

LINUX STARTER KIT

LINUX PRIMER

MANUAL

By

Graeme Cross

v1.1, 22 April 1999

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

2390 EMAC Way, Carbondale, IL 62901
World Wide Web: www.emacinc.com
(618) 529-4525 FAX: (618) 457-0110

Linux Primer Overview

Paths

The first concept that is essential to understand Linux is that of **paths**. A path is the means by which one indicates a file. There are two kinds of path: absolute and relative. An absolute path is one that indicates a file's location from the 'root directory.' An absolute path for a file in my directory might look like this: /home/lester/xjewel.scores

This means that the file I'm interested in referring to (xjewel.scores) is in my directory (lester/), which is in the directory containing user directories (home/) which is in the root directory (/). Alternatively, if I were in my directory (/home/lester), I could refer to xjewel.scores simply by typing its name, thus: xjewel.scores

This is an example of relative paths. But, from /home, I could refer to the same file by using this path: /lester/xjewel.scores

There are two other 'directories' in each directory. One is . , which indicates the current directory, so ./xjewel.scores is the same as xjewel.scores, and, more importantly, .. refers to the directory in which the current directory is contained, called the *parent directory*. That way, if my current directory was /home/lester/whitesocks, I could call the example file ../xjewel.scores.

While all this may seem somewhat involved and nebulous, paths are central both to using Linux and to programming in HTML. For instance, why don't we start with a bunch of commands that use paths as arguments.

Path Commands

cd - Change directories

cd is one of those commands that's used with alarming frequency. What cd does is to change one's current directory to the directory indicated by the path one gives to cd. For instance: cd will take you to your home directory. cd ~[username] will take you to the directory of the user [username]. cd .. will move one up to the parent directory of the current one. cd / will change the current directory to the root directory. cd . has no effect. The idea expressed is 'change from this directory to this directory.'

mkdir dirname

mkdir is short for make directory. (If you were wondering how these directories came from, mkdir is the answer.) Typing mkdir dirname adds dirname to the current directory. If one calls mkdir with the -p option, thus: mkdir -p

onedir/twodir/dirnamemkdir will make all the non-existent directories as well. If onedir existed, but twodir didn't, mkdir would create twodir in onedir, and then dirname in twodir.

pwd

pwd is an acronym for 'present working directory,' and returns the path of the directory one is in at the moment, viz: whiskey:lester:7:pwd /usr/people/lester

cp source dest

cp is short for copy. The cp command works two different ways depending on what source and dest are.

1. If source and dest are both paths that indicate a file, cp copies the contents of source into dest. Be careful, because the result will be that you have two copies of source, and the information in dest is gone.
2. If source is a file, and dest is a path to a non-existent file, cp will make a copy of a file at dest.
3. If source is a list of files and dest is a directory, cp will make copies of the files with the same names, but in the dest directory.

mv source dest

mv is short for move. Like cp, mv works differently depending on the arguments passed to it. Thus:

1. If source and dest are paths to files, the source is moved into dest, erasing dest and replacing it with source. Note that source will no longer exist under that name, and that the contents of dest will be replaced with those of source.
2. If source is a file, and dest is a path to a non-existent file, mv will change the name of source to dest (changing its path if necessary.) Think of this as moving the file between file folders.
3. If source is a list of files, and dest is a directory, the files in source will be moved into dest, using the same filenames.

ls - list contents of a directory

ls is another command that is used with frightening regularity. Again, ls works differently under different conditions.

1. If typed by itself ('ls'), ls will display a list of the contents of the current directory.

2. If entered with the `-l` option (`ls -l`), `ls` displays a list of all files, with additional information about each, like owner's name, the permissions of the file, modification date, what kind of file it is, and its size. (Some of these ideas will be explained later, others aren't terribly important, and others are assumed to be evident.)
3. The `-a` option (`ls -a`) gets `ls` to display all the contents of a directory, including any file that begins with a `.` (including `.login`, `.` and `...`) Note that `ls -la` displays a long view (i.e. with lots of semi-relevant information) of all the contents of a directory.
4. If you give `ls` a directory (`ls /home/lester`), it will list the contents of that directory. (Adding the `-a` and `-l` options (or `-la`) will do appropriate things to the listing.)
5. Giving `ls` a wildcard expression (one that contains `*` or `?`), it will return a list of all files that match that expression in the current directory. Further, if the expression is a directory, all the contents of that directory that match the expression will be listed. Also, any directories that match the expression will have their contents listed, with the name of the directory as a header.

The discussion of `ls` brings up several topics of confusion which will now be expounded upon:

Permissions, Owners and Wildcards

In Unix, and in Linux, files have *permissions*. Permissions indicate who gets to use the file, and what they can do with it. There are nine permissions that can either be granted or removed: The owner's permission to read the file, write to it, or execute it, the owner's group's (a concept seldom used) permissions to read, write or execute, and the global permissions do the same. (Global permissions apply to everyone, except the owner and his group.)

Permissions are usually written one of two ways. The first way is used by `ls` to describe permissions. It consists of ten characters, the first indicates the type of file (`-` for regular files, `d` for directories, and `l` for symbolic links (don't worry too much about links)), the next three indicate the user's permissions, the next three are group, and the next three are global permissions. In this context, permissions are indicated either by a `-` or a letter, an `r` in the read position, a `w` in the write position, and an `x` in the execute position. Thus an example permission string: `-r-xrw--x`.

The second way of indicating permissions is the one used to send a permission set to various commands, notably `chmod` (which we'll get to soon). This method uses three digits, where the first digit is the owner's permissions, the second is his group's, and the last is the world's. Two determine what a digit is, add the values of the permissions: 4 is read, 2 is write, and 1 is execute. Therefore, the permission of the previous example in this form is 561. This form is used whenever a permission set is sent to a command.

Another important concept are those of wildcards. Wildcards save time, effort, and allow one to limit the breadth of one's search. Essentially, a wildcard expression is a regular path, but with *'s and ?'s in place of letters. A command to which a wildcard expression is passed will be applied to all paths which match the expression. An * matches any number of characters, with any character in it's place, while a ? matches any one character. pl* matches ply, plague, place, plateau, etc. pl? matches pla, plb, plc etc.

Equally important are regular expressions. Regular expressions (or regexs) are like wildcards, but more powerful. There are three basic elements of regexs: ., *, and []. '.' allows for matching of any number of the character that follows it. Thus foo.b matches foo, foob, foobb, foobbb, etc. * matches any single character. foob*r matches foobar, foobbr, foobcr, etc. [] contain ranges to be matched. For example [a-z] matches any lower case letter, [a-z, A-Z] matches any letter, [0-9] matches any number. foo[0-9] matches foo0, foo1, foo2, foo3, foo4, foo5, foo6, foo7, foo8, and foo9. These elements can be combined: .* will match any number of any character, .[A-Z] will match any number of upper case letters.

Permissions Commands

chmod permission file -change permissions

chmod is the means by which the permissions of files changes. The permission argument is a numerical representation permission, and the file is the file whose permissions you want changed.

chown user file

This command changes the owner of a file. Usually the person who created the file is it's owner, but sometimes you have to make a file for someone else, and then 'give' it to them. This is accomplished via chown.

Generally Really Useful Commands

man commandname

man is short for manual. It produces (usually) a helpful, if somewhat cryptic descriptions of what various commands do. man accepts the name of the command you need information about as its argument. If one's question is of the type "What command does ... ", there are two options: 1) Enter a related command. man pages often have See Also sections at the bottom, or 2) man -k word, where the word relates to what you want done. The result will be a list of pages that contain that word, from which you can select likely candidates. man is exceptionally useful for increasing one's facility with Linux. As an exercise, use man to get more information about the command is this tutorial.

passwd

Typing passwd will begin a brief dialog that will allow you to change your passwd, something that should be done with relative frequency. Keep in mind that a password should not be any actual English word, and that the use of non-letter characters (i.e. numbers and punctuation) is encouraged, as this will make the password more difficult to guess. Contrariwise, you need to be able to remember your password, because: a) you shouldn't write it down and b) if you forget it, you'll need to get the sysadmin to reset your password.

Linux Primer

Author: Graeme Cross, <gcross@netspace.net.au>

v1.1, 22 April 1999

This primer is an introduction for people new to both Linux and Unix, explaining the basic commands needed to use Linux.

1. Preamble
 - 1.1 Copyright
 - 1.2 Credits
 - 1.3 Assumptions
 - 1.4 What this primer does not cover
2. What is Linux?
3. Linux: multi-user & multi-tasking
4. Logging in, logging out
5. Directories
6. Moving around
7. Some basic commands
8. Getting help about commands
9. Shutting down a Linux computer
10. Where to from here?

1. Preamble

1.1 Copyright

Copyright © Graeme Cross, 1999.

You may redistribute and/or modify this document as long as you comply with the terms of the GNU General Public Licence, version 2 or later.

1.2 Credits

My thanks to the following people for their additions, comments and corrections: Tim Abbott, Mike Battersby, Geoffrey Bennett, Peter Moulder.

1.3 Assumptions

I am assuming that you have installed a recent distribution of Linux, such as Red Hat (<http://www.redhat.com/>) or Debian (<http://www.debian.org/>).

I also assume that you have basic computer skills: using a keyboard and mouse, etc.

1.4 What this primer does not cover

This primer does not cover installing Linux, using X-Windows, system administration, compiling programs or using networking tools.

2. What is Linux?

Linux is a computer operating system, like Unix, Windows 95/NT, MacOS, and OS/2. It was originally developed for Intel x86 processors, but now runs on Motorola 68k, Digital Alpha, Sun Sparc, MIPS, Motorola PowerPC and a number of other machines. Linux is source compatible with Unix (this means that programs which are written for Unix can be recompiled with no changes to run under Linux) and runs nearly all of the same software as commercial Unix versions, but it is not descended from the non-free Unix sources.

It supports a wide range of software, including graphical user interfaces, text editors, word processors, databases, web servers, and an extensive collection of networking software (including PPP and ISDN). Many people have run benchmarks on Pentium Linux systems and found them comparable with mid-range workstations from Sun and Digital.

Linux (often pronounced with a short "i" and with the first syllable stressed -- LIH-nucks) is available over the Internet from hundreds of ftp sites, and from various vendors on CD-ROM.

The Linux kernel is covered by the GNU General Public License (GPL), and is usually bundled with various programs that comprise a working UNIX operating system. These software bundles are called "distributions" and come in many sizes and arrangements.

Linux is being used today by millions of people around the world. It is used for software development, networking (intra-office and Internet), and as an end-user platform. Linux has become a cost-effective alternative to expensive UNIX and Windows NT systems.

(Adapted from <http://www.linuxresources.com/what.html>).

3. Linux: multi-user & multi-tasking

Linux is a multi-user and multi-tasking operating system. This means that the computer can be used by more than one user simultaneously, and the computer can also run multiple programs simultaneously.

To co-ordinate this, each user has a user name and a corresponding password. There is also a number of special users, the most important being the **root** user, who is the system administrator with complete access to the computer. The root user account is usually only used for system administration tasks, such as adding or removing user accounts.

4. Logging in, logging out

To gain access to your account, enter your user name at the login prompt, which will look something like this:

```
Red Hat Linux release 4.2 (Biltmore)
Kernel 2.0.36 on an i586
```

```
guava login:
```

In this example, the computer is running Red Hat Linux version 4.2, and the word "guava" is the hostname of the computer.

As soon as you have entered your user name, you will be prompted for your password, as shown below:

```
Red Hat Linux release 4.2 (Biltmore)
Kernel 2.0.36 on an i586
```

```
guava login: graeme
Password:
```

For security reasons, the password is not displayed on the screen as it is entered.

The username and password, like most parts of Linux, are case-sensitive, so make sure you enter your user name correctly: GRAEME and graeme are two different user names! This also applies to passwords.

Once the username and password are entered correctly, you will be logged into your account, and have full access to a shell, such as bash. The shell accepts user commands and displays information, and will be familiar to anyone who has used DOS (but is far more powerful!).

Because Linux is a multi-user system, you can be logged into the computer on multiple virtual consoles: pressing Alt-F1 switches you to the first virtual console, Alt-F2 switches you to the second virtual console, etc.

This demonstrates one aspect of multi-tasking: for example, you can read your e-mail in one console, edit files in another console, while playing Quake in a third console!

To exit the shell, type exit.

5. Directories

Linux keeps files in a single hierarchical directory structure, that (behind the scenes) is made up of a combination of hard drives, CD-ROMs, floppy disks, ZIP drives, etc.

The typical Linux filesystem looks like this:

```
/
|- bin          Essential programs (or binaries)
|- boot        Startup (or boot) information
|- dev         Devices
|- etc         Configuration files
|- home        Users' home directories
|  |- user1
|  |- user2
|  |- user3
|
|- lib         System libraries and other various program files
|- lost+found  Files recovered after filesystem checks
|- mnt         Mount point for removable disks
|  |- cdrom    (An alternative is /cdrom and /floppy)
|  |- floppy
|
|- proc        A special directory with file-based system information
|- root        Home directory for the root user
|- sbin        Essential system programs
|- tmp         Temporary work space
|- usr
|  |- X11R6    X-Windows files
|  |  |- bin   X-Windows programs
|  |  |- include
|  |  |- lib
|  |  |- man
|  |  |- share
|  |
|  |- bin     General programs
|  |- dict   Dictionary
|  |- doc    Documentation
|  |
|  |- etc    Additional configuration files
|  |- include Include files for the C preprocessor
|  |- info   GNU info files
|  |- lib    Additional libraries
|  |- local  Files generally not supplied by the distribution
|  |  |- bin
|  |  |- doc
|  |  |- etc
|  |  |- include
|  |  |- lib
|  |  |- man
|  |  |- sbin
|  |  |- share
|  |
|  |- man    Man (ie. user manual) pages
|  |- share  Files that can be shared between computers of different architectures
|  |- sbin  Additional system programs
```

- | |- src Source code directories (eg. /usr/src/linux/)
- |
- | |- var
- |- lock Lock files
- |- log Log files (system messages, error logs, etc.)
- |- spool Spool files
- |- mail Users' mailbox files

6. Moving around

pwd

Prints the current working directory

cd name

Change to a new directory

cd

Return to your home directory

There are three special directories: / refers to the root directory, which is the top of the directory tree. A single period (.) refers to the current directory, while two periods (..) refers to the directory above the current one.

7. Some basic commands

ls

List the contents of the current directory

ls name

List the contents of the named directory (eg. ls /etc)

ls -l

List the directory contents in "long" format, including file permissions, ownership details and file size

ls -a

List all the files in a directory, including files that start with a '.'

cat filename

Display the contents of a file to the screen

more filename

Display a file on the screen, with scrolling and search facilities. less is an enhanced version of more

cp source destination

Copy the source file to the destination (eg. cp /etc/passwd . copies the password file to the current directory)

mv source destination

Move (or rename) the source file to the destination, which can be a directory or another file

rm filename

Remove (ie. delete) a file - use with caution!

mkdir name

Make a directory

rmdir name

Remove an empty directory

locate keyword

Displays a list of files that contain the keyword in their filename.

su

Temporarily become the superuser - useful for system administration tasks (can also be used to become another user - see the man page for further information)

passwd

Change the password used to access your user account

who

Display a list of users currently logged onto this computer

8. Getting help about commands

Linux has two on-line documentation systems: man and info. There is a man page for every command on the system, while info is normally used to document applications from the Free Software Foundation, such as EMACS (a text editor) and gcc (a C compiler for programming).

man command

Display a manual page for the specified command. The "man" page provides a brief explanation about the command, possible options and switches and detailed information about using the command

apropos keyword

Displays a list of man pages that contain information about the supplied keyword

9. Shutting down a Linux computer

A Linux computer needs to be turned off correctly, to ensure that files are correctly written to the hard drive, and that programs are closed properly.

Reboot the computer

`/sbin/shutdown -r now`

Shutdown (halt) the computer

`/sbin/shutdown -h now`

Reboot the computer

Press Control-Alt-Delete

When the shutdown command starts, it will display information as each step of the shutdown sequence is completed, until the computer finally displays:

The system is halted

in which case you can turn off the computer, or

Please stand by while rebooting the system...

for a computer that is being rebooted.

10. Where to from here?

The Linux Documentation Project (<http://metalab.unc.edu/LDP/>) has an extensive collection of links and documents, including the Linux HOWTOs (<http://metalab.unc.edu/LDP/HOWTO/>), which provide information on a number of Linux issues, such as configuring networks, Linux-compatible hardware, and setting up a PPP link to the Internet.

MS-DOS and MS Windows users should have a look at the "From DOS/Windows to Linux" HOWTO (<http://metalab.unc.edu/LDP/HOWTO/DOS-Win-to-Linux-HOWTO.html>).

There are a number of freely-available books for Linux, published under the auspices of the Linux Documentation Project. An excellent free book for Linux beginners is Matt Welsh's Installation and Getting Started guide (<http://metalab.unc.edu/LDP/LDP/gs/gs.html>).

There are some good on-line introductions to Unix available at the Unix Reference Desk (<http://www.geek-girl.com/unix.html>).

Other Linux web sites worth a look are:

- Linux Resources: <http://www.linuxresources.com/>
- The Linux Journal: <http://www.linuxjournal.com/>
- slashdot.org (Linux news): <http://slashdot.org/>

If you want to buy a Linux book, two excellent introductions to Linux are:

- *Running Linux* (2nd edition) by Matt Welsh and Lar Kaufmann, published by O'Reilly and Associates
- *A Practical Guide to Linux* by Mark Sobell, published by Addison-Wesley

In addition, there are some useful mailing lists and USENET news groups that provide information and support for new Linux users, such as:

- comp.os.linux.announce
- comp.os.linux.misc
- comp.os.linux.setup