Unix Commands :

(1) ls -l -- long listing

ls → list of files & dir from current working dir

filetype permissions, no of hardlinks, owner, groupowner, size in bytes, last modific^n time, Name of files & directories

* -a → all files including . and ..

→
-	regular file
d	directory file
s	socket file
b	block special file
c	character special file
l	symbolic link
p	pipe

* -r sort in reverse alphabetically
* -1 list names one per line
* -u (with lt) sort by access time
* -t sort by modific^n time
* ul } sheet.
* ult }
* ls c* - list all files/dir starting with letter c
* ls [a-d]* list all files/dir starting with

1/14

* To know the location of an executable program i.e command use type command

$ type ls ↵

(2) who : who is online
- displays all users who are currently logged in to the system
- It returns user's name (ID), terminal and time at which he/she was logged in
* $ who ↵
* $ who -u ↵
- to know since what how long there was no activity (idle
eg. o/p * $who -uH ↵
- displays header for each column above
* $ who am i ↵

(3) pwd - print working directory
- It prints absolute path name of our working directory
$pwd ↵
/home/chirag
It means that we are in home dir of user ID chirag

(4) $ date → displays system date and time

$ date d
→ displays current date & time

$ date +%m d (1-12)
→ prints only month (1-12)

$ date +%b d
→ prints month name

$ date +%Y d
→ prints year

$ date +%d d → prints day (01-31)

$ date +"%I %p" → prints hour with am/pm.

(5) cal → display calendar

$ cal d → display current month calendar.

$ cal 3 2017 d → display calendar of month March 2017

$ cal 2017 d → calendar of whole year 2017

$ cal -j 2018 → displays calendar of 2018 in julean form.

$ cal -h → turns off highlighting today

$ cal -3 → displays prev, current & next month.

(6) clear - clear the terminal screen

(7) echo - displays my P results of computa on screen.
by default interpretation of backslash escapes is disabled

: $echo "Hello \n World" ↵
displays
Hello \n World

- To enable interpretation of \ type
$ echo "Hello -e "Hello \n World" ↵
o/p Hello
World

$ echo -e "Hello \\ World" ↵
$ echo -e "Hello \\World ↵
o/p Hello \ World
$ echo -e "Hello \v World" (vertical tab

$ echo -e "Hello \b World" ↵

(8) bc - activates basic calculator

$ bc ↵ It goes in interactive mode and waits for input
type 7+6 as i/p ↵ press enter
o/p 13

type quit to quit calculator

$ bc -l ↵
- initializes scale to 20 instead of default zero.

$ bc 5/3 ↵
O/P 1.66666666666666666666
type scale=2 ↵
5/3 ↵
O/P 1.66

functions available with bc command.
sqrt() → calculates square root
s() → calculates sine value. argument should be in radians
c() → cosine value

(9) login : begin session on the

In root type login
give userid and password
$ login -f → to skip authentication. User is pre-authenticated

(10) To logout : logout

(11) cd - change directory
cd - → go to prev directory
cd ~ → navigate to home directory.
cd .. → navigate one level up in directory
cd / → go to root directory 5/14

'>' O/P redirection operator
- used to send O/P of command to a file

(12) cat → showing creating and concatenating files

$ cat > file1 ↵
This is my first file. ↵
I have created it ↵
^d ↵
} creates file named file1

$ cat file1 ↵ will show the contents of file file1

$ cat -n file1 ↵ will display contents with line numbering
1. This is my first file
2 I have written it.

$ cat file1 file2 ↵ will display contents of file1 followed by contents of file2.

$ cat file1 file2 > file3 ↵
concatenates contents of file1 & file2 in file3.

$ cat -ve file1 ↵ - will show hidden characters in the file
This is my first file.$
I have written it.$
newline characters are shown by $.

1 Pipe operator

used to send o/p of command to another command

use >> to append to the file

$ cat >> file1 ↵
This (tab) third (tab) line (tab) in (tab) file ↵
^d
$ cat -vt file1 ↵ (displays tab as ^I)

o/p This is my first file. ↵
I have written it.
This^I third^I line^I in^I file.

(13) cp → to copy contents of one file to another.
(source) (dst)

$ cp file1 file2 ↵ will copy contents of file 1 to file 2

$ cat file2 ↵ to view contents of file

$ cat -i file2 file4 ↵ will ask before overwriting file (interactive copying)

$ mkdir mydir1 ↵
$ mv file1 mydir1 ↵
$ mkdir mydir2 ↵
$ mv file2 mydir2 ↵
$ cp -r mydir1 mydir2 ↵
will copy the directory mydir1 along with its files to mydir2.

7/14

(14) mv - removes the file from current one location and copies to another.

$ mv oldname newname

(15) ~~mkdir~~ rm - remove file

$rm -i file3 ↵
will ask ~~what~~ whether to remove file before removing.

$rm -i file4 file5 ↵
will remove file4 and file5

$rm -r mydir2 ↵
will recursively remove ~~mydir2~~ files & directory & all from mydir2 and finally the directory mydir2

(16) ~~rmdir~~ rmdir - will remove the directory only if it is empty.

$ rmdir mydir1 ↵
- will give following error if mydir1 not empty
rmdir: failed to remove 'mydir1': directory not empty

$ rmdir dir1 dir2 ↵ (multiple directory names)
will remove dir1 and dir2.

$ rmdir -p dir1/dir3 ↵
will first remove dir3 then dir1.

(17) mkdir - make directory.

$mkdir dir1 dir2 ↵
will create dir1 and dir2 in current working directory
$ mkdir -p dir1/dir2 ↵ will create parent directory dir1 first and then dir2 in it

(18) history → will give the history of all commands executed in the session.

$history ↵

$history 50 ↵ will display 50th line in history of commands

time, more, ps, chown, finger, shutdown, login, logout, last, kill, chmod.

first

* more than one command can be combined in one line by giving semicolon.

date

9/14

(19) more - more is a filter for paging through text one screenful at a time.

It following to understand more

create a file 'file1' and write some lines of text into it

$ cat > file1 ↵
This is my first file ↵
file is created in labform ↵
to explore the use of more commands ↵
if required we will add more data to it ↵
123 ↵
234 ↵
345 ↵
456 ↵
567 ↵
^d

Now give the command as follows

$ more -3 file1 ↵

- This will display of file1 on screen with three lines at a time on screen. press enter to see next lines

- Then $ more -number filenames displays 'number' lines on screen at a time

10/14

$ more -p file1 ↵
→ Don't scroll, instead clear the whole screen and then display text

Now append ~~four~~ 4 blank lines in file 1 as:

$ cat >> file1 ↵
↵
↵
↵
↵
i have inserted 4 blank lines ↵
^d

Display file 1 :
$ cat file1 ↵

Now give command
$ ~~man~~ more -s file1 ↵
This will merge 4 blank lines to 1 and show the o/p.

$ ~~man~~ more +3 file1 ↵
will display file1 ~~from~~ starting from line # 3.

$ ls -1 ↵ displays list of files one per line
$ ls | more ↵ will display it one page & then more file will be displayed when u press enter.

11/14

(20) time → displays the time required to execute the command.

e.g. $ time sort file1 ↵
o/p = } sorted contents of file1

real 0m 0.002 s
user 0m 0.000 s
sys 0m 0.000 s

real time shows the clock elapsed time from invocation of command till termination.
user time shows time spent by program in executing itself (i.e. in user mode)
sys indicates time used by kernel in doing work on behalf of user process (in kernel mode)
Sum of user time + systime actually represents CPU time.

e.g. $ time ls -l ↵
$ time ls -l > list ↵

(21) wc - prints no. of lines, words & characters in a file

```
$wc file1
o/p  2  15  92  file1
     ↓  ↓   ↓
   no. of lines  words  characters
```

```
$wc < file1
2  15  92
```

```
$ wc -l file1 ↵        (only no. of lines)
o/p   2 file1
```

```
$ wc -w file1 ↵        (only no. of words)
o/p   15 file1
```

```
$ wc -c file1 ↵        (only no. of characters)
o/p   92 file1
```

```
$ wc ↵                 } Instead of file name directly text is i/p
enter text here ↵ ^d ↵ }
```

(22) login - begin session on the system.

You have to a root user to begin a new session

```
# login ↵
login: lablco6 ↵      (asks for username
password:  ↵          (asks for password)
   (If login successful, new session
   begins and # changes to $)
$
```

(23) logout - exits the session

(24) finger - displays information of users who are logged in

Note: In normal user mode, finger command may not work.
- switch to root user by
$ sudo su ↵
# apt-get install finger ↵

finger ↵
// displays:
login, name, terminal name, idle time, login time, etc

finger labtab6 ↵ (username)

o/p
login: labtab6 Name: MUM089
Directory: /home/labtab6 shell: /bin/bash
On since sat Jan 30 9:58 on :1 from :1 (message #)
On since sat Jan 30 10:03 on pts/26 5 seconds idle
(message #1)
No mail.
No plan.