Unix/Linux Command Reference

File Commands		
1.	ls	Directory listing
2.	ls -al	Formatted listing with hidden files
3.	ls -lt	Sorting the Formatted listing by time modification
4.	cd dir	Change directory to dir
5.	cd	Change to home directory
6.	pwd	Show current working directory
7.	mkdir dir	Creating a directory dir
8.	cat >file	Places the standard input into the file
9.	more file	Output the contents of the file
10.	head file	Output the first 10 lines of the file
11.	tail file	Output the last 10 lines of the file
12.	tail -f file	Output the contents of file as it grows,starting with the last 10 lines
13.	touch file	Create or update file
14.	rm file	Deleting the file
15.	rm -r dir	Deleting the directory
16.	rm -f file	Force to remove the file
17.	rm -rf dir	Force to remove the directory dir
18.	cp file1 file2	Copy the contents of file1 to file2
19.	cp -r dir1 dir2	Copy dir1 to dir2;create dir2 if not present
20.	mv file1 file2	Rename or move file1 to file2,if file2 is an existing directory
21.	ln -s file link	Create symbolic link link to file
Process management		
1.	ps	To display the currently working processes
2.	top	Display all running process

3.	kill pid	Kill the process with given pid
4.	killall proc	Kill all the process named proc
5.	pkill pattern	Will kill all processes matching the pattern
6.	bg	List stopped or background jobs,resume a stopped job in the background
7.	fg	Brings the most recent job to foreground
8.	fg n	Brings job n to the foreground
File permission		
1.	chmod octal file	Change the permission of file to octal,which can be found separately for user,group,world by adding, • 4-read(r) • 2-write(w) • 1-execute(x)
Searching		
1.	grep pattern file	Search for pattern in file
2.	grep -r pattern dir	Search recursively for pattern in dir
3.	command | grep pattern	Search pattern in the output of a command
4.	locate file	Find all instances of file
5.	find . -name filename	Searches in the current directory (represented by a period) and below it, for files and directories with names starting with filename
6.	pgrep pattern	Searches for all the named processes , that matches with the pattern and, by default, returns their ID
System Info		
1.	date	Show the current date and time
2.	cal	Show this month's calender
3.	uptime	Show current uptime
4.	w	Display who is on line
5.	whoami	Who you are logged in as

6.	finger user	Display information about user
7.	uname -a	Show kernel information
8.	cat /proc/cpuinfo	Cpu information
9.	cat proc/meminfo	Memory information
10.	man command	Show the manual for command
11.	df	Show the disk usage
12.	du	Show directory space usage
13.	free	Show memory and swap usage
14.	whereis app	Show possible locations of app
15.	which app	Show which applications will be run by default
Compression		
1.	tar cf file.tar file	Create tar named file.tar containing file
2.	tar xf file.tar	Extract the files from file.tar
3.	tar czf file.tar.gz files	Create a tar with Gzip compression
4.	tar xzf file.tar.gz	Extract a tar using Gzip
5.	tar cjf file.tar.bz2	Create tar with Bzip2 compression
6.	tar xjf file.tar.bz2	Extract a tar using Bzip2
7.	gzip file	Compresses file and renames it to file.gz
8.	gzip -d file.gz	Decompresses file.gz back to file
Network		
1.	ping host	Ping host and output results
2.	whois domain	Get whois information for domains
3.	dig domain	Get DNS information for domain
4.	dig -x host	Reverse lookup host
5.	wget file	Download file
6.	wget -c file	Continue a stopped download

Shortcuts		
1.	ctrl+c	Halts the current command
2.	ctrl+z	Stops the current command, resume with fg in the foreground or bg in the background
3.	ctrl+d	Logout the current session, similar to exit
4.	ctrl+w	Erases one word in the current line
5.	ctrl+u	Erases the whole line
6.	ctrl+r	Type to bring up a recent command
7.	!!	Repeats the last command
8.	exit	Logout the current session

" Linux Utilities"

Introduction to Linux

Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released 5 October 1991 by Linus Torvalds.

Linux Advantages

1. **Low cost:** You don't need to spend time and money to obtain licenses since Linux and much of its software come with the GNU General Public License. You can start to work immediately without worrying that your software may stop working anytime because the free trial version expires. Additionally, there are large repositories from which you can freely download high quality software for almost any task you can think of.
2. **Stability:** Linux doesn't need to be rebooted periodically to maintain performance levels. It doesn't freeze up or slow down over time due to memory leaks and such. Continuous up-times of hundreds of days (up to a year or more) are not uncommon.
3. **Performance:** Linux provides persistent high performance on workstations and on networks. It can handle unusually large numbers of users simultaneously, and can make old computers sufficiently responsive to be useful again.
4. **Network friendliness:** Linux was developed by a group of programmers over the Internet and has therefore strong support for network functionality; client and server systems can be easily set up on any computer running Linux. It can perform tasks such as network backups faster and more reliably than alternative systems.
5. **Flexibility:** Linux can be used for high performance server applications, desktop applications, and embedded systems. You can save disk space by only installing the components needed for a particular use. You can restrict the use of specific computers by installing for example only selected office applications instead of the whole suite.
6. **Compatibility:** It runs all common Unix software packages and can process all common file formats.
7. **Choice:** The large number of Linux distributions gives you a choice. Each distribution is developed and supported by a different organization. You can pick the one you like best; the core functionalities are the same; most software runs on most distributions.
8. **Fast and easy installation:** Most Linux distributions come with user-friendly installation and setup programs. Popular Linux distributions come with tools that make installation of additional software very user friendly as well.
9. **Full use of hard disk:** Linux continues work well even when the hard disk is almost full.
10. **Multitasking:** Linux is designed to do many things at the same time; e.g., a large printing job in the background won't slow down your other work.
11. **Security:** Linux is one of the most secure operating systems. "Walls" and flexible file access permission systems prevent access by unwanted visitors or viruses. Linux users have to option to select and safely download software, free of charge, from online

repositories containing thousands of high quality packages. No purchase transactions requiring credit card numbers or other sensitive personal information are necessary.

12. **Open Source:** If you develop software that requires knowledge or modification of the operating system code, Linux's source code is at your fingertips. Most Linux applications are Open Source as well.

File Handling utilities:

cat COMMAND:

cat linux command concatenates files and print it on the standard output.

SYNTAX:

The Syntax is

cat [OPTIONS] [FILE]...

OPTIONS:

-A	Show all.
-b	Omits line numbers for blank space in the output.
-e	A $ character will be printed at the end of each line prior to a new line.
-E	Displays a $ (dollar sign) at the end of each line.
-n	Line numbers for all the output lines.
-s	If the output has multiple empty lines it replaces it with one empty line.
-T	Displays the tab characters in the output.
-v	Non-printing characters (with the exception of tabs, new-lines and form-feeds) are printed visibly.

EXAMPLE:

1. To Create a new file:

```
cat > file1.txt
```

This command creates a new file file1.txt. After typing into the file press control+d (^d) simultaneously to end the file.

2. To Append data into the file:

   cat >> file1.txt

   To append data into the same file use append operator >> to write into the file, else the file will be overwritten (i.e., all of its contents will be erased).

3. To display a file:

   cat file1.txt

   This command displays the data in the file.

4. To concatenate several files and display:

   cat file1.txt file2.txt

   The above cat command will concatenate the two files (file1.txt and file2.txt) and it will display the output in the screen. Some times the output may not fit the monitor screen. In such situation you can print those files in a new file or display the file using less command.

   cat file1.txt file2.txt | less

5. To concatenate several files and to transfer the output to another file.

   cat file1.txt file2.txt > file3.txt

   In the above example the output is redirected to new file file3.txt. The cat command will create new file file3.txt and store the concatenated output into file3.txt.

rm COMMAND:

rm linux command is used to remove/delete the file from the directory.

SYNTAX:

The Syntax is

rm [options..] [file | directory]

OPTIONS:

-f	Remove all files in a directory without prompting the user.
-i	Interactive. With this option, rm prompts for confirmation before removing any files.
-r (or) -R	Recursively remove directories and subdirectories in the argument list. The directory will be emptied of files and removed. The user is normally prompted for removal of any write-protected files which the directory contains.

EXAMPLE:

1. To Remove / Delete a file:

   rm file1.txt

   Here rm command will remove/delete the file file1.txt.

2. To delete a directory tree:

   rm -ir tmp

   This rm command recursively removes the contents of all subdirectories of the tmp directory, prompting you regarding the removal of each file, and then removes the tmp directory itself.

3. To remove more files at once

   rm file1.txt file2.txt

   rm command removes file1.txt and file2.txt files at the same time.

cd COMMAND:

cd command is used to change the directory.

SYNTAX:

The Syntax is

cd [directory | ~ | ./ | ../ | -]

OPTIONS:

-L Use the physical directory structure.

-P Forces symbolic links.

EXAMPLE:

1. cd linux-command

   This command will take you to the sub-directory(linux-command) from its parent directory.

2. cd **..**

   This will change to the parent-directory from the current working directory/sub-directory.

3. cd ~

   This command will move to the user's home directory which is "/home/username".

cp COMMAND:

cp command copy files from one location to another. If the destination is an existing file, then the file is overwritten; if the destination is an existing directory, the file is copied into the directory (the directory is not overwritten).

SYNTAX:

The Syntax is

cp [OPTIONS]... SOURCE DEST

cp [OPTIONS]... SOURCE... DIRECTORY

cp [OPTIONS]... --target-directory=DIRECTORY SOURCE...

OPTIONS:

-a	same as -dpR.
--backup[=CONTROL]	make a backup of each existing destination file
-b	like --backup but does not accept an argument.
-f	if an existing destination file cannot be opened, remove it and try again.
-p	same as --preserve=mode,ownership,timestamps.
--preserve[=ATTR_LIST]	preserve the specified attributes (default: mode,ownership,timestamps) and security contexts, if possible additional attributes: links, all.
--no-preserve=ATTR_LIST	don't preserve the specified attribute.
--parents	append source path to DIRECTORY.

EXAMPLE:

1. Copy two files:

   cp file1 file2

   The above cp command copies the content of file1.php to file2.php.

2. To backup the copied file:

   cp -b file1.php file2.php

   Backup of file1.php will be created with '~' symbol as file2.php~.

3. Copy folder and subfolders:

   cp -R scripts scripts1

The above cp command copy the folder and subfolders from scripts to scripts1.

ls COMMAND:

ls command lists the files and directories under current working directory.

SYNTAX:

The Syntax is

ls [OPTIONS]... [FILE]

OPTIONS:

-l	Lists all the files, directories and their mode, Number of links, owner of the file, file size, Modified date and time and filename.
-t	Lists in order of last modification time.
-a	Lists all entries including hidden files.
-d	Lists directory files instead of contents.
-p	Puts slash at the end of each directories.
-u	List in order of last access time.
-i	Display inode information.
-ltr	List files order by date.
-lSr	List files order by file size.

EXAMPLE:

1. Display root directory contents:

   ls /

   lists the contents of root directory.

2. Display hidden files and directories:

   ls -a

   lists all entries including hidden files and directories.

3. Display inode information:

   ls -i

   7373073 book.gif

   7373074 clock.gif

   7373082 globe.gif

   7373078 pencil.gif

   7373080 child.gif

   7373081 email.gif

   7373076 indigo.gif

   The above command displays filename with inode value.

ln COMMAND:

ln command is used to create link to a file (or) directory. It helps to provide soft link for desired files. Inode will be different for source and destination.

SYNTAX:

The Syntax is

ln [options] existingfile(or directory)name newfile(or directory)name

OPTIONS:

-f	Link files without questioning the user, even if the mode of target forbids writing. This is the default if the standard input is not a terminal.
-n	Does not overwrite existing files.
-s	Used to create soft links.

EXAMPLE:

1. ln -s file1.txt file2.txt

   Creates a symbolic link to 'file1.txt' with the name of 'file2.txt'. Here inode for 'file1.txt' and 'file2.txt' will be different.

2. ln -s nimi nimi1

Creates a symbolic link to 'nimi' with the name of 'nimi1'.

chown COMMAND:

chown command is used to change the owner / user of the file or directory. This is an admin command, root user only can change the owner of a file or directory.

SYNTAX:

The Syntax is

chown [options] newowner filename/directoryname

OPTIONS:

-R	Change the permission on files that are in the subdirectories of the directory that you are currently in.
-c	Change the permission for each file.
-f	Prevents chown from displaying error messages when it is unable to change the ownership of a file.

EXAMPLE:

1. chown hiox test.txt

   The owner of the 'test.txt' file is root, Change to new user hiox.

2. chown -R hiox test

   The owner of the 'test' directory is root, With -R option the files and subdirectories user also gets changed.

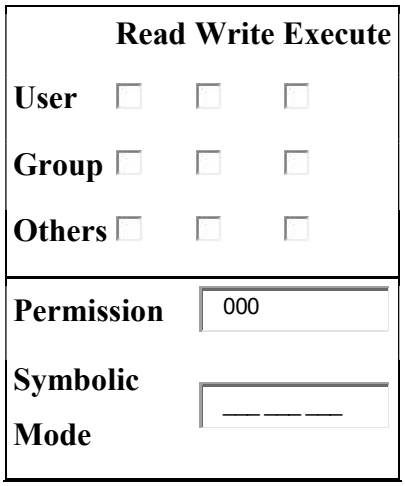
3. chown -c hiox calc.txt

   Here change the owner for the specific 'calc.txt' file only.

chmod COMMAND:

chmod command allows you to alter / Change access rights to files and directories.

File Permission is given for users,group and others as,



SYNTAX:

The Syntax is

chmod [options] [MODE] FileName

File Permission

#	File Permission
0	none
1	execute only
2	write only
3	write and execute
4	read only
5	read and execute
6	read and write
7	set all permissions

OPTIONS:

-c	Displays names of only those files whose permissions are being changed

-f Suppress most error messages

-R Change files and directories recursively

-v Output version information and exit.

EXAMPLE:

1. To view your files with what permission they are:

   ls -alt

   This command is used to view your files with what permission they are.

2. To make a file readable and writable by the group and others.

   chmod 066 file1.txt

3. To allow everyone to read, write, and execute the file

   chmod 777 file1.txt

mkdir COMMAND:

mkdir command is used to create one or more directories.

SYNTAX:

The Syntax is

mkdir [options] directories

OPTIONS:

-m Set the access mode for the new directories.

-p Create intervening parent directories if they don't exist.

-v Print help message for each directory created.

EXAMPLE:

1. Create directory:

mkdir test

The above command is used to create the directory 'test'.

2. Create directory and set permissions:

   mkdir -m 666 test

   The above command is used to create the directory 'test' and set the read and write permission.

rmdir COMMAND:

rmdir command is used to delete/remove a directory and its subdirectories.

SYNTAX:

The Syntax is

rmdir [options..] Directory

OPTIONS:

-p	Allow users to remove the directory dirname and its parent directories which become empty.

EXAMPLE:

1. To delete/remove a directory

   rmdir tmp

   rmdir command will remove/delete the directory tmp if the directory is empty.

2. To delete a directory tree:

   rm -ir tmp

This command recursively removes the contents of all subdirectories of the tmp directory, prompting you regarding the removal of each file, and then removes the tmp directory itself.

mv COMMAND:

mv command which is short for move. It is used to move/rename file from one directory to another. mv command is different from cp command as it completely removes the file from the source and moves to the directory specified, where cp command just copies the content from one file to another.

SYNTAX:

The Syntax is

mv [-f] [-i] oldname newname

OPTIONS:

-f	This will not prompt before overwriting (equivalent to --reply=yes). mv -f will move the file(s) without prompting even if it is writing over an existing target.
-i	Prompts before overwriting another file.

EXAMPLE:

1. To Rename / Move a file:

   mv file1.txt file2.txt

   This command renames file1.txt as file2.txt

2. To move a directory

   mv hscripts tmp

   In the above line mv command moves all the files, directories and sub-directories from hscripts folder/directory to tmp directory if the tmp directory already exists. If there is no tmp directory it rename's the hscripts directory as tmp directory.

3. To Move multiple files/More files into another directory

   mv file1.txt tmp/file2.txt newdir

   This command moves the files file1.txt from the current directory and file2.txt from the tmp folder/directory to newdir.

   .

About wc

Short for word count, wc displays a count of lines, words, and characters in a file.

Syntax

wc [-c | -m | -C] [-l] [-w] [file ...]

-c	Count bytes.
-m	Count characters.
-C	Same as -m.
-l	Count lines.
-w	Count words delimited by white space characters or new line characters. Delimiting characters are Extended Unix Code (EUC) characters from any code set defined by iswspace()
File	Name of file to word count.

Examples

wc myfile.txt - Displays information about the file myfile.txt. Below is an example of the output.

5 13 57 myfile.txt

5 = Lines

13 = Words

57 = Characters

About split

touch newfile.txt

Creates a file known as "newfile.txt", if the file does not already exist. If the file already exists the accessed / modification time is updated for the file newfile.txt

About comm

Select or reject lines common to two files.

Syntax

comm [-1] [-2] [-3] file1 file2

-1	Suppress the output column of lines unique to file1.
-2	Suppress the output column of lines unique to file2.
-3	Suppress the output column of lines duplicated in file1 and file2.
file1	Name of the first file to compare.
file2	Name of the second file to compare.

Examples

comm myfile1.txt myfile2.txt

The above example would compare the two files myfile1.txt and myfile2.txt.

Process utilities:

ps COMMAND:

ps command is used to report the process status. ps is the short name for Process Status.

SYNTAX:

The Syntax is

ps [options]

OPTIONS:

-a	List information about all processes most frequently requested: all those except process group leaders and processes not associated with a terminal..
-A or e	List information for all processes.
-d	List information about all processes except session leaders.
-e	List information about every process now running.
-f	Generates a full listing.
-j	Print session ID and process group ID.
-l	Generate a long listing.

EXAMPLE:

1. ps

   Output:

   ```
   PID TTY          TIME CMD
   2540 pts/1    00:00:00 bash
   2621 pts/1    00:00:00 ps
   ```

   In the above example, typing ps alone would list the current running processes.

2. ps -f

   Output:

   ```
   UID        PID  PPID  C STIME TTY          TIME CMD
   nirmala   2540  2536  0 15:31 pts/1    00:00:00 bash
   ```

nirmala 2639 2540 0 15:51 pts/1 00:00:00 ps -f

Displays full information about currently running processes.

kill COMMAND:

kill command is used to kill the background process.

SYNTAX:

The Syntax is

kill [-s] [-l] %pid

OPTIONS:

-s	Specify the signal to send. The signal may be given as a signal name or number.
-l	Write all values of signal supported by the implementation, if no operand is given.
-pid	Process id or job id.
-9	Force to kill a process.

EXAMPLE:

Step by Step process:

- Open a process music player.
  xmms

  press ctrl+z to stop the process.

- To know group id or job id of the background task.
  jobs -l
- It will list the background jobs with its job id as,
- xmms 3956
  kmail 3467

- To kill a job or process.
  kill 3956

  kill command kills or terminates the background process xmms.

Filters:

more COMMAND:

more command is used to display text in the terminal screen. It allows only backward movement.

SYNTAX:

The Syntax is

more [options] filename

OPTIONS:

-c	Clear screen before displaying.
-e	Exit immediately after writing the last line of the last file in the argument list.
-n	Specify how many lines are printed in the screen for a given file.
+n	Starts up the file from the given number.

EXAMPLE:

1. more -c index.php

   Clears the screen before printing the file .

2. more -3 index.php

   Prints first three lines of the given file. Press **Enter** to display the file line by line.

head COMMAND:

head command is used to display the first ten lines of a file, and also specifies how many lines to display.

SYNTAX:

The Syntax is

head [options] filename

OPTIONS:

-n	To specify how many lines you want to display.
-n number	The number option-argument must be a decimal integer whose sign affects the location in the file, measured in lines.
-c number	The number option-argument must be a decimal integer whose sign affects the location in the file, measured in bytes.

EXAMPLE:

1. head index.php

   This command prints the first 10 lines of 'index.php'.

2. head -5 index.php

   The head command displays the first 5 lines of 'index.php'.

3. head -c 5 index.php

   The above command displays the first 5 characters of 'index.php'.

tail COMMAND:

tail command is used to display the last or bottom part of the file. By default it displays last 10 lines of a file.

SYNTAX:

The Syntax is

tail [options] filename

OPTIONS:

-l	To specify the units of lines.
-b	To specify the units of blocks.
-n	To specify how many lines you want to display.
-c number	The number option-argument must be a decimal integer whose sign affects the location in the file, measured in bytes.
-n number	The number option-argument must be a decimal integer whose sign affects the location in the file, measured in lines.

EXAMPLE:

1. tail index.php

   It displays the last 10 lines of 'index.php'.

2. tail -2 index.php

   It displays the last 2 lines of 'index.php'.

3. tail -n 5 index.php

   It displays the last 5 lines of 'index.php'.

4. tail -c 5 index.php

   It displays the last 5 characters of 'index.php'.

cut COMMAND:

cut command is used to cut out selected fields of each line of a file. The cut command uses delimiters to determine where to split fields.

SYNTAX:

The Syntax is

cut [options]

OPTIONS:

-c Specifies character positions.

-b Specifies byte positions.

-d flags Specifies the delimiters and fields.

EXAMPLE:

1. cut -c1-3 text.txt

   Output:

   Thi

   Cut the first three letters from the above line.

2. cut -d, -f1,2 text.txt

   Output:

   This is, an example program

   The above command is used to split the fields using delimiter and cut the first two fields.

paste COMMAND:

paste command is used to paste the content from one file to another file. It is also used to set column format for each line.

SYNTAX:

The Syntax is

paste [options]

OPTIONS:

-s Paste one file at a time instead of in parallel.

-d Reuse characters from LIST instead of TABs .

EXAMPLE:

1. paste test.txt>test1.txt

   Paste the content from 'test.txt' file to 'test1.txt' file.

2. ls | paste - - - -

   List all files and directories in four columns for each line.

sort COMMAND:

sort command is used to sort the lines in a text file.

SYNTAX:

The Syntax is

sort [options] filename

OPTIONS:

-r	Sorts in reverse order.
-u	If line is duplicated display only once.
-o filename	Sends sorted output to a file.

EXAMPLE:

1. sort test.txt

   Sorts the 'test.txt'file and prints result in the screen.

2. sort -r test.txt

   Sorts the 'test.txt' file in reverse order and prints result in the screen.

General Commands:

date COMMAND:

date command prints the date and time.

SYNTAX:

The Syntax is

date [options] [+format] [date]

OPTIONS:

-a	Slowly adjust the time by sss.fff seconds (fff represents fractions of a second). This adjustment can be positive or negative.Only system admin/ super user can adjust the time.
-s date-string	Sets the time and date to the value specfied in the datestring. The datestr may contain the month names, timezones, 'am', 'pm', etc.
-u	Display (or set) the date in Greenwich Mean Time (GMT-universal time).

Format:

%a	Abbreviated weekday(Tue).
%A	Full weekday(Tuesday).
%b	Abbreviated month name(Jan).
%B	Full month name(January).
%c	Country-specific date and time format..
%D	Date in the format %m/%d/%y.
%j	Julian day of year (001-366).
%n	Insert a new line.
%p	String to indicate a.m. or p.m.
%T	Time in the format %H:%M:%S.
%t	Tab space.

%V Week number in year (01-52); start week on Monday.

EXAMPLE:

1. date command

   date

   The above command will print Wed Jul 23 10:52:34 IST 2008

2. To use tab space:

   date +"Date is %D %t Time is %T"

   The above command will remove space and print as Date is 07/23/08 Time is 10:52:34

3. To know the week number of the year,

   date -V

   The above command will print 30

4. To set the date,

   date -s "10/08/2008 11:37:23"

   The above command will print Wed Oct 08 11:37:23 IST 2008

who COMMAND:

who command can list the names of users currently logged in, their terminal, the time they have been logged in, and the name of the host from which they have logged in.

SYNTAX:

The Syntax is

who [options] [file]

OPTIONS:

am i	Print the username of the invoking user, The 'am' and 'i' must be space separated.
-b	Prints time of last system boot.
-d	print dead processes.
-H	Print column headings above the output.
-i	Include idle time as HOURS:MINUTES. An idle time of . indicates activity within the last minute.
-m	Same as who am i.
-q	Prints only the usernames and the user count/total no of users logged in.
-T,-w	Include user's message status in the output.

EXAMPLE:

1. who -uH

   Output:

   NAME LINE TIME IDLE PID COMMENT
   hiox ttyp3 Jul 10 11:08 . 4578

   This sample output was produced at 11 a.m. The "." indiacates activity within the last minute.

2. who am i

   who am i command prints the user name.

echo COMMAND:

echo command prints the given input string to standard output.

SYNTAX:

The Syntax is

echo [options..] [string]

OPTIONS:

-n	do not output the trailing newline
-e	enable interpretation of the backslash-escaped characters listed below
-E	disable interpretation of those sequences in STRINGs

Without -E, the following sequences are recognized and interpolated:

\NNN	the character whose ASCII code is NNN (octal)
\a	alert (BEL)
\\	backslash
\b	backspace
\c	suppress trailing newline
\f	form feed
\n	new line
\r	carriage return
\t	horizontal tab
\v	vertical tab

EXAMPLE:

1. echo command

   echo "hscripts Hiox India"

   The above command will print as hscripts Hiox India

2. To use backspace:

   echo -e "hscripts \bHiox \bIndia"

   The above command will remove space and print as hscriptsHioxIndia

3. To use tab space in echo command

   echo -e "hscripts\tHiox\tIndia"

   The above command will print as hscripts Hiox India

passwd COMMAND:

passwd command is used to change your password.

SYNTAX:

The Syntax is

passwd [options]

OPTIONS:

-a	Show password attributes for all entries.
-l	Locks password entry for name.
-d	Deletes password for name. The login name will not be prompted for password.
-f	Force the user to change password at the next login by expiring the password for name.

EXAMPLE:

1. passwd

   Entering just passwd would allow you to change the password. After entering passwd you will receive the following three prompts:

   Current Password:
   New Password:
   Confirm New Password:

   Each of these prompts must be entered correctly for the password to be successfully changed.

pwd COMMAND:

pwd - Print Working Directory. pwd command prints the full filename of the current working directory.

SYNTAX:

The Syntax is

pwd [options]

OPTIONS:

-P The pathname printed will not contain symbolic links.

-L The pathname printed may contain symbolic links.

EXAMPLE:

1. Displays the current working directory.

   pwd

   If you are working in home directory then, pwd command displays the current working directory as /home.

cal COMMAND:

cal command is used to display the calendar.

SYNTAX:

The Syntax is

cal [options] [month] [year]

OPTIONS:

-1 Displays single month as output.

-3 Displays prev/current/next month output.

-s Displays sunday as the first day of the week.

-m Displays Monday as the first day of the week.

-j Displays Julian dates (days one-based, numbered from January 1).

-y Displays a calendar for the current year.

EXAMPLE:

1. cal

   Output:

   September 2008
   Su Mo Tu We Th Fr Sa
   1 2 3 4 5 6
   7 8 9 10 11 12 13
   14 15 16 17 18 19 20
   21 22 23 24 25 26 27
   28 29 30

   cal command displays the current month calendar.

2. cal -3 5 2008

   Output:

   April 2008 May 2008 June 2008
   Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
   1 2 3 4 5 1 2 3 1 2 3 4 5 6 7
   6 7 8 9 10 11 12 4 5 6 7 8 9 10 8 9 10 11 12 13 14
   13 14 15 16 17 18 19 11 12 13 14 15 16 17 15 16 17 18 19 20 21
   20 21 22 23 24 25 26 18 19 20 21 22 23 24 22 23 24 25 26 27 28
   27 28 29 30 25 26 27 28 29 30 31 29 30

   Here the cal command displays the calendar of April, May and June month of year 2008.