

Chapter 1

Introduction to Linux



This chapter introduces the new user to the basic system concepts and background of the Linux System.

Concepts Learned in this Chapter

- Why one might use Unix / Linux
- Linux History
- Architectural Support
- Linux Distributions

Table of Contents

Introduction to Linux.....	1
1.1 Introduction.....	3
1.1.1 Linux Operating System	3
1.1.2 Installation Requirements.....	4
1.1.3 System Requirements	4
1.1.3.1 Recommended Hardware	5
1.1.3.2 Justification for Recommendation	5
1.1.4 Text Conventions	6
1.1.5 Wild Card Characters.....	7
1.1.6 File Name Completion.....	7
1.1.7 Megacharacters.....	7
1.1.8 Terminating an Application	8
1.1.9 System Lockup	8
1.1.10 Power of the Administrator ROOT	9
1.1.11 Recommendation.....	9
1.1.12 Corrections and Updates.....	9
1.2 General Organization.....	9
1.3 Why use Unix / Linux	9
1.3.1 Which Version Should I Use?	12
1.3.1.1 Capabilities.....	12
1.3.1.2 Installation Time.....	12
1.4 Linux Standardization	12
1.5 Multi-Architecture Support.....	13
1.6 Linux Distributions	14
1.6.1 Linux Versions.....	14
1.6.2 Linux Standardization	16
1.7 FLOSS.....	17
1.8 Commands Used in this Chapter.....	17
1.9 Chapter Review Questions.....	17

1.1 Introduction

There are numerous books published regarding the installation and operation of Linux. These tell the reader how to install and nearly all of the commands that are needed to run, but to the author's observation, virtually none have a simple procedure to have the new user perform the various commands while in front of a Linux system. This book, along with the Lab Manual has been designed to provide the new Linux user a simple exercise process where they can perform the basic commands and observe the results. This book is not to be considered to be complete nor exhaustive, but to give the user a simple, fundamental and initial understanding of the operation. Additional practice is required to become proficient, and further investigation is required to understand the full features of all options available. The user is referred to the online man (manual) and info (information) pages for additional command clarification. More will be learned these commands starting with Chapter 3. Lab exercises start with Chapter 3, and are a separate manual.

It is the intent of this manual to take the Linux user from the beginning of how to log on all the way through to the ability to set up basic servers. The user will learn the basics, but specifically to that of servers, it does not cover all of the options to improve security or improve performance – these are advanced topics that are beyond the scope of this book. It is believed that a little bit of success leads one to want to learn more, thus getting a server to work as a basic system makes one feel good – and to want to go forward. In all cases, the exercises are not exhaustive, additional practice and study beyond this manual is required. For those that are really interested in Linux, and you have a local Linux Users Group (LUG), it is highly recommended that you join; one can learn more advanced topics and and have fun with a group of great people.

Lastly, this manual is not to be considered complete, many more commands and options exist than what are presented here. It is left to the student of Unix and Linux to investigate and experiment in order to become more proficient. As a general concept, Chapter 3 provides the basic commands to navigate and perform the very basic functions to operate on the Linux system; Chapter 6 provides additional administrator commands; and Chapter 9 provides many more user commands. Neither are exhaustive.

1.1.1 Linux Operating System

The basic Linux Operating System chosen to run in a classroom environment is Fedora Core 5 (this will always be out of date). Within the classroom, it is the intent that a YUM repository server be established and a Workstation installation by the student be made of Fedora be made. When an additional function is required, the installation is made from the local repository. (Instructions for setting up a repository server are in the Appendix.) Starting with Chapter 3, the instructions presented here are generally independent of the operating system (there are a few specific Red Hat commands). Later versions are available, or you may desire to use another distribution, but this should not be a problem, as we are attempting to learn the fundamentals.

The reason that Red Hat / Fedora has been chosen is fairly simple – the vast majority of books presently on the market are in reference to Red Hat. Hence the author has been able to learn the most about Linux while reviewing the 7-foot high stack of books. Other distributions are just as good as Red Hat. Each has their own highlights. If the student should install the Red Hat 7.3 package (yes, very outdated), it installs in about 1 hour using an “Install Everything” process, where the students actually install the operating system. Installing Fedora Core 5 or later distributions can require up to three hours, which may be longer than allowed for class time. By using the YUM repository and a minimum installation, the installation time in classroom can be less than 1 hour. If one is installing Fedora at home for the first time, it is recommended that a full installation be made, thus minimizing any frustration during the initial learning curve.

1.1.2 Installation Requirements

This book does not cover the installation details, that is, a graphical point by point direction of the installation procedure for Linux. This is covered numerous times by virtually all of the books on the market. Appendix A does provide an outline to answering the basic questions of installing the Fedora operating system if such is desired. Later versions of Red Hat are fairly equivalent in installation, although there may be a few minor differences. Installation is fairly straight forward.

In an earlier version of Red Hat, several options were provided that are no longer available in the advanced versions, specifically whether to use the **shadow** file and if the operation is to be in either Command Line Interface (CLI) or X Windows mode. Because it is used as an exercise in one of the labs, it is not to be created during the installation process. This is not a normal procedure – only necessary to support this learning process. During the normal installation, one should create the shadow file since it provides considerably enhanced security. The newer Fedora Core distributions do not include this option, the shadow file is automatically installed without an option.

Additionally, initially the user will log on as the administrator, which again is not a recommended procedure. We do this to learn the basic administrative tasks. This procedure is later modified to log in as a normal user and then switch to the administrator.

Finally, the system should be set to boot up under the CLI Text mode. Although one can easily use the GUI mode, the vast majority of commands used to learn Linux are from the CLI mode. If one insists on using the GUI, the XTerminal will be used to issue the commands. Fedora Core automatically sets the user to boot under the GUI mode, so the user must either use the XTerminal, or modify the system to boot to the CLI mode (this is covered).

1.1.3 System Requirements

Due to the lowering cost of hardware for the personal computer, the cost of a decent system is reasonable. This does not imply that the latest hardware is necessary. In fact Linux can be set up on a very minimum system – even may be set up to run off a floppy disk. What we are doing here is to learn the commands and make a system available for further use at a later date. Reference to Knoppix Linux (utilizing the Debian Linux Operating System) is

made in Appendix C. Knoppix is a excellent system for booting on to a Microsoft™ (MS) Windows system without having to do an installation – great for working from at home and still maintaining your base MS system.

1.1.3.1 Recommended Hardware

The following hardware is recommended as a minimum system. A slower system will will perform very well as a learning system, or a learning server, but if you wish to have minimum productivity, the following is a minimum. Naturally, a faster system will make you feel better.

- Processor: Pentium 266 MHz minimum, 400 MHz or better more desirable
- Memory: 64 Mbytes, 256 Mbytes is really recommended, and more is better
- Hard Drive: 10 Gbytes (suggested minimum), for Fedora Core
- Video Card: Make sure it works with Linux
- NIC Card: Plug-n-Play Ethernet 10 / 100 Mbps card
- Sound Card: Sound Blaster compatible
- Modem: I prefer external for the learning process, otherwise internal is OK
Do not use a “WIN-Modem” – Linux does not natively support it! (Linux now supports it, but it takes extra effort.)

1.1.3.2 Justification for Recommendation

The reason for these recommendations is for present and future use. These are noted as follows:

1. Although a 386 / 486 computer will support Linux, it is by itself not Plug-n-Play compatible. Linux supports the detection of most Plug-n-Play hardware when a Pentium system is utilized. Hence to make it easier during the installation, a Pentium system is recommended. The faster the system, the more you will enjoy X Windows later. Additionally, recall that a 386 does not support multi-users operation, as it will not support process threading.
2. Only 32 Mbytes of memory is really required, but due to the cost of memory and better systems, a greater amount naturally makes the system operate faster. It is recommended that you have at least 64 Mbytes. Today, having a system with 256 Mbytes is reasonable. 32 Mbytes will support the CLI mode, but running a GUI (X Windows) will run very sluggish at best. It is recommended that the minimum memory on a system that is to run a GUI interface be at least 64 Mbytes, preferably 128 Mbytes. Today, to have a system with 512 or more Mbytes is cost effective.
3. When installing Red Hat, setting the system to install “everything” insures that all file dependents are installed for all future desires. Additionally, in a learning curve, one does not need to know if a specific command will or will not be installed. This is especially true when attempting to install user applications. For an-everything installation, Red Hat 7.2 requires slightly less than 3 GBytes. There are some applications, such as pico, linuxconf, and

chkconfig that are not necessarily installed unless you do an “everything”. Later versions of Red Hat require considerably more Hard Drive space and a longer time to install. Typical installation time for a 400 MHz system is approximately 1 hour.

4. The most difficult part of installing Linux, when using the X Windows GUI, is the video card. The latest and greatest card most likely will not work. If you are into downloading drivers and going through the configuration setup, then go for it. Otherwise make sure the video card you are using is on the Linux hardware acceptance list for your preferred vendor.
5. At this time, Linux is best in the server world. This requires network connection and learning the basics of setting up an IP address. It is not the intent to provide an in depth understanding of an IP address, but one does need to understand the difference between a domain address and a local host address (this is explained in a simple way in the Appendixes). When setting up Linux as a server, connection is necessary to other hosts. Many companies are starting to produce more workstation user applications, so the use of Linux will grow more and more in the desktop. Chapter 8 provides a greater in-depth understanding of the network addressing.
6. We all enjoy a little music. Make sure the sound card is acceptable. Most will be if they are not the latest and greatest.
7. Connecting to the Internet is one of the great joys of using the computer. External modems with blinking lights allow the user to learn a lot more with respect to what is happening (I am a technical person and enjoy doing things like that). An internal modem will do just as well if Linux detects it. If you select an internal modem, make sure it is not a “WIN-Modem”. The problem with a WIN-Modem is that it is not a stand-alone modem, rather it depends upon the system processor to perform various functions to make it operational. Although drivers are now starting to be available for the WIN-Modem, it is best to avoid them unless one is willing to work with the units to make them operational.

These are a few of the requirements and recommendations for using this book. Enjoy the process of learning the basic commands necessary to navigate through the system and setting up your system configuration. It is not necessary for the user to follow this book chapter by chapter, as it is oriented to allow a user to either install a new system or work from an existing one. Please keep in mind that this is just an initial learning – you are required to perform more investigation on your part to learn more. This is only the start – go for more!

1.1.4 Text Conventions

The notes and labs presented here are set up with the following conventions:

- | | |
|------------------|--|
| Bold Text | This is what the user keys into the computer. |
| | Bold text may also be used within a sentence to highlight a fact or path/file. |

Host Response	This is the response from the computer to a user command.
{Input-Response}	This is a variable that you need to input, such as your user name. When typing in your response, do not type the curly brackets.
®	This symbol is sometimes used in the document to specify the ENTER key. It is used initially, but not later on. Just remember, you must hit the ENTER key to make the command work.

1.1.5 Wild Card Characters

Sometimes we desire to search for a file but may not know the exact spelling. The following characters are considered “wildcards”.

*	Accepts all following characters or preceding characters
?	Accepts only one character at the specified location
[. . .]	Accept only one of the characters within the brackets

For example in the **/etc** directory, there is the file **inittab**; we might search for this file by typing

```
ls init*
```

And we would get:

```
inittab
```

The “*****” indicates to take any characters to complete the command.

1.1.6 File Name Completion

We are all tired of typing file names, so we have a quick means to improve the process – and to minimize typing errors. If you type several characters of the file and hit the TAB key, Linux will attempt to complete the typing. Linux will complete as many of the letters as it can up to ambiguity, at which time it will stop and beep. Hitting the **TAB Key** again will present all of the options when ambiguity exists. Typing in enough characters to get past the ambiguity and hitting the TAB again will then complete the file name. This is great to insure that you do not improperly key in a filename. One might think of Unix / Linux as being “intelligently lazy”.

1.1.7 Megacharacters

When typing in commands or file names, we often like to fine short cuts. This is especially true if we have a name that is especially long and / or hard to remember. Sometimes we want the system to perform multiple simultaneous tasks.

Several short cut characters have been set up to perform special tasks – or have a special meaning. These are known as the “mega-characters”.

!	Exclamation Mark – It is too hard to say, so we call it the “ bang ”. Commonly used to negate a condition.
&	Ampersand – Used to run a task in the background so we can open another task.

- >, < Greater / Less Than – Takes the information from the greater side and directs it to the lesser side.
- >> Double Greater Than – Takes the information from the greater side and appends it to the lesser item.
- !! Double Bang – Search and complete the command.
- TAB** Name Completion – Completes the typing of a filename when there is no conflict. If indeterminate, there is no response, but hitting the TAB key second time will create a list of options.
- # Hash Mark – Comment – Most configuration files allow for user comment lines. If the first character is a “#”, then the remainder of the line is a comment or remark and is ignored by the OS.
- ; Semi-colon – Comment – Most configuration files allow for configuration code options that may be enabled if necessary, but are commented out and ignored.
- | Pipe – Used to direct the output of the first command into the second command. (This key is above the back-slash “\”.)
- ~ Tilde – Is used as a substitute for the user's home directory. If logged in as the administrator, it represents “/root”, if logged in as a regular user, it represents “/home/username”.

1.1.8 Terminating an Application

All applications have a way to terminate. Very easy if you know how to do it. There are several options – one has to know which one to use.

Most will terminate with a “q” at the application prompt. Some may require a “bye”, “exit”, or “quit”. With one of these, you should be able to return to the system Command Line Prompt. Applications that take input from the keyboard typically are terminated by keying in a CTRL-D (^D) on a new line. Unfortunately, there is not a consistency between programs.

When this fails, we can resort to another means – **CTRL C (^C)**. This should terminate the application. There are a few situations where this may not work, but that will be covered in more detail in a lab.

1.1.9 System Lockup

Every once in a while, you do something, and the system hangs – you are stuck in an application, not knowing how to exit, or everything quits working.

As noted above, you should be able to enter **CTRL – C** and you should be able to exit the program. Remember to try quitting the program first.

And then comes the time when all of a sudden everything quits, the keyboard appears locked up – nothing seems to work! Chances are that are that you may have accidentally pressed the **CTRL – S** key. CTRL-S is the command to “Pause” the system keyboard input – everything stops. Actually, the computer is still operating, it is just ignoring the keyboard. Well, almost everything, the keyboard is monitored for one specific input, **CTRL – Q**, which is the resume command. All of the commands that you typed in during the “pause” stage were in fact stored in memory. They are now displayed at the prompt.

1.1.10 Power of the Administrator ROOT

This is a very important issue. As the **root** administrator of a system, you have total power! Whatever you do, that is what is done. This means you can go anywhere, do anything, change whatever you want, and delete anything – or everything. There are very few checks to keep you from doing something – especially deleting. Approach your system with appropriate responsibilities.

You will learn early on that the word “root” has four different meanings. You must learn to listen to what is being said in order to fully understand the meaning. Learn what each means and how to interpret what is being said.

1.1.11 Recommendation

It is recommended that the new user start a quick reference command dictionary. It might be made up something like the following:

ls	list	lists files in a directory
ls -a	list all	lists all files (including hidden) in a directory

This could be provided for you, but the process of writing it down yourself is an additional learning process.

1.1.12 Corrections and Updates

This book and lab documents are under continual improvement. If you find an error (very possible) or have a comment, please send an email to dennis@dearroz.net. The latest updates are maintained on the web site www.dearroz.com.

1.2 General Organization

The book is divided into multiple sections, where each section is centered around a given topic. Sections include:

- General Background and Requirements
- Basic Operational Commands
- Administrator Utilities
- User Applications
- Server Setup
- System Initialization
- System Security
- Hardware Configuration
- Office Software
- Programming and Scripting
- System Upgrade
- Applications and Other Things

1.3 Why use Unix / Linux

Why should one desire to use a Unix / Linux system rather than a Microsoft Windows system? The answer lies with another question – What do you want to do?

Today, most business workstation applications are written to be run on a MS Windows OS platform. Although there is a slow migration to other platforms – namely Macintosh and Unix / Linux, there is a long way to go.

What are the strengths of the other OS's?

The Apple / Macintosh computers maintain their history based on the Motorola 68 series processor, although this has now been modified, and they are now using the Intel processor. The Motorola processor might be considered by some to be the first "RISC" processor – using larger code words and being hardware I/O oriented. This made them much more powerful at manipulating information for display and sound. And thus we saw the first production GUI interfaces (the first true GUI and sound was on the Amiga). Thus if you wish to generate multimedia applications, then a Macintosh is probably the optimum system to utilize. Note that Apple has now elected to utilize the Intel processor, so we may see additional changes.

Microsoft elected to support the Intel processor, whose strengths lay in numeric processing – which is a strength required by business. This support was developed for the word processor, spreadsheet, and database applications. If all you are worried about is the business workstation, and you have users that only know of Bill Gates, then you need to stick with the Microsoft base.

Unix was kind of in between. It was developed on early processors, but held properties designed for both data manipulation and hardware I/O. The difference was that Unix was traditionally a "scientific" or background business systems. It might be considered a miniature main frame – able to perform multiple tasks and support multiple users but on much a smaller scale.

Hence we have seen that Unix was designed to process the manipulation of information – taking raw data and formatting it for a user or other applications. As examples of where Unix demonstrates its strength:

- Telephone Network Manipulating telephone number to route a call to its destination. Collect user call information, extracting usage and creating a billing statement.
- Internet Network Manipulating data packets to extract address information, route packets, and verification of data integrity.
- Scientific Data Manipulate data to format it into standard blocks, extracting needed information, and make it user readable.

Some of the better "well known" reasons for using Unix or Linux are:

- Preemptive Multitasking
All processes run independently of one another, each in their own protected memory area.
- Multi-Processor Design
Linux is designed for Symmetric Multi-Processing (SMP) operation since the 2.1 Kernel.
- Multi-User
Unix and Linux have always supported multiple users logged onto the same host. Each user has appropriately designated rights, and can even work in a remote GUI mode (X-Windows) if the appropriate software has been installed and configured.
- Multi-Platform

Linux has been ported to more platform architectures than any other operating system. These include (but not limited to): Intel x86, AMD x86, MIPS, PA-RISC, Alpha, Sun Sparc, Apple Power PC, Motorola M68, and others.

➤ Unix Based System

Linux is a Unix clone, based on the POSIX-1300.1 standard. ¹

➤ Network Functionality

The Linux architecture supports an extensive number of network protocols and Internet functions.

➤ Open Source

The Kernel software code is freely available and may be modified under the GNU Public License. Programmers around the world have supported this concept, having put in an extraordinary amount of time and effort, without compensation, to assist in the improvement or enhancement of the Kernel's operation.²

➤ Efficient Implementation

The Linux Kernel supports a “well-structured” implementation for the support of network functionality and adaptability to meet the specifications of the Internet Engineering Task Force (IETF), IEE, and ISO. Configuration of the Kernel may be individually configured and optimized for various specific functions.

➤ Internet Adaption

Linux has been designed for the future Internet, supporting IPv6.

➤ Cost of Linux

Best of all, most distributions of Linux may be downloaded from the Internet for free. You can freely copy and distribute the distributions without worrying about going to jail.

Until the late 1980s or so, Unix was not very interested in a GUI format, but that has now changed. But if you look at the capabilities of Unix / Linux using a windows GUI, one still has to revert to a terminal mode in order to utilize all of the applications that have been developed.

Today, the Unix / Linux base is that of a server and router. All of the Internet Network is based on Unix. All of the routers, Frame Relay, ATM, and Label switches use a modified version of Unix – modified for the specific application and commands needed to configure the system for the functional application.

In the business world, Unix and more specifically Linux are gaining in the applications available for the business user. There are applications available for nearly all of the major needs of the business user, although they may not have the flash and flexibility of those developed for the Microsoft platform.

As a background server in the business environment, Unix / Linux is a much stronger platform. The foundation of the Internet world, DNS, DHCP, Web, Mail, and FTP are built on the reliability of Unix / Linux. Security of the Unix / Linux system is much higher than that of the Microsoft system.

1 Portable Operating System Interface based on Unix – POSIX, which defines the minimum standards for a Unix like-system.

2 For a better understanding of what makes up those that put forth the effort to do Open Source software, read the book, **The Cathedral and the Bazaar**, by Eric Raymond.

Again, we ask the question, what do you want to do? If you are concerned only with the business user, then you should base your system on Microsoft, but if you are in need of reliability and security, server based system, then the Unix / Linux platform is the more optimum.

After all this discussion, do not completely put the Unix / Linux platform down in the business world, as it is an upcoming base. Things are improving.

1.3.1 Which Version Should I Use?

Originally, the initial instruction was performed using Red Hat Version 7, rather than a later version. This has been upgraded to Fedora Core 3 or even a later version (they change faster than one keep up on writing), but even now this has been outdated. Which version that one wishes to utilize is not a real issue when first learning how to use Linux. Very few commands (if any) are different between various distributions, and some applications are installed in different directories, but not enough to hinder one from learning the basics. The main issue to be considered is what what one wants to achieve. If one is looking at a system to learn on, any distribution will do, if one wants to have a production system, then a more detailed review is required to insure that the optimum distribution is utilized.

1.3.1.1 Capabilities

With respect to an instructional point of view on how to use Linux, the later versions of Linux do not provide any additional features with respect to learning the basics. For additional features and improved performance, security, and enhanced GUI functionality, then one should choose to use a later version. One does need to be aware that later versions may have changes in the file structure. An important advantage to using a later version of Linux is improved hardware drivers and enhanced support for newer hardware.

1.3.1.2 Installation Time

After lots of practice, full (install everything) of Red Hat 7.2 can be installed in approximately 45 minutes; Fedora Core 3 will require about 3 hours. For courses that allow students to install, a two-hour class period would be difficult for an advanced version for most students. A minimal installation with the support of a lab YUM repository can provide an excellent installation base and the ability to easily install additional packages.

1.4 Linux Standardization

Unix was originally developed by AT&T around 1969 as an operating system to support their digital switch. It was designed as a “multi-user”, “multi-tasking” OS. Over the years, it was released to other educational and corporate users, who made variations to suit their particular requirements. Today we have several different “flavors” of the Unix OS, each operating on its specific hardware platform – and with small variations in where it maintains its information.

In 1991, Linus Torvalds, as a graduate student in Finland decided that the “Unix clone” he was using was not operating like a true Unix OS, so he decided

to create a new OS that looked and acted like Unix on the surface, but is in fact a whole new OS.

Some of the benefits that Torvalds maintained was the concept that Linux would be able to run programs written for Unix after they had been recompiled for the specific hardware. Thus Linux shares a compatibility with Unix – but it is different.

Torvalds used another concept that was promoted by Richard Stallman (originator of the idea of “Open Source” and free software). With this, Torvalds maintained the copyright, but gave the source code out to anyone who wanted it, so that they could make improvements and fixes. He also allowed commercial ventures to enhance the core, called the Kernel, by external features and pre-compiled applications for Linux. This concept is now known as **copy-left**.

One of the major enhancements that vendors made was to develop installation packages for installing Linux on a hardware system. Over the years, significant improvements have been made that provided additional hardware drivers and expanded the devices that are supported. But problems arose because there were no specific rules which specified how the various applications were to be installed—one of the original problems of Unix.

Up through Red Hat version 6, and the equivalent versions for the other vendors, there were different approaches to where information should be placed. But at the same time, the action of the almighty committee was also taking place, and the result was that some information was shifted around. Due to the committee’s actions, in June 2001 the committee released Linux Standard Base 1.0. In addition to the Linux Standard Base, another standard, FHS 2.2 also specifies a directory file structure, which the various vendors of Linux are approaching adherence to. Starting with Red Hat version 7. The way information is stored may be different from previous releases, and hence we may need to alter previous documentation. Later versions, and different distributions, are improving in terms of standardization, but there are still some small differences.

Where this leads to is an improvement in user operation, application installation, problems of trouble shooting, and overall piece of mind. It could be observed that Linux has learned what Microsoft established back with its Windows OS – establish a standard for the other vendors to write their software for – and you will obtain improved sales.

This also might mean that Linux will take a different design path from Unix. Although it will remain highly compatible, it will develop its own unique style. Where it will have its major difference will forever make it stand out against all other OS’s, Linux is Open Source for everyone to provide improvements and to solve bugs, whereas all other OS’s are proprietary.

Long rule to believe in – Open Source.

1.5 Multi-Architecture Support

The Linux Kernel is written in the C Language, and is easily ported to many other systems. It has been asked why it was not written in C++, and the answer is that the extensions available to the C++ language are not needed for the

Kernel. This makes the porting of the Kernel to various processors very easy, as only the base language commands are required.

The Linux Kernel was originally written to support the Intel 386 processor. Today, the compiled Kernel used by Red Hat, Suse, and many others are still compiled to support the 386, thus are able to be installed on nearly all Intel and compatible products, including the now Pentium processors.

Mandrake originally used the Red Hat source code, and compiled it to support the later Pentium processors, thus gaining improved performance by using the extended commands that were available. Of course, this made the Mandrake version incompatible with the 386 and 486 processors.

Linux has also been ported to other processors, including the Sun Sparc processor, SGI processor, Motorola 68-series processors, and now even the IBM mainframe. Yellow Dog Linux was designed to run on the Apple Macintosh processor.

1.6 Linux Distributions

As we have noted, the Kernel for Linux was originally developed by Linus Torvalds. Today, there are thousands of individuals around the world that have contributed, but still under the watchful eye of Linus. The question always arises – why are there so many different vendors of Linux, and what are the differences?

First, all of the different versions of Linux contain the same basic Kernel. From that point, each vendor adds their own special flavor in terms of installation, looks, and operational enhancements. To start, there was quite a bit of divergence between the vendors, but today the fundamentals are generally the same.

Some vendors will be very conservative by providing a solid, well-supported packages, while others are supported by experimental groups and are always attempting to support the latest hardware and user needs. Depending upon what you want to do – solid base or bleeding edge, will determine which vendor you might desire to utilize.

One consideration is to ask what one wishes to accomplish. Is the system to be used for the desktop, server base, productivity, learning, or some other requirement; then ask which distribution provides the required features. Live CD versions is an excellent start as an introduction, but if one wishes to learn the full power of a Unix / Linux system, one must learn the power of the command line, and the utilities available. The full strength resides in the command line to manipulate data.

1.6.1 Linux Versions

Several vendors are reviewed here. The following list is not complete, nor should it be. There are over 250 distributions, so one has lots to think about and chose from. Hopefully there will always be new development going on. Check out www.distrowatch.com, www.linux.org, and www.linuxiso.org for the latest distributions.

Red Hat Linuxwww.redhat.com

Probably the most widely distributed version. A very sound product that is supported on a corporate basis. Quite easy to use as a desktop system and excellent server support.

Fedora Corewww.fedora.org

Red Hat elected to discontinue the consumer version of their distribution, and focus on the enterprise (commercial) version. In this move, they decided to provide corporate support to a user group called **Fedora Core**, which had been working on various applications. The Fedora distribution now is a little more on the advanced side, but still providing a very stable version. Both Red Hat and Fedora base their distributions on using totally free software with the package. Fedora is not sold in retail stores, thus must be either downloaded or obtained from another source.

SUSEwww.suse.com

Probably the second most popular distribution. It is supported in Germany (and now owned by Novell). A very sound product that provides an excellent base for corporate user and server support. Suse provides some commercial products with their distribution.

Open Linux / Caldera Linuxwww.sco.com

A distribution that is aimed at the business community. Relatively easy to use as a desktop system. Caldera recently changed its corporate name to SCO (formerly Santa Clara Operations) after having purchased SCO.

United Linuxwww.unitedlinux.com

A collaboration of multiple vendors to establish a base foundation by which new applications may be developed given a standard design, lead by SUSE and SCO. This distribution has kind of degenerated due to the failing of the alliance.

Turbo Linuxwww.turbolinux.com

A version whose support comes from Asia and intended to support the server market.

Mandrake / Mandrivawww.mandrakelinux.comwww.mandriva.com

Originally Mandrake was a version of Red Hat that was optimized for the Pentium processor. It has now taken its own flavor. It is very user friendly. Mandrake is officially no longer available. It has subsequently been converted to Mandriva Linux, which is intended to be a very user friendly version.

Stampedewww.stamped.org

A user supported version optimized for speed.

Debianwww.debian.org

A user supported version aimed at new hardware and software development.

Yellow Dog Linuxwww.yellowdoglinux.com

A version of Linux that was developed specifically to support the Macintosh PowerPC processor.

Slackwarewww.slackware.com

A user-supported version, generally considered to be on the “bleeding edge” of hardware and software support. Hardware drivers are typically developed on Slackware, and then released for others to incorporate into their products.

Linux Router Projectwww.linuxrouter.org

A user supported version, designed as a minimal version to implement a router function only. Versions have been developed that can be supported on a floppy disk.

Knoppix Linuxwww.knoppix.org

A version of Debian Linux designed to boot and operate from a CD. Excellent for booting a MS Windows system to Linux without having to do an installation to the hard drive. It is capable of reading an MS NTFS file system, but will not write to it, thus protecting the MS Windows system. On rebooting, without the CD, the system will return to the MS Windows OS.

ClarkConnectwww.clarkconnect.org

A version of Red Hat Linux that has been optimized to operate as a router and firewall, supporting all of the server functions (when activated). To minimize operational overhead, X Windows is not supported.

Ubuntuwww.ubuntu.com

Ubuntu is a variation of Debian that allows one to initially operate as a “Live CD”, and later install on one's hard drive, if desired. Because of the data capacity of a single CD, all utilities are not available. It is an excellent desktop system that provides the necessary tools for the common user, but limits utilities that one might wish to utilize.

Aurora Linuxwww.auroralinux.org

A special version of Red Hat Linux that has been optimized to operate on a Sun Sparc processor.

1.6.2 Linux Standardization

As observed above, there is a divergence in the different vendors of Linux. Recently, a forum was established to develop a common base for software development. From the forum came the **Linux Standard Base (LSB)**. Although

many vendors are standing up to the standard, several vendors are not completely supporting it.

A second consortium of vendors was created to establish a standard application development base. The leaders in the consortium are SCO, Suse, Turbolinux, and Conectiva. Together, they developed the United Linux version. Unfortunately, SCO has elected to alienate itself from the rest of the Unix / Linux community, and the outcome of this situation is yet to be determined.

1.7 FLOSS

FLOSS stands for **Free / Libre – Open Source Software**, or the right of the user to choose software that supports Open Non-Proprietary standards and protocols. The terms “Free / Libre” do not imply free of cost, but rather free of proprietary standards. The ramification of this manifest itself may be viewed in two parts.

First, by storing data as an **open standard**, the user may import / export their data to any vendor application that they chose. This is very important because the user has freedom of choice rather than being forced into a single vendor.

Second, when a vendor encodes the user’s data in a proprietary format, they force the user to exclusively use their application – and hence the vendor “owns” the user’s data. In the (near) future, some major vendors may design their applications to force the user to pay an annual fee to continue to use the application – thus locking the user’s data and making it inaccessible if the fee is not paid!

Linux, as an open source concept, totally encompasses the open source concept. All Open Source software applications also encompass the usage of open standards for the user’s data.

1.8 Commands Used in this Chapter

No commands were introduced in this chapter.

1.9 Chapter Review Questions

1. What is the recommended minimal processor speed?
 - a. 100 MHz
 - b. 266 MHz
 - c. 350 MHz
 - d. 500 MHz
2. What type of modem should not be utilized?
 - a. Internal
 - b. External
 - c. Winmodem
 - d. Voice
3. The “*” character represents what?
 - a. Wildcard representing one character
 - b. Wildcard representing a set of characters
 - c. Wildcard representing any set of characters
 - d. Just an ASCII character

4. The “[...]” represents what?
 - a. Wildcard representing one character
 - b. Wildcard representing a set of characters
 - c. Wildcard representing any set of characters
 - d. Just an ASCII character
5. What character can be used to complete a file name?
 - a. TAB
 - b. ESCape
 - c. Control
 - d. ALT
6. What character is used for user comments?
 - a. @
 - b. #
 - c. &
 - d. *
7. What character is known as the bang?
 - a. !
 - b. #
 - c. |
 - d. %
8. What character directs the output of one command into the second?
 - a. >
 - b. &
 - c. !
 - d. |
9. What key stroke is used to Pause the system output?
 - a. ^ A
 - b. ^ S
 - c. ^ Q
 - d. ^ Z
10. What is the base OS of routers and switches?
 - a. Unix
 - b. Linux
 - c. Microsoft
 - d. VAX
11. What is the notation for the user's home directory?
 - a. Bang
 - b. Pipe
 - c. Shebang
 - d. Tilde
12. Which distribution of Linux would you prefer to use?
 - a. Aurora
 - b. Fedora Core
 - c. Red Hat
 - d. Suse

(Is there a true real answer?)

Chapter Index

A		United Linux	15
Aurora Linux	16	Yellow Dog Linux	16
C		Distrowatch.com	
Character		URL	14
Name Completion TAB	8	F	
Pipe	8	Fedora Core Linux	15
TAB	8	File	
; Comment	8	/etc/inittab	7
! bang	7	File Name Completion	7
!! Search and Complete	8	H	
& Background Operation	7	Hardware	
# Comment	8	Memory	5
> < Redirect	8	Pentium	5
>> Append	8	Hardware Justification	5
Pipe	8	Hardware Requirements	5
~ Tilde	8	K	
Chosen Linux OS	3	Knoppix Linux	16
Clark Connect Linux	16	L	
CLI	4	Linux Distributions	14
Command Line Interface	4	Linux Router Project	16
Copy-Left	13	Linux Standard Base	16
CTRL		Linux Standardization	12, 16
S 8		Linux Versions	14
D		Linux, Use of	
Debian Linux	16	Cost of	11
Directory		Efficient Implementation	11
/etc	7	Internet Adaption	11
Distribtuion		Multi-Platform	10
Open Linux	15	Multi-Processor Design	10
Distribution		Multi-User	10
Aurora Linux	16	Network Functionality	11
ClarkConnect Linux	16	Open Source	11
Debian Linux	16	Preemptive Multitasking	10
Fedora Core Linux	15	Unix Based System	11
Knoppix Linux	16	Linux.org	
Linux Router Project	16	URL	14
Mandrake	15	Linuxiso.org	
Mandriva Linux	15	FLOSS	17
Red Hat Linux	15	Free Libre Open Source Source	
SCO Linux	15	Software	17
Slackware Linux	16	Open Standards	17
Stampede Linux	15	URL	14
Suse Linux	15	LSB - Linux Standard Base	16
Turbo Linux	15	M	
Ubuntu Linux	16	Mandrake Linux	15

Mandriva Linux	15	Www.clarkconnect.org	16
Megacharacters	7	Www.debian.org	16
Multi-Architecture Support	14	Www.fedora.org	15
N		Www.knoppix.org	16
Name Completion TAB	8	Www.linuxrouter.org	16
P		Www.mandrakelinux.com	15
Pipe	8	Www.mandriva.com	15
POSIX	11	Www.redhat.com	15
Power of the administrator ROOT	9	Www.sco.com	15
R		Www.slackware.com	16
Recommended Hardware	5	Www.stamped.org	15
Hard Drive	5	Www.suse.com	15
Memeory	5	Www.turbolinux.com	15
Modem	5	Www.ubuntu.com	16
NIC Card	5	Www.unitedlinux.com	15
Processor	5	Www.yellowdoglinux.com	16
Sound Card	5	User Dictionary Recommendation	9
Video Card	5	W	
Red Hat Linux	15	Why Red Hat Ver 7	12
S		Why use Unix / Linux	9
SCO Linux	15	Wild Card Characters	7
Shadow file	4	WIN-Modem	6
Slackware Linux	16	X	
Stampede Linux	15	X Windows	4
Suse Linux	15	Y	
System Lockup	8	Yellow Dog Linux	16
System Requirements	4	;	
T		; Comment	8
TAB	8	!	
TAB Key	7	! Bang	7
Terminating an Application	8	!! Search and Complete	8
Termination		&	
Bye	8	& Background Operation	7
^C	8	#	
Text Conventions	6	# Comment	8
Turbo Linux	15	>	
U		> < Redirect	8
Ubuntu Linux	16	>> Append	8
United Linux	15		
URL		Pipe	8
Dearroz.pointclark.net/book.html	9	~	
Distrowatch.com	14	~ Tilde	8
Www.auroralinux.org	16		