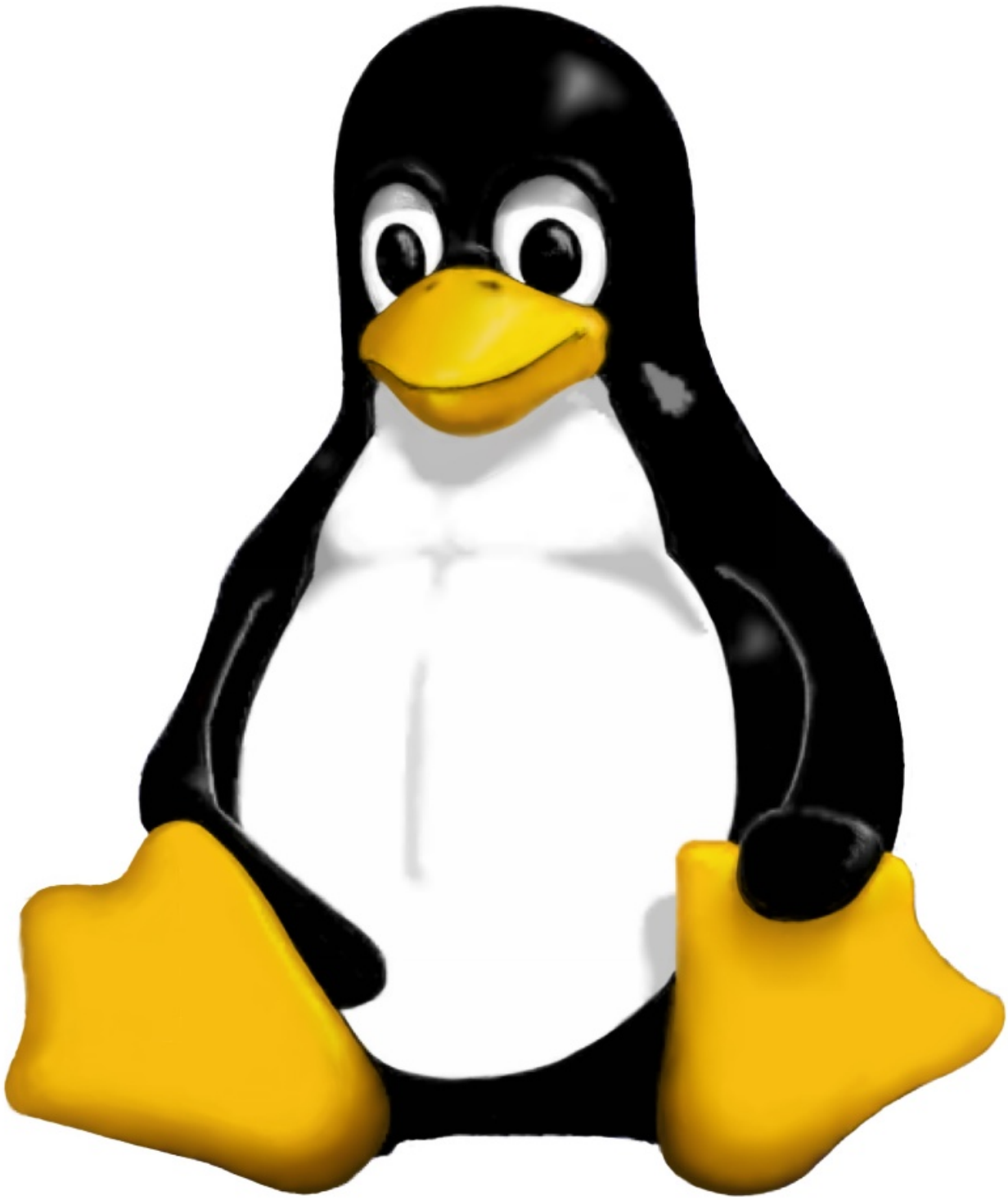
Introduction to Linux

Brandon Chisham

May 15, 2004

Contents



Chapter 1

Introduction

This book is intended as a practical guide to the Linux operating system for those with a moderate amount of technical experience. My intention is to make this book as friendly to the beginning user as possible. If you are already a Unix uber-geek, then this book will likely be boring to you although it has been my experience that many times those who have been using *nix systems for a long time are often less familiar with the available (Graphical User Interface) tools than those who began using those systems after GUI's became commonplace, full-featured, and easy to use.

The first part of the book will describe the history of Unix and Linux. This background information will help the reader to understand how these systems are related, and how they are different from other systems originally developed for personal computers This section also covers the development of the GNU (GNU's Not Unix) and the free software foundation. It then goes on to describe why one would want to switch to Linux, how to acquire Linux, and how to install Linux.

Chapter 2

What is Linux

The Linux itself is not directly interacted with by the user; it is responsible for handling requests, load balancing, and hardware support. None of these activities are visible to the end user. When people talk about Linux they are usually referring to a complete GNU/Linux system. Linux is a clone of the Unix operating system. It has nearly all of the same functionality, but its source code is freely available. Most of the common applications available for Linux are licensed under the terms of the GNU Public License. This License is technically a copyright that grants the right to freely distribute the software, but it prohibits selling the software *without making the source code to the program available*. The, "If its not source its not software!" doctrine is central to the idea of the GPL. It protects users' rights and prevents software that is licensed with it from becoming proprietary.

2.1 Who Created Linux?

2.1.1 Kernel

Linus Torvalds created the first version of Linux, and he is still the final authority on what changes make it into versions now. He created Linux while in graduate school in Finland. He had been using a terminal system called Minix, and had found some bugs. He wrote the maintainer of Minix who rebuffed him, and so he decided to write a system that would suit his needs. At first he intended to simply write a terminal program like the one that he was using with Minix, but the project took on a life of its own and became the Linux kernel. Many new users are confused about what a kernel is and why it is important. As stated the user does not ever directly interact with the kernel. Instead the kernel manages system resources, handles communication with devices, and manages information flow in the system. It provides a standard interface for programs to communicate with when making system requests. This feature allows program development to be streamlined because the program does not have to handle the details of the hardware that it is operating on. In 1991, the Linux was born. It was released under the GPL (GNU Public License) and became the force behind the modern open source movement. When the Linux system was combined with the nearly complete

GNU implementation of the Unix interface a complete Unix clone was created. This system did not contain proprietary Unix code, but provided the same functionality, and because of its similarity to the Unix system it was easy for someone who was familiar with Unix to use Linux. This also made it easy to port (adapt) applications originally designed for Unix to Linux.

2.1.2 Environment

While the Kernel provides the basic system services, the GNU system forms the evnironment that one will interact with directly when Linux. For this reason many insist that Linux is more properly referred to as **GNU**/Linux. The history of the GNU system is outlined later in this section, or more information is also available from the GNU website *http://www.gnu.org*.

2.2 Unix

Unix is an operating system that was originally developed by bell labs in the late 1960's It is widely regarded as a powerful system for a number of reasons first it can run on a wide variety of computers from an x486 to a Supercomputer. It also has been an early leader in computer networking. The client server model has been a powerful influence on its development, and has resulted in a system that can network readily and today powers much of the Internet's backbone. It is also a true multiuser multitasking operating system. With Unix multiple users can be logged into and executing programs on the system at any one time. Also a user does not have to be in the same physical location as the machine to use it.

2.2.1 Early History

Unix grew out of an MIT project called Multics (Multiplexed Information and Computing Service) in 1965 GE and Bell Labs joined this effort. Before this time operating systems were not nearly as inter-operable as they are today. Even systems from the same company needed special programs to communicate with each other, and systems from different companies had almost no chance of operating with each other. Worst of all upgrading could cause an organization to lose all of its data! If the the organization wanted to keep the data they would need to manually re-enter it. The Multics

system sought to do away with these problems, but development was slow and Bell Labs stopped work on the project in 1969. A few Bell Labs employees continued working on a system that would solve the problems that Multics tried to address. Leading the effort were Murry Hill, Ken Thompson, Dennis Ritchie, Doug McIlroy, and J.F. Ossanna.

During the summer of 1969, Unix development began in earnest. Thompson took the lead in development and reserved a day a week to work on some aspect of the new system. Though a self supporting system (it had its own assembler) was developed in 1969 it wasn't until 1970 that Unix was given and name and finished. Later that year a new PDP-11 was purchased to help continue the development into a commercially viable operating system. Development on the new system did not begin until that December however because a disk was not yet available for it. During this time the Bell Labs patent department became the first Unix Customer. Text formatting programs were developed for the existing system and used to automatically do much of the formatting required for patent applications. This use gave the new system credibility and help it to gain the support of management.

The next step was to develop a high level language that could be used to simplify further development. The original system was written in assembly language which is very close to the machines native language and is very difficult to work with. Fortran was tried and was rejected after less than a day. Instead Ken Thompson developed has own language which he called B. The original language was interpreted rather than compiled and resulted in a slow system. Dennis Ritchie continued developing it into a language that became known as NB or New B for the PDP-11. Finally some more enhancements were added and the C language was born! C remains in widespread use today, and is the foundation of other modern languages like C++ and Java.

Translating the Unix system into the C language was slow. During this period a number of important aspects of Unix were completed including the use of pipes. Pipes are used to string several small programs together to do a task. This process is described in detail in the section introducing the command line. This advance has shaped Unix and related systems greatly. It meant that a program only had to do one small thing because its output could be sent to another program that would perform some other operation on it. It also meant that text streams which were easily standardized became an important interface for these programs.

2.2.2 Spreading

In 1976-77 Ken Thompson took a temporary teaching position at UC Berkeley and brought Unix with him. The system rapidly became accepted. Berkeley was so enthused about the new operating system that it obtained a license from Bell Labs to help develop it into a system for academic use. This system became the BSD, Berkeley Standard Distribution, version of Unix. This system is the foundation of many open Operating Systems including Free-BSD, Open-BSD, and Net-BSD.

After the Berkeley version, several other organizations created their own versions of Unix including IBM (AIX), Sun (Solaris), and Hewlett-Packard (HP-UX). This fragmentation lead to a large number of versions of Unix that were not always compatible with each other. (The very problem that Unix was created to solve).

After years of development AT&T sold its interest in Unix to Novell in 1993. Later that same year Novell sold its interest in Unix to Santa Cruz Corporation. Vendors such as IBM, HP, Sun, and Compaq still produce there own versions of Unix, and it or one of its descendants still from the foundation of the Internet, and company/university mainframes.

2.3 GNU

2.3.1 Brief History

The GNU, **G**NU's **N**ot **U**nix, system had been under development long before Linux was invented. GNU was an effort to create an entirely free clone of the powerful Unix operating system. Which was described in the previous section. The effort was begun in the early 1980's by Richard Stallman. He was alarmed by the trend toward closed source systems and nondisclosure agreements. He was determined to create a system that would permit users to share and modify its components freely. By the time that Linux had been created the effort had produced a nearly complete Unix clone all that remained to create was the system's . This is the part of the operating system that actually handles user and program requests. It is also responsible for interfacing with devices, writing output to them and taking input from them. This is a very important and complex task. When Linux was released, it

became that core component. In actual fact a distribution of Linux is more properly called a GNU-Linux system. Nearly all of the software that is bundled with the Linux kernel to form a distribution was produced as part of the GNU project, and is open source too.

The free software foundation is the organizational structure behind the GNU project. The foundation was created to help to further the project's open source objectives. The foundation is responsible for the development and defense of GNU software licenses. The GPL is the most common license used for GNU software. It is also likely the most well known open source license. It's basic purpose is to protect the code from becoming part of a closed source commercial application. It declares that any modifications made to a program that is licensed under it must also remain open source. There are also other licenses offered by the free software foundation. The LGPL or **L**esser **G**NU **P**ublic **L**icense is used to permit developers to create free applications that are linked with non-free libraries. It also offers the Free Documentation License which despite its name can be used for any written work.

2.3.2 Philosophy

Many people are confused by the term . Software that is "free" can be redistributed without penalty, and more importantly its source code must be made available. This is important for a number of reasons. First if you have a program that doesn't meet your needs exactly you are free to change it yourself or hire someone to change it for you rather than waiting for the next product release. Second since the source code is available anyone reviewing it can spot and report bugs. This distributed code auditing has resulted in an extremely capable and secure operating environment when compared to other consumer level systems. It is important to note that "free" does not necessarily referr to the price to the software, but instead to freedoms that the software allows one to have. For more information about the free software philosophy visit *http://www.gnu.org/philosophy/philosophy.html*, or Richard Stallman's book Free Software, Free Society

Chapter 3

Why Choose Linux

As previously mentioned, Linux is a very stable and secure system. This means to the average user that he/she will not have to reboot as frequently as with alternative systems. It is also a very capable system with lots of applications. Also Linux like its Unix cousin is a true multiuser, multitasking operating system. This means that several users can be logged into and using the system at the same time! It also means that one user can log into the machine several times at once using the virtual terminal feature that is described in the command line introduction section of this document. Also like Unix, Linux is from the ground up a networking system. Unix grew up with the client/server computing model, and Linux has continued this tradition, it offers several key features that are lacking in other systems. These include transparent access to applications on remote machines. The ability to log on to and execute commands on a remote server, and the ability to access files from a variety of clients. Linux is also a very scalable system. It can run on everything from an old 486 to the latest supercomputers. It is a very lean system too there are even some complete systems, designed to run routers, that can be entirely contained on a single standard floppy diskette!

3.1 Freedom

Because it is free software licensed under terms of the GPL, Linux offers a level of freedom that isn't allowed with closed source products. With Linux you are free to change the source code in any way that you choose, and you can distribute you changes as long as you include the source code and give

the original author credit. This aspect of the system should not be under estimated; even if you never make a modification to the source code the freedom to view it grants the user an enormous amount of power. It means that if there is a bug in the software, and the original maker won't help one is free to make the modification (or have it made for them). It also means that all of the Application Program Interfaces are available to developers when writing programs. With other system,the original vendor does not release all of these interfaces called API's this means that accessory programs from competing vendors might not run as well on the computer.

Linux also offers a great deal of freedom to users. At first this freedom may seem intimidating to those who have used more closed systems in the past. Every aspect of the system can be altered to suit ones own individual needs and tastes. Even if you don't know how to program there are an almost endless number of configuration and administrative changes that can be made to a Linux system. Because of its Unix heritage, Linux also inherited the Unix philosophy of application development. As described in the Unix history section, Linux applications tend to focus on one clearly defined task and perform exceptionally at that task. This is in contrast to some current consumer operating systems and applications that attempt to do too many things and end up only being adequate at them. "A Jack of all trades is a master of none," This axiom applies to all things in life including software applications. Linux and Unix give one a number of tools that excel at one logically defined task and share a common interface that allows them to communicate through pipes and command redirection. This means that they can be combined in any way that the user needs or desires. This additional freedom imparts a great deal of power to the user, but it is up to the user to take advantage of it. Users that are new to this system will have to invest some time in learning it, but how many of you instantly knew how to use your current system. An even better question might be if the Microsoft system is so easy and intuitive to use how come all of those *Windows for Dummies* books do so well. Just because Linux is different doesn't mean that it is harder to use, and with the new enhancements made to Gnome and Kde someone that is already comfortable using their computer will have very little trouble making the adjustment.

Linux also gives users the freedom to learn. Though this might sound like a specious comment it is not. Though the use of multiple user accounts and file permissions Linux allows new users the freedom to experiment with and learn the system without fear of damaging it or deleting any critical system

files. As long as the user does not log in to the "root" account he or she can not hurt the system though he or she can still delete his or her own files.

With additional freedom comes responsibility, it is the responsibility of the user to decide what application will best suit his or her needs. It is the responsibility of the user to help support this additional freedom by contributing to the community. Contributing to the community can be as simple as helping a new user get his or her system setup or answering a question. There is a great deal of misunderstanding surrounding open source and free software, just helping to dispell myths can be the greatest contribution of all.

3.2 Software Costs

Linux can be obtained at reasonable cost or for free. Most distributions can be downloaded freely from the internet if you have a fast connection. *If you are using a phone modem to connect to the internet do not attempt to download Linux!* An entire distribution is usually several CD's full of software. Even though it can be downloaded freely there are a number of reasons why one would want to purchase a copy. First buying a copy will help to support the developers who make Linux possible. Second if you do not have a broadband connection it could take a week or more to completely download a distribution, and finally a purchased distribution almost always comes with extra tech support. Also as mentioned in the introduction to this chapter, Linux is a very lean system that runs very well on older hardware. One caution, don't expect your 486 with Linux to be as speed demon, but it will run effectively.

For some this low cost has actually been a barrier to trying Linux, many have confused it with the some times questionable "freeware" that is available for the Windows platform. Though it is *free software* **Linux is not FREEWARE!!** Linux is produced by a large group of developers all over the world, most volunteer their time to work on the project. Linux is free software for idealogical reasons and it is certainly not second rate software.

3.3 Stability

Linux is a very stable operating system. In fact some of the early versions have been known to run for a more than a year at a time (the newer releases haven't existed long enough to break these records). Though most end users don't leave their system running nonstop for an entire year, many people become frustrated with other operating systems that lock-up, have blue screens, and have to be rebooted to install most programs.

Because of its stability, much of the internet is powered by Linux, BSD, and other Unix type systems. Web, FTP, mail, and other network servers need to be placed on a stable system to limit the amount of time that the server is not operational. In fact many people are surprised to learn that Microsoft uses Free BSD for a number of purposes, as reported by the Netcraft survey, *http://www.netcraft.com/survey*.

Now with file systems Linux is even more stable. File systems like ext-3, reiserfs, XFS, and JFS make it possible to safely reboot without running a disk check if the system is suddenly shut down from something like a power failure. These systems keep a record of changes that are going to be made to the hard drive before they are actually made. While this slows disk access down slightly it helps to prevent damage to the data on the disk if the system is unexpectedly shutdown. If the system is unexpectidely shutdown as in a power failure then the transaction log of the journal is replayed at startup so that the state of the disk is never unknown. If damage does occur then there are a number of powerful filesystem repair tools such as fsck that allow the filesystem to be checked and recovered.

3.4 Security

Linux provides a more secure environment than most users are accustomed to. Unlike the 95, 98, and Me versions of Windows. Linux operates in a true multiuser, multitasking environment. A user must have a valid login ID and password to log-on to the system. After logging on only the system administrator, the "root" user can make modifications to critical system files, or access another user's files. This means that everyone that logs on to the system has his/her own home directory, and settings.

3.4.1 Viruses and Immunity

Linux is also immune to most programs currently being used to exploit systems. This is because most virus makers focus on Windows. When a program is created from source code in a process called compiling it is it made in such a way that it will only run on the type of system that created it. This means that almost without exception that a virus that was created for Windows will not be capable of infecting a Linux system.

Computer viruses share many properties with human disease, and disease prevention concepts developed for human disease can be applied to them. Just as with human disease there are several layers of immunity. The first and best layer is called genetic immunity. This level of immunity refers to immunity that is conferred to a person just because he of she is human. For example most pathogens (disease causing organisms) that infect animals are incapable of infecting humans. In the computer world this level of immunity is equivalent to operating system level immunity. For example just by using Linux, Unix, or a Mac you are automatically immune to all but a handful of viruses (most are written for windows). The next level of immunity is establishing barriers to entry. In humans this means avoiding getting cuts. The equivalent in Linux is avoiding programs with known security problems if at all possible. For example most risk can be avoided by not installing BIND or WU-FTP which have the most vulnerabilities. Finally there are several features like file permissions (described in the next section) which make it more difficult for a virus writer to create a virus for Linux.

The hardest part about securing any system is between the keyboard and the chair. The users of the system must be made aware of the importance of not opening things like executable e-mail attachments. It is also important to restrict access to program installation otherwise users could install "spyware" programs that could compromise system security. Often users are duped into installing these programs to gain access to some service like peer to peer file sharing and the application installs additional software that reports on the user's activities, or in the worst case could serve as a backdoor access point to the system.

3.4.2 Other Security Features

In addition to virus protection (by virtue of not being Windows), Linux includes a security feature called . File permissions tell the system what a

file is allowed to do and who is allowed to access and modify it. At first many new users are intimidated by the concept of file permissions, but once one gets the hang of it they are quite useful. This is example of a file entry in Linux. The first section of r's and w's tells the computer that the file can be read and written to, but not executed like a program.

```
-rw-rw-r--    1 brandon  brandon      7478 Apr 13 02:33 Using_Linux.tex
```

If this file could be executed by a user then the entry would appear like this one.

```
-rwxrwxr-x    1 brandon  brandon         0 Apr 13 02:42 fake_file.sh*
```

The x's tell the computer that the program can be executed like a program. If a file is not marked with these x's, then the computer will not allow it to run. For more information about file permissions please read the introduction to the command line section of the book.

3.5 Software Development

Linux also has the advantage of being a great development program. The access to source code, stability, and accessibility of the command line make it a great development system especially for new programmers who might not yet be ready to produce graphical applications. It has a number of different editors and compilers available, and it supports a wide variety of formatting, scripting, and programming languages. Linux has support for document formatting languages like html, and LaTeX . In addition it supports an enormous number of scripting languages. It also has compliers available for a number of languages like C, C++, Java, Fortran, and Cobol.

Chapter 4

How to Acquire & Install Linux

Checking the compatibility of your hardware is the first step in preparing to install Linux. In general the devices that you should pay most attention to are internal phone modems, and USB devices. Linux does have support for many of these devices but one would be well advised to check compatibility first. There are several resources on the internet that can be used for this purpose. One that I have used in the past is at *http://lhd.datapower.com*. One can also go the web sites of the makers of individual components to make sure that the manufacturer makes a Linux driver for the device.

4.1 Distributions

Unlike Microsoft and Apple products, Linux is not owned or produced by a single company. A is a collection of the Linux operating system and associated applications. Even though most of the software contained in each distribution is the same. Each has its own look and feel. I've used the Mandrake version of Linux more than any other. Mandrake's homepage is *http://www.linux-mandrake.com/en/*. Mandrake is more user friendly than many of the older distributions because it includes a number of configuration tools for the graphical environment. This allows users who are not yet comfortable with the command line to still make adjustments to the system as needed. It also includes a number of tools that aren't available in the other distributions. Some examples are the Mandrake Control Center and DiskDrake. The Mandrake control center functions as a central point for configuring many of the system's features. The DiskDrake Utility is similar

to the Partition Magic program in Windows. This utility allows one to create, format and resize a variety of file systems. The Mandrake system also includes its own system for managing rpm packages, called DiskDrake. This program makes it easy to add, remove, and update all of the programs on the system including the kernel itself.

4.2 Acquisition Method

As mentioned one can get Linux either by downloading it, or purchasing a copy. is a good option for someone that has already had experience with Linux or Unix, but it is not the best solution for someone that is unfamiliar with these systems. If you choose to download it, you should locate and contact the nearest Linux user group since they may have the distribution that you want already, and you will be able to download it much faster from their site which will almost certainly be less busy. To find the group nearest to you Linux.org maintains a list of user groups ordered by location. This listing can be found at *http://www.linux.org/groups/index.html*. In addition to providing you with a faster download contacting a Linux User Group will give you access to expert assistance that is only a call or an email away. In many cases groups will allow you to bring your computer to a meeting and will help you install it or solve a problem that you are having. Often these groups also have "install fests." These events allow those interested in installing Linux, but worried about trying to do it themselves to bring there computer and get one on one help with the actual install and configuration of the system.

If you choose to purchase Linux you have several choices. Most computer and office supply stores have at least one version of Linux available, usually Red hat or Mandrake. Copies can also be purchased from a distribution's homepage for usually a small fee to cover the costs of producing the CD's. Other on-line and computer stores also sell versions of Linux. In many cases Linux books also come with a distribution.

4.3 An Example Install

4.3.1 Preparing to Install

Before beginning the install, it is a good idea to make sure that you have backups of your important documents, and that you have ran scandisk and diskdefragmenter on all of your Windows partitions. If you already have Partition Magic, then you can use it to resize your Windows portion of the disk and create a section for Linux. If you do not have this tool the Red Hat and Mandrake distributions both have tools that will do the same job. I have had much more experience with Mandrake's DiskDrake Utility, so that is the one that I will describe in this section. You should also check to make sure that your computer is set up to boot from the CD-ROM drive, and that the CD-ROM is checked before the hard drive. To do this restart you computer and look at the first screen as it is starting up. It will usually display the name of the computer maker and tell you which button to press to enter the BIOS setup. On most systems this key is either "Delete" or "F1". *Your computer may have a different key that performs this function.* If the screen disappears too quickly to read, press the "Pause" key on the keyboard when you first see it and the computer will wait until you hit another key before resuming the boot process.

4.3.2 Beginning the Install

Now that the system is ready, you can place the first installation CD in the CD-ROM drive and restart the computer. When the computer starts you will see a penguin in front of a yellow star (in the case of the Mandrake version). From here press the "enter" key to continue. You will see some text scroll by quickly as the system prepares for the install. Then a graphical installer will appear on the screen. The installer will then ask you to select your location so that the language settings can be properly set, and it will then ask you to accept the license agreement. With this stage complete, the installer will then attempt to auto-detect your mouse and keyboard. It will also detect the types of disks that you have and load the correct modules to access them. In almost all cases this part of the install is complete automatically, and the user will have a chance to view and amend the results when it has completed.

Now the installation moves on to the critical step of partitioning the disk. The utility is automatically loaded from within the installer, and it

will display a menu of choices.

1. Use the Entire Disk
2. Use Free Space on the Windows Partition
3. Use existing Partition
4. Custom Disk Partitioning

I recommend using the free space option or the custom option if you have **not** already set up a place for Linux using partition magic. Both of these choices will take you to a screen displaying a graphical representation of how the disk is laid out. You can select a partition (section of the disk) by clicking on it with the mouse pointer. When a partition is selected a series of buttons will appear in a panel to the left. If the partition is formatted for Windows 95 or 98, you can use this utility to resize it **without losing any data!** The application will calculate how much the partition can be reduced and then display a dialog box where you can specify the new size by either entering it or clicking and dragging a slider bar.

After a short time the disk partition will be resized and the extra space that it previously occupied will appear as a white block. You can then click on this block and specify how you want to set up the Linux part of the disk. The easiest thing to do is to allocate it all as one large partition called "/" which will contain the entire Linux system, but this configuration is less then ideal for a number of reasons that will be discussed later in the system administration section. A configuration that I have used for quite some time and find to be very versatile is as follows.

Mount Point[1]	*Size*
/	250MB
/boot	8MB
/home	5GB or more
/usr	3-4GB
/var	500MB
/var/www [2]	500MB
/var/ftp [3]	3-6GB

At first the idea of mount points may be a bit confusing to the new user, but after some getting used to they are actually much more descriptive than

the method that other systems use to name there drives. As mentioned the "/" mount point is the uppermost. This is analogous to the My Computer icon that Windows users will be familiar with. The "/boot" mount point is where all of the programs needed to boot the system are stored. The "/home" point is where the personal files of all of the system's users except root are stored. To Windows users this will be similar to the "My Documents" folder. The "/usr" point is where most of the programs that are installed on the computer are stored. For those familiar with Windows this is like the "Programs" folder. The "/etc" point contains information about the system configuration The "/var" point is were variable data is stored. This includes mail and printer spools, databases, system logs, and web and FTP data. The /var/www and /var/ftp points are optional and are contained within the /var directory there are a number of reasons why one would want to place them on a separate partition, but unless you are running one of these servers you do not even need to have them at all. The Mandrake installer makes the selection of mount points easy and will even auto allocate your disk if you want. It will also suggest mount points for you so that you do not have to worry about remembering all of these.

After setting the partition sizes, the new partitions that you just created will be formatted, and the install will move to the next phase. In this part of the install the user selects which packages (applications) to install. Mandrake also makes this set easy by giving several general package categories that can be selected depending on what kind of applications that you plan on using. Don't worry about going through them all for now. There are literally thousands of packages available, and any of them can be added later using Mandrake's Update utility.

After selecting the packages, the computer will install them. This process should only take a few minutes depending on the speed of your computer and the number of packages that you have selected, but it generally takes between 10-30 minutes.

Once the packages are installed the installer program begins to make the final configurations to the system. The first step in this process is to set the root password. Root is a special user on the system that is permitted to make

[1]A mount point is a drives' name this is similar to the concept in Windows of giving drives the name of a letter.

[2]This partition will not be needed on every system and you should only include them if you are planning on installing a web server.

[3]This partition is only needed if you are planning on running an FTP server

changes freely and perform administrative tasks. *You should never login as root to do your day to day work!* Choose a good password for the root user it should contain letters and numbers, be at least 8 characters long, and easy to remember. One caution to new users, Linux passwords are case sensitive. Once the root password is set then you can add yourself as a regular user and create a password for that account using similar rules. You can then add another user or continue to the next phase of the installation.

After adding users, the installer continues the installation by setting up the system. X-Windows has no relationship to Microsoft Windows and actually existed prior to it. X-Windows which is commonly called simply X is the standard for graphical interfaces in Linux. It provides a common interface for graphical environments to communicate with while it handles all of the low level interaction with the hardware installed on the system. This step is usually automated and will almost always properly detect your video card and monitor, and automatically select a setting that will be compatible with both. If the installer fails to properly detect your video card then it will display a list of possible matches and ask you which one it is. It will then ask if you want to test the configuration to make sure that it works. It is always a good idea to go ahead with this test to make sure that everything is configured as it should be.

Next the installer configures the printer if you have one installed. Again this task is accomplished with a wizard style interface that would be familiar to Windows users. The installer next prepares the network. It first auto detects which devices you have, and then allows you to choose a configuration that best meets your individual needs. Configuring the network and approving the settings are the last interactive steps in the install. Next the boot-loader "LILO" is added. This is a small program that allows you to select whither you want to boot Linux or Windows when you start your computer.

The computer will then eject the CD-ROM and begin to shutdown. When it restarts you will see a menu that will allow you to select with the arrow keys which Operating System you want to use at the moment.

Part I

Getting to Know Linux

Chapter 5

The Graphical Environment

The Graphical environment is an important part of nearly all modern distributions. Though the myth is still oft perpetuated that Linux is a command line only system, graphical environments for it have existed for a number of years. Two of the most common are Gnome and Kde. Both have a huge developer base and are growing easier to use with every release. At present neither is significantly harder to navigate than the standard Windows interface. Underneath these desktops runs the X-Windows system. This system acts as a common interface for programs to interact with, so that they do not have to interact directly with what ever hardware you happen to have installed.

5.1 Choices

Unlike other systems that you may be familiar with Linux offers a number of choices in graphical environments, and even after one chooses his or her preference, he or she can still configure it in almost endless number of ways. Choices are a common theme in many aspects of Linux. This level of choice my be intimidating to new users who are familiar with more restrictive environments, but the defaults are quite easy to use making choice a luxury not a chore.

5.1.1 Gnome

The Gnome desktop is a mature environment that offers a number of important features. First it makes the transition from Windows easy by offering a system that resembles the "start" button for launching applications. It also allows one to embed small programs called applets in the taskbar. This is similar to the idea of tool bars, but is much more powerful. There are a few of these applets that are enabled by default such as the quick-launch applet and the task monitoring applet. In addition to these, you can add applets that will check your email (even when your e-mail client program is not running) and notify you when a message arrives. Gnome also places by default icons that link to your home directory and any Windows partitions on to your desktop so that you can easily access them. Gnome can be extensively configured and almost every aspect of the environment can be changed with point and click ease. Gnome has extensive documentation and step by step tutorials on its website *http://www.gnome.org*. There are also links on this site to download it if your distribution doesn't have it, or you want a different version from the one that comes with the distribution.

5.1.2 Kde

Kde is an alternative to the Gnome desktop environment. It shares much of the functionality of the Gnome environment though it is implemented differently. Despite their differences applications developed for either environment will work fine not matter which one you happen to be using. You also don't have to make a definite choice when starting out either because both can be installed on a machine at the same time without interfering with each other. Kde's basic setup resembles that of Gnome's except that it does not by default place icons for Windows partitions on the desktop, but they are easy to add by drag and drop. It also does not place the same emphasis on applets that Gnome does. I have kept both systems on my machine and which one I use depends on what mood I'm in on a given day. Both systems also offer a nice feature called virtual desktops. This feature can be conceived of as having a series of desktops available. This allows one to spread his or her work out across them without having one overly cluttered workspace. One can easily switch between them by clicking with the mouse pointer on any of the virtual desktop boxes at the bottom of the screen. You can even customize these desktops separately (each can have a different wallpaper ect).

You can even move an application window to another desktop to get it out of the way by right clicking on it and selecting "send to" and the desktop number. This will move the window to the other desktop, but leave your current workspace on the screen.

Additional help for the Kde desktop can by obtained at the Kde homepage *http://www.Kde.org*.

5.1.3 Others

As previously mentioned Linux is all about choice. To reinforce the point a number of lesser known graphical environments were developed. Many of these are not full desktop environments like Gnome of Kde, and they are called "Window Managers." Most of these are not horribly difficult to learn, but they aren't quite as user friendly as gnome or Kde. They do have some advantages though in some specific situations. For example if you are using your computer as a server many of these use fewer system resources than either gnome or Kde. They also perform well on older/slower computers. Some examples of these window managers include xfce, IceWM, blackbox, enlightenment, and WindowMaker. There are also many others too numerous to mention. The best advice for new users is to not start out with these environments unless you are having trouble with both gnome and Kde until you have some experience with Linux.

5.2 Navigating

Navigating the computer is fairly straight forward, but there are several marked differences from the Windows environment that many new users will have had previous experience with. First is the Unix style layout for the file system. At first this may be confusing to a new user because it is more independent from the physical device that the data is actually stored. This system has a number of advantages over the DOS/Windows style of naming drives by letter and having directories with in each drive letter. In Linux, and Unix in general all folders are kept in one root (top level) directory named "/" this is somewhat analogous to the My Computer icon that Windows users my have some experience with. There are several important directories to note as well. The directory named "/home" contains all of the personal files of all of the machine's users except root. This folder like any folder in the system can

actually be its own partition. For example the folder "/home/foo" could be stored on a separate disk from the folder "/home/bar." At first this concept may be a bit confusing, but it is actually a very powerful feature that adds greatly to the flexibility of Linux.

In addition to "/home" there are a number of other important directories. These include */usr*, */boot*, */etc*, */var*, and */mnt*. There are a few others, but normally you will not have to interact with them. "/usr" stores most of the programs on the computer, and is generally read-only to normal users. Normal users may or may not have access to the "/boot" directory depending on the security level that you choose in the installation. "/etc" is the folder where configuration about the machine and various programs is stored. Normal users will have read-only assess to this folder. "/var" stores information that changes frequently. This is were the printer and mail queue's are. If you have a web, FTP, or news server installed it will also serve its files from a subdirectory of "/var." Finally "/mnt" is where Windows partitions, the cd-ROM, and floppy drives are accessed from.

5.3 Productivity Software

There is a large amount of productivity software available for Linux systems. There are three major office suites Koffice, StarOffice, and gnome office applications. Each of these suites offers the traditional word processing, spread sheet, and presentation software. But there are also others available.

5.3.1 Koffice

Koffice is the Kde office suite its applications can import Microsoft Office formats, but it can not save them. It can export to a number of alternative formats though including rich text, html, and LaTeX. It has the features that users have come to demand from an office suite including the ability to embed objects created with one office application into a file created by another.

5.3.2 StarOffice

Staroffice is a very powerful office suite for Linux or Windows. It is capable of reading and writing MS Office file formats. It has a look and feel that is very comfortable to those who have used Microsoft Office before. It also allows

content created with one of its applications to be embedded into another. The other advantage of this office suite is that versions of it are available for Linux and Windows which eliminates any problem with sharing files between the two platforms if you decide to use it for both.

5.3.3 OpenOffice

OpenOffice is the opensource incarnation of StarOffice. After verson 5.2, Sun decided to begin distributing that suite for a fee, but at the same time released much of its codebase to the OpenOffice project. OpenOffice offers a similiar look and feel to StarOffice. It handles a variety of external file formats, and it can export documents to a variety of formats. With continued development from the developer community it has come to have all the features that all but a handful of users require.

5.3.4 Gnome Tools

Gnome also has a number of office tools associated with it. Abiword is a word processor with most of the features that one would commonly use, and it has the ability to import and export MS Word format. In addition it can export to rich text, html, and LaTeX. Its native file format is based on the XML standard which means that it can more easily be translated into other formats by another application or the user.

In addition to Abiword, Gnumeric is also another powerful office application for Linux. It is a spread sheet program that has a number of formulas built in. It allows one to have multiple worksheets with custom names. It can generate charts and graphs from data entered, and export the file into a variety of formats including MS-Excel. It can also save files in generic formats such as CSV (comma separated values) or tab deliminated values. It also has very good support for a variety of data types times, percents, ect. I've found through my informal testing with my running log that it actually handles calculations with time much better than I was ever able to accomplish with MS-Excel. Its user interface is easy to learn especially form someone who has had experience with other spreadsheet applications.

In addition to these standard office applications, gnucash, a personal finance manager is available. Its interface is designed to resemble a standard check register, but it has a number of other powerful functions. It has support

for a variety of account types and loan/bill tracking available. It can also import Quicken files which makes the transition to it even more easy.

5.4 Networking/Internet Software

There is also a large amount of networking software available for Linux. These include both client and server applications. With Linux, unlike some other systems you can actually take an active role in the internet by setting up your own web, mail, FTP, or news server. You will need the approval of your ISP if you want offer these services to the Internet in general, but if you have a home network or you want to see how your site will look when loaded over the internet then these tools can be very useful.

5.4.1 Browsing Software

There are a number web browsers available for Linux. The most common ones include Mozilla is a open source browser based on the Netscape code base. It looks similar to Netscape for this reason, but I like it better for several reasons. In my informal testing it renders pages better, and seems to be more standards compliant. It supports a variety of plug-ins designed for Netscape including flash and the Java Virtual Machine.

Konqueror is the default browser for Kde. It can display a variety of file types in addition to html because it is also the default file browser for Kde. It is pretty good at displaying a variety of sites, and the newest versions have support for Netscape plug-ins. It also supports changing the browser identification. This can be an important feature to get around the "designed for Internet Explorer" websites that disallow other types of browsers from accessing the site.

Opera is a fast and lean browser. It has a number of unique features including navigation by "mouse gestures." It also claims to be the "fastest browser on earth." In my experience it is very fast, and makes a big difference when one is connecting to the internet with a dial-up connection. It is free to download and use, but the free version has a banner ad in the upper-right corner of its window. One can purchase an ad free version for a small fee.

Lynx is a text only browser. It is very good when one wants to just read textual information without having a lot of pictures in the way. Because it doesn't support graphics, it loads pages very quickly. Newer versions have

support for SSL encryption allowing the user the ability to connect to secure sites. I've even used it to connect to hotmail!

Links is another text based browser that is similar to lynx, except that it is better able to handle webpages that contain frames or tables. It also has the ability to download files as a background task. This feature allows one to start several simultaneous downloads and continue browsing while they complete.

5.4.2 FTP Clients

There are a number of dedicated FTP clients available for Linux. These include lftp, gftp, and ftp. Lftp and gftp are both graphical clients that allow one to connect to an FTP server and download or upload files. FTP is a text based client that one can use to connect to, download, and upload files from an FTP server. For those who don't want to take the time to learn a new application it is useful to know that most of the web browsers can be used to browse FTP sites. Netscape, Mozilla, and Konqueror can all be used to connect to, browse, download, and upload files from FTP sites. Using this method one can drag and drop the files to or from the remote site. Lynx can be used to connect to, browse, and download from FTP sites. I haven't attempted to upload files using it because I thought that it would be awkward.

5.4.3 Mail Clients

There are mail clients to suit almost everyone's tastes. For those that want an outlook like interface the evolution client is a good choice. This client can manage multiple mail accounts, can be used to read newsgroups, and can be used as a calendar. Additionally with the Xiaman Connector plugin one can connect to Microsoft Exchange servers from evolution, and one use the server's appointment scheduling features.

Kmail is kde's default mail client, it does not offer all of the features that are bundled into the evolution client, but it is very good at managing multiple mail accounts. It is also very easy to setup and learn to use. It works with GPG security program allowing it to exchange encrypted mail, or to digitally sign messages.

There are also a number of console based mail clients including mutt which is my favorite. This client, besides being console based, is quite pow-

erful. Its interface is clean as well as being easy to navigate and understand. Despite this simplicity it is very powerful. It allows messages to be sorted in a variety of ways including grouping by subject, which allows "Threading." It also allows you to create a special signature file that you can automatically append to every message that sent. Mutt does not do any mail filtering on its own, but the procmail utility which is available for all distributions will allow one to do complex mail filtering tasks.

5.4.4 Servers

This section is meant to serve only as an introduction to these servers, and more information about these will be presented in the System Administration section. There are a number of powerful servers that are available for Linux including apache, MySQL, PostgreSQL, CUPS, sendmail, Postfix, INN, proFTP, and squid. There are many others, but these are the most commonly encountered.

Apache is an entirely modular webserver, and is the most commonly use webserver on the internet today. It is reasonably easy to configure, and a number of graphical configuration tools exist so that you don't have to edit the configuration file directly if you don't want to. For more information about apache, visit the project homepage at *http://www.apache.org*.

MySQL and PostgreSQL are both fast and powerful database tools that are available for Linux. While doing research for this paper, I've found that MySQL now offers support for (Binary Large OBjects). BLOB's can be used to store data other than just text into a data base. This is useful if you want to store and associate a picture or some other file with other data. For detailed information about MySQL, visit its homepage at *http://www.MySQL.org*.

CUPS, the (Common Unix Printing System), is a set of printer drivers that can be used to print to local or remote printers. It even supports printing sending and receiving print jobs across the a network, the Internet, or a VPN (Virtual Private Network). It is very easy to configure through a web browser by connecting to port 631 on the computer that is running it. Users can access it by entering this URL into their webbrowser *http://127.0.0.1:631*. It also supports SSL so the information sent to it can be encrypted in transit to the remote system.

Sendmail and Postfix are both powerful mail server applications that are capable of supporting large numbers of users. These servers also have a number of add on packages including automatic virus scanning of e-mail

attachments. They also have a number of easy to use configuration tools so that one does not have to interact directly with the configuration unless you just want to.

is a versatile tool that is often used with apache or other servers. It is a primarily a proxy server. It can be used as a security layer or it can cache frequently requested data.

5.5 Graphical Development Tools

5.5.1 Editors

There are a number of powerful editors that are available for the graphical environment in Linux. Including emacs, gvim, and kwrite. Each of these editors does syntax highlighting for a variety of languages. Emacs and gvim also offer several advanced features as well.

Emacs This editor has had a long history. It was originally created by Richard Stallman, and was one of the first programs that was licensed under the terms of the GPL. It has undergone much development since it's creation and has a number of features that one would not expect from a text editor. In addition to highlighting syntax for a variety of programming and mark-up languages, it includes macros to check spelling, automatically compile a program or LaTeX document, do Lisp evaluation (Lisp is a language used in many artificial intelligence applications), and it even includes a mail and newsgroup reader. It is easy to learn when using it in a graphical environment, but it can be difficult to learn in console mode where one does not have the benefit of using the mouse to activate menus.

Vi & Gvim Vi has been a popular editor for Unix systems for a long time. It has many powerful functions while at the same time requiring very few system resources to operate. Like emacs it can be confusing for new users to learn because it requires the memorization of several keystroke commands, but it can be a very good choice if you are concerned about system resources, you want an extensible editor with many powerful editing functions built-in, or if you are connecting to a remote computer with ssh or telnet and are having problems getting the "Ctrl" and "Alt" /"Meta" keys to work properly. On most current systems typing vi will actually invoke vim. Vim

is the successor to vi. It does all the things that vi can do, and has some added features too. It accepts all the same commands that vi does, so those familiar with vi do not have to spend any time re-learning features that they already know. Gvim is the latest in the vi family of editors. It is substantially easier to use than either vi or vim because it is a graphical application. It accepts all the commands that vi or vim do, and it has most of them built in to menus that make it easier for new users to access. It has a very good tutorial that is included with the application that guides a new user through how to get started with gvim. It can be accessed by opening the gvim application and typing ":help" as soon as the application opens.

Kwrite KWrite is another graphical text editor. It does not have as many features as emacs or gvim, but it can be less intimidating for new users. It has an interface similar to notepad in Windows except it includes syntax highlighting for a number of languages, spell checking, and it can convert a file with Unix end of line markers to Windows end of line markers to make it more easily viewed from the Windows platform. It is part of the Kde and is the default viewer for a number of plain text file types in Kde.

5.5.2 Compilers

There are a number of compliers and interpreters available for Linux. Gcc is the most common it originally was an acronym from **G**nu **C** **C**ompiler, but now stands for **G**nu **C**ompiler **C**ollection. It includes compilers for C, C++, Fortran, Java, and can be adapted for Ada95. There are also a number of interpreters for scripting languages. These include python, perl, ruby, php, and batch (shell) scripts. Though you still have learn the languages Linux provides all of these options for free, and many of them are usually installed by default, so you don't have to worry about downloading a large development kit.

Chapter 6

The Command Line Environment

The command line environment is the part of Linux that is most often misunderstood or misrepresented by those who are not familiar with Linux or Unix already. In fact some still believe that Linux is a command line only system which hasn't been true for many years now. The command line is a powerful part of Linux and can be used to exercise great control over your system, but new users are not obligated to use it in their day to day tasks. For those that decide to use it, the command line offers a number of features that are quite useful.

6.1 Common Commands

6.1.1 Navigation

The commands that are used to navigate to different directories are likely the ones that one will use most frequently. Most navigation commands are two letters that are usually closely associated with the operation that they perform. The following list details these commands.

1. ls

   This command is quite useful it will **lis**t the contents of the current directory. It can be modified using attributes. Typing "ls -l" for example produces a list of all the files their owner, permissions, size, date of creation, and name. The command "ls -a" produces a complete listing

of files in the directory included ones that are designated as hidden. (Hidden files all begin with a '.'). You can even combine these two commands! By typing "ls -la" the computer will list all files, including hidden ones, and display all the additional information about them.

2. cd

   This command will be familiar to those that have used DOS before and it functions in much the same way as it does in that system. Typing cd and a directory name will move you to that directory. Linux has built in a number of enhancements to it though. For example if you wanted to change to "ReallyLongDirectoryNameThatIsHardToType" you don't have to type the whole thing. Simply by typing "cd Rea

   $$Tab$$

   " Linux will autocomplete the rest of the name for you. If there are two directories with similar names it will fill in as much as it can and then ask you which one you want. You can then simply type the next letter of two and then hit

   $$tab$$

   again and it will finish filling in the name for you. There are also a number of other short-cuts that can be used with this command. For example typing "cd .." will take you up one directory. Typing "cd " will move you to your home directory no matter where you currently are. One of the most useful shortcuts is "cd -" this will take you back to the last directory that you were in. The cd command even allows you to go through several layers of directories at once. For example the following example will illustrate this point. Say one is in a directory called /home/user/source_code/ and he/she wants to go to a webpage that is saved on the desktop. He/She can move directly to it by typing "cd /Desktop/Web_Pages/Current_Research/". Linux would then take the user directly to that folder, and using the autocomplete feature too he or she would not even have to type the entire command.

6.1.2 Useful Operations

In addition to navigation commands there are also a number of things that one might want to do from the command line. The following list outlines some of these commands and how they can be used.

1. man & info

   This pair of commands can be used to obtain information about a command. Both have a similar information format and automatically display the results a page at a time. If you wanted more information about the cd command for example typing "man cd" would cause information about the cd command to be displayed on the screen. When you are done reading about the command you can return to the command prompt by typing the "q" key.

2. cat

   This command allows one to view the contents of a file without opening it. It can be executed on any file but only text files will contain human readable content. This command is often combined with others using the

   ‘‘|’’

   character. This practice is referred to as piping. For example if the contents of the file are too large to fit on to the screen, the cat command can be combined with the less command to make the contents be displayed one page at a time. The command would be entered like this: "cat foobar.txt — less" the contents of the file would then be displayed one page at a time on the screen. The user can move back and forth through the pages while reading using this method.

3. strings This command is similiar to cat except that when displaying the file it skips control characters. This is useful when displaying files that may not contain only normal text, such as executable files. Using cat on such files can scramble your termanal settings, which generally requires a terminal reset to cure.

4. wc

   The wc command reads a file and outputs the number of words that are in it. This command is also frequently combined with others often as part of a shell script.

5. lpr

The lpr command can be used to print documents. It can be used to print a variety of documents in addition to plain text including .pdf, .dvi, and postscript formats.

6. ispell

   If the ispell program is installed on your machine this command will activate a console based spell checking program that you can use to check the spelling of your documents. Ispell is very userfriendly and clean. It highlights possible mistakes and displays suggestions the user can then choose to skip the word, replace the word, add the word to the dictionary, or exit the program. When displaying the possible error the ispell shows the sentence proceeding it and the sentence containing it. It also ignores mark-up tags making it great for checking XML, HTML, or LaTeX documents.

In addition to the basic operation commands there are a number of programs that are designed to allow one to monitor the system.

1. top

   This command produces a table that shows how long the system has been running, the amount of processor usage, memory utilization, and all the processes that are currently running. The processes can be sorted by user, memory usage, or processor usage. Top even allows one to kill processes by entering k and then the process number that you want to terminate. Top also automatically updates the screen after a predefined time interval so that current information about the machine's status is always displayed.

2. ps

   The ps command, can be used to display all the processes that one are currently running. The default is to only display processes that are running on the terminal that one are currently logged into, but by using some simple attributes in addition to it a person can view all of the processes he or she is running or all of the processes running on the machine. Modifying the command to be "ps -ux" will display all of the programs that the current user is running. By typing "ps -aux" the computer will display a list of all the processes that are users are currently running. If you have used the top program that was described

just before this one then the output from the ps command will look very familiar. It includes not only the process name but its process number, and its status. Using the technique of piping that was described earlier one can easily tailor the output to get just the information that he or she wants. For example if one wanted to see if the program "foobar" was running issuing the following command would show only programs named foobar.

```
ps -aux | grep foobar
```

Unlike the top command the list that is generated by the ps command does not update itself, so the command must be rerun to get updates about the state of the system.

3. Memory and Storage Commands

   The free command generates a table that shows how much free memory is available on the system. It displays information about RAM and Swap (virtual memory), and how much of each is used.

   The df command generates and easy to read table that allows one to quickly see how much space is available on all of the system's disks it even displays information about floppies if there are any in use at the time that the command is issued.

   The du command generates a complete listing of all of the files on the system and how much space each takes up. This command can be modified to only display the contents of a particular directory or mount point. It can be useful for generating a complete list of files in a directory because unlike the ls command it also displays files that may be contained in subdirectories. It can also be useful when combined with other commands using pipes. One can also use command line redirection to store the output of the command in to a plain text file. For example the command:

   ```
   du ~/mp3 > song_list.txt
   ```

   will generate a complete list of all of the files in the mp3 directory, and store that list in a file named "song_list.txt" one can then view this

file with a standard text editor. As previously stated if there are files stored in subdirectories within this folder they will also be listed.

6.1.3 Finding Information

There are several ways to find out what commands are available on the computer that you are using and how to use them. First by using the ls command on the /usr, /usr/bin, and /usr/local/bin directories you can view all of the commands that are available. Since there may be a large number redirecting the output of the command into a file will allow you to view the information in a text editor like gvim.

```
ls -la /usr/bin > usr_list.txt
```

This command will generate a list of all files in the /usr/bin directory. If you have followed the naming conventions for the /usr mount point this will also be a list of commands that are available to you. With this list you can then use the "man" and "info" commands to look up the manual pages for these programs. Most distributions also install documentation in either /usr/doc, or /usr/local/doc. This documentation can be very helpful with the process of learning to use your Linux system in general and will have specific up to date information for your distribution and configuration.

Chapter 7

Basic System Administration

System Administration is one of the most important aspects of computing. It is also a fairly broad category of tasks. The goal of system administration should be to take advantage of all of the system's resources to the fullest extent possible for the completion of what ever computing task one is involved with. This proposition can be more difficult in a multiuser/multitasking environment such as Linux because there may be several users logged into and using a system at one time. The Linux kernel handles the low level problems of deciding which applications will receive which resources, but there are still steps that need to be taken as the system administrator to help ensure that all of the users of the system have access to its resources.

7.1 Tools

There are several tools that are available to make the task of system administration easier. This section will cover some of the most important concepts, and introduce the reader to several common system administration tools.

7.1.1 RPM Packages

RPM or the **R**edhat **P**ackage **M**anager makes the installation and removal of programs easy. This system for program installation greatly simplifies the task of installing new applications lessening the amount of work that the system administrator needs to do to add, remove, or upgrade an application.

Many distributions, in addition to Redhat, are compatible with RPM

programs. Nearly any RPM package will work on the Mandrake Distribution, and RPM's are used exclusively in the initial install process. RPM includes a command line tool called rpm. If one wanted to install a package named foobar.rpm, the command would be: "rpm -ivh foobar.rpm" The foobar application would then be installed. Most of the time you will need to be logged in as root to install rpm packages.

As with other configuration tasks there are a number of graphical applications that simplify this process. Mandrake's update utility called rpmdrake can install, update, or remove rpm packages. Kpackage can also be used to install rpm packages.

7.1.2 Linux Conf

This tool can be used by the system administrator to configure a variety of system services such as webservers, ftpservers, mail servers, and file servers. It can also be used to managage users. It can add new users, remove users, or revoke user privelages. In addition it can be used to restrict actions that an individual is allowed to perform on the system. For example, the "ftp" user is usually not allowed to open a command shell. This is a security feature that makes it more difficult for an intruder to access system services even if that individual has found a way to compromise the ftp server.

Linux Conf also offers a tool to let the system administrator read the system logs, messages, and warnings.

7.2 Printers

7.2.1 CUPS

There are several ways to install a printer using Linux, and your choice will largely depend on which distribution you choose. The Mandrake distribution uses a system called CUPS this stands for **C**ommon **U**nix **P**rinting **S**ystem. This is a very easy to use system that can be configured using a web browser. It also allows one to configure the computer to be a print server that can accept jobs from other computers on the network or even over the internet! It supports SSL encryption for data that is being transmitted to it for added security. To configure CUPS using a web browser. After opening the browser enter the address *http://127.0.0.1:631*. If you are connecting from a remote

machine enter the address as *http://machinename.domain:631*. Either way after entering the root password you can configure your system to print.

CUPS can even be configured to print to allow you to print to another printer on the network. This remote printer can either be a stand alone network printer, another CUPS system, or a Windows machine that has printer sharing activated.

Before deciding to use this system one should visit the CUPS homepage at *http://www.cups.org* here you will find information about the latest CUPS version and a list of supported printers.

7.2.2 LPD

The lpd option can be more complicated. LPD is simply the name of a printer daemon, but there are many different implementations of this system. Each of these systems has similar functionality and options, but different configuration methods.

7.3 Samba (Windows File-sharing)

The Samba system allows Linux users to share files on a Windows network with Windows machines. It contains three main parts the config file this is usually called smb.conf and stored in the /etc directory. In addition it includes a server and a client program. The server allows Windows computers to connect to your Linux machine and the client allows you to access shared files and printers on Windows machines.

7.3.1 Samba Configuration

The Samba system can be configured by directly editing the smb.conf file, or by using a graphical configuration utility that will write the configuration for you using information that it gathers from you. Among the easiest of these utilities is the SWAT program. It allows you to access information about your configuration and to change your configuration using a web browser. Because it is network based it has the advantage of allowing easy remote configuration. If you have SWAT installed on your system you can open a web browser and enter the following address: *http://127.0.0.1:901/* and the SWAT configuration tool will appear as a webpage. This format makes it easy

to enter the required information in the same manner that you would fill out a form on the Internet. If you are connecting to a business or campus network you may need to get some of the required information from your system administrator, or you can log in using a Windows machine and write down any information that you need to have and don't already know. The address of the network's WINS server falls into this category. Not all networks will have such a server, but if yours does then you will need to enter it's IP address for Samba to work properly. You should also pay attention to the "OS Level" option set this to 0 unless your machine is going to be the domain controller and set the domain controller to false. Also set the WINS support to false unless your machine is the WINS server. If you don't know what a WINS server is then your machine is almost certainly not performing that function on the network. There should only be one such server on the network and outages can result if an improperly configured machine attempts to take on this role. It is important to note that the WINS support option only refers to the server not to the client. If you select false the Samba client will still be able to browse the network.

SWAT also allows you to define file and printer shares. It makes the process easy most of the time you just have to enter the name that you want to call the share, and the path to it. There are then some security options that you can select. For example you can make the share invisible or you can only allow users with user accounts to connect. In addition you can reset the read/write status of the share by default shares are read-only, but this can be modified.

The Samba system can also be configured using the Linuxconf program. This program offers most of the basic configuration options, but I have found that it is somewhat harder to use than the SWAT utility that I previously described.

7.3.2 Samba Server

Once configured the Samba server can be started by logging in as root and typing the commands: "nmbd start" and then "smbd start" if you change the configuration while the server is running then you will have to issue these commands: "nmbd restart" and "smbd restart." If you are uncomfortable with the command line the SWAT utility has a tool that will perform this operation for you. With the SWAT page open click on the "Status" link at the top of the page this will take you to a page that displays information

about the current status of the server and everyone that is connected to your computer, and it also has two buttons that allow you to start or restart these services.

7.3.3 Samba Clients

There are a number of clients designed to work with Samba. The smbclient that comes with the Samba package is a text based program that allows one to navigate the network. This is okay if you are using a small network or know exactly where you are going, but it is not very good for browsing. To help make network browsing easier there are several graphical browsers. My favorite is called smbrowser. It is a graphical front end to the smbclient program. It uses scripts to interact with the smbclient program, and presents the user with a clean easy to understand graphical interface to browse through the network. Because it interacts with the smbclient program it requires very little configuration by the user. If Samba is properly installed and running it will automatically use those predefined settings to setup the browsing session.

Another browsing program is called gnomba. This program does not interact directly with the smbclient program, and requires the user to enter some configuration information about the network. Also as of the last time I used it the program did not support browsing on networks with a WINS server.

There is also a Kde application called komba. I have not used this program at all, but I've read about it and it's goal is to present the user with a Network Neighborhood type interface. In addition to komba, kde's Konqueror web browser is also capable of connecting to Windows shares thought it does not allow Network Neighborhood style browsing through available shares.

7.4 Apache

Apache is quite possibly the premier webserver today. Like other popular software titles available for Linux it is also Free Software. Most likely your distribution will already have this available as a package, and in many cases it is installed by default.

7.4.1 Description

Apache makes it possible for you to create a website on your computer. This does not automatically make your computer a webserver however. Depending on other aspects of your configuration the page may be viewable to any computer on the Internet, only computers on your home network, or possibly only the computer on which it is installed. One might wonder what the point of having a webserver that only served pages to the machine that it was installed on. The answer is actually quite simple **Debugging** like many web developers I have my public site mirrored on my computer so that I can try things out before adding them to my site. With static content this process is not as important, but if one wants to test CGI scripts or other dynamic content a webserver is essential.

7.4.2 Configuration

A number of tools exist to make it possible for new users to configure apache without having to learn the details of how its configuration file works. It is a good idea to look at this file called httpd.conf and familiarize yourself with it, if you want to get serious about customizing your setup. The two configuration tools that I have used the most are Webmin and Linuxconf. Both of these programs are described elsewhere in this text. Each contains some specific modules which one can use to configure the server to meet your needs. Each is easy to use, providing form fields and radio buttons for most options. After collecting the information, each then writes the configuration file. Linuxconf will even offer to restart the server for you (necessary after making configuration changes). If it doesn't restarting the server is a simple task. First log in as root or another superuser account and open a terminal window; type **httpd restart** and the server will restart!

7.4.3 Security

As with any server one needs to be careful to keep up with security updates and warnings. If you only want to server files on your local machine, then use iptables to close the port to outside connections. This command will accomplish this objective *iptables -t filter -A -s !127.0.0.1 -j DROP*. There are numerous other ways to accomplish this goal. It is also possible to formulate rules that will filter incoming requests to your machine so that they

will not reach the server. A very good tutorial about setting up iptables rules can be found at $http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html$. This site does a much better job explaining what iptables is and how to use it.

If you are creating a site that will be available to the Internet at large there are some important things to take into consideration. First don't assume that just because Apache has a good security track record that you are safe, when people feel safe they get complacent about security. The following are some general guidelines that will help to improve the security of your site.

1. Don't advertise your configuration. If you like me then you take pride in your system, and you want to show it off. This isn't a good idea on a website however. banners that indicate the version of Apache you are using or which Operating System you have only saves time for a potential cracker. While this information can be obtained by other means, with-holding it will force the attacker to use these other means.

2. Use chroot

   Chroot is a utility that changes the root directory of the webserver. It can also be used for other severs as well, basically what it does is limit the applications ability to access the file system. If apache were *chroot-ed* in the directory where the files for the site are stored it would only be allowed to access directories contained with in this area of the system. From the point of view of the application $/var/public_html/$ would be the equivalent of the root directory /. This approach is helpful to control the damage if the server becomes compromised in some way.

3. Don't grant more access than you have to.

   Linux has inherited the concept of file permissions from Unix, which is described at the beginning of this text. File permissions are useful tools that enable one to restrict the access visitors have to the machine. For example one should almost never allow write permission on any of the files in the site directory. For sites with only static content mount the drive as read-only, which will prevent anything from being written to it. If you want to modify something on your site later you can unmount it and remount it with write capability.

4. Use *good* passwords

A good password goes along way to improving your security. One should avoid using dictionary words, names, jargon, etc. A password should contain uppercase, lowercase letters, numbers, and symbols. It should be reasonably long at least 8-10 characters, but more is better. Along with choosing a good password, management of passwords is vitally important! Crackers often try and use social engineering to trick someone into telling them the password to a machine rather than actually mounting a brute-force attack which could take a long time with a well chosen password.

5. Avoid Scams

   As mentioned in the previous point, hackers often don't want to take the time to mount a brute force attack; doing so often consumes large amounts of effort and time if one has adequate security. The next weakest link is the human factor, and physically securing the computer. Don't tell your password to any one. If someone has a legitimate reason to use the machine, give them a limited user account, restrict their access permissions, and limit the amount of time that the account is valid. The truth of the matter is most crackers' are actually nothing more than con-artists. While jigalos specialize in wealthy older women, crackers specialize in conning techie's. It is important that everyone involved with your project has a commitment to security and understands this risk that crackers pose.

Chapter 8

Summary

8.1 General Description

Linux is a stable, low cost, easy to install alternative to proprietary systems which are available. Like many other systems today it comes from the Unix tradition of operating systems though it was developed independently of Unix. Some other operating systems that are based on Unix are Solaris, HP-UX, AIX, and anything with BSD in the title. In addition OS X from Apple is based on the BSD version of Unix with the familiar Mac interface pasted on top of it.

8.2 Philosophy Overview

Unlike other systems Linux is not controlled by a single manufacturer, so it is available in a variety of versions that are called distributions. Many of these are tailored to work best for a specific purpose or to suit the individual needs of a developer. The GNU Public License which governs the distribution of Linux base products allows individuals to make modifications to the system to suit their needs provided that they also make these available under the same license terms. This licensing scheme ensures that software like Linux can not be made proprietary in the future.

8.2.1 What is Free Software

Much misunderstanding exists about the label free software. The *free* in free software refers to speech and not price. Though many programs licensed under the GPL are free to download and use, it is not unreasonable, or uncommon to pay for *free* software. One might want to do such a thing for a variety of reasons. First if one has a slow internet connection it might not be practical to download the software in a reasonable period of time. Second most free software projects are the work of volunteers and often purchases are more or less considered donations that help to support those who work to develop the application that you use. Finally the GPL license makes no claim of warranty or support, and many companies, RedHat for example sell service as a primary product.

Free software is also about empowering the user of a piece of software. The analogy of a car is often used to illustrate this point many people don't know how to work on their own cars, just as most computer users can't program a computer. Even though one doesn't work on the car personally, it is generally regarded as one's right to have the work performed by the mechanic of one's own choosing. If cars were like most commercial software is today the hood would be locked at the factory, and could only be opened by a dealer! Imagine having to got to the dealer anytime you wanted any work done at all. Would you trust that dealer to charge a fair price, or not try and pressure you into buying a bunch of parts that you don't need. Remember in this analogy the dealer would be the only one who would every be able to see the engine; you would have no way to independently verify what had been told to you. This is the situation that exists with proprietary software today. If one has a problem, it is up to the publisher to decide when or if the problem will be addressed. The publisher will likely even disavow any knowledge that a problem even exists! Since the source code (access to the engine) is not provided you can't verify this fact for yourself or even (take the application to another shop for a second opinion). While the proponents of proprietary software often make the claim that free software is anticapitolistic the opposite condition actually exists. Free software opens a market and allows greater competition. If a competitor can fix a bug in software product A and product A is free software, then they would be permitted to do so and sell the product to new customers. To carry the car analogy further imagine if GM was the only company that produced cars on a large scale. Do you think that it would have much incentive to innovate or keep prices

low? Proprietary vendors claim that allowing monopolization of the market in what ever area they work in will improve development and keep costs down. They further claim that allowing competition would be desasterous because they would lose the investment they made in the product's development, and the product's interface would fragment and become difficult for users to transition between versions. Again cars cost as much or more to develop than the average piece of software, and while many vendors exist they have a remarkablely similar interface because of user demand. Fragmentation of a products interface will occur only if users find it undesirable, in which case it should be changed anyway!

Free software and security is another contentious issue. Proprietary vendors often employ what is known as "security by obscurity" which means if no one knows how it works then they won't know how to break it. The flaw in this thinking is that one doesn't necessarily have to know the details of something operation to disrupt that operation. Knowing only how to drive a vehicle (use a product), one can destroy that vehicle or at least make it inoperable without ever knowing how its underlying parts work. The same concept is true for software; the true test of a software product is not how it reacts to normal conditions, it is how it handles errors. If a malicious user is able to put a program into an error state the chances that he/she will be able to exploit that state increase tremendously. Keep in mind this disruption does not at all depend of knowing the innerworkings of the product only interacting with its interface. The strength of free software is the level of code auditing. If a particular program has a bug, one can often contact the programmer who created that code directly and relay the problem. This ability is especially important for security applications when it is critical to verify that a program truly does what it claims to do, and is free from obvious errors.

Just as most people don't work on their own cars; most users of free software will never modify their own code, However having the source available makes it possible to hire someone to work on the code; just like one can hire a mechanic to work on a car. Microsoft's "Software as Service" campaign actually had substance, though I don't believe it would actually give users more freedom. Software should be a product that one purchases, and then can have serviced by the programmer of their choice. If that programmer is the original vendor then so be it. If however the original vendor is unable or unwilling to satisfy the demands of its customers, then that customer should have the right to take their business elsewhere. Free software creates

competition it doesn't destroy markets.

8.3 Relationship to GNU

Unlike proprietary software no single company is responsible for the development of Linux. GNU which was founded by Richard Stallman is the creator of the GPL license which Linux is developed under. GNU is not responsible for the development of Linux nor does it provide a warranty for it. Support is the job of individual vendors GNU helps to ensure that Linux and other free applications remain free.

Chapter 9

GNU Free Documentation License

Version 1.1, March 2000

Preamble

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software

does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

9.1 Applicability and Definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or

discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

9.2 Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

9.3 Copying in Quantity

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts:

Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

9.4 Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if

there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- Include an unaltered copy of this License.
- Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

- Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

- Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

- Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties – for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

9.5 Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

9.6 Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

9.7 Aggregation With Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution

medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

9.8 Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9.9 Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9.10 Future Revisions of This License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

Typeset with LaTeX