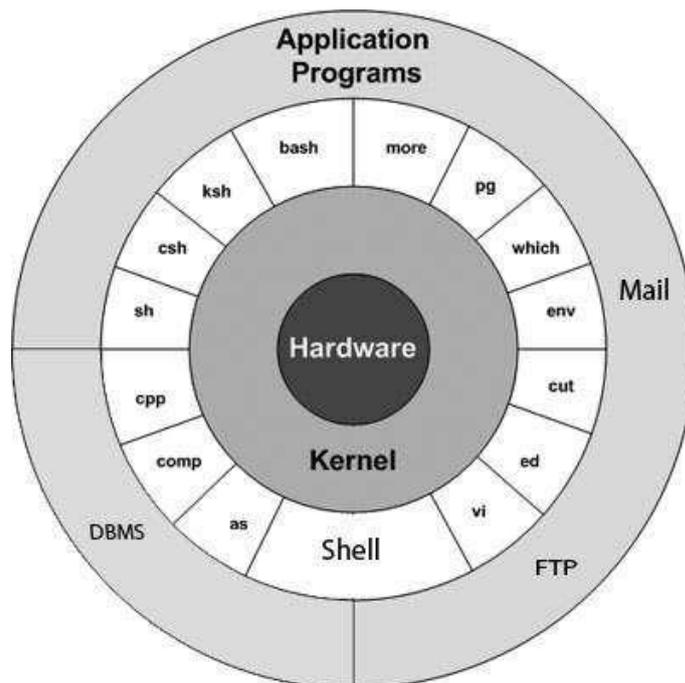


LINUX OS Manual for Students, faculty and Staff



(Ravi K. Walia)
Assistant Professor & Incharge
Computer & Instrumentation Centre
Dr. Y. S. Parmar University of Horticulture & Forestry,
Nauni Solan INDIA (HP)

PREFACE

This document has been prepared for students, faculty, and staff at Dr. Y. S. Parmar University of Horticulture & Forestry, Nauni, Solan (HP) India.

Linux, based on the Unix is one of most powerful and versatile operating system. A research team at AT&T's Bell Labs developed Unix in the late 1960s and early 1970s with a focus on creating an operating system that would be accessible and secure for multiple users. In 1991, **Linus Torvalds** released the Linux kernel as free, open-source software. Open source means that the code is fully visible, and can be modified and redistributed.

Linux is used in most of the servers like web Server, Email Server, DNS Server, File Servers, workstations, home user PC, client and even more than 80% of supercomputers.

Most of Linux version has GUI like MS Windows, however to explore the full capability of Linux user need to learn the command mode. Each command in Linux has many options which effect the output. This manual cover main features of Linux, basic concepts, file system and security access, most of the commands with example and vi editor. To explore more about command such as attribute and parameters user can use inbuilt help using man command.

Feature of Linux

- Written in high level language “C”
- Use hierarchical file system allows easy file maintenance
- Use consistent file format : byte stream
- Multiuser and multi-process system
- Use round robin scheduling with multilevel feedback. Process aging is employed to prevent starvation
- Memory management employ swapping and demand paging

BASIC CONCEPTS

Kernel: The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

Shell: The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells.

Commands and Utilities: There are various commands and utilities which you can make use of in your day to day activities. **cp**, **mv**, **cat** and **grep**, etc. are few examples of commands and utilities.

Files and Directories: All the data of Linux is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the filesystem. There are three basic types of files –

Ordinary Files – An ordinary file is a file on the system that contains data, text,

or program instructions. In this tutorial, you look at working with ordinary files.

Directories – Directories store both special and ordinary files. For users familiar with Windows or Mac OS, Unix directories are equivalent to folders.

Special Files – Some special files provide access to hardware such as hard drives, CD-ROM drives, modems, and Ethernet adapters. Other special files are similar to aliases or shortcuts and enable you to access a single file using different names.

SYSTEM BOOTUP

If you have a computer which has the Linux operating system installed in it, then you simply need to turn on the system to make it live.

As soon as you turn on the system, it starts booting up and finally it prompts you to log into the system, which is an activity to log into the system and use it for your day-to-day activities.

Login Linux

When you first connect to a Linux system, you usually see a prompt such as the following:

login:

To log in

- Have your userid (user identification) and password ready. Contact your system administrator if you don't have these yet.
- Type your userid at the login prompt, then press ENTER. Your userid is case-sensitive, so be sure you type it exactly as your system administrator has instructed.
- Type your password at the password prompt, then press ENTER. Your password is also case-sensitive.
- If you provide the correct userid and password, then you will be allowed to

enter into the system. Read the information and messages that comes up on the screen, which is as follows.

You will be provided with a command prompt (sometime called the \$ prompt or % prompt) where you type all your commands.

Linux Command Structure:

The linux command structure is

```
command [ -options] [arguments]
```

The options are specified by “-“ followed by one letter per option. The option change the output pattern of command. Argument specifies that the command will work on the argument supplied if any.

e.g. \$ ls ;List information about directories & files of current working directory

\$ ls -l ;display long list contents of current working directory

\$ ls -l /etc ;display long list contents of directory /etc

List of important command in Linux:

Command	Example	Explanation
<u>cal</u>		Display calendar
	\$ cal	Display calendar of current month and current year
	\$ cal 2016	Display the calendar of year 2016
	\$ cal 4 2016	Display the calendar of April, 2016
<u>clear</u>	\$ clear	Clear the screen
<u>pwd</u>	\$ pwd	Print working directory. When you first login, your current working directory is your home directory. Your home directory has the same

name as your user-name

date

\$ date Display system date and system time

who

\$ who Display users who are currently working on the system

\$ who am i Display your login id

\$who -q Display number of users currently working on the system

banner

\$ banner UHF Display text in the form of banner

man

Display manual/help of a command, use q to quit from help

\$ man cal Display manual/help of a command cal

ls

List contents of directory

\$ ls List the contents of current working directory excluding hidden files (File beginning with dot (.) are hidden file)

\$ ls -a List all the files in current working directory including hidden files

\$ ls -l Long Listing of files

\$ls /etc List the contents of directory /etc

logout

\$ logout Logout from the system

or

\$ ^+d

shutdown

\$ shutdown shutdown/ power off the system

or

\$ init 0

reboot

\$ reboot Reboot the system

mkdir

Make a directory

\$mkdir UHF Make directory UHF under current directory

\$mkdir /UHF/COH Make directory COH & COF under /UHF
/UHF/COF

cd

Change the current working director

\$ cd / Make root (/) as current working directory

\$ cd /UHF/COH Change directory to /UHF/COH

\$ cd FLS Change to subdirectory FLS under current working directory

\$ cd .. Go one step up in the directory tree

\$ cd ./COF Go to directory COF under current directory

\$ cd Change to home directory

cp

To make a copy of a file

\$ cp abc xyz Copy file abc to xyz

\$cp abc /xyz Make copy of file abc in the directory xyz with name abc (if directory xyz exist) else make copy of file abc with name xyz in the root directory (/).

mv

Move a file from one directory to other or change the name of a file in the same directory

\$ mv abc xyz Change the name of file abc to xyz

\$ mv abc /etc Move the file abc from current directory to /etc directory

rm

Remove a file or directory

\$ rm abc Remove file abc under present working directory

\$ rm /etc/xyz Remove file xyz from directory /etc

\$ rm -r /etc/abc Remove directory abc form /etc . Remove directory even if it is not empty.

rmdir

Remove directory (directory must be empty ie. It should not have any subdirectory or file)

\$ rm abc Remove empty directory abc from present working directory

\$ rm /etc/abc Remove empty directory abc from /etc

history

\$ history List all previously executed commands

!

\$!4 Recall 4th command

\$!-4 Recall the 4th most recent command

cat

Concatenate files: take screen as standard input output

\$ cat myfile Display the content of myfile on the screen

\$ cat myfile1
myfile2 Display the content of myfile1 and myfile2 on the screen

\$ cat myfile1
myfile2 > myfile3 Concatenate(Combine) files myfile1 and
myfile2 and create a new file myfile3

\$cat >newfile Create a new file with name newfile by taking
Some text.... input from standard input and redirecting the
Sometext... output to file newfile
Ctrl+d

more

Display the contents of a file on the screen., a page at a time. Press the [space-bar] if you want to see another page, and type [q] if you want to quit reading.

\$ more myfile Display myfile on the screen

less

Display the contents of a file on the screen., a page at a time. Press the [space-bar] if you want to see another page, and type [q] if you want to quit reading.

\$ less myfile Display myfile on the screen

head

Display first few line of file

\$ head myfile The head command writes the first ten lines
of a file to the screen.

\$ head -12 myfile Display all line starting from line 12 on the
screen

tail

Display last few line of a file

\$ tail myfile The tail command writes the last ten lines of a
file to the screen.

\$ tail -8 myfile Display the last 8 lines on the screen

pg

Display file in formatted form with header, footer, page number etc.

	\$ pg abc	Display file abc in formatted form
<u>lpr or lp</u>		Print the file on printer
	\$ lprt myfile	Print the file myfile
	\$ pg myfile lpr	Print the file myfile with formatted output
<u>lpq</u>	\$lpq	List print queue
<u>cancel</u>	\$cancel abc-123	Terminate print request (abc-123) from lp command
<u>wc</u>		Count characters, words and lines in a text file
	\$wc filename	Display number of characters, words and line in a text file
	\$wc -c filename	Display number of characters in file filename
	\$wc -w filename	Display number of words in file filename
	\$wc -l filename	Display number of lines in file filename
	\$wc -wl filename	Display number of words and number of lines in file filename
>		Used to redirect the output
	\$cat abc > xyz	Redirect the output of cat command to file xyz, First remove the content of file xyz if any
>>		To append the content of redirection
	\$cat xyz >>newfile	Append the content of file xyz at the end of newfile
<		Used to redirect the input
	\$ sort < abc	Sort the file abc and display the content of screen
	\$ sort <abc > sabc	Redirect the input from file abc and redirect the output to file sabc
<u>sort</u>		Sort/arrange the contents
	sort abc	Sort the file on string/ascii code

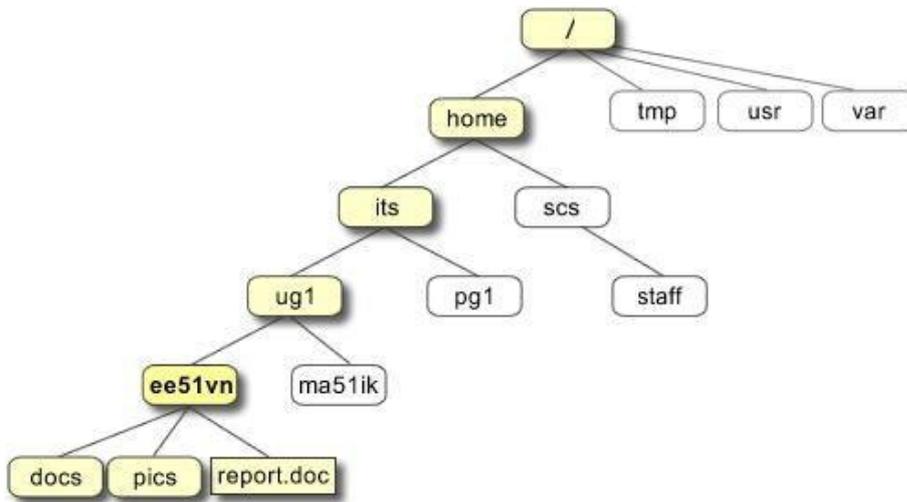
	<code>sort -n abc</code>	Sort the file numerically
<u>I</u>		Piping is used to send output of one command as input of another command
	<code>\$who sort</code>	Send the output of who command i.e list of user currently working on the system and send the output to sort command which sort the results. Thus instead of storing intermediate results in a file (abc.txt) as below <code>\$who >abc.txt</code> <code>\$sort >abc.txt</code> We use piping <code>\$who sort</code>
<u>ps</u>		To see information about your processes, with their associated PID and status, type
	<code>\$ ps</code>	To see information about user own processes
	<code>\$ ps -e</code>	To see information about all user processes
<u>kill</u>		kill (terminate or signal a process)
	<code>\$ kill 123</code>	To kill process with PID 123
	<code>\$kill -9 145</code>	If a process refuses to be killed, uses the -9 option to forcibly kill process.
<u>&</u>		To run process in the background
	<code>\$ ls -l / >xyz &</code>	Run the ls -l process in the background
<u>chown</u>		Change the owner of file. NOTE: The super user, root, has the unrestricted capability to change the ownership of any file but normal users can change the ownership of only those files that they own.
	<code>\$ chwon rkw myfile</code>	Change the owner of file myfile to rkw
<u>chgrp</u>		Change group name of file

	<code>\$chgrp group2 myfile</code>	Change group name to group2 for file myfile
<u>Jobs</u>	<code>\$ jobs</code>	List the current jobs
<u>df</u>	<code>\$ df</code>	The dfcommand reports on the space left on the file system
<u>du</u>	<code>\$du</code>	The ducommand outputs the number of kilobytes used by each subdirectory
	<code>\$ du -s *</code>	Display only summery of all files and directories
<u>diff</u>	<code>\$ df file1 file2</code>	This command compares the contents of two files and displays the differences
<u>find</u>	<code>\$ find . -name "*.txt" -print</code>	This searches through the directories for files and directories with a given name, date, size To search for all fies with the extention .txt, starting at the current directory (.) and working through all sub-directories, then printing the name of the file to the screen
<u>file</u>	<code>\$ file *</code>	fileclassifies the named files according to the type of data they contain, for example ascii (text), pictures, compressed data, etc..
<u>fsck</u>	<code>\$ fsck</code>	To check the file system for any error
<u>write</u>	<code>\$ fsck -y</code>	Try to recover from errors in file system For online communication
	<code>\$ write user1</code>	Communicate online with user user1
	<code>\$ write user1 tty06</code>	Communicate online with user user1 on terminal tty06
<u>mesg</u>		Allow or deny online communication
	<code>\$ mesg -n</code>	Deny online communication

<u>mail</u>	<p>\$ mesg -y</p> <p>\$ mail</p> <p>\$ mail user6</p> <p>Welcome to UHF</p> <p>Welcome to UHF</p> <p>.....some text</p> <p>Ctrl+d</p>	<p>Allow online communication on</p> <p>For offline communication to read or write message</p> <p>To read mail</p> <p>To Send mmessage/mail to user with user id user6</p>
<u>ping</u>	<p>\$ ping hostname</p> <p>or</p> <p>\$ ping ip-address</p>	<p>The ping command sends an echo request to a host available on the network. Using this command, you can check if your remote host is responding well or not.</p>
<u>finger</u>	<p>\$ finger user1</p>	<p>The finger command displays information about users on a given host. The host can be either local or remote.</p>
<u>netstat</u>	<p>\$ netstat</p>	<p>Display network status: network connections, routing tables, interface statistics, masquerade connections, and multicast memberships</p>
<u>nice/renice</u>		<p>Runs a program with modified scheduling priority</p>
<u>useradd</u>		<p>Adds accounts to the system</p>
<u>usermod</u>		<p>Modifies account attributes</p>
<u>userdel</u>		<p>Deletes accounts from the system</p>
<u>groupadd</u>		<p>Adds groups to the system</p>
<u>groupmod</u>		<p>Modifies group attributes</p>
<u>groupdel</u>		<p>Removes groups from the system</p>

File system security (access rights)

Linux use the inverted tree type structure (like windows) for directories as shown below



Following are the directories that exist on the major versions of linux –

Directory	Description
/	This is the root directory which should contain only the directories needed at the top level of the file structure
/bin	This is where the executable files are located. These files are available to all users
/dev	These are device drivers
/etc	Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hosts, where to send critical messages
/lib	Contains shared library files and sometimes other kernel-related files
/boot	Contains files for booting the system

/home	Contains the home directory for users and other accounts
/mnt	Used to mount other temporary file systems, such as cdrom and floppy for the CD-ROM drive and floppy diskette drive, respectively
/proc	Contains all processes marked as a file by process number or other information that is dynamic to the system
/tmp	Holds temporary files used between system boots
/usr	Used for miscellaneous purposes, and can be used by many users. Includes administrative commands, shared files, library files, and others
/var	Typically contains variable-length files such as log and print files and any other type of file that may contain a variable amount of data
/sbin	Contains binary (executable) files, usually for system administration. For example, fdisk and ifconfig utilities
/kernel	Contains kernel files

Use the following long listing filename command

```
$ ls -l
```

You will see that you now get lots of details about the contents of your directory, similar to the example below.

```
-rw-rw-r-- 1 coh uhf 166255 Aug 9 2007 myfile1
drwxr-xr-x 11 cof uhf 4096 May29 2007 myfile2
```

Here is the information about all the listed columns –

Column 1: Represents the file type and the permission given on the file Every file line begins with a d, -, or l. These characters indicate the type of the file that's listed.

PrefixDescription

- Regular file, such as an ASCII text file, binary executable, or hard link
- b Block special file. Block input/output device file such as a physical hard drive
- c Character special file. Raw input/output device file such as a physical hard drive
- d Directory file that contains a listing of other files and directories
- l Symbolic link file. Links on any regular file
- p Named pipe. A mechanism for interprocess communications
- s Socket used for interprocess communication

The 9 remaining symbols indicate the permissions, or access rights, and are taken as three groups of 3.

- The left group of 3 gives the file permissions for the user that owns the file (or directory) (coh in the above example);
- the middle group gives the permissions for the group of people to whom the file (or directory) belongs (uhf in the above example);
- the rightmost group gives the permissions for all others.

The symbols r, w, etc., have slightly different meanings depending on whether they refer to a simple file or to a directory.

Access rights on files.

- r (or -), indicates read permission (or otherwise), that is, the presence or absence of permission to read and copy the file
- w (or -), indicates write permission (or otherwise), that is, the permission (or otherwise) to change a file
- x (or -), indicates execution permission (or otherwise)

otherwise), that is, the permission to execute a file, where appropriate

Access rights on directories.

- r allows users to list files in the directory;
- w means that users may delete files from the directory or move files into it;
- x means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files. So, in order to read a file, you must have execute permission on the directory containing that file, and hence on any directory containing that directory as a subdirectory, and so on, up the tree.

Column 2: Represents the number of memory blocks taken by the file or directory.

Column 3: Represents the owner of the file. This is the Linux user who created this file.

Column 4: Represents the group of the owner. Every Linux user will have an associated group.

Column 5: Represents the file size in bytes.

Column 6: Represents the date and the time when this file was created or modified for the last time.

Column 7: Represents the file or the directory name.

Changing access rights

chmod (changing a file mode)

Only the owner of a file can use chmod to change the permissions of a file.

The options of chmod are as follows

u : user	g : group	o : other	a : all
r : read	w : write /delete	x : execute/ accessdirectory	
+ : add permission	- : take away permission		

For example, to remove read write and execute permissions on the file **myfile** for the group and others, type

\$ chmod go-rwx myfile

Grep Command

Grep is one of many standard linux utilities. It searches files for specified words or patterns.

\$ grep "string1" filename	List the lines containing string1 in file
\$ grep "string1" filename*	List the lines containing string1 in multiple files
\$ grep -i "string1" filename	Case insensitive search
\$ grep "^string1" filename	List the lines containing string1 in the beginning
\$ grep "string1\$" filename	List the lines containing string1 in the end
\$ grep -c "string1" filename	Give count of lines containing string1 in file
\$ grep -v "string1" filename	List the lines not containing string1 in file
\$ grep "string1" *	Search all the files in current directory
\$ grep -r "string1" *	Search all the files in current directory and all sub directory
\$ grep -l "string1" *	List on the file names containing string1
\$ grep -o "is.*this" filename	List only the match string of pattern
\$ grep -n "string1" filename	Show also the line number
\$ find . -name "*.mp3" grep -i JayZ grep -vi "remix"	Find all mp3 file then find file from JayZ then remove remix file from the list

vi editor commands

Use the **vi** editor to create ordinary files on any linux system. You simply need to give the following command

```
$vi abc
```

The above command will open a file with the given filename. Now, press the key **i** to come into the edit mode. Once you are in the edit mode, you can start writing your content in the file.

Press the key **esc** to come out of the edit mode. Press two keys **Shift + ZZ** together to come out of the file completely

Cursor Movement Commands	Editing Commands
h Left	A Append to end of current line
j Down	i Insert before cursor
k Up	I Insert at beginning of line
l Right	o Open line above cursor
w Forward one word	O Open line below cursor
b Back one word	ESC End of insert mode
0 Start of current line	Backspace Move back one character
\$ End of current line	Ctrl-U Delete current line
Ctrl-F Forward one screen	cw Change word
Ctrl-B Back one screen	cc Change line
Ctrl-G Display line number of cursor	C Change from current position to end of line
nG Move cursor to line number n	dd Delete current line
:n Move cursor to line number n	ndd Delete n lines
G Move to last line in file	D Delete remainder of line
File Handling	dw Delete word
:w Write file	J Join two lines

:w!	Write file (ignoring warnings)	p	Insert buffer after cursor
:w!	file Overwrite file (ignoring warnings)	P	Insert buffer before cursor
:wq	Write file and quit	rx	Replace character with x
:q	Quit	Rtext	Replace text beginning at cursor
:q!	Quit (even if changes not saved)	u	Undo last change
:w file	Write file as file, leaving original untouched	U	Restore current line
ZZ	Quit, only writing file if changed	x	Delete current cursor position
:x	Quit, only writing file if changed	X	Delete back one character
		nX	Delete previous n characters
		y	Copy current line to new buffer
		yy	Copy current line