

Advance topics in NepaLinux 2.0
Training Manual

Madan Puraskar Pustakalaya, Nepal

Table of contents

- NepaLinux command Line Basics
- Virtual consoles
- Shell basics
- Linux File Hierarchy Concepts
- Command line filesystem browsing
- Standard Input/Output and pipes
- Users, groups and permissions
- File permissions and user categories
- vi Editor Basics
- Archive and compression utilities in Linux
- Checking free space
- Removable media CD/DVD and USB media
- Fonts installation
- Network configuration
- File sharing NFS/SAMBA
- Searching files and directory
- CVS - Concurrent versioning system
- Package management
- Installing source and binary packages
- NepaLinux - Basic troubleshooting

NepaLinux command Line Basics:

Virtual consoles:

Multiple non-GUI logins are possible through the use of virtual consoles. There are by default 6 available virtual consoles. These are available through CTRL-ALT-F[1-6]. If X(GUI) is running it is available as CTRL-ALT-F7.

Shell basics

Commands are entered in the shell. The default shell in NepaLinux is Bash shell. Simply shell can be defined as a program that provides the traditional text only user interface for UNIX like operating system. Shell's primary function is to execute i.e run the commands type in the console or Terminal window.

The default shell prompt for a user is

\$

eg. for general user, maya,

```
maya@nepalinux:~$
```

The default shell prompt for the root user is

#

eg.

```
root@nepalinux:~#
```

For most of Linux system, the \$ or # is preceded by your user name, system name, and current directory name. So, for example, a login prompt for user name "maya" on a computer nepalinux with /tmp as current directory would appear as,

```
maya@nepalinux:tmp$
```

If user maya is in his home directory, the login prompt appears as,

```
maya@nepalinux:~$
```

Here the tilde(~) refers to the home directory of user maya. i.e /home/maya.

Running simple commands and getting help

Generally Linux commands have the following syntax.

Command [options] [arguments]

Each item is modified by a space. Options modify the command's behaviour.

Words options usually preceded by "-" sign. Arguments are the file name or other data needed by the command.

Interrupting command execution

If you enter a command and don't get the output, then the command may be busy waiting executing or waiting for the input.

To interrupt the command taking too long to respond, type <Ctrl -c>

Some simple commands:

date --display date and time

cal-- display calendar

```
maya@nepalinux:~$ date
Sat Sep 22 23:11:16 NPT 2007
```

Getting Help

you don't have to memorize each and every Linux command. Many levels of help is available.

whatis

<command> --help

man and info

/usr/share/doc

whatis ----> displays the short description of command.

eg. maya@nepalinux:~\$ whatis date

date (1) - print or set the system date and time

the --help option

Displays the usage summary and argument list

```
maya@nepalinux:~$ date --help
```

```
Usage: date [OPTION]... [+FORMAT]
```

```
or: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
```

Display the current time in the given FORMAT, or set the system date.

```
-d, --date=STRING      display time described by STRING, not `now`
-f, --file=DATEFILE    like --date once for each line of DATEFILE
-r, --reference=FILE    display the last modification time of FILE
< output truncated >
```

Reading man page

The man page provides the detail help about a command.

eg.

```
$ man date
```

Navigating man pages.

While viewing the man pages, navigate with arrows, pgup, pgdown.

/<text> to search for the text.

n/N Next or previous match.

q Quit viewing the page.

The info command

Similar to the man pages, but more in-depth.

```
Info < command >
```

The /usr/share/doc directory contains documentation for most of the installed packages.

Linux File Hierarchy Concepts

Windows and Linux use different concepts for their file hierarchy. Windows uses a volume-based file hierarchy, Linux uses a unified scheme. Windows uses letters of the alphabet to represent different devices and different hard disk partitions. Under Windows, you need to know what volume (C:, D:,...) a file resides on to select it, the file's physical location is part of its name. In Linux all directories are attached to the root directory, which is identified by a forward-slash, "/". For example, below are some second-level directories:

/bin/ ---- system binaries, user programs with normal user permissions
/sbin/ --- executables that need root permission
/data/ --- a user defined directory
/dev/ ---- system device tree
/etc/ ---- system configuration
/home/ --- users' subdirectories
/home/{username} home directory of users
/tmp/ ---- system temporary files
/usr/ ---- applications software
/usr/bin - executables for programs with user permission
/var/ ---- system variables
/lib/ --- libraries needed for installed programs to run
/root/ ----Root's home directory
/media and /mnt – Foreign filesystem mount point.

Every device and hard disk partition is represented in the Linux file system as a subdirectory of the root directory. The root directory lives in the root partition, but other directories (and the devices they represent) can reside anywhere. Removable devices and hard disk partitions other than the root are attached (i.e., "mounted") to subdirectories in the directory tree. This is done either at system initialization or in response to a mount command.

In brief:

Files and directories are organized into a single-rooted inverted tree structure including distinct physical volumes such as CD-ROM and Hard Drives.

The base of the inverted tree is known as root or "/" to top of the file structure.

Paths are delimited by "/"

Names are case sensitive.

.. refers to the parent directory of any particular directory.

. refers to the current directory.

Files and directories whose names begin with a . are hidden, i.e they are not listed by default while listing.

Command Line Filesystem Browsing

Current working directory

Each shell and system process has a current working directory.

pwd. Displays the absolute path to the current directory.
Current working directory is the directory in which you are working.
eg. maya@nepalinux:demo\$ pwd
/home/maya/demo

Absolute Pathnames and Relative Pathnames.

The location of a file or directory can be specified either by absolute path or relative path names.

An absolute path names begins with slash (/). It contains the name of the directory that must be traversed from the root file system.
eg. /var/lib/alsa/asound.state

Relative path names don't begin with slash (/). It specifies the location relative to your current working directory. It can be used as a shorter way to specify a file name.

Changing Directories.

cd changes the directories to an absolute path or relative path
maya@nepalinux:~\$ cd /home/maya/nepalinux_guide
maya@nepalinux:~\$ cd project/forensics

To a directory one level up

cd ..
eg.
maya@nepalinux:nepalinux_guide\$ cd ..
maya@nepalinux:Manual\$

To your home directory

cd
eg.
maya@nepalinux:Manual\$ cd
maya@nepalinux:~\$

To your previous working directory.

cd -
eg.
maya@nepalinux:nepalinux_guide\$ cd -
/home/maya/Manual

maya@nepalinux:Manual\$

Listing Directory contents

ls

usage

ls [options] [arguments]

ls without any options and arguments list the file and directory names in the current working directory.

eg.

```
maya@nepalinux:~$ ls
```

```
Manual research test
```

ls list the contents of the directory given as argument.

eg.

```
maya@nepalinux:~$ ls /
```

```
bin dev hda5 lib mnt sys var
```

```
boot etc hdb8 lost+found opt root t1 vmlinuz
```

```
cdrom floppy home media patch sam tmp
```

ls -a includes so called hidden files whose names begin with .

eg.

```
maya@nepalinux:~$ ls -a
```

```
. .. .bash_history .lesshst .viminfo Manual research test
```

ls -l for a more detailed long listing

eg.

```
maya@nepalinux:~$ ls -l /bin/
```

```
total 3476
```

```
-rwxr-xr-x 1 root root 3200 2006-05-16 06:05 arch
```

```
-rwxr-xr-x 1 root root 684724 2006-03-23 06:08 bash
```

```
-rwxr-xr-x 1 root root 25352 2006-07-05 16:36 bunzip2
```

```
-rwxr-xr-x 1 root root 25352 2006-07-05 16:36 bzip2
```

```
-rwxr-xr-x 1 root root 2105 2006-07-05 16:36 bzip2
```

For other more options please refer to the man pages.

Creating and Removing files and directory.

mkdir ---> makes a directory

touch ----> makes an empty file or updates the timestamps

rmdir ----> removes the directory.

rm -r ----> recursively removes the directory and all its contents.

other rm options

-i ---> interactive

-f ---> force

rm can also be used to remove files.

eg.

```
maya@nepalinux:~$ mkdir test
```

This command creates one directory named test.

{ } Curly braces are useful for generating pattern strings.

eg.

```
maya@nepalinux:forensics$ mkdir -p tools/{research1,research2}/papers
```

```
maya@nepalinux:forensics$ ls tools/
```

```
research1 research2
```

```
maya@nepalinux:forensics$ ls tools/research1/
```

```
papers
```

Files are usually created with the help of editors. We try to explain them later.

Copying files and directories

cp copy the files and directories.

Usage.

cp [options] file destination

more than one file may be copied at a time if the destination is a directory.

eg.

```
cp [option] file1 file2 file3 destination
```

cp must always be given at least two arguments. First is source file or directory and other last one is destination. If the destination is a directory, file is copied in that directory. If the destination is not directory destination is interpreted as filename and a copy of the source file is created with the destination name.

A few common options

-i interactive ask before overwriting a file

-r recursive: recursively copy entire directory tree.

-p preserve permission, timestamps, ownership

eg.

```
maya@nepalinux:~$ ls
Manual forensics research test
maya@nepalinux:~$ cp research forensics
maya@nepalinux:~$ cd forensics/
maya@nepalinux:forensics$ ls
research
```

Here “research” file is copied in the directory forensics.

For copying directory use options either -a or -r.

Moving and renaming Files and Directory.

mv – move and or rename files and directory

usage

mv file destination

more than one file may be moved at a time if the destination is a directory.

```
mv file1 file2 file3 destination
```

eg.

```
maya@nepalinux:~$ ls
Manual forensics research test
maya@nepalinux:~$ mv Manual NepaLinux_Manual
maya@nepalinux:~$ ls
NepaLinux_Manual forensics research test
```

Here “Manual” is renamed to “NepaLinux_Manual”

another eg.

```
maya@nepalinux:~$ ls
NepaLinux_Manual gnome_manual.txt research
forensics      kde_manual.txt test
maya@nepalinux:~$ mv gnome_manual.txt kde_manual.txt NepaLinux_Manual/
```

Determining the file content

To determine the type of file you have, use the command “**file**”

eg.

```
maya@nepalinux:Desktop$ file LTSP
LTSP: Zip archive data, at least v2.0 to extract
```

maya@nepalinux:Desktop\$

Viewing an entire text file

cat

contents of files are displayed with no break

eg.

```
$ cat research.txt
```

Viewing text page by page.

Use “less” command

less is the pager used by man.

Use q to quit from viewing the file.

To view only few upper portion of a file use, **head** command and to view only few lower portion of the file use **tail** command.

eg. tail research.txt

to view exactly 20 lower lines, use

```
tail -20 research.txt
```

Command Line shortcuts and automatic file name completions.

It is wonderful features for the Bash shell. You don't have to type the whole command and whole file name, just press <Tab> key, it will be completed automatically. In case of matching with more than more than one command or files are displayed for selection.

eg.

```
root@nepalinux:~# user<Tab>
```

```
useradd  userdel  usermod  users  users-admin
```

History

Bash stores the history of command that you entered, which can be used to repeat command.

Use “history” command to see the commands that you entered previously.

Standard I/O and Pipes

Standard Input and Output

Linux system provides three I/O channels to processes.

Standard Input – Keyboard is default

Standard Output – Terminal window is default

Standard error --- Terminal window is default

The standard output of commands, which ordinarily displays on the terminal, can be redirected into a file or piped into another command.

Standard error, which also ordinarily displays on the terminal, can be redirected into a file. Although it is also possible to pipe standard error into a file using some fairly complex syntax, this is generally not done.

Standard input, ordinarily coming from the keyboard, can be redirected from a file. More commonly, the standard output of one command can be piped into the standard input of another command

Common Redirection Operators

>	command > file	Directs standard output of command to file.
>>	command>>file	Appends standard oupput of command to file.
<	command < file	command receives its input from a file.
2>	command 2> file	Error messages from command are directed to file.
2>>	command 2>>file	Error messages from command are appended to file.

Piping

| command1 | command2 Pipes the standard output of command1 into the standard input of command2.

eg.

Redirecting the standard output

```
maya@nepalinux:~$ history > histfile
```

Here rather than displaying the output of history command , it is redirected to histfile.

Redirecting Input

eg.

```
maya@nepalinux:~$ cat < research.txt
```

cat command may work here without < operator. But there are some commands that only operate on < operator.

eg. **tr** (translate command)

```
maya@nepalinux:~$ cat < research.txt
```

The research paper on Digital Forensics.

```
maya@nepalinux:~$ tr 'a-z' 'A-Z' < research.txt
```

THE RESEARCH PAPER ON DIGITAL FORENSICS.

Redirecting standard Error

eg.

```
maya@nepalinux:~$ find /etc -name passwd
```

```
/etc/passwd
```

```
find: /etc/cups/ssl: Permission denied
```

```
/etc/pam.d/passwd
```

```
find: /etc/ppp/peers: Permission denied
```

```
find: /etc/ssl/private: Permission denied
```

Here both standard error and output are presented together in the terminal window. To redirect the standard error to a file use `2>` operator.

eg.

```
maya@nepalinux:~$ find /etc -name passwd 2> find_error
```

```
/etc/passwd
```

```
/etc/pam.d/passwd
```

Now in this case only standard output is displayed in the console.

Verify the error are redirected to the file, use `cat` command.

eg.

```
maya@nepalinux:~$ cat find_error
```

```
find: /etc/cups/ssl: Permission denied
```

```
find: /etc/ppp/peers: Permission denied
```

```
find: /etc/ssl/private: Permission denied
```

Using pipes to connect Processes.

Pipes let you to redirect the output from one command to become the input for another command.

eg.

```
maya@nepalinux:~$ ls /usr/lib | less
```

maya@nepalinux:~\$ ls | grep pass

Users, Groups and Permissions

Users

- To Login in the Linux system, you need User account.
- Each user is assigned a UID. UID and username is stored in /etc/passwd file.
- Users are assigned a home directory and a shell, which is run when they login in.

Groups

- Users are assigned with unique group ID.
- Group names and GID are stored in /etc/group file.
- Each user is given their own group with the same name as that of their account. They can also be added to other groups to give additional access.
- All users in a group share a file that belongs to the group.

The root user

- The root user sometimes called a super user.
- Root has the complete control over the system.
- You should not login as root unless you have to do administrative jobs.
- Normal (unprivileged) users potential to do damage is limited.
- To switch to the power user (root) from the normal user, use **su** – command.

Adding a New user account

useradd

useradd <username>

eg. root@nepalinux:~# useradd maya

Here user maya is added.

The next step in creating a user account is to assign password.

Use **passwd** command.

```
eg. root@nepalinux:~# passwd maya
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Modifying and Deleting the user account

use **usermod** command to modify the user account or directly edit the `/etc/passwd` file.

usermod options:

- d <home directory> change the home directory.
- g <group> change the initial login group.
- G < group, [,] > A command separated list of supplementary group for user.
- s <shell>
- l < login name>

```
root@nepalinux:~# usermod -G account,finance maya
```

Deleting account

use **userdel** [-r] username

```
eg. root@nepalinux:~# userdel -r test
```

Here -r is used to remove the home directories of the user as well.

Adding, removing and modifying the group

To add group use **groupadd** <group-name>

```
eg. root@nepalinux:~# groupadd finance
```

To modify group use, **groupmod** command.

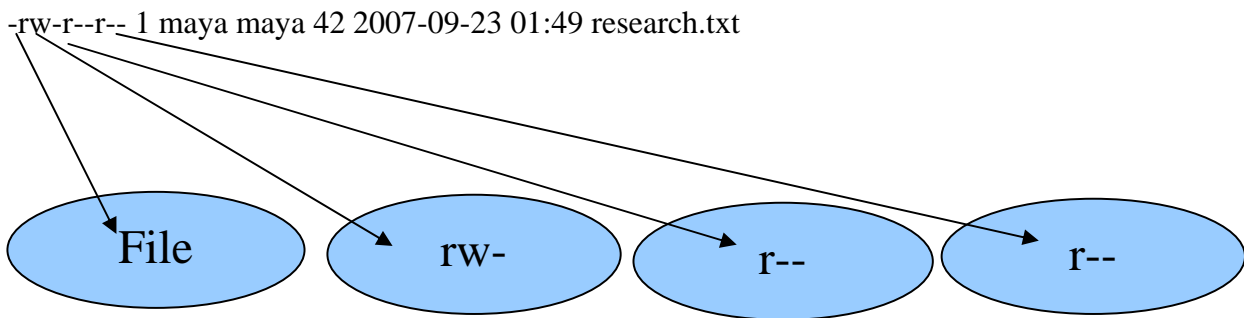
Usage. **groupmod** -n newname oldname.

To delete a group use **groupdel** < group name>

File Permission

File permission may be viewed using “**ls -l**” command.

```
maya@nepalinux:~$ ls -l research.txt
```



File type and File access permissions are symbolised by 10 character string. The graphical representation may give the insight of it.

User categories

File permissions are established for each of the three user categories.

Each category also has a symbol.

- u – the files owner(user)
- g- other users in the file group
- o- everyone else (others)

File Permission Types:

Four symbols are used while displaying the permission.

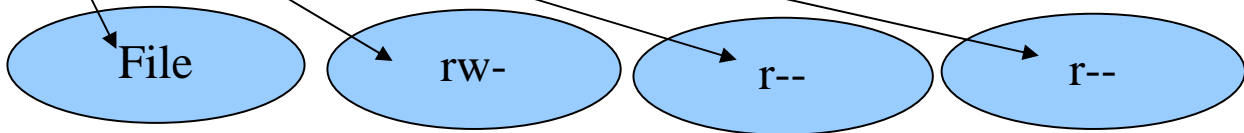
- r permission to read a file or list a direcotory's content.
- w permission to write a file or create or remove files in the direcotory.
- x permission to execute a file or change into the direcotory and do long listing.
- no permission(in place of r, w, x)

Of the 10 character in the file permission, first character represents the file type.

-, represent the regular file, d represent directory, b represent block special file and c represent character special file, l represent symbolic link.

Examining user, group and other permission.

```
-rw-r--r-- 1 maya maya 42 2007-09-23 01:49 research.txt
```



characters 2,3, and 4 in the access mode identify the user permission, characters 5, 6, 7 in the access mode identify the group permission and characters 8, 9, 10 identify the others permission.

From the permission observation, it is clear that this file can be read and write by the user, and only read by the group members and others.

File Ownership

Every file has both user and group ownership.

A newly created file is owned by the user who creates it, and the primary group of that users.

Changing the ownership

for changing the ownership of the file use, chown command.

eg.

```
root@nepalinux:scalpel# ls -ld scalpel-1.60
```

```
drwxr-xr-x 3 root root 4096 2007-09-22 13:30 scalpel-1.60
```

```
root@nepalinux:scalpel# chown maya scalpel-1.60
```

```
root@nepalinux:scalpel# ls -ld scalpel-1.60
```

```
drwxr-xr-x 3 maya root 4096 2007-09-22 13:30 scalpel-1.60
```

for recursive option use, -R option.

For changing the group ownership of a file, use chgrp command, or use chown command.

- chown username.groupname <file or directory>
- chgrp groupname <filename or directory>

Changing the permission

Symblic Method

u –users	+ Add a permission	r	-Read
g---groups	- Remover a permission		w -Write
o---others	= Assign a permission	x	-Execute
a----all			

eg.1

```
maya@nepalinux:~$ ls -l research.txt
-rw-r--r-- 1 maya maya 42 2007-09-23 01:49 research.txt
maya@nepalinux:~$ chmod g+w research.txt
maya@nepalinux:~$ ls -l research.txt
-rw-rw-r-- 1 maya maya 42 2007-09-23 01:49 research.txt
Here chmod g+w gives the write access to th group.
```

eg.2

```
maya@nepalinux:~$ ls -l research.txt
-rw-rw-r-- 1 maya maya 42 2007-09-23 01:49 research.txt
maya@nepalinux:~$ chmod o-r research.txt
maya@nepalinux:~$ ls -l research.txt
-rw-rw---- 1 maya maya 42 2007-09-23 01:49 research.txt
```

Here chmod o-r removes the read permission from the other.

Numeric Method

uses a three digit number.

- First digit specifies the owner's permission.
- Second digit specifies the group's permission.
- Third digit specifies the other's permission.

Permission are calculated by adding

4 (for read)

2 (for write)

1 (for execute)

0 implies no permission.

eg.

```
maya@nepalinux:~$ ls -l research.txt
-rwxr-xr-x 1 maya maya 42 2007-09-23 01:49 research.txt
maya@nepalinux:~$ chmod 700 research.txt
maya@nepalinux:~$ ls -l research.txt
-rwx----- 1 maya maya 42 2007-09-23 01:49 research.txt
```

Here Full permission is given only to the user. Groups and others are assigned 0, i.e no permission.

vi Editor Basics

vi is the most popular text editor in Linux.

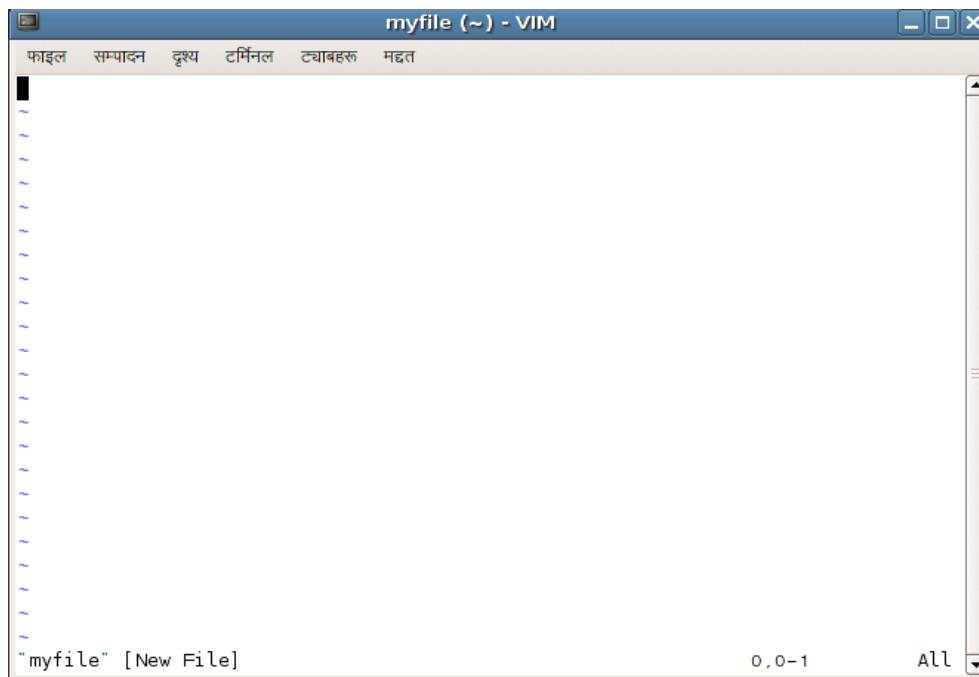
- **Starting vi**
vi <filename>

If the file exists, the file is opened and the contents are displayed and if the file doesn't exist, vi creates it when the edit are saved for the first time.

Three modes of vi

- **Command Mode**
Cursor Movement
Change, delete, yank, put and search
- **Insert Mode**
Type in new text
Return to command mode with <ESC>
- **ex Mode**
Configuring, exiting, and Saving
Search and replace

```
nepali@nepalinux:~$ vi myfile
```



empty line is shown as a tilde (~).

Entering Insert Mode -----> Press i

Leaving Insert Mode -----> hit the <Esc> key Esc key takes you from the insert mode back to the command mode.

Hint: when in trouble, press <Esc> and then press <Esc> again.

Saving and exiting mode

	Save Changes	Abandon Changes
Exit	:wq	:q or :q!
Don't Exit	:w	:e!

Forcing Changes:

	Save Changes
Exit	:wq!
Do not Exit	:w!

Yank (copy)

Searching for Text

press <Esc> and go to the ex mode.

/string -----> search down for the string

?string -----> search up for the string

n continues search in the same direction.
N continues search in the opposite direction.

Delete, Yank (copy) and Put (paste)

dd -----> delete line

5dd deletes 5 lines from the cursor.

yy -----> copy one line

5yy copies 5 lines.

p----->put the general buffer back after the current line

P-----> paste the line after the cursor.

Search and Replace

1, \$ ----> for changes to entire file.

8, 12 ---> change lines 8 through 12

Example

:1,\$s/kathmandu/pokhara/g -----> replace globally kathmandu by pokhar

Undoing changes.

u – undo most recent changes.

<Ctrl-r> redo last “undone” change.

Symbolic Link

A symbolic link points to another file.

ls -l displays the link name and referenced file.

To create symbolic link use ln -s command.

ln -s <filename> <link name>

eg.

```
maya@nepalinux:~$ ls -l research.txt
```

```
-rwx----- 1 maya maya 42 2007-09-23 01:49 research.txt
```

```
maya@nepalinux:~$ ln -s research.txt research_link
```

```
maya@nepalinux:~$ ls -l research_link
```

```
lrwxrwxrwx 1 maya maya 12 2007-09-23 06:19 research_link -> research.txt
```

Removing a symbolic link removes the actual link itself, not the actual file.

Archive and compression utilities in Linux

Archive

Archive places many files into one target file.

tar --- standard Linux archiving command.

Creating an archive

syntax.

```
tar cvf archive_name files .....
```

c creates a new archive

v produces verbose message

f archive_name is name of new file.

option don't need a leading dash.

eg.

```
maya@nepalinux:~$ tar cvf mytar.tar forensics research.txt
```

Inspecting Archives

syntax

```
tar tvf archive_name.tar
```

shows the lists of the files the tar file contains.

Extracting an Archives

Syntax.

```
tar xvf archive_name.tar
```

The archive will be extracted to the current directory. So first change to the target directory and then run the extracting command.

Compression Utilities

gzip, gunzip

standard Linux compression utility

bzip2, bunzip2

newer compression utility.

zip, unzip

eg. maya@nepalinux:~\$ gzip mytar.tar

When a filename is compressed with gzip , the compressed file is named with filename.gz. The file can be uncompressed using gunzip, recreating the original file.

bzip2 can also be used to compress files, and bzip2 to decompress. Similarly you can use zip and unzip.

Compressing Archives

Often tar archives are compress/uncompress

options

z for gzip compression

j for bzip2 compression

tar zcvf filename.tar.gz files.

to extract,

tar zxvf filename.tar.gz

Similar for bz2

tar jcvf filename.tar.bz2 files.

tar jxvf filename.tar.bz2

Checking Free Space

df

report file system disk space usage.

maya@nepalinux:~\$ df -h

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hda6	4.2G	3.6G	330M	92%	/
tmpfs	249M	4.0K	249M	1%	/dev/shm

```
/dev/hdb10      9.8G 2.3G 7.6G 23% /mnt/hdb10
/dev/hdb1       5.9G 5.5G 450M 93% /mnt/hdb1
```

-h displays sizes in easier to read units.

du

estimate file space usage

eg.

```
maya@nepalinux:~$ du -h forensics
4.0K  forensics/tools/research1/papers
8.0K  forensics/tools/research1
4.0K  forensics/tools/research2/papers
8.0K  forensics/tools/research2
20K   forensics/tools
24K   forensics
maya@nepalinux:~$ du -hs forensics
24K   forensics
maya@nepalinux:~$
```

-h human readable.

-s single directory summary

free

Display amount of free and used memory in the system

-m switch to megabytes.

Removable Media

CD/DVD and USB media,

Mount can be defined as making the physical partition available to the local file system. Unless you mount a partition or device, the system doesn't know the existence that partition or device.

By default USB pen drive and CD or DVD are automounted in NepaLinux i.e we don't have to mount it. To access them simply click them on the icon appeared on the Desktop (such as CD-ROM icon and pendrive Icon).

In any worst case if they are not mounted by default when you insert cd or pen drive then you have to mount it manually.

Any things you mount must be unmounted. You can unmount the the mounted CD/DVD and USB pen drive by right clicking the icon and clicking the “umount volume” for pendrive and “Eject” for CD/DVD.

Mount can be done by only the power user i.e root due to security reason.

mount command syntax

```
#mount <physical partion name> <the local path where you want to mount>
```

note:

after you mount the partition it must be unmounted, to do it, use umount command.

Syntax:umount <partion name> or <absolute path where the partition is mounted>

For pen drive mount /dev/sda1 /media/pen

or

```
mount /dev/sda2 /mnt/pen
```

for cd mount /dev/cdrom /media/cdrom

Mounting Windows partitions in NepaLinux.

If you want to mount your windows partitions it have to be done manually.

Listing Partion table

- **fdisk -l**

list the available partitions in your HardDrive, run the command fdisk -l as root user.

```
nepali@test:~$ su -
```

```
Password:
```

```
root@test:~# fdisk -l
```

```
Disk /dev/hda: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Device Boot    Start      End   Blocks  Id System
/dev/hda1      1        1216  9767488+ 83 Linux
/dev/hda2 *    1217      3040  14651280  c W95 FAT32 (LBA)
/dev/hda3      3041      3869   6658942+ 5 Extended
/dev/hda4      3870      4865   8000370 83 Linux
/dev/hda5      3041      3805   6144831 83 Linux
/dev/hda6      3806      3869   514048+ 82 Linux swap / Solaris
root@test:~#
```

The command fdisk -l list all the partitions of your HardDrive and the partitions types.

Here the physical partition /dev/hda2 is W95 FAT32 (LBA) type. i.e our windows partition. To mount it run the following commands.

```
root@test:~# mkdir /mnt/win
root@test:~# mount /dev/hda2 /mnt/win/
root@test:~#
```

```
root@test:win# umount /mnt/win/
umount: /mnt/win: device is busy
umount: /mnt/win: device is busy
root@test:win#
```

note that here we can't umount /mnt/win
the status show device is busy. Why?
Because we try to umount it by accessing it, i.e we are in the win directory and trying to umount it.

```
Just run pwd
root@test:win# pwd
/mnt/win
root@test:win#
```

so we must leave this directory.
Type cd to return to the home directory of the root user and run umount command.

```
root@test:win# cd
root@test:~# umount /mnt/win/
root@test:~#
```

The above command only mount the windows partitions temporarily that is , next time you boot your system you have to mount it again by running the command.

To mount it permanently you have to edit the file /etc/fstab. But take care while editing this file not to make any mistakes, otherwise your sytem may not boot in Linux.

```

/dev/hda1 / ext3 defaults 0 1
/dev/hda3 none swap sw 0 0
proc /proc proc defaults 0 0
/dev/fd0 /floppy auto user,noauto 0 0
/dev/cdrom /cdrom auto ro,user,noauto 0 0
none /sys sysfs defaults 0 0

/dev/hda2 /mnt/win vfat defaults 0 0

```

here to mount the windows partitions /dev/hda2 permanently I have added the line
/dev/hda2 /mnt/win vfat defaults 0 0
in fstab.

Here vfat is the file system type , defaults is the mount options and 0 and 0 are fsck and dump value

Fonts installation:

1. Open the terminal
2. Switch to the root user by the command su
3. Copy the fonts you want, to the directory /usr/share/fonts/
cp <fonts> /usr/share/fonts
4. then run the following comands
fc-cache -f
5. Verify the fonts installation by fc-list | grep <fonts name>

Network configuration

Assigning IP

edit the the file /etc/network/interfaces as root users.

```

#vi /etc/network/interfaces
add the following lines
auto eth0
iface eth0 inet static
address 192.168.2.9
netmask 255.255.255.0
gateway 192.168.2.1

```

(here we set an IP address 192.168.2.1, gateway 192.168.2.1 and netmask 255.255.255.0)
for DNS setting edit the file /etc/resolv.conf as root)

```
#vi /etc/resolv.conf
```

and add the following lines.

```
nameserver 192.168.2.1
```

(here it set the address 192.168.2.1 as your DNS address)

now run these commands as root.

```
#ifdown eth0
```

```
#ifup eth0
```

For assigning IP address through dhcp,
use ,
iface eth0 inet dhcp in network configuration file.

Ping

ping command is used to test whether there is network connectivity or not.

For eg. Ping 192.168.2.1 tests whether the host 192.168.2.1 is accessible or not.

Secure Shell

ssh

allows secure access to the remote system.

```
ssh 192.168.2.19 -l maya --> login to 192.168.2.19 with user ID maya.
```

Or simply ssh maya@192.168.2.19

To exit from the remote system type <Ctrl-d> or exit.

scp

Secure copy

syntax

```
scp [source] [destination]
```

for copying directory use -r option for recursive.

wget

retrieves files via HTTP and FTP.

eg. wget -c <http://www.forensicswiki.org/wiki/Scalpel>

File Sharing NFS and SAMBA

NFS

The Network File System (NFS) was developed to allow machines to mount a disk partition on a remote machine as if it were a local disk. It allows for fast, seamless sharing of files across a network.

For NFS you need to install `nfs-kernel-package`. If you have Internet connection simply run `apt-get update`, and `apt-get install nfs-kernel-package` as root user.

```
root@nepalinux:~# apt-get update
root@nepalinux:~# apt-get install nfs-kernel-server
```

NFS Server Configuration:

Command Line method (For advance users)

NFS exports from a server are controlled by the file `/etc/exports`. Each line begins with the absolute path of a directory to be exported, followed by a space-separated list of allowed clients.

An entry in `/etc/exports` will typically look like this:

```
directory machine1(option11,option12) machine2(option21,option22)
```

where

directory

the directory that you want to share.

machine1 and machine2

client machines that will have access to the directory. The machines may be listed by their DNS address or their IP address (e.g., *machine.company.com* or *192.168.0.8*). It can indicate a range of client using the subnet address(eg. *192.168.2.0/255.255.255.0*.)

Wildcard mask `*` can also be used to indicate the whole network client lists.

optionxx

the option listing for each machine will describe what kind of access that machine will have. Important options are:

- **ro:** The directory is shared read only; the client machine will not be able to write it. This is the default.
- **rw:** The client machine will have read and write access to the directory.

Examples:

```
/share 192.168.0.1(ro) 192.168.0.2(ro)
/data station1.mpp.org(rw)
/home *(rw)
```

/share directory is shared to the client 192.168.0.1 and 192.168.0.2 as read only.
 /data is shared to only one station i.e station1.mpp.org as read write.
 /home is shared to whole network as read write

Now run “`exportfs -a`” command as root users, or restart the `nfs-common` and `nfs-kernel-server` services.

```
nepali@nepalinux:~$ su -
Password:
root@nepalinux:~# cd /etc/init.d
root@nepalinux:~# ./nfs-common restart
root@nepalinux:~# ./nfs-kernel-server restart
root@nepalinux:~#
```

To find which files are exported shared in the remote server run the command “`showmount -e`”

eg.

```
root@nepalinux:~# showmount -e 192.168.2.9
Export list for 192.168.2.9:
/share *
```

NFS Client configuration:

For accessing the directories that is shared with the nfs in the remote server, you have to mount it in the local directory.

Syntax:

```
mount -t nfs <server's IP or Domain name>:/<shared directory name> <local mount point>
```

Here for eg to mount the /share directory in my local system I follow these steps.

```
nepali@nepalinux:~$ su -  
Password:  
root@nepalinux:~# mkdir /mnt/nfs/  
root@nepalinux:~# mount -t nfs 192.168.2.9:/share /mnt/nfs  
root@nepalinux:~#
```

For permanent mounting edit the fstab file and append the following lines.

```
192.168.2.9:/share /mnt/nfs nfs rw,soft,intr 0 0
```

SAMBA

Linux to windows and vice versa (samba)

Samba is Used for sharing files between Windows and Linux

Server configuration:

Edit the smb.conf file (/etc/samba/smb.conf) as root user.

For example if you want to share /share file to windows client , then put these portion at the bottom of the file.

```
[share]  
path = /share  
public = yes
```

To give write access to the directory you are sharing ,provide write permission to the others.

```
Chmod 777 /share
```

Restart your samba services.

```
root@nepalinux:~# cd /etc/init.d/  
root@nepalinux:~# ./samba restart  
Stopping Samba daemons: nmbd smbd.  
Starting Samba daemons: nmbd smbd.  
root@nepalinux:~#
```

In Windows to access the share just run and //IP

In Linux to access the samba share you can either run `smb://<ip>` in the Nautilus window or you mount manually as ,

```
mount -t smbfs //<ip>/<shared-name> <mount -point>
```

You may asked for the password, just press enter since we have shared the folder to public by indicating `public = yes` in the `smb.conf` file.

Searching file and directory.

For searching files and directories Linux have `find`, and `locate` command.

`Locate` command is faster but not efficient than that of `find` command to search for the files.

eg. `$ locate myfile.txt`

It searches the files that are updated in the database. So for new files to search update the database using the command `updatedb`.

Using find command.

Syntax,

```
find <dirs> [condition] [action]
```

eg.

```
find . -name "*.html"
```

It searches for the all files end with `.html`.

For additional `find` options, please refer to the man page.

Task Automation

Two facilities can be used to automate tasks.

`at`: It is used to execute a task at a specific time.

`cron`: It is used to execute tasks at scheduled intervals.

Using “at”:

It is useful for scheduling a single future event. The command “atq” is used to list defined jobs while “atrm” used to delete a defined job. Example:

```
at 21:00
```

```
at > mail ram < ram.msg
```

Note: Input can be ended with Ctrl + D

The /etc/at.allow and /etc/at.deny files can be used to allow or deny users to use “at”. The users listed in /etc/at.deny are not allowed to use “cron” and all other users are allowed. If file /etc/cron.allow exists, this supersedes cron.deny (ie, cron.deny is ignored) and allows only those listed in it to use “cron”.

Using cron:

It is useful for scheduling regularly **occurring** events. The crontab command can be used to edit, install and view the job schedules. The /etc/at.allow and /etc/at.deny files can be used in the same way as 'at' to allow or deny users to use cron.

Syntax

```
crontab [-u user] file
```

```
crontab [-l | -r | -e]
```

-u user to install crontab for user, who must be privileged to use

-e to create or edit the current crontab

-l to list the contents of the current crontab

-r to delete the current crontab

Each line of the file, consisting of six fields, is used for definition of one job. The first five fields define the time while the sixth field contains the command to run. This can contain any command or the full path of the shell script.

Format for the first five fields:

Field: Range

Minutes:0-59

Hours: 0-23

Day of Month: 1-31

Month: 0-12

Weekday: 0-7

A field may be an asterisk (*), which stands for every value in the corresponding field. In the weekday field, 0 or 7 is Sunday.

Example:

```
*/5 8-17 * * 1-5 fetchmail mailserver
```

From Monday to Friday every five minutes between 8:00 to 17:00, the command fetchmail is run to fetch incoming emails from the computer mailserver.

cron.hourly, cron.daily and cron.monthly for system jobs:

Most GNU/Linux distributions have three directories in /etc called *cron.hourly*, *cron.daily* and *cron.monthly*. These let the system administrator to run jobs on an hourly, daily or monthly basis. Simply by placing a shell script here, jobs can be carried out at those intervals. There is no need to have a crontab entry for these jobs.

grep

It is used to search through one or many files for a specific string and have the matching lines displayed. Some of the common options are:

-r: searches entire directory trees recursively

-i: ignores case

-v: gives all lines that do not contain the search string

Eg. `grep nepal /etc/testfile.conf`

CVS

Concurrent Versioning System

Generally cvs is used for version control and generally used in project.

Simple cvs command.

First of all export the CVSROOT variable.

```
export CVSROOT=:pserver:<username>@<server>:/<cvsrepo_name>
```

eg. `export CVSROOT=:pserver:daya@192.168.2.19:/cvsrep`

For exporting CVSROOT variable permanently edit .basrc file.

CVS checkout

go to the directory where you want to check out.

```
cvs co <model -name>
```

eg. `cvs co PAN/ubuntu/livecd/research`

To update the CVS run “cvs update” command.

and then to commit the changes you have made, run cvs ci command.

Eg. `cvscv -m "committed by daya" research2.txt`

Linux process.

Any program in execution is called process. Init is the first process started at boot time and always have PID 1.

Viewing processes

syntax,

`ps [options]`

generally aux option is used.

-a all process

-u users process

-x include process not controlled by terminal, such as daemon processes.

For getting information about the process running mozilla, run

Eg `ps aux | grep mozilla`

Sending signals to processes

Syntax,

`kill [-signal] [pids]`

`kill [-signal] %jobID`

eg.

`kill -9 3453`

`kill -l` list all the available signals.

Terminating a process

use `<Ctrl-c>` or use kill signal.

Running a process in a background.

To run a process in a background from the terminal windows, launch the program by appending an ampersand (&) to the end of a command.

eg.

`mozilla &`

Suspending a Process.

To suspend a process use `<Ctrl-z>`

Listing background and suspended jobs.

`jobs` command displays all the process running in the background.

Jobs number are referenced with %.

Resuming Suspended jobs.

Use fg command to foreground the jobs.

Syntax

fg [% job number]

bg [% job number]

bg is used to background the jobs.

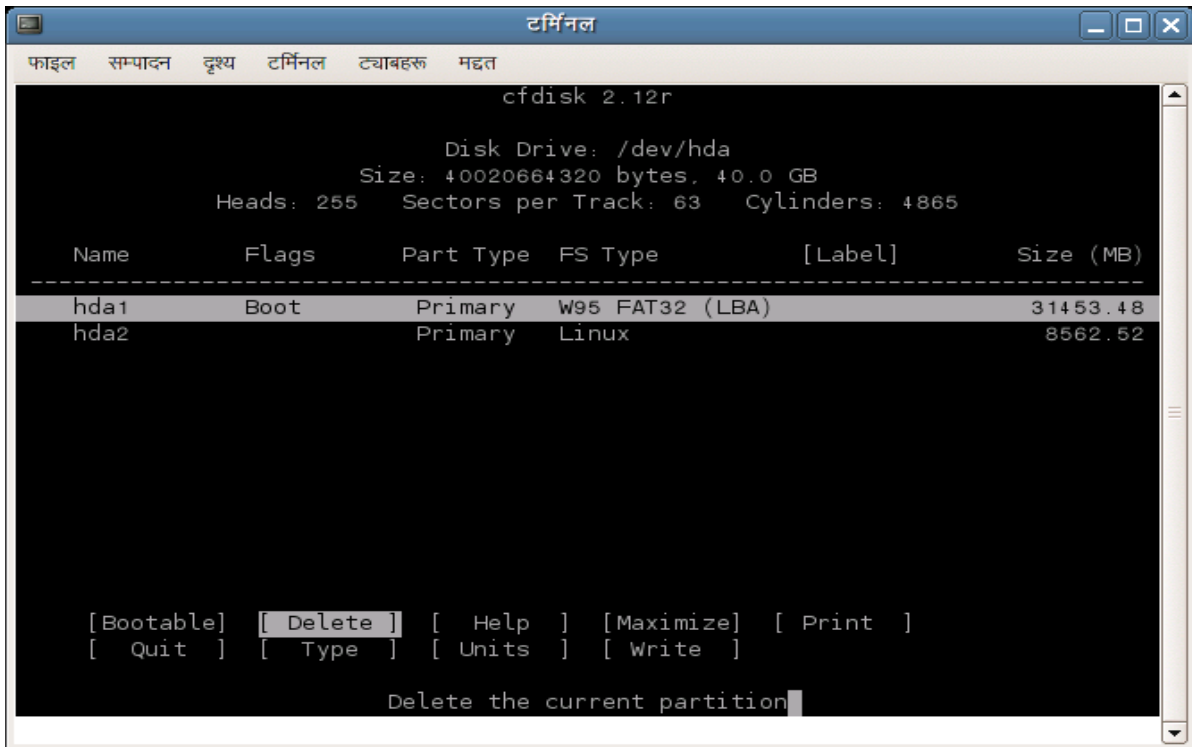
top

The top program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of tasks currently being managed by the Linux kernel. Simply type top command in the terminal window to run it. To kill any process type k and provide PID to exit from the top, process listing type q.

Creating and Deleting Partitions

To make a new partition or delete a new partition, you can use either cfdisk or fdisk command utility in Nepalinux. I prefer cfdisk as it is easy and more interactive than fdisk.

To run cfdisk open the terminal and switch to the root user, and type cfdisk command. A Terminal window is opened listing a partition table.



To delete a partition select the partition and go the Delete tab. Similarly to make a new partition select Free partition and go to the New tab.

Package management

Source packages versus binary packages

A source package is a package that is made up of completely source code. Source code is useful for programmers but must be built or compiled before it can be run. Building packages can take longer periods of time even on fast machines.

Binary packages are already compiled and are distributed in a ready-to-use manner.

Installing source package

For installing source packages, you have to compile it manually.

Packages in source code are either in .tar.gz or .tar.bz2 compressed archives.

The installation procedure for software that comes in tar.gz and tar.bz2 packages isn't always the same, but usually it's like this:

```
# tar xvzf package.tar.gz (or tar xvjf package.tar.bz2)
```

```
# cd package
# ./configure
# make
# make install
```

(for more information see README or INSTALL file)

Uninstalling

If you know where the program was installed, you'll have to manually delete the installed files or the directory where your program is. If you have no idea where all the files are, you'll have to read the Makefile and see where all the files got installed, and then delete them.

If you didn't delete your Makefile, you may be able to remove the program by doing a `make uninstall`:

```
root@puter: /home/me/dls/pkg# make uninstall
```

Installing binary packages

Installation of binary package is easy unlike the source packages. But the most harrasing things are dependecy problems often ending into displaying information such as the following library is missing , please install the library, etc.

There are two package managers in NepaLinux:

1. dpkg (debian package manager)
2. apt-get tool (APT – Advanced Packaging tool)

dpkg

The binary packages have .deb extensions. It is very easy to use dpkg tool. Just run the command

`dpkg -i <package.deb>` as root users to install a package.

`dpkg -i packagename-version -->` installs/upgrade the package, upgrades if not installed previously or the package is newer than the installed one.

`dpkg -r packagename -->` removes the package

`dpkg -l -->` Lists the installed packages

`dpkg -L packagename -->` Shows the files and directories created by package

`dpkg -s packagename -->` Shows the information of the package

`dpkg -S filename -->` Finds package owning the file

`dpkg -reconfigure packagename -->` Reconfigures the package's configuration

apt-get tool

APT is the *Advanced Package Tool* and provides the `apt-get` program. `apt-get` provides a simple way to retrieve and install packages from multiple sources using the command line. Unlike `dpkg`, `apt-get` does not understand `.deb` files, it works with the packages proper name and can only install `.deb` archives from a source specified in `/etc/apt/sources.list`. `apt-get` will call `dpkg` directly after downloading the `.deb` archives from the configured sources. For `apt-get` tool to run, you must have the Internet connection.

Editing the `/etc/apt/sources.list` file

open the `/etc/apt/sources.list` file in your favourite editor.

You get these lines

```
# deb http://ftp.debian.org/debian/ etch main contrib
# deb http://security.debian.org/ etch/updates main contrib
# deb-src ftp://ftp.debian.org/debian/ etch main contrib
```

just remove the comment sign (`#`) from the first line.

Then run the command `apt-get update` as root.

```
root@nepalinux:~# apt-get update
```

This command updates the list of the packages know by your system.

To install the `nis` package (for example) and all its dependencies, run

```
apt-get install nis
```

```
root@nepalinux:~# apt-get install nis
```

To remove the `nis` package from your system, run:

```
apt-get remove nis
```

```
root@nepalinux:~# apt-get remove nis
```

To remove the `nis` package and its configuration files from your system, run:

```
apt-get --purge remove nis
```

```
root@nepalinux:~# apt-get --purge remove nis
```

The `apt` tool suite also includes the `apt-cache` tool to query the package lists. You can use it to find packages providing specific functionality through simple text or regular expression queries and through queries of dependencies in the package management system. Some common ways to use `apt-cache` are:

To find packages whose description contain *word*:

`apt-cache search word`

- To print the detailed information of a package:

`apt-cache show package`

- To print the packages a given package depends on:

`apt-cache depends package`

- To print detailed information of the versions available for a package and the packages that reverse-depend on it:

`apt-cache showpkg package`

NepaLinux Basic Troubleshooting

a. Changing the root password.

If you forgot the root password, or if your root password is lost, you can change it through the single user mode.

- While computer boot in the Grub menu press `e` to edit the grub menu.
- In the kernel section edit and append `rw init=/bin/bash`
- Then you are provided with the shell prompt.
- Just issue the `passwd` command to change the root password.

b. Installing grub in MBR

When you install Windows system after you install NepaLinux, By default Windows system, doesn't keep entry in its MBR for Linux. GRUB keeps the entry for both windows and Linux.

To install MBR, you need NepaLinux Live CD. After the NepaLinux Live CD/DVD is fully loaded.

1. Open the terminal window.
2. Switch to the root user, `su -` (password of root for LiveCD is nepali)
3. Then mount the partition which contains the NepaLinux root partition. For eg. In `/mnt` (Identify the NepaLinux root partition by `fdisk -l` command)
4. `chroot` to the `/mnt` command **chroot /mnt** does it.
5. Then issue the command `grub-install /dev/hda`.

GRUB can be installed in the MBR by another method also.

1. Open the terminal
2. switch to the root user su -
3. type the command **grub**
4. you are prompted gurb> command prompt.
5. Identify the root partition of the NepaLinux (fdisk -l command)
6. If for eg. Your NepaLinux partition is /dev/hda3)
7. type the command root (hd0,2) Note 0---> a , and 2 ----> 2+1. i.e hda3, for hda5
you have to enter root (hd0,4).
8. issue the command setup (hd0).
9. Type quit

Acknowledgement

The PAN L10n Project works have been carried out with the aid of a grant from the International Development Research Centre, Ottawa, Canada, administered through the Center for Research in Urdu Language Processing (CRLUP), National University of Computing and Emerging Sciences, Lahore, Pakistan (NUCES).