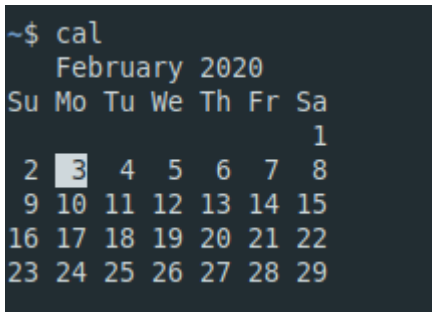
EXPERIMENT NO:1

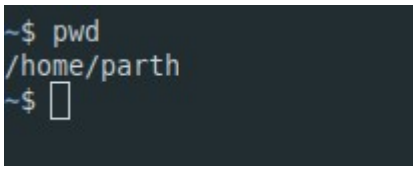
AIM: To study and execute various Linux commands.

THEORY:

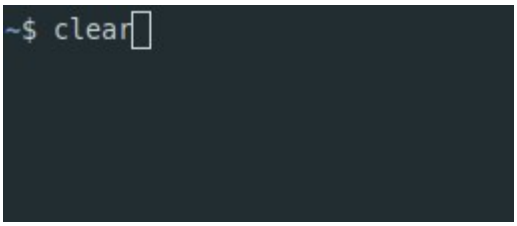
1. cal:
   It displays current month's calendar with current day highlighted.



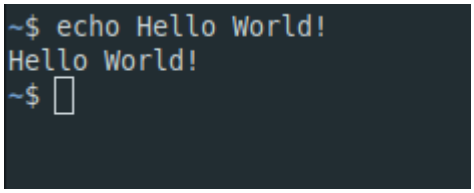
2. pwd:
   pwd (print working directory) command displays your location currently you are working on. It will give the whole path starting from the root ending to the directory.



3. clear:
   The clear command clears the terminal.



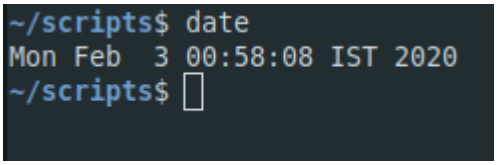
4. echo:
   Echo as a command prints the string passed to it as an argument.



5. ls:
   The ls is the list command. It will show the full list or content of your directory. Just type *ls* and press the enter key. The whole content will be shown.

ls command	Description
ls -a	In Linux, hidden files start with . (dot) symbol and they are not visible in the regular directory. The (ls -a) command will enlist the whole list of the current directory including the hidden files.
ls -l	It will show the list in a long list format.
ls –lh	This command will show you the file sizes in human readable format. Size of the file is very difficult to read when displayed in terms of byte. The (ls -lh)command will give you the data in terms of Mb, Gb, Tb, etc.
ls -d */	It is used to display only subdirectories.
ls –r	It is used to print the list in reverse order.
ls ~	It gives the contents of home directory.
ls ../	It give the contents of parent directory.

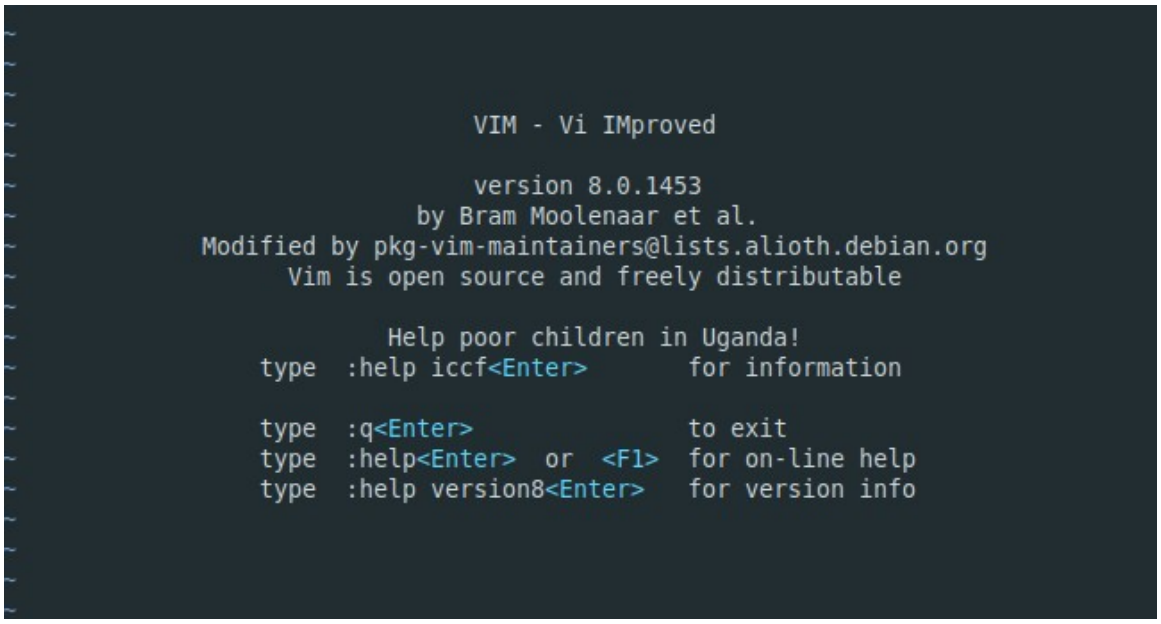
6. Date:
   The date command displays date, time and time zone.



7. History:
   Displays older commands from shell command history.

```
~/scripts$ history
  565  exit
  566  conda activate tensorflow
  567  python -m ipykernel install --user --name tensorflow --display-name "Python 3.7 (tensorflow)"
  568  jupyter
  569  jupyter --version
  570  jupyter-notebook
  571  conda deactivate
  572  exit
```

8. Vi:
   Opens vi editor to write a program.



9. Head:
   head print the first n lines of a file, where n is 10 by default.

```
~/scripts$ head -3 evenodd.sh
echo -n Enter a number:
read num
r=$(( $num % 2 ))
~/scripts$
```

10. Cat:
    It can be used to display the content of a file, copy content from one file to another, concatenate the contents of multiple files, display the line number, display $ at the end of the line, etc.

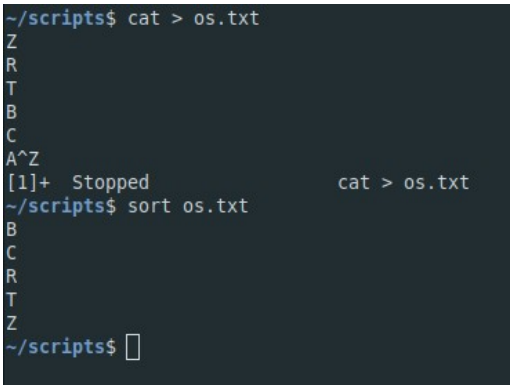
```
~/scripts$ cat helloworld.sh
#!/bin/sh
echo Hello World
~/scripts$
```

Option	Function
cat > [fileName]	To create a file.
cat [oldfile] > [newfile]	To copy content from older to new file.
cat [file1 file2 and so on] > [new file name]	To concatenate contents of multiple files into one.
cat -n/cat -b [fileName]	To display line numbers.
cat -e [fileName]	To display $ character at the end of each line.
cat [fileName] <<EOF	Used as page end marker.

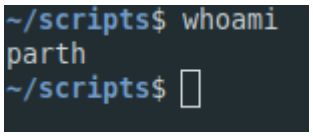
11. W:
    w shows a list of users currently logged into the system along with the commands they are running.

```
~/scripts$ w
 01:03:24 up 15:15,  1 user,  load average: 0.65, 0.55, 0.61
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
parth    tty7     :0               Sun10   15:14m 29:07   6.06s cinnamon-session -
~/scripts$
```

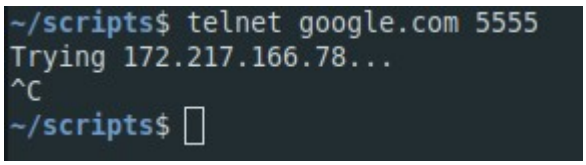
12. Sort:
    the command `sort` sorts files. The default behavior of `sort` is to *dictionary sort* the rows of a file according to what's in the first column, then second column, and so on.



13. Whoami:
    Shows your linux username.



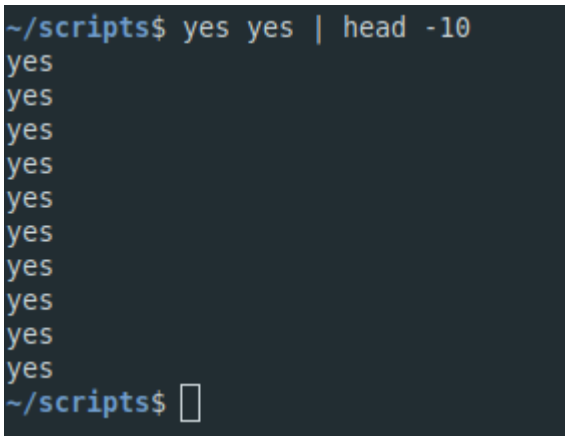
14. telnet:
    telnet connects destination host port via a telnet protocol if connection establishes means connectivity between the two hosts is working fine.



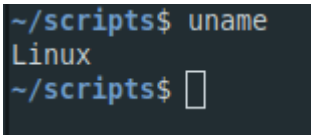
- -c: disables the reading of the users .telnetrc file
- -d:sets the initial value of the debug toggle to TRUE

- -e: escapechar: sets the initial telnet escape character to escapechar. If escapechar is omitted then there will be no escape character.

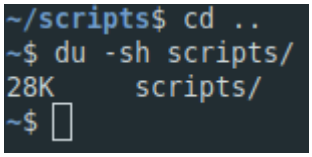
15. Yes:
    yes prints out the character *y* in an infinite loop (so be careful - you'll have to stop it with *Cntrl-c*). If you give yes an argument, it will print out the argument rather than *y*.



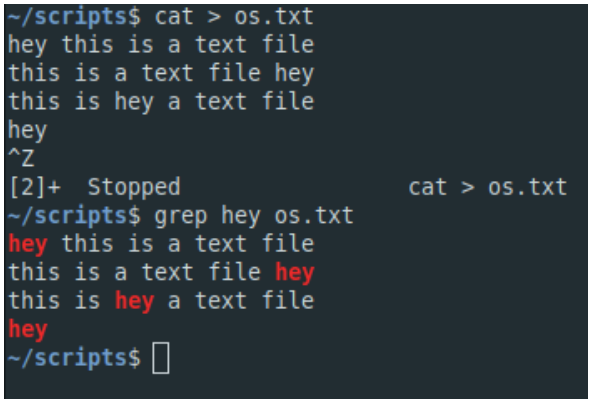
16. Uname:
    As its man page says, uname prints out various system information. it prints out the name but not the version of the kernel.



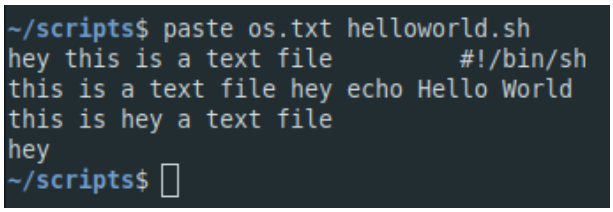
17. Du:
    it's for checking the sizes of individual directories.



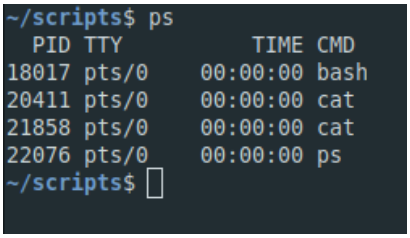
18. Grep:
    grep searches for text in a file and returns the line(s) where it finds a match.



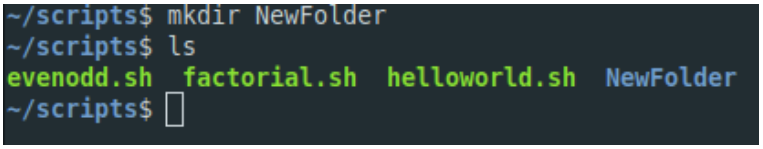
19. Paste:
    `paste` joins files together in a column-wise fashion. Another way to think about this is in contrast to `cat`, which joins files vertically.



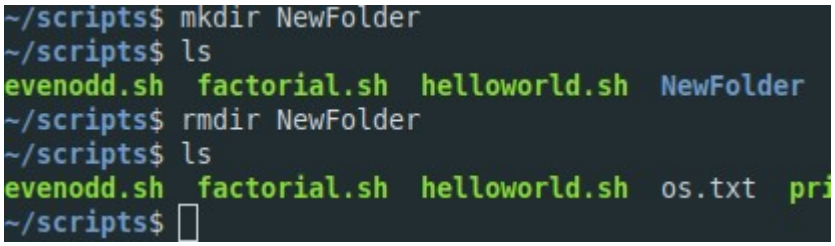
20. Ps:
    The command ps displays information about your processes.



21. Mkdir:
    To *make directory*—i.e., create a new folder



22. Rmdir:
    Used to delete empty directories. rm -r is used to delete non-empty ones.



23. Ifconfig:
    The command ifconfig stands for interface configuration. This command enables us to initialize an interface, assign IP address, enable or disable an

interface. It displays route and network interface. You can view IP address, MAC address and MTU (maximum transition unit) with this command.

```
~/scripts$ ifconfig
enp1s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether c8:5b:76:d9:5d:7c  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 24790  bytes 7744114 (7.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24790  bytes 7744114 (7.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.8  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::16d8:5a7a:388b:1d7  prefixlen 64  scopeid 0x20<link>
        ether c8:3d:d4:80:27:f7  txqueuelen 1000  (Ethernet)
        RX packets 450420  bytes 253268669 (253.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 343698  bytes 51310380 (51.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

24. Rm:

    The command `rm` removes the files you pass to it as arguments

```
~/scripts$ ls
evenodd.sh  factorial.sh  helloworld.sh  os.txt  prime.sh  sqrt.sh
~/scripts$ rm helloworld.sh
~/scripts$ ls
evenodd.sh  factorial.sh  os.txt  prime.sh  sqrt.sh
~/scripts$
```

25. Touch:

    `touch` makes an empty file.

```
~/scripts$ touch NewFile.txt
~/scripts$ ls
evenodd.sh  factorial.sh  NewFile.txt  os.txt  prime.sh  sqrt.sh
~/scripts$
```

26. Bzip2:
    Compresses a file

```
~/scripts$ bzip2 os.txt
~/scripts$ ls
evenodd.sh  factorial.sh  NewFile.txt  os.txt.bz2  prime.sh  sqrt.sh
~/scripts$
```

27. Bunzip2:
    Deompress a bzip2 file.

```
~/scripts$ bunzip2 os.txt.bz2
~/scripts$ ls
evenodd.sh  factorial.sh  NewFile.txt  os.txt  prime.sh  sqrt.sh
~/scripts$
```

28. Ip:
    ip address:it is used to show all ip addresses associated on all network devices. This will show the information related to all interfaces available on the system.

29. Locate:
    Searches a file in the database.

```
~/scripts$ locate os.txt
/home/parth/nltk_data/corpora/pros_cons/IntegratedPros.txt
/home/parth/nltk_data/corpora/udhr2/cos.txt
/home/parth/nltk_data/corpora/udhr2/mos.txt
/usr/share/vim/vim80/doc/os_beos.txt
/usr/share/vim/vim80/doc/os_dos.txt
/usr/share/vim/vim80/doc/os_msdos.txt
~/scripts$
```

30. Netsat:
    the netsat command stands for network static. It displays information about different interface statics including open sockets, routine tables and connection information.

```
~/scripts$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 parth-ideapad:48158     bom05s12-in-f2.1e:https ESTABLISHED
tcp        0      0 parth-ideapad:51478     ec2-35-160-108-61:https ESTABLISHED
tcp        0      0 parth-ideapad:48790     74.125.24.189:https     ESTABLISHED
tcp        0      0 parth-ideapad:53982     kul01s10-in-f35.1:https ESTABLISHED
tcp        0      0 parth-ideapad:50870     del03s16-in-f5.1e:https ESTABLISHED
tcp        0      0 parth-ideapad:45838     del03s10-in-f14.1:https ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    32828    /run/user/1000/systemd/n
otify
unix  3      [ ]         DGRAM                    14865    /run/systemd/notify
unix  3      [ ]         SEQPACKET  CONNECTED     452393   @0001f
unix  4      [ ]         DGRAM                    29262    @var/run/nvidia-xdriver-
```

31. route :

    it displays and manipulates IP routing table for your system. A router is a device which is basically used to determine the best way to route packets to a destination.

    - -net target: target is a network
    - -net host: target is a host

```
~/scripts$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    600    0        0 wlp2s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 wlp2s0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlp2s0
~/scripts$
```

32. Hostname:

    with the help of hostname command you can set and view hostname of the system.

    - -d: displays the domain name the machine belongs to
    - -f: displays fully qualified hostname and domain name

```
~/scripts$ hostname
parth-ideapad
~/scripts$
```

33. Id:

    Tells about the user's id in the system.

```
~/scripts$ id
uid=1000(parth) gid=1000(parth) groups=1000(parth),4(adm),24(cdrom),27(sudo),30(dip
),46(plugdev),112(lpadmin),128(sambashare)
~/scripts$
```

34. Wc:

```
~/scripts$ cat os.txt
hey this is a text file
this is a text file hey
this is hey a text file
hey
~/scripts$ wc os.txt
 4 19 77 os.txt
~/scripts$
```

35. Which:

`which` shows you the path of a command in your `PATH`.

```
~/scripts$ which clear
/usr/bin/clear
~/scripts$
```

36. Seq:

`seq` prints a sequence of numbers.

```
~/scripts$ seq 1 7
1
2
3
4
5
6
7
~/scripts$
```

37. Shred:

Securely remove your file by overwriting then removing

```
~/scripts$ shred NewFile.txt
~/scripts$ ls
evenodd.sh  factorial.sh  NewFile.txt  os.txt  prime.sh  sqrt.sh
~/scripts$
```

38. Sleep:

The command sleep pauses for an amount of time specified in seconds.

39. Split:

`split` splits up a file.

40. Df:

    `df` reports "file system disk space usage". This is the command you use to see how many much space you have left on your hard disk. The -h flag means "human readable," as usual.

```
~/scripts$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G     0  3.9G   0% /dev
tmpfs           788M  1.6M  787M   1% /run
/dev/sda10       88G   65G   19G  78% /
tmpfs           3.9G   90M  3.8G   3% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/sda1       256M   35M  222M  14% /boot/efi
tmpfs           788M   52K  788M   1% /run/user/1000
/dev/sda4       348G  220G  128G  64% /media/parth/DATA_VOL
~/scripts$
```

41. Diff:

    `diff` prints out the differences between two files. It's the best way to find discrepancies between files you think are the same.

```
~/scripts$ diff factorial.sh evenodd.sh
1c1
< echo -n Enter a number:
---
> echo -n Enter a number:
3,10c3,9
< fact=1
< while [ $num -ge 1 ]
< do
<       fact=$(( $fact * $num ))
<       num=$(( $num - 1 ))
< done
< echo Factorial: $fact
<
---
> r=$(( $num % 2 ))
> if [ $r -eq 0 ]
> then
>       echo EVEN
> else
>       echo ODD
> fi
```

42. Env:

    There are 3 notable things about env. First, if you run it as standalone, it will print out all the variables and functions set in your environment. secondly, you can use env to avoid hard-wired paths and lastly env can run a utility "in an altered environment without having to modify the currently existing environment.

```
~/scripts$ env
CONDA_SHLVL=0
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40
;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=0
1;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=
01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zi
p=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;3
1:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;
31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;
31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;
31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=0
1;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.p
pm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*
.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;3
5:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=0
1;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=0
```

43. Man:

    This command will display all the information about '**ls**' command as shown in the screen sh

```
NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       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.

       -a, --all
              do not ignore entries starting with .
```

44. Tail:

    Displays last ten lines of a file.

```
~/scripts$ tail -3 evenodd.sh
else
        echo ODD
fi
~/scripts$
```

45. Type:

    Tells us whether a command is built in or external

```
~/scripts$ type pwd
pwd is a shell builtin
~/scripts$
```

46. Alias:
    The ‘alias’ is another name for a command. If no argument is given, it shows current aliases. Aliases can be used for short names of commands. For example, you might use the clear command frequently. You can create an alias for it:

```
~/scripts$ alias c="clear"
~/scripts$
```

47. Tty:
    Displays the current terminal

```
~/scripts$ tty
/dev/pts/0
~/scripts$
```

48. Ps –a:
    It gives information about all the processes.

```
~/scripts$ ps -a
  PID TTY          TIME CMD
20411 pts/0    00:00:00 cat
21858 pts/0    00:00:00 cat
25324 pts/0    00:00:00 ps
~/scripts$
```

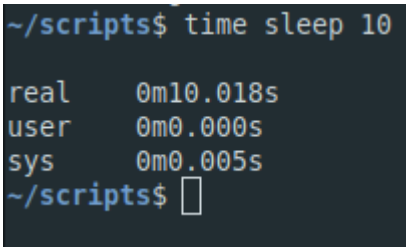
49. uname -a:
    It displays all the system information

```
~/scripts$ uname -a
Linux parth-ideapad 4.15.0-54-generic #58-Ubuntu SMP Mon Jun 24 10:55:24 UTC 2019 x
86_64 x86_64 x86_64 GNU/Linux
~/scripts$
```

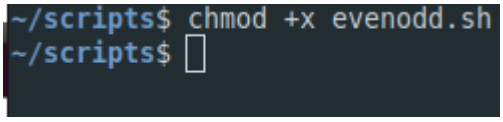
50. ping: checks the connectivity between two nodes

```
~/scripts$ ping google.com
PING google.com (172.217.167.174) 56(84) bytes of data.
64 bytes from bom12s01-in-f14.1e100.net (172.217.167.174): icmp_seq=1 ttl=56 time=1
41 ms
64 bytes from bom12s01-in-f14.1e100.net (172.217.167.174): icmp_seq=2 ttl=56 time=6
1.9 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 61.931/101.843/141.755/39.912 ms
~/scripts$
```

51.time :
It denotes the amount of time taken for the command or script to run.



52. chmod:
Used to change permissions of any file or folder.



CONCLUSION:
We have successfully studied basic Linux commands and implemented them respectively.