
Objective:**1. Study of Unix/Linux command list:**

cd, cal ,cat, banner, touch, man, who, cat, cp, ls, mv, rm, mkdir, rmdir, echo, more, date, history, chmod, pwd, wc, sort, cut, paste, logout, shutdown.

- a) cd - To change directory - change the current working directory to a specific Folder.

Description:

If directory is given, changes the shell's working directory to directory. If not, changes to HOME (shell variable). If the shell variable CDPATH exists, it is used as a search path. If directory begins with a slash, CDPATH is not used. If directory is '-', this will change to the previous directory location (equivalent to \$OLDPWD). The return status is zero if the directory is successfully changed, non-zero otherwise.

Options:

Tag	Description
-P	Do not follow symbolic links
-L	Follow symbolic links (default)

Procedure:

1. Open terminal and create a directory
2. change the directory by using the cd command
3. check whether directory changed or not using pwd

Examples:

```
Move to the sample folder  
$ cd /usr/local/sample  
$ pwd  
/usr/local/sample  
Change to another folder  
$ cd /var/local/logs  
$ pwd  
/var/local/logs  
Quickly get back  
$ cd -  
$ pwd  
/usr/local/sample  
Move up one folder  
$ cd ..  
$ pwd  
/usr/local/
```

Get back to original location

```
$ cd  
$ pwd  
/usr/local/
```

- b) cal - To display a calendar

Syntax: cal [-m] [month] [year]

Description:

A single parameter specifies the 4 digit year (1 - 9999) to be displayed. Two parameters denote the Month (1 - 12) and Year (1 - 9999). If arguments are not specified, the current month is displayed. A year starts on 01 Jan.

Options:

Tag	Description
-m	Display monday as the first day of the week.
-j	Display julian dates (days one-based, numbered from January 1).
-y	Display a calendar for the current year.

Procedure:

- 1.Open terminal and enter cal command
- 2.It will display the calendar

Examples:

To display current month's calendar
\$ cal
April 2016
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
To display feb 2015 calendar
\$ cal 2 2015
February 2015
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
To display complete year calendar.
\$ cal -y

c) cat - Concatenate and print the content of files.

Syntax:

cat [Options] [File]...

Description:

Cat command concatenate FILE(s), or standard input, to standard output. With no FILE, or when FILE is -, it reads standard input.

Options:

Tag	Description
-A, --show-all	equivalent to -Vet
-b, --number-nonblank	number nonblank output lines
-e	equivalent to -vE
-E, --show-ends	display \$ at end of each line
-n, --number	number all output lines
-s, --squeeze-blank	never more than one single blank line
-t	equivalent to -vT
-T,--show-tabs	display TAB characters as ^I
-u	(ignored)
-v, --show-nonprinting	use ^ and M- notation, except for LFD and TAB. display this help and exit
--help	display this help and exit
--version	output version information and exit

Procedure:

- 1.Create a file using cat command
- 2.check the file created or not using cat command

Examples:

Create two sample files
#sample.txt
This is a sample text file
#sample1.txt
This is another sample text file
To display content of a file.
\$ cat sample.txt
This is a sample text file
To display content of all txt files.
\$ cat *.txt

d) Banner –banner command is used to print the ASCII character string in large letter to standard output.

Syntax:
Banner <text>

Procedure:

- 1.Enter any content using Banner command
- 2.It display the content in large letters.

Example:
\$ Banner Raghu Engineering College

e) touch - change file timestamps

Syntax:
touch [*OPTION*]... *FILE*...

Tag	Description
-a	change only the access time
-c, --no-create	do not create any files
-d, --date= <i>STRING</i>	parse <i>STRING</i> and use it instead of current time
-f	(ignored)
-m	change only the modification time
-r, --reference= <i>FILE</i>	use this file's times instead of current time
-t <i>STAMP</i>	use [[CC]YY]MMDDhhmm[.ss] instead of current time
--time= <i>WORD</i>	

f) man - it is the interface used to view the system's reference manuals.
Syntax: man [-C file][-d][-D][--warnings[=warnings]]

g) cp - To copy one or more files to another location.

Syntax:
cp [options] ... source destination
cp [options] ... source ..directory

Description :

By default, sparse SOURCE files are detected by a crude heuristic and the corresponding DEST file is made sparse as well. That is the behavior selected by --sparse=auto. Specify --sparse=always to create a sparse DEST file whenever the SOURCE file contains a long enough sequence of zero bytes. Use --sparse=never to inhibit creation of sparse files.

Options:

Tag	Description
-a, --archive	same as -dpR
-b, --backup	make backup before removal
-d, --no-dereference	preserve links
-f, --force	remove existing destinations, never prompt
-i, --interactive	prompt before overwrite
-l, --link	link files instead of copying
-p, --preserve	preserve file attributes if possible
-P, --parents	append source path to DIRECTORY
-r	copy recursively, non-directories as files
--sparse= <i>WHEN</i>	control creation of sparse files

-r	copy recursively, non-directories as files
--sparse=WHEN	control creation of sparse files
-R, --recursive	copy directories recursively
-s, --symbolic-link	make symbolic links instead of copying
-S, --suffix=SUFFIX	override the usual backup suffix
-u, --update	copy only when the SOURCE file is newer than the destination file or when the destination file is missing
-v, --verbose	explain what is being done
-V, --version-control=WORD	override the usual version control



-x, --one-file-system	stay on this file system
--help	display this help and exit
--version	output version information and exit.

Examples :

```
Copy sample.txt to sample.bak.  
$ cat sample.txt  
This is a sample file  
$ cp sample.txt sample.bak  
$ cat sample.bak  
This is a sample file  
Copy sample directory to home directory  
$ cp -f /user/sample/* >
```

h) who - show who is logged on

Syntax :

who [*OPTION*]... [*FILE / ARG1 ARG2*]

Options :

-a, --all	same as -b -d --login -p -r -t -T -u
-b, --boot	time of last system boot
-d, --dead	print dead processes
-H, --heading	
	print line of column headings
-l, --login	
	print system login processes
--lookup	
	attempt to canonicalize hostnames via DNS
-m	only hostname and user associated with stdin
-p, --process	
	print active processes spawned by init
-q, --count	
	all login names and number of users logged on
-r, --runlevel	
	print current runlevel
-s, --short	

	print only name, line, and time (default)
-t, --time	print last system clock change

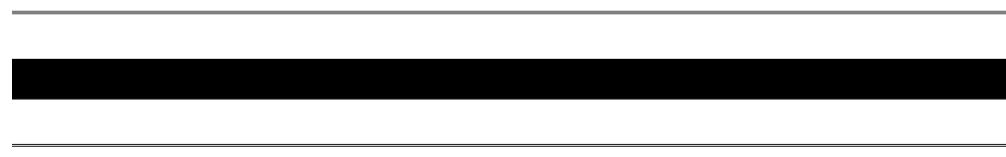
i) mv - move (rename) files

Syntax:

mv [OPTION]... [-T] SOURCE DEST
mv [OPTION]... SOURCE... DIRECTORY
mv [OPTION]... -t DIRECTORY SOURCE...

Options:

Tag	Description
--backup[=CONTROL]	
	make a backup of each existing destination file
-b	like --backup but does not accept an argument
-f, --force	
	do not prompt before overwriting
-i, --interactive	
	prompt before overwrite
--strip-trailing-slashes	remove any trailing slashes from each SOURCE
	argument
-S, --suffix=SUFFIX	
	override the usual backup suffix
-t, --target-directory=DIRECTORY	
	move all SOURCE arguments into DIRECTORY
-T, --no-target-directory	
	treat DEST as a normal file
-u, --update	
	move only when the SOURCE file is newer than the destination file or when the destination file is missing
-v, --verbose	
	explain what is being done
--help	display this help and exit



--version
output version information and exit

j) ls list directory contents.

Syntax:

ls [option][file]

Description:

ls List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified. Mandatory arguments to long options are mandatory for short options too.

Options:

Tag	Description
-a, --all	do not ignore entries starting with .
-A, --almost-all	do not list implied . and ..
--author	with -l, print the author of each file
-b, --escape	print octal escapes for nongraphic characters
--block-size=SIZE	use SIZE-byte blocks
-B, --ignore-backups	do not list implied entries ending with ~
-c	with -lt: sort by, and show, ctime (time of last modification of file status

-B, --ignore-backups	do not list implied entries ending with ~
-c	with -lt: sort by, and show, ctime (time of last modification of file status information) with -l: show ctime and sort by name otherwise: sort by ctime
-C	list entries by columns
--color[=WHEN]	control whether color is used to distinguish file types. WHEN may be `never', `always', or `auto'
-d, --directory	list directory entries instead of contents, and do not dereference symbolic links
-D, --dired	generate output designed for Emacs' dired mode
-f	do not sort, enable -aU, disable -ls --color

Example:

Example-1:

To list all files of current directory:

```
$ ls
output:
# ls
bin dev home lib lost+found mnt proc run srv tmp var
boot etc initrd.img lib64 media opt root sbin sys usr vmlinuz
```

k) mkdir - make directories

Syntax:

mkdir [OPTION] ... DIRECTORY

Description:

Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.

Options :

TAG	DESCRIPTION
-m, --mode=MODE	set file mode (as in chmod), not a=rwx - umask
-p, --parents	no error if existing, make parent directories as needed
-v, --verbose	print a message for each created directory
-Z, --context=CTX	set the SELinux security context of each created directory to CTX
--help	display this help and exit
--version	output version information and exit
--kernelargs args	Adds args to the arguments appended on the kernel command line. If this is not specified mkbootdisk uses grubpy to parse the arguments for the default kernel from grub.conf, if possible.
--size size	Uses size (in kilobytes) as the size of the image to use for the boot disk. If this is not specified, mkbootdisk will assume a standard 1.44Mb floppy device.

Examples:

Example-1:

Creates a new directory called mydir whose parent is the current directory.

\$ mkdir mydir

output:

\$ls

mydir

Example-2:

Create the mydir directory, and set its permissions such that all users may read, write, and execute the contents.

\$ mkdir -m a=rwx mydir

I) rm remove files or directories

Syntax

rm [OPTION]... FILE...

Options:

TAG	DESCRIPTION
-f, --force	ignore nonexistent files, never prompt.
-i	prompt before every removal.
-I	prompt once before removing more than three files, or when removing recursively. Less intrusive than -i, while still giving protection against most mistakes.
--interactive[=WHEN]	prompt according to WHEN: never, once (-I), or always (-i). Without WHEN, prompt always.
--one-file-system	when removing a hierarchy recursively, skip any directory that is on a file system different from that of the corresponding command line argument.
--no-preserve-root	do not treat '/' specially.
--preserve-root	do not remove '/' (default).
-r, -R, --recursive	remove directories and their contents recursively.
-v, --verbose	explain what is being done.
--help	display this help and exit.
--version	output version information and exit.

Examples:

Example-1:

Remove the file myfile.txt. If the file is write-protected, you will be prompted to confirm that you really want to delete it:

\$ rm myfile.txt

Example-2:

Remove the file myfile.txt. You will not be prompted, even if the file is write-protected; if rm can delete the file, it will:

\$ rm -f myfile.txt

Example-3:

Remove all files in the working directory. If it is write-protected, you will be prompted before rm removes it:

\$ rm *

m) rmdir - remove empty directories

Syntax:

rmdir [OPTION]... DIRECTORY...

Description :

Remove the DIRECTORY(ies), if they are empty.

Options :

TAG	DESCRIPTION
--ignore-fail-on-non-empty	ignore each failure that is solely because a directory is non-empty.
-p, --parents	remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b/c' is similar to 'rmdir a/b/c a/b a'.
-v, --verbose	output a diagnostic for every directory processed.
--help	display this help and exit.
--version	output version information and exit.

Examples :

Example-1:

rmdir command will delete the empty directories. i.e directory without any sub-directories or files:

\$ rmdir test

Example-2:

To Delete Nested Empty Directories in Linux:

\$ rmdir -p dir1/dir2/dir3

n) echo - display a line of text.

Syntax : echo [SHORT-OPTION]... [STRING]...
echo LONG-OPTION

Description :

echo is a fundamental command found in most operating systems that offer a command line. It is frequently used in scripts, batch files, and as part of individual commands; anywhere you may need to insert text. Many command shells such as bash, ksh and csh implement echo as a built-in command.

Options

Tag	Description
-n	Do not output a trailing newline.
-e	Enable interpretation of backslash escape sequences (see below for a list of these).
-E	Disable interpretation of backslash escape sequences (this is the default).
--help	Display a help message and exit.
--version	Output version information and exit.
\\	A literal backslash character ("\\").
\a	An alert (The BELL character).
\b	Backspace
\c	Produce no further output after this.
\e	The escape character; equivalent to pressing the escape key.
\f	A form feed.
\n	A newline.
\r	A carriage return
\t	A horizontal tab.
\v	A vertical tab.
\0NNN	byte with octal value NNN (which can be 1 to 3 digits).
\xHH	byte with hexadecimal value HH (which can be either 1 or 2 digits)

Examples :

Example-1:

To print string "Hello, World!" on console
\$ echo "Hello, World!"

output:

Hello, World!

Example-2:

To print value of x, where x=10.

```
$ echo $x
```

output:
10

o) more - file perusal filter for crt viewing

Syntax :

more [-dlfpcsu] [-num] [+/- pattern] [+ linenum] []

Description:

More is a filter for paging through text one screenful at a time. This version is especially primitive. Users should realize that less(1) provides more(1) emulation and extensive enhancements.

Options:

Command line options are described below. Options are also taken from the environment variable *MORE* (make sure to precede them with a dash ("-'")) but command line options will override them.

Tag	Description
-num	This option specifies an integer which is the screen size (in lines).
-d	more will prompt the user with the message "[Press space to continue, 'q' to quit.]" and will display "[Press 'h' for instructions.]" instead of ringing the bell when an illegal key is pressed.
-l	more usually treats ^L (form feed) as a special character, and will pause after any line that contains a form feed. The -l option will prevent this behavior.
-f	Causes more to count logical, rather than screen lines (i.e., long lines are not folded).
-p	Do not scroll. Instead, clear the whole screen and then display the text.
-c	Do not scroll. Instead, paint each screen from the top, clearing the remainder of each line as it is displayed.
-s	Squeeze multiple blank lines into one.
-u	Suppress underlining.
+/	The +/- option specifies a string that will be searched for before each file is displayed.
+num	Start at line number num.

p) date : date command is used to display system date and time.

Syntax:

date [OPTION]... [+FORMAT]
date [-u|-utc|--universal] [MMDDhhmm[CC]YY][.ss]]

Examples:

Example 1

\$date

Output:

Tue Oct 10 22:55:01 PDT 2017

Example 2

\$date -u

Output :

Wed Oct 11 06:11:31 UTC 2017

Example 3

\$date --date="2/02/2010"

\$date --date="Feb 2 2010"

Output:

Tue Feb 2 00:00:00 PST 2010

Tue Feb 2 00:00:00 PST 2010

q) history – history command is used to view the previously executed commands.

This feature was not available in the Bourne shell. Bash and Korn support this feature in which every command executed is treated as the event as is associated with an event number using which they can be recalled and change if required. These command are saved in a history file. In bash shell history command shows the whole list of the command.

Syntax:

\$ history

q) chmod - To change owner, change the user and/or group ownership of each given File to a new Owner. Chown can also change the ownership of a file to match the user/group of an existing reference file.

Syntax: chown [Options]... NewOwner File...

chown [Options]... :Group File...

new Owner. Chown can also change the ownership of a file to match the user/group of an existing reference file.

Syntax: chown [Options]... NewOwner File...
chown [Options]... :Group File...
chown [Options]... --reference=FILE File...

Description :

If used, NewOwner specifies the new owner and/or group as follows(with no embedded white space):

chown [OWNER] [[:] [GROUP]]

Following are the examples of how the owner/group can be specified:

- If only an OWNER (a user name or numeric user id) is given, that user is made the owner of each given file, and the files' group is not changed.

chown OWNER

- If the OWNER is followed by a colon or dot and a GROUP (a group name or numeric group id), with no spaces between them, the group ownership of the files is changed as well (to GROUP).

• chown OWNER.GROUP

chown OWNER:GROUP

- If a colon or dot but no group name follows OWNER, that user is made the owner of the files and the group of the files is changed to OWNER's login group.

• chown OWNER.

chown OWNER:

- If the colon or dot and following GROUP are given, but the owner is omitted, only the group of the files is changed; in this case, 'chown' performs the same function as 'chgrp'.

• chown .GROUP

chown :GROUP

Options:

Tag	Description
-c --changes	Verbosely describe the action for each File whose ownership actually changes.
--dereference	Do not act on symbolic links themselves but rather on what they point to.
-f --silent --quiet	Do not print error messages about files whose ownership cannot be changed.
-h --no-dereference	Act on symbolic links themselves instead of what they point to. This is the default. This mode relies on the 'lchown' system call. On systems that do not provide the 'lchown' system call, 'chown' fails when a file specified on the command line is a symbolic link. By default, no diagnostic is issued for symbolic links encountered during a recursive traversal, but see '--verbose'.
--reference=FILE	Use the user and group of the reference FILE instead of an explicit NewOwner value.
-R --recursive	Recursively change ownership of directories and their contents.
-v --verbose	Verbosely describe the action (or non-action) taken for every FILE. If a symbolic link is encountered during a recursive traversal on a system without the 'lchown' system call, and '--no-dereference' is in effect, then issue a diagnostic saying neither the symbolic link nor its referent is being changed.

Examples:

Change the owner of file.

\$ chown user1 sample.txt

Change the group of file.

```
$ chown :mygroup file.txt
```

Change both the owner and group of file in single command.

```
$ chown user1:mygroup file.txt
```

r) pwd - print name of current/working directory

Print the full filename of the current working directory.

Tag	Description
--help	display this help and exit
--version	
	output version information and exit

s) wc - print the number of newlines, words, and bytes in files

Syntax : wc [*OPTION*]... [*FILE*]...

Options :

Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified. With no FILE, or when FILE is -, read standard input.

Tag	Description
-c, --bytes	
	print the byte counts
-m, --chars	
	print the character counts
-l, --lines	
	print the newline counts
-L, --max-line-length	
	print the length of the longest line
-w, --words	
	print the word counts
--help	display this help and exit
--version	
	output version information and exit

t) sort - sort lines of text files

Syntax :

```
sort [OPTION]... [FILE]...
```

Options:

Tag	Description
-b, --ignore-leading-blanks	
	ignore leading blanks
-d, --dictionary-order	
	consider only blanks and alphanumeric characters
-f, --ignore-case	
	fold lower case to upper case characters
-g, --general-numeric-sort	
	compare according to general numerical value
-i, --ignore-nonprinting	
	consider only printable characters
-M, --month-sort	
	compare (unknown) < 'JAN' < ... < 'DEC'
-n, --numeric-sort	
	compare according to string numerical value

-d INPUT_DELIM_BYTE --delimiter=INPUT_DELIM_BYTE	For '-f', fields are separated in the input by the first character in INPUT_DELIM_BYTE (default is TAB).
-n	Do not split multi-byte characters (no-op for now).
-s	For '-f', do not print lines that do not contain the field separator character.
--only-delimited	For '-f', output fields are separated by OUTPUT_DELIM_STRING The default is to use the input delimiter.

Examples:

```
Let's have a sample file sample.txt
$ cat sample.txt
1;2;3;4;5;6;7;8;9
To Parse out column 2 from a semicolon (;) delimited file:
$ cat sample.txt | cut -d \; -f 2 > output.txt
$ cat output.txt
2
```

v) paste : merge lines of files

Syntax :
paste [*OPTION*]... [*FILE*]..

Options:

Tag	Description
-d, --delimiters= <i>LIST</i>	reuse characters from LIST instead of TABs
-s, --serial	paste one file at a time instead of in parallel
--help	display this help and exit
--version	output version information and exit

w) logout - session-logout.

Syntax:

session-logout[options]

Description:logout command allows you to programmatically logout from your session. causes the session manager to take the requested action immediately.

Examples:

Example-1:

To logout from current user session:

```
$ logout
output:
no output on screen, current user session will be logged out.
```

x) shutdown - bring the system down

Options:

Tag	Description
-a	Use /etc/shutdown.allow.
-t sec	Tell init(8) to wait sec seconds between sending processes the warning and the kill signal, before changing to another runlevel.
-k	Don't really shutdown; only send the warning messages to everybody.
-r	Reboot after shutdown.
-h	Halt or poweroff after shutdown.
-H	Halt action is to halt or drop into boot monitor on systems that support it.
-P	Halt action is to turn off the power.
-n	[DEPRECATED] Don't call init(8) to do the shutdown but do it ourself. The use of this option is discouraged, and its results are not always what you'd expect.
-f	Skip fsck on reboot.
-F	Force fsck on reboot.
-c	Cancel an already running shutdown. With this option it is of course not possible to give the time argument, but you can enter a explanatory message on the command line that will be sent to all users.
Time	When to shutdown.
warning-message	Message to send to all users.