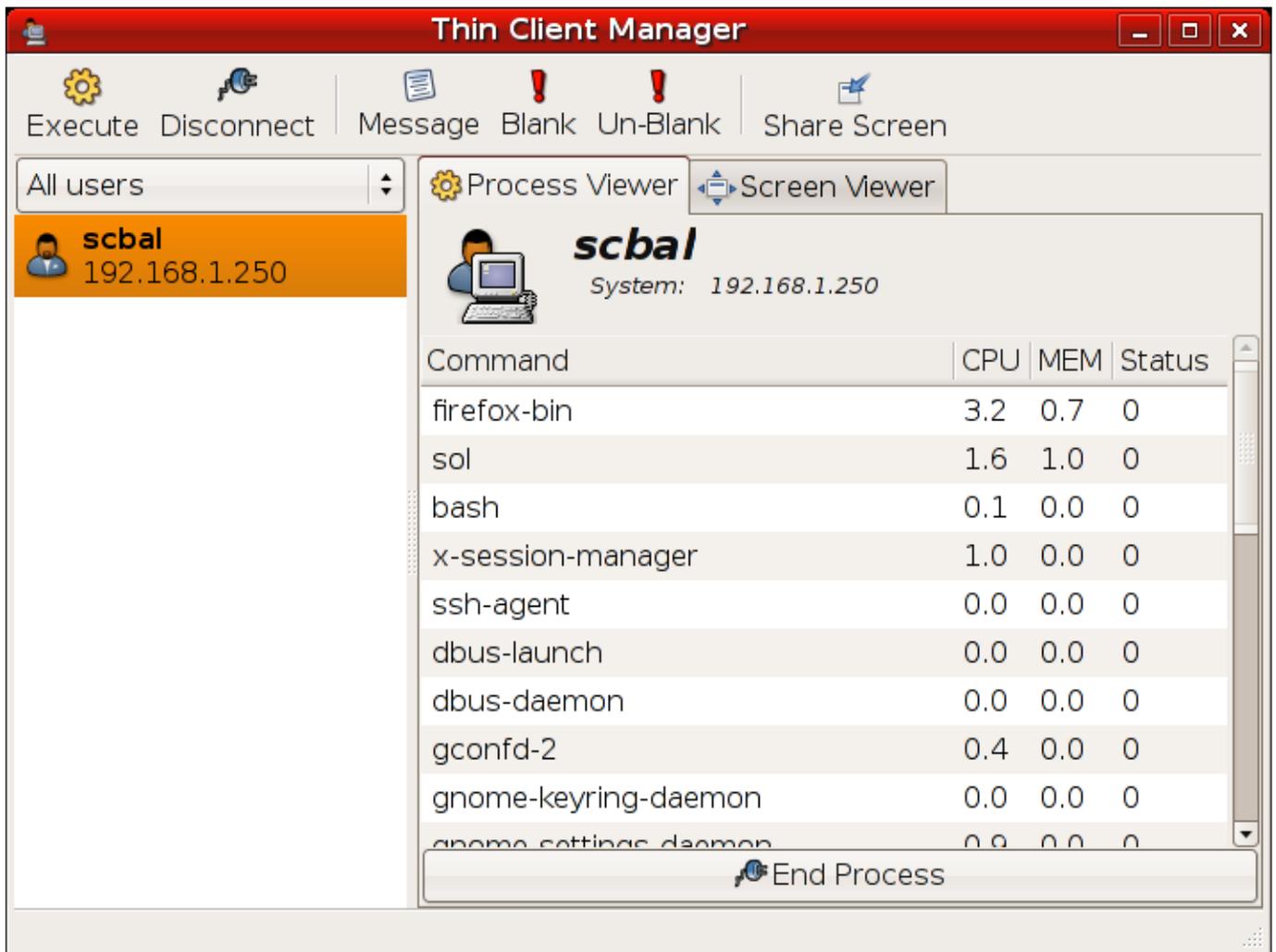
3.8.1 Using Thin Client Manager

Managing an LTSP setup effectively Thin Client Manager (TCM) is a system administration utility to enable a teacher to easily control the machines in their class, by utilising the following features:

- Remote program execution
- Simple message sending
- Remote application closure
- Quick access to the lockdown editor

- Remote logout execution
- Process listings for each user
- Plugin framework to expand functionality

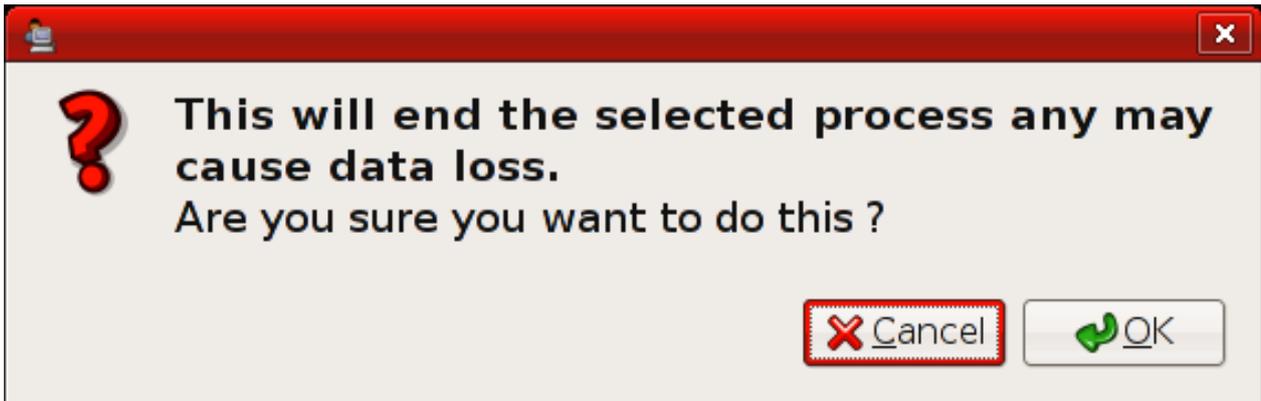
When first loaded, TCM shows the clients currently logged onto the server on the right hand side, along with their IP address and username, as in Fig 1. This list is dynamically updated and will change as users log into and out of the system. Clicking on an entry in this list will highlight it, and enable certain functions in the user interface. Clicking on a user will also fill the window on the right with all the processes that are running in that users session.



3.8.2 Operations

3.8.2.1 Process Management

You can end a users applications by first choosing a user, selecting a process from the right hand side, and then clicking on the "End Process" button. You will then be asked to confirm your actions and, upon confirmation, a message is sent to the clients session asking for the program to terminate. (Note: You can select multiple users here)



3.8.2.2 Logging Users Out

If desired, you can log a user out of their session, by first selecting a user and click on the "Disconnect" button. You will then be asked to confirm your actions and, upon confirmation, a message is sent to end the users session. This will log the user out of their current session. (Note: You can select multiple users here)

3.8.2.3 Sending A Message

You are able to send short messages to users, for example, "You have 5 minutes left of this lesson". To do this, first select a user and then click on the send message button. You will then be presented with a box to type in your message. Upon clicking "Ok", the message will be sent to the selected user. (Note: You can select multiple users here)



3.8.2.4 Starting A Program

It is also possible to start an application or process in a clients session from TCM. To do this simply select the user, and click on the "Execute" button. You will then be presented with a dialog box to enter a command to be run in the users session. Upon clicking "Ok", a message will be sent to the users session asking for the chosen command to be run. (Note: You can select multiple users here)



3.8.2.5 Lockdown Editor

By choosing a single user and right clicking on that users name, you will open up the context menu. From here you can choose "Lockdown", which will allow you to set options to restrict a particular user. Clicking this menu item will invoke the "Pessulus" program, which is the Gnome lockdown editor. Ticking and unticking options in Pessulus will enable and disable certain functions for that particular user. There is a padlock next to each option in Pessulus. Ticking this will make the option unchangeable by the user. This is called a mandatory setting. Pessulus has been altered for integration with Thin Client Manager so that mandatory keys are now per user, instead of per system. For further help with Pessulus, please refer to the Pessulus documentation.

3.8.2.6 Managing users with groups

The latest version of Thin Client Manager comes with a user group and filtering system. Right clicking in the user list, will present you with a menu which has an option called Groups. From this menu you can create new groups, delete old groups and assign users to groups.

To assign users to a group simply select the users required from the user list and then right click to enter context menu. Now you can move through the menu, Groups → Add user to group → Group Name The process for removing members from a group is exceedingly similar.

Once you have assigned your users into groups, you can use the filter combo box above the user list to show only members of that particular group. Groups and members are persistent across Thin Client Manager sessions and are automatically saved once you have altered them. If you ever need to manually access the file which stores this information, it is located at `/etc/tcm/users.conf`.

3.8.2.7 Plugins

The plugin framework allows you to expand the way TCM works. By selecting one or more users in the left hand panel you can right click in the left hand area and be presented with the context menu. This menu has an option called "Plugins" and from that there is a list of all the plugins installed in TCM. On a fresh installation, this will consist of a single plugin, which is used purely as an example. For the more advanced user, please see the example plugin file located at `/usr/lib/python2.4/site-packages/studentcontrolpanel/plugins/cheap_plugin`. A plugin consists simply of a class and a registration function. The plugin is provided with a list of users, which you can use to write a code to perform functions based on that list.

3.8.2.8 Screen Viewing

This functionality requires some set up by the system administrator, please see the Ubuntu wiki for more information.

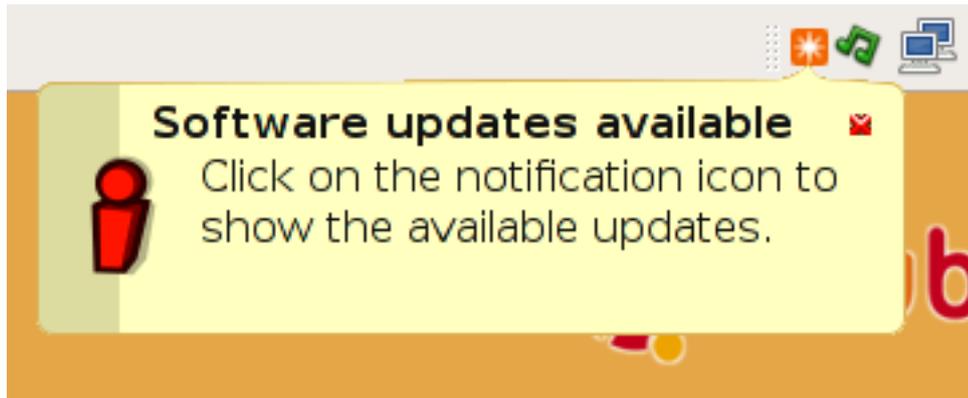
By clicking on the Screen Viewer tab, you can view four client screens at a time. By using the buttons at the bottom of the screen you can move between the currently logged in users. If a user becomes disconnected, you can use the refresh option to reconnect all screens. Connections are closed when switching to the Process Viewer to save bandwidth, but are reconnected again upon switching to the Screen Viewer tab.

3.9 Keeping your Edubuntu server in shape

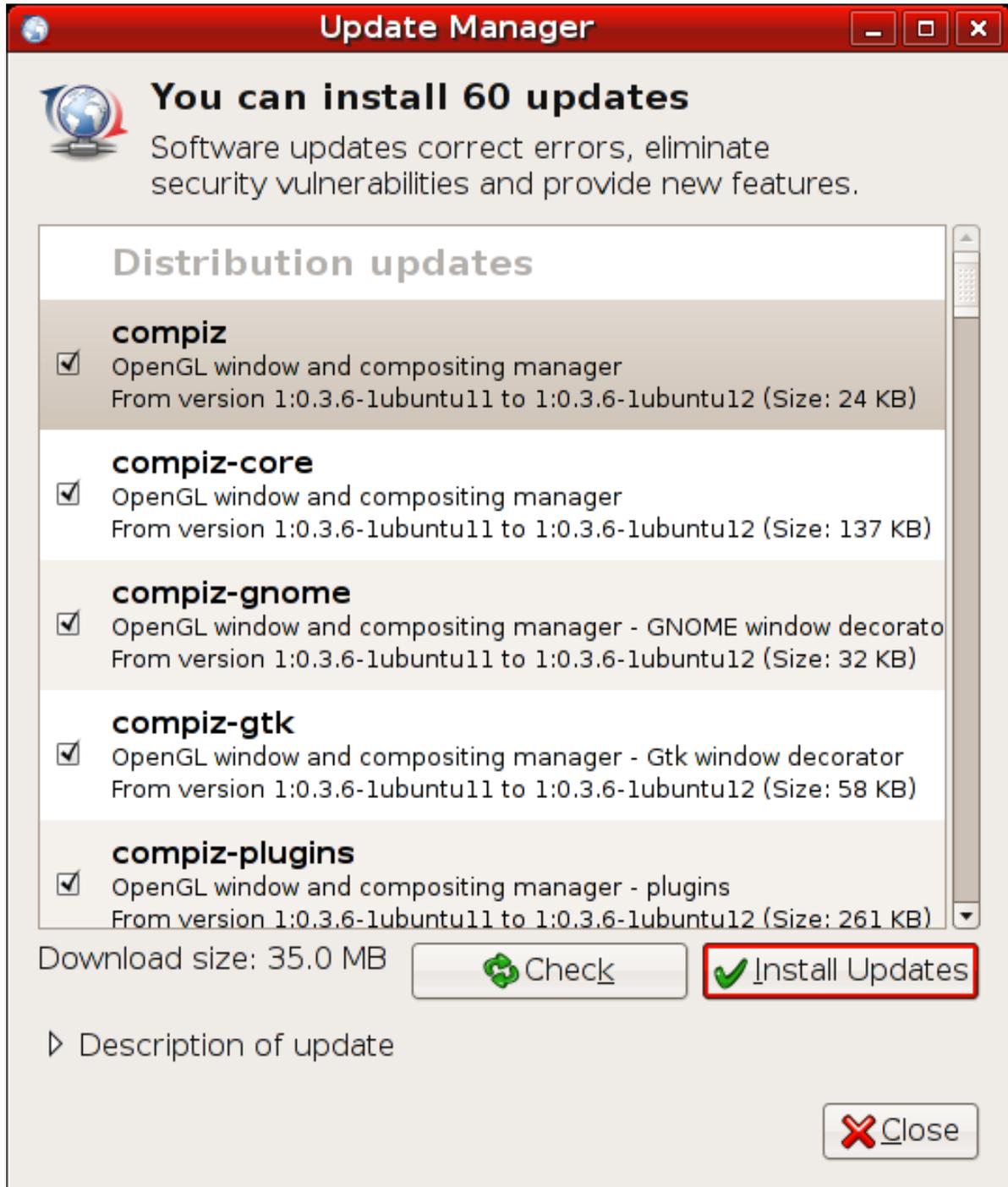
Security, as it's often been said, is a process, and not an end result. Fortunately, Ubuntu makes it easy to keep up to date with the latest security patches.

3.9.1 Edubuntu Server Management

When logging onto the LTSP network as an administrator you will sometimes find a bubble appear in the top right informing you that there are updates available for your computer.

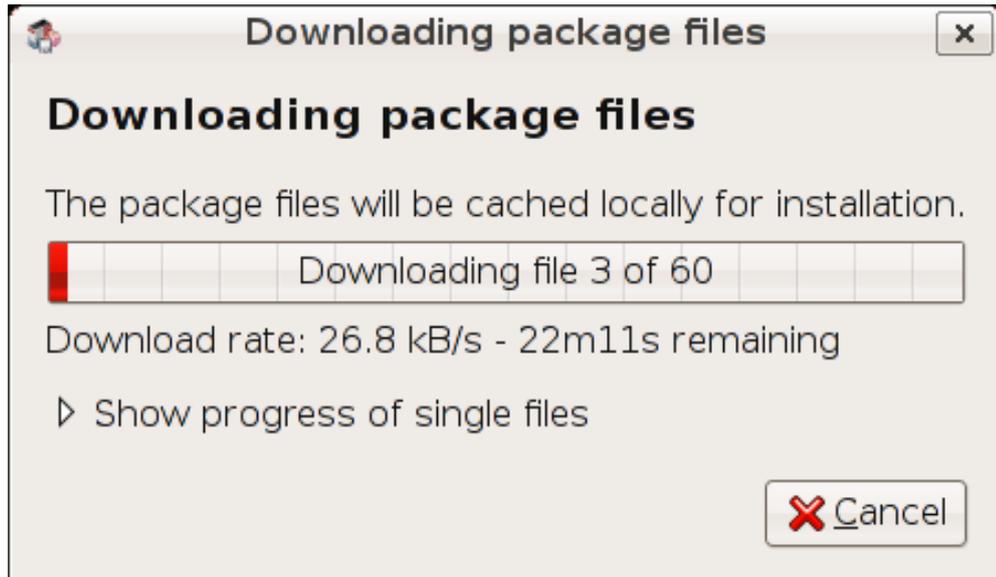


Any updates you apply here will automatically be applied to all clients, though they may require a reboot. This is because LTSP clients all use the same set of applications on the server. When this prompt appears click the small red icon, and enter your password to display the following application.

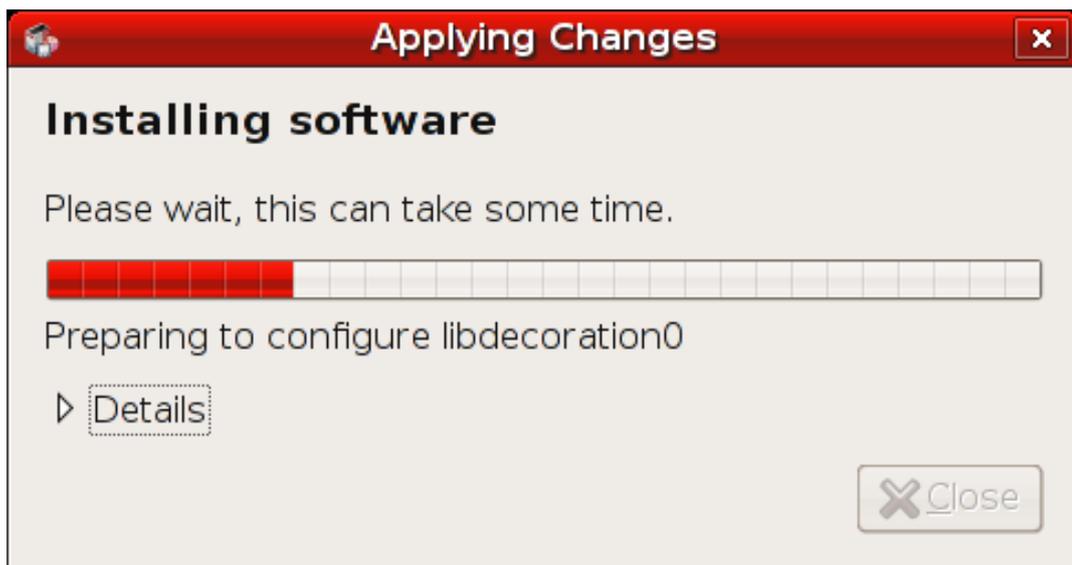


This application allows you to apply all available updates for your computer quickly and easily. Clicking on the **Check** button, checks to see if there are any more updates since the computer last checked. If you are going to perform the updates, it's a good idea to click this button. Once the check is completed, you are ready to update your system.

Be sure to check how much information is going to be downloaded, if you are low on bandwidth, or using a slow connection. Click on the **Update** button, to set the computer downloading and installing updates. Once clicked, if there are any verification issues, these will be shown. If you are happy that the packages come from a trusted source, you can click **OK** to continue. First, your computer will download all available packages and display a screen similar to that below.



After all packages have finished downloading, you will be presented with the progress box, which shows how far the computer has come in updating. If you wish you can click the word **terminal** to show more detailed information about the updates as shown below.



After the updates have completed, the computer will inform you if you need to restart in order for the updates to come into effect. You can either shutdown when you wish, or click the small blue icon in the top righthand corner, and you will be presented with this question.

**Important**

If you restart your computer without shutting down your clients, their computers will stop responding and they may lose data. Be sure all clients are logged out before restarting the server.

3.9.2 Updating your LTSP chroot

At some point in the future, updates will become available for your LTSP server. You must remember that although you may have applied all the updates to the server itself, as in the instructions....HERE it is likely that the LTSP chroot will also need updating. To do this you must open up a terminal and use the following commands.

First make sure the Client environment has the same Package lists as the Server, to achieve that, you will copy the sources.list file from the Server to the Client environment.

```
sudo cp /etc/apt/sources.list /opt/ltsp/i386/etc/apt/
```

Now issue the command below.

```
sudo chroot /opt/ltsp/i386
```

This will change your root directory to be the LTSP clients root directory. In essence, anything you now do inside here, will be applied to the LTSP clients NFS root. This is a separate small set of files that are used to boot the clients into a usable, and enable them to contact the LTSP server. Once inside this shell, we must type the following command to obtain the latest list of packages from the apt servers.

```
mount -t proc proc /proc  
apt-get update
```

You need to mount /proc in the chroot before beginning, as some of the packages you install may need resources in /proc to install correctly.

Once this has completed you will have to upgrade the software in the chroot by running the following command:

```
apt-get upgrade
```

After the upgrade is completed, don't forget to unmount /proc with:

```
umount /proc
```

Once you're done, you must leave the chroot by either typing **exit** or by using the key combination Ctrl+D. This will return you to the root of the server.

If your kernel has been upgraded you must run the LTSP kernel upgrade script, to ensure that your LTSP chroot uses the latest version. This is performed by running the command below:

```
sudo ltsp-update-kernels
```

All of your clients will now use the latest kernel upon their next reboot.

Finally, you must remember to rebuild the NBD boot image from your chroot with the following command:

```
sudo ltsp-update-image
```

Be advised that this may take a few minutes, depending on the speed of your server.

3.9.3 Changing the IP of your LTSP server

At some point in time, it may become necessary to change the IP address of your LTSP server. Normally this does not present an issue, but LTSP servers and clients communicate over an encrypted channel and require all SSL certificates to be updated. Without this update, **no LTSP clients will be able to log in**. This is done by simply opening a terminal and running the following command.

```
sudo ltsp-update-sshkeys
```

3.10 Backing Up

Now that you have your Edubuntu server working, and are keeping it updated, you're going to want to back up your users' data.

There are a variety of options available for external backup media. SCSI tape drives, DVD RAM or ROM's, or even CD/R's may all be used to back up your server on a regular basis.

3.10.1 What needs to be backed up?

When backing up a server, there are typically two approaches:

- Back up everything.
- Back up config files, and user data.

We'll look at each briefly.

3.10.1.1 Back up everything

Typically, by backing up everything, the administrator is looking to do a restore of the entire operating system, as well as user data.

Typically, you'll want to use a SCSI tape drive, and a program like **amanda** (available in the universe repository) to do this kind of backup strategy.

3.10.1.2 Back up configs and data

Backing up only config files and data is an acceptable strategy for sites on a budget. On a typical Edubuntu system, without a lot of customization, backing up the following files and directories should capture most of what's on your system. Following a disaster, you'll need to re-install Edubuntu from the CD, re-install any programs you added, and re-apply all updates. At that point, you can pull the config files and user data from your backup.

```
/etc  
/home  
/var/lib/tftpboot/ltsp/i386/lts.conf
```

Chapter 4

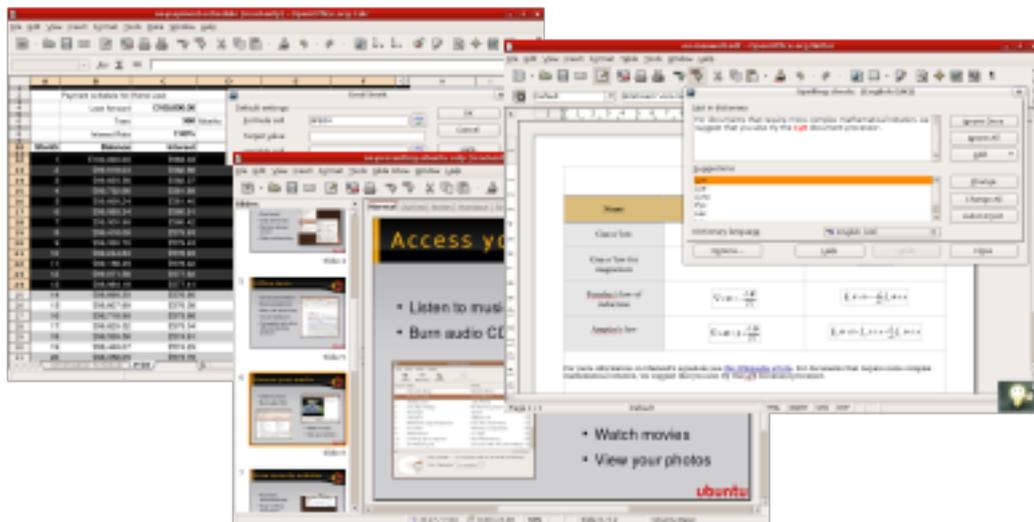
Use, Configure, Support, and Contribute

4.1 Introduction to available software

Edubuntu builds to be a complete and ready to use educational environment. As such, Edubuntu comes pre-installed with a complete office suite, teaching and learning programs, pre-school resources and much more across a range of categories. Here is a brief summary and some screenshots for the most popular Edubuntu packages, and if you need more, Edubuntu has thousands of additional applications ready for you to install.

4.1.1 Office Suite

OpenOffice.org is a complete office suite similar to other, commercial office suites. It features a word processor, as well as spreadsheet, presentation, drawing, database, and mathematical formula applications. OpenOffice.org can open and convert most documents from other commercial office suites and has support for many different languages.



4.1.2 Internet

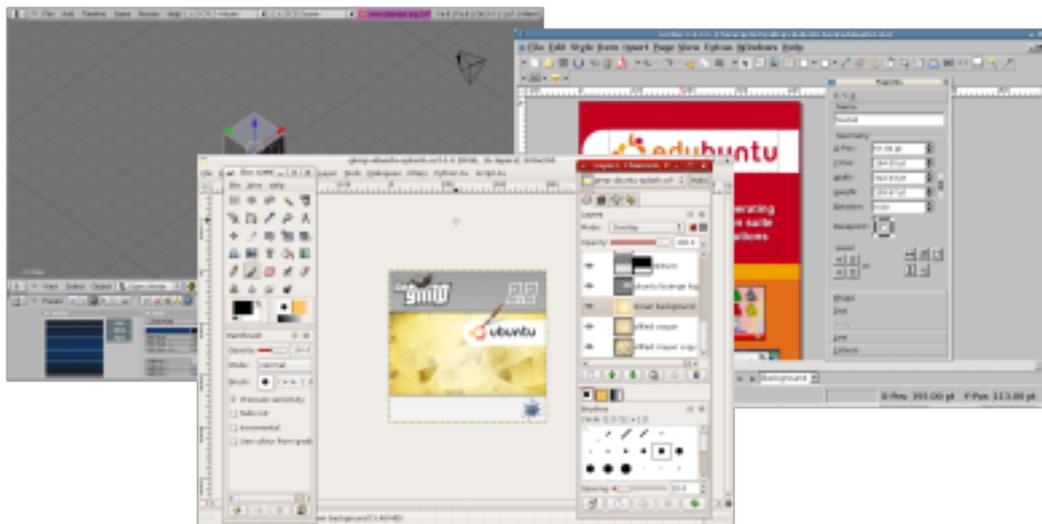
Firefox is the incredible successor to the hugely popular Mozilla browser. Amongst its many features are tabbed browsing, which makes viewing multiple sites a breeze, pop-up blocking, live bookmarks and great accessibility support. Combine these with its fantastic customization engine and you have the most powerful web browsing at your fingertips.

Moving beyond the simple web browser, Edubuntu also comes with an email client and personal information manager, Evolution. Video conferencing with Ekiga as well as instant messaging and chat with Gaim provide communications tools that teachers and students need in an Internet age.



4.1.3 Graphics

In the graphics department, diagrams and flowcharts are a breeze with Dia. Xsane provides an easy to use scanner interface. 3D modeling and animation are Blender's speciality. Image editing similar to other professional photo editors is done using the GIMP. Scribus provides layout and publishing capabilities.



4.1.4 Sound and Video

For all your video and audio play and editing needs, Edubuntu provides several programs including Rhythmbox for music playing and Serpentine for burning audio CDs. Sound Juicer rips audio CD tracks to your computer. Totem is an excellent video and movie player and you can use Kino to edit video clips. Sound Recorder can be used to record audio clips to your computer.



4.1.5 Science

4.1.5.1 Explore the stars

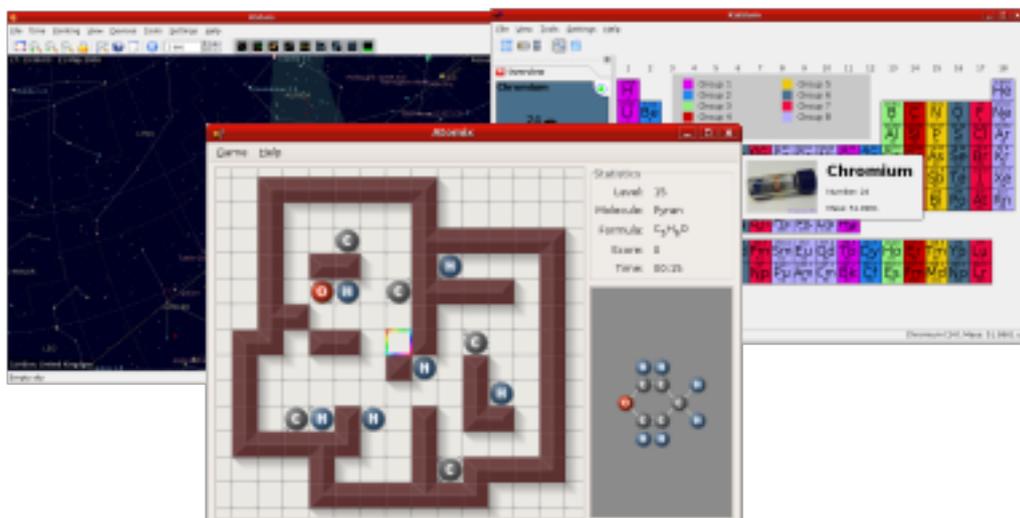
KStars is a Desktop Planetarium that provides an accurate graphical simulation of the night sky. The program includes details for 130000 stars, 13000 deep-sky objects and all of the planets. KStars has an easy to use interface, so it can be used by everyone, from amateurs through to astronomy experts. You will also find a great deal of general information regarding telescopes and other astronomy related matter.

4.1.5.2 Research the periodic table

Kalziium is a package for discovering and researching information about the periodic table and the elements. It includes pictures for most of the 111 elements present, along with more detailed information including atomic models, spectrum analysis, chemical data and energies.

4.1.5.3 Test your chemistry skills

Atomix is a puzzle game, the purpose of which is to make up the molecule displayed from the pieces scattered throughout the level. You can only move them up, down, left and right and they don't stop moving until they hit something.



4.1.6 Mathematics

4.1.6.1 Program in Logo

KTurtle is a Logo programming language interpreter. The Logo programming language is very easy to learn and thus it can be used by young children. A unique quality of Logo is that the commands or instructions can be translated, so the 'programmer' can program in his or her native language. This makes Logo ideal for teaching kids the basics of programming, mathematics and geometry. One of the reasons many children warm to Logo is due to fact that the programmable icon is a small turtle, which can be moved around the screen with simple commands and can be programmed to draw objects.

4.1.6.2 Create geometric constructions

Kig is an application that allows teachers and students to create high precision geometrical constructions. These can be built on and used to explain concepts like perpendicular bisectors, tangents and arcs.

4.1.6.3 Plot mathematical equations

KmPlot is a mathematical function graphing and plotting package. KmPlot has a built in powerful expression parser and you can plot different functions simultaneously and combine their terms to build more complex mathematical functions. It supports functions with parameters and functions in polar coordinates. Plots may also be printed with high precision.

4.1.6.4 Calculate percentages

KPercentage is a mathematical application that helps pupils improve their skills in calculating percentages. Percentages are split into three types and KPercentage tests pupils on these in three training modes. In addition to this, there is a random mode which will pick questions from each of the three percentage question variations.

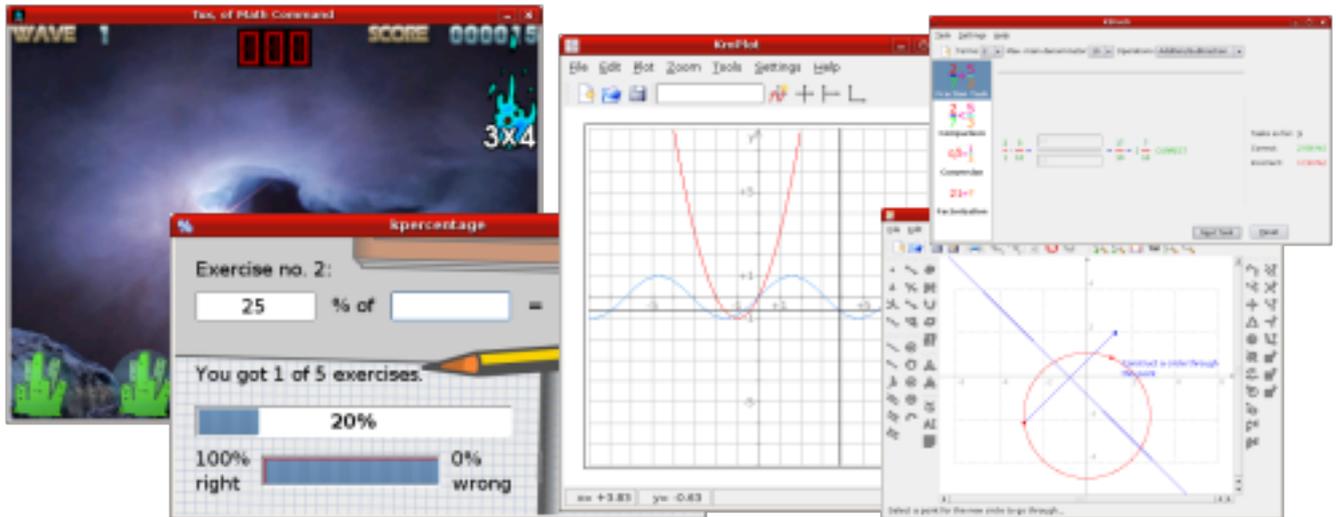
4.1.6.5 Practice Fractions

KBruch is a small program to help pupils practice calculations involving fractions. There are four different types of fraction exercise:

- Exercise Fraction Task; where pupils have to solve a given fraction task.
- Exercise Comparison; where pupils have to compare 2 given fractions sizes.
- Exercise Conversion; where pupils have to convert an already given number into a fraction.
- Exercise Factorization; where pupils have to factorize a given number into its prime factor.

4.1.6.6 Play at Math

TuxMath is an educational arcade game starring Tux, the Linux mascot! Based on the classic arcade game "Missile Command," Tux must defend his cities by solving arithmetic problems.



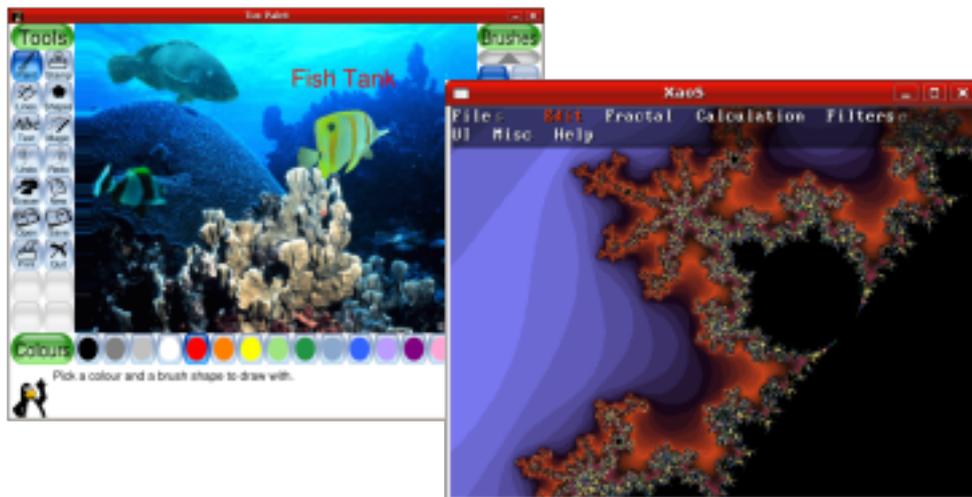
4.1.7 Drawing

4.1.7.1 Tux Paint

Tux Paint is a free drawing program designed for young children. The program provides an easy to use interface with fun sound effects and an encouraging cartoon mascot that gives help to children while they are painting.

4.1.7.2 Generate Fractals

Xaos is a fractal generator with a whole host of extra features for teaching pupils about fractal patterns.



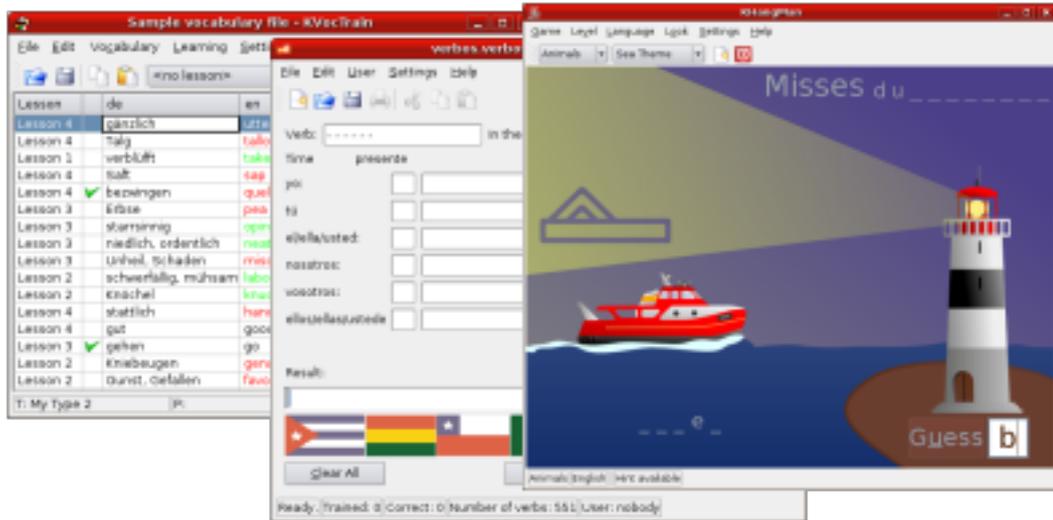
4.1.8 Language

4.1.8.1 Practice Spanish verbs

With KVerbos you can practice Spanish verb conjugation. The program comes with a large set of over 9,000 Spanish verbs that you can choose from to begin your verb training. The package also keeps track of previous scores so that pupils can identify where their strengths and weaknesses are. Some exercises also have a time limit to raise the level of difficulty for more advanced users.

4.1.8.2 Play Hangman

KHangman is the classic hangman game with pupils try to guess a word, letter by letter. At each miss, the picture of a hangman is added to. After 10 incorrect tries the game is over.



4.1.9 Basic Skills

4.1.9.1 Generate and give tests

KEduca is an educational testing package, allowing teachers to create tests for pupils to take. KEduca includes a module for constructing and saving new tests, as well as a separate module for loading and running the exams. Questions can be enhanced with images, multiple choice answers with varying grades, and time limits.

4.1.9.2 Learn touch typing

KTouch helps pupils to learn to type quickly and correctly. With many different modes and detailed reporting, KTouch truly makes learning touching typing easily.

4.1.9.3 Gcompris

Gcompris is a suite of over 80 educational games and activities for kids age 4 to 10 to learn with. These include:

- Computer discovery: keyboard, mouse.
- Algebra: table memory, enumeration, double entry table, mirror image.
- Science: the canal lock, the water cycle, the submarine, electric simulation.
- Geography: place the country on the map.
- Games: chess, memory, connect 4, oware, sudoku.
- Reading: reading practice.
- Other: learn to tell time, puzzle of famous paintings, vector drawing.

4.2 Documentation and Support

This book only provides a basic introduction to Edubuntu. To learn more you may read the extensive documentation included with the system itself, or access the constantly expanding material that community members maintain on the Edubuntu Web site.

Documentation for Other Systems

Documentation written for Ubuntu also applies to the Edubuntu system. Edubuntu shares a common layout and software with the other operating systems in the Ubuntu family. Material that describes the Debian system may also apply to Edubuntu, although the Ubuntu family differs from Debian in some respects.

You may also use a range of facilities to communicate with fellow Edubuntu and Ubuntu users over the Internet. As a global community with a strong commitment to sharing and cooperation, Ubuntu has many volunteers who contribute their time to support others. In many cases a polite query may receive an answer in minutes.

This chapter explains the resources available to help you develop your technical skills and overcome any difficulties that you may encounter. For information on how to participate in the Edubuntu community yourself, refer to Chapter 5.

4.2.1 Supplied Documentation

Edubuntu includes several sets of documentation by default. To read these, select System → Help and choose the relevant link in the Ubuntu Help Center.

4.2.1.1 Accessing Documentation from the Command-line

You may also use command-line utilities to read the **man** and **info** documentation in a terminal such as a text-only virtual console, or a desktop terminal window.

Use the **man** command to read a **man** page in a terminal. For example, to read the page for the file `sources.list`, type:

```
man sources.list
```

To see the various search options available with **man**, type:

```
man man
```

To load the GNU help system in a terminal, run the **info** command:

```
info
```

To open the manual for a specific GNU utility, add the name of the program after **info**. For example, this command displays the manual for the **chown** utility:

```
info chown
```

4.2.2 Online Resources

The Ubuntu project and many other groups offer documentation on their Web sites for Debian, Ubuntu, and related Linux systems. All of the providers listed here offer documentation without charge, under licences that enable you freely copy and distribute the material.

4.2.2.1 Ubuntu Web Sites

Refer to these pages for both on-line documentation and downloadable books produced by the Ubuntu project:

The main Ubuntu documentation page [Ubuntu Documentation](#)

The Edubuntu documentation page [Edubuntu Documentation](#)

4.2.2.2 Other Documentation Web Sites

Some of the most well-known providers of free Linux documentation include:

The Debian Project [Debian Documentation](#)

The Linux Documentation Project [TLDP](#)

O'Reilly Media [Open Book](#)

4.2.3 Asking Questions Online

The large number of Linux users today means that any issue that arises is likely to have already been discussed on an Internet venue. Web search engines such as Google are particularly useful for finding relevant information in the text of the numerous on-line discussions held on forums, mailing lists, blogs, and project bug tracking systems.

If you cannot find the answer to your question in the documentation or with a search engine, choose a support venue and post your question there. The section below explains the support venues for Edubuntu.

4.2.3.1 Using Web Search Engines

Search engines provide as many matches as possible for the words or phrases that you specify. To get the most useful and relevant possible set of results from your searches, try to follow these general principles:

- Use the most specific words that you can.
- Include at least three key words in your search.
- If possible use a specific phrase, such as text from an error message.
- Consider whether the advanced or power-user options of the search engine might help you.

Use whichever search engine you prefer. Google users may find that the dedicated Linux search facility at <http://www.google.com/linux> provides better results than the main Google search page.

4.2.3.2 Community Support Facilities

The main venues for discussing Edubuntu on-line are:

The mailing list for Edubuntu users [Edubuntu-Users List](#)

The Ubuntu Forums [Ubuntu Forums](#)

The IRC channel for live discussions on using and developing Edubuntu [Community](#)

When you use these facilities, remember that the individuals that answer questions are freely donating their time to help others, and may see many requests for help. If you are new to on-line discussions, please read these pages before you post your question:

The Ubuntu Code of Conduct [Code of Conduct](#)

How to Ask Questions the Smart Way [How to file a bug](#)

4.2.4 Commercial Support and Training Providers

Refer to this page for information about obtaining commercial support for Edubuntu: [Paid Support](#)

Chapter 5

Contributing to Edubuntu

Edubuntu is maintained by people just like you who use it and want to keep the project going strong. We love our developers, but we need many other kinds of volunteers, too -- "you do not have to be a computer programmer to contribute to Edubuntu".

5.1 Are you a user?

Every day users and administrators turn to our community for support. There may be a user out there who has a problem this week you learned to solve last week. You can help Edubuntu by [joining our mailing list](#) and/or hanging out in #edubuntu on irc.freenode.net and helping users just like you. You can also contribute to our [wiki](#) at, or help us [test the latest edubuntu-in-progress](#).

5.2 Are you a teacher, parent, or student?

We are always interested in hearing about how Edubuntu is working out there in the wide world. [Join our mailing list](#) and let us know about how you are using Edubuntu, what you like, and what you'd like to see improve. Add to our [list of recommended projects](#) so we know what kind of software you want to see in the future of Edubuntu.

5.3 Are you multilingual?

Edubuntu has users around the world, and we are always looking for volunteers to help translate web pages, documentation, and more. Volunteer via our [email list](#).

5.4 Are you a writer, editor, or proofreader?

Introduce yourself [on our email list](#) or visit <http://www.edubuntu.org/help> to find out who our current documentation contact is and offer to lend a hand, or just jump in and help out with the [documentation on our wiki](#).

5.5 Are you an artist?

Contribute some [artwork](#) to Edubuntu.

5.6 Are you a coder?

Join the [mailing list](#) or visit us on IRC ([#edubuntu](#) on [irc.freenode.net](#)) and see where your skills and interests lead you. We are always in need of good coders.
