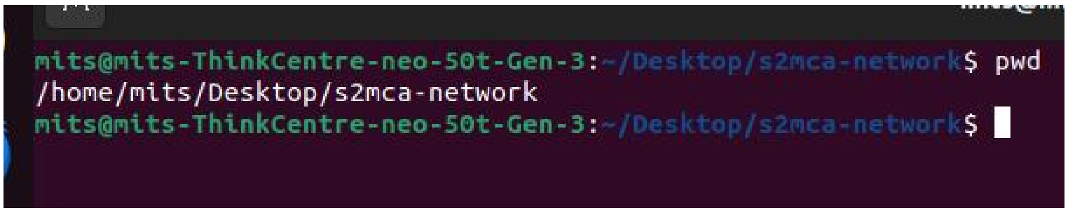
2. Basic Linux Commands

Study of a terminal based text editor such as Vim or Emacs. (By the end of the course, students are expected to acquire following skills in using the editor: cursor operations, manipulate text, search for patterns, global search and replace)
Basic Linux commands, familiarity with following commands/operations expected

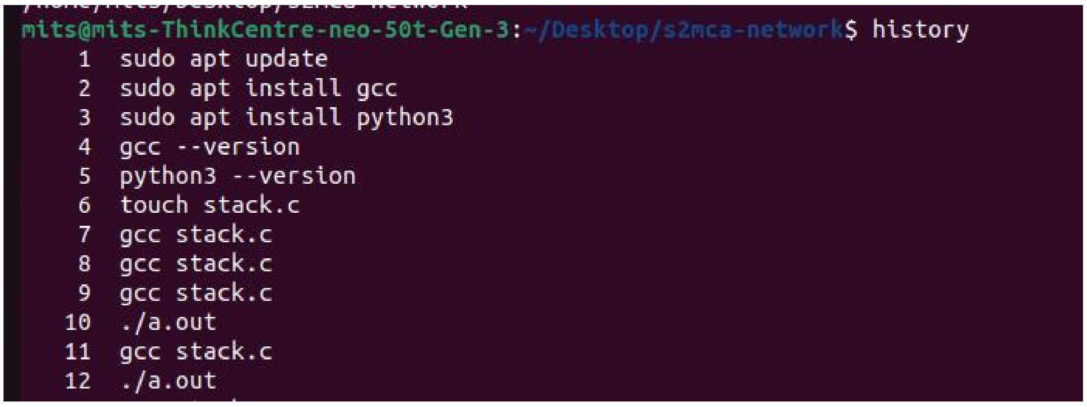
1. man
2. ls, echo, read
3. more, less, cat,
4. cd, mkdir, pwd, find
5. mv, cp, rm ,tar
6. wc, cut, paste
7. head, tail, grep, expr
8. chmod, chown
9. Redirections & Piping
10. useradd, usermod, userdel, passwd
11. df,top, ps
12. ssh, scp, ssh-keygen, ssh-copy-id

1. **pwd (Print Working Directory):** Use the pwd command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash (/). An example of an absolute path is /home/username.



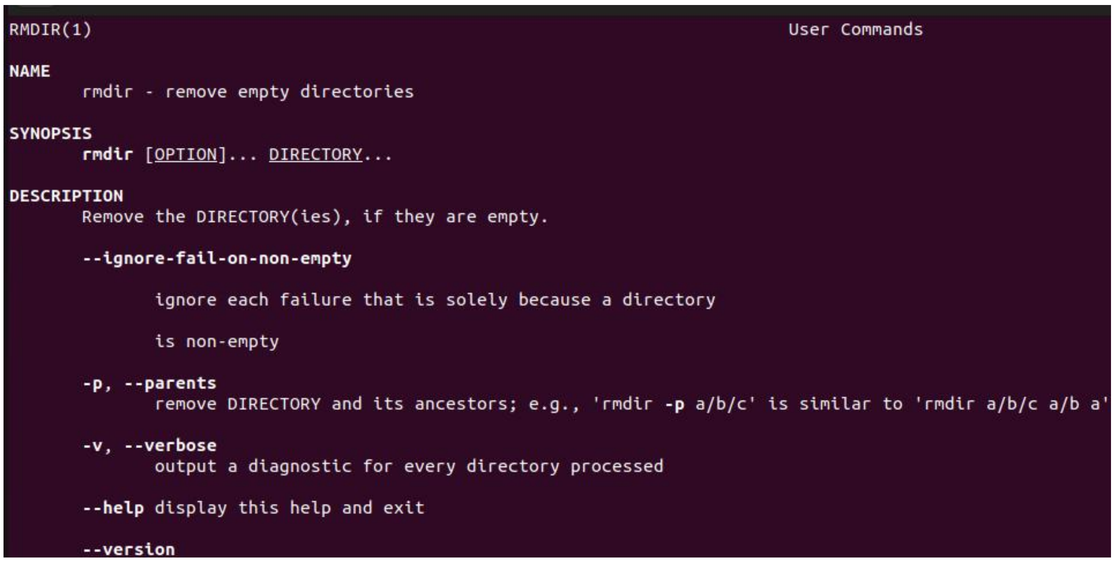
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network$ pwd
/home/mits/Desktop/s2mca-network
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network$ 
```

2.history : When you have been using Linux for a certain period oftime, you will quickly notice that you can run hundreds of commands every day. As such, running history command is particularly useful if you want to review the commands you have entered ,.before.



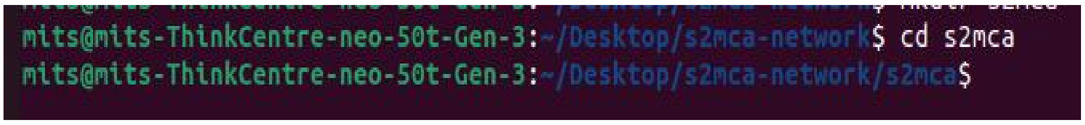
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network$ history
    1  sudo apt update
    2  sudo apt install gcc
    3  sudo apt install python3
    4  gcc --version
    5  python3 --version
    6  touch stack.c
    7  gcc stack.c
    8  gcc stack.c
    9  gcc stack.c
   10  ./a.out
   11  gcc stack.c
   12  ./a.out
```

2. **man :**by using this command you can easily learn how to use



```
RMDIR(1)                                                   User Commands

NAME
       rmdir - remove empty directories

SYNOPSIS
       rmdir [OPTION]... DIRECTORY...

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

       --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
```

3. **cd** :To navigate through the Linux files and directories, use the cd .It requires either the full path or the name of the directory, depending on the current working directory that you’re in.

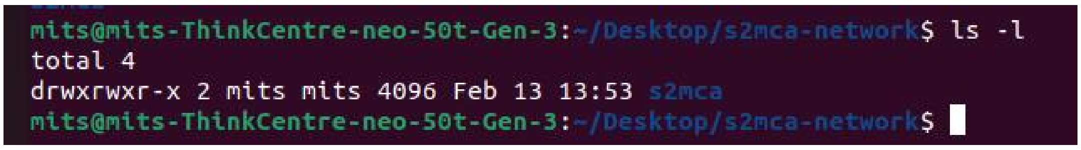


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network$ cd s2mca
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca$
```

5. **ls**:The ls command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.If you want to see the content of other directories, type ls and then the directory's path.

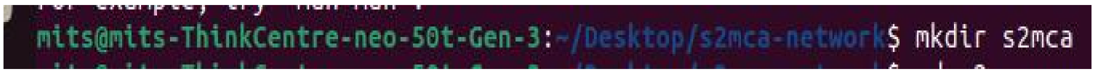
There are variations you can use with the ls command:

- ls -R will list all the files in the sub-directories as well
- ls –l – long listing
- ls -a will show the hidden files
- ls -al will list the files and directories with detailed information like the
- permissions, size, owner, etc.
- ls -t lists files sorted in the order of "last modified"
- ls -r option will reverse the natural sorting order. Usually used in combination withother switches such as ls -tr. This will reverse the time-wise listing.



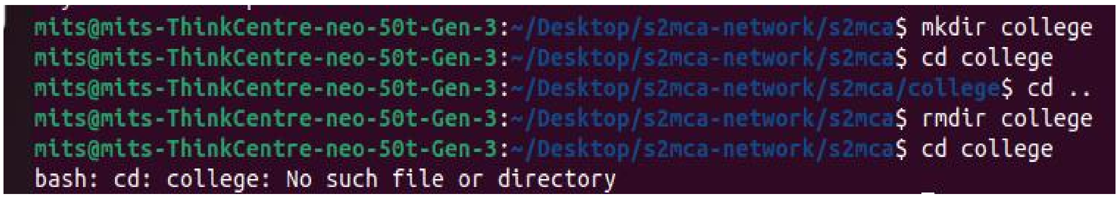
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network$ ls -l
total 4
drwxrwxr-x 2 mits mits 4096 Feb 13 13:53 s2mca
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network$
```

6. **mkdir :** Use mkdir command to make a new directory — if you type mkdir Music it will create a directory called Music.To generate a new directory inside another directory, use this Linux basic command.



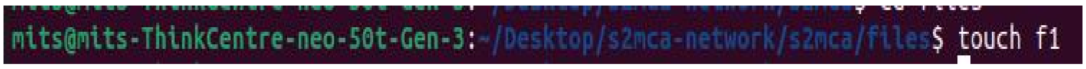
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network$ mkdir s2mca
```

7. **rmdir:**If you need to delete a directory, use the rmdir command.However, rmdir only allows you to delete empty directories.



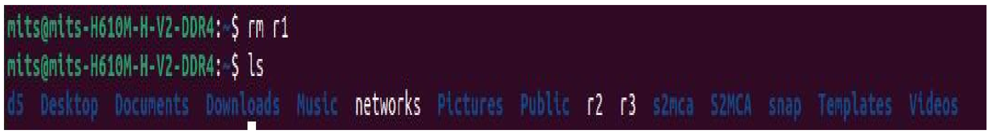
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca$ mkdir college
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca$ cd college
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/college$ cd ..
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca$ rmdir college
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca$ cd college
bash: cd: college: No such file or directory
```

8.touch: The touch command allows you to create a blank new filethrough the Linux command line.



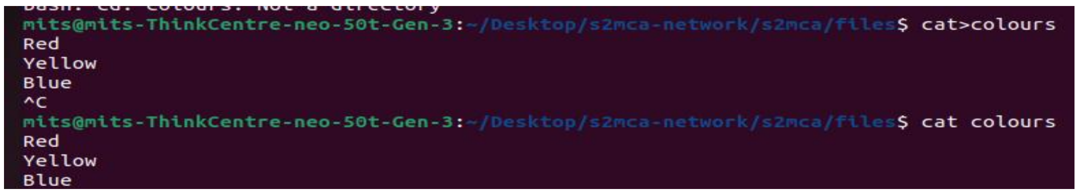
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$ touch f1
```

9.rm : The rm command is used to delete directories and the contents within them. If you only want to delete the directory —as an alternative to rmdir — use rm -r.Be very careful with this command and double-check which directory you are in. This will delete everything and there is no undo. To remove a file use rm filename.



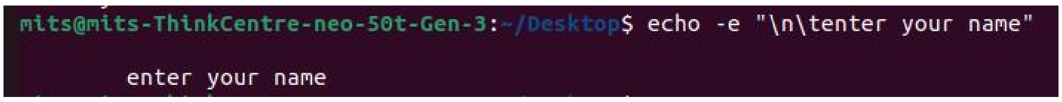
```
mits@mits-H610M-H-V2-DDR4:~$ rm r1
mits@mits-H610M-H-V2-DDR4:~$ ls
d5  Desktop  Documents  Downloads  Music  networks  Pictures  Public  r2  r3  s2mca  S2MCA  snap  Templates  Videos
```

10.Cat:cat (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output stdout . To run this command, type cat followed by the file's name and its extension.



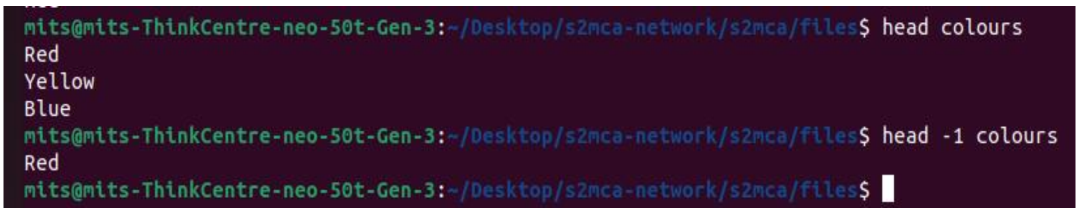
```
bash: cd: colours: Not a directory
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$ cat>colours
Red
Yellow
Blue
^C
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$ cat colours
Red
Yellow
Blue
```

11.echo: echo command is used to move some data into a file. If you want to add the text, "Hello, my name is John" into a file called name.txt, you would type echo Hello, myname is John >> name.txt 2. head.



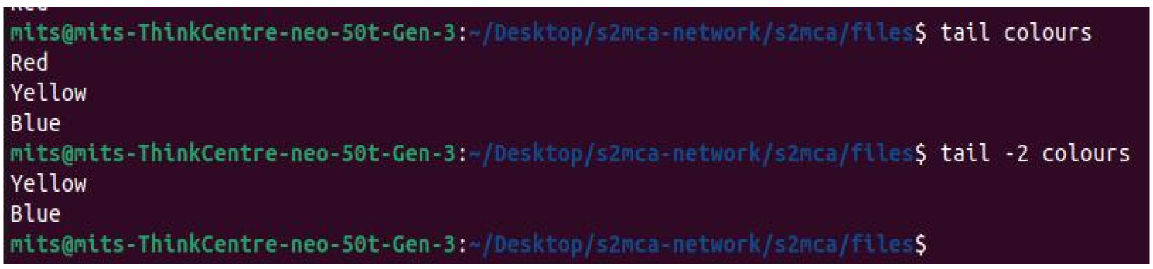
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop$ echo -e "\n\tenter your name"

        enter your name
```

12.head: The head command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. If youonly want to show the first five lines, type head -n 5 file name.txt.



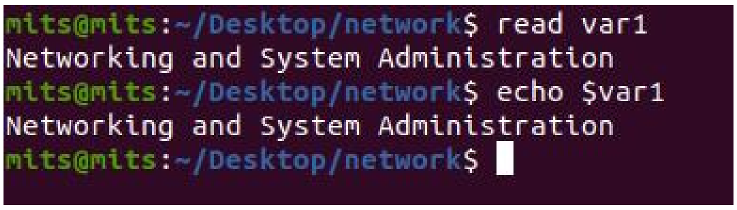
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$ head colours
Red
Yellow
Blue
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$ head -1 colours
Red
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$ 
```

13.tail: This one has a similar function to the head command, but instead of showing the first lines, the tail command will display the last ten lines of a text file. tail -n filename.txt.



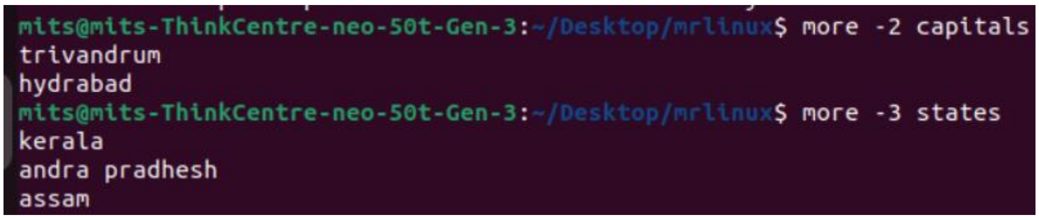
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$ tail colours
Red
Yellow
Blue
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$ tail -2 colours
Yellow
Blue
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca/files$
```

14.read: read the contents of a line into a variable. The read command can be used with and without arguments. read command is used to read [options] [name...] . $read $read var1 var2 var3. $echo &quot;[$var1] [$var2] [$var3].



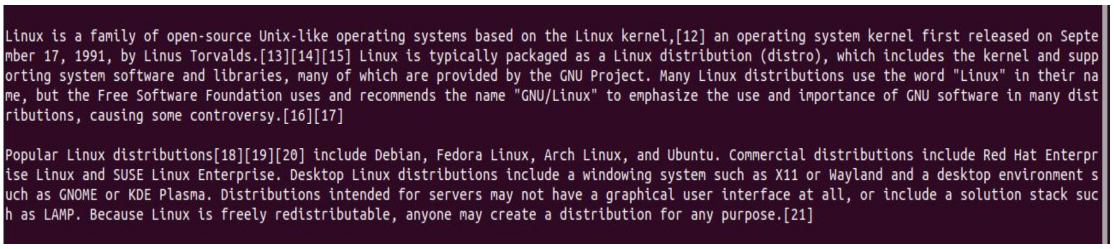
```
mits@mits:~/Desktop/network$ read var1
Networking and System Administration
mits@mits:~/Desktop/network$ echo $var1
Networking and System Administration
mits@mits:~/Desktop/network$ 
```

15. more: Like cat command, more command displays the content of a file. Only difference is that, in case of larger files, &#39; cat&#39; command output will scroll off your screen while &#39; more&#39; command displays output one screenful at a time. Enter key



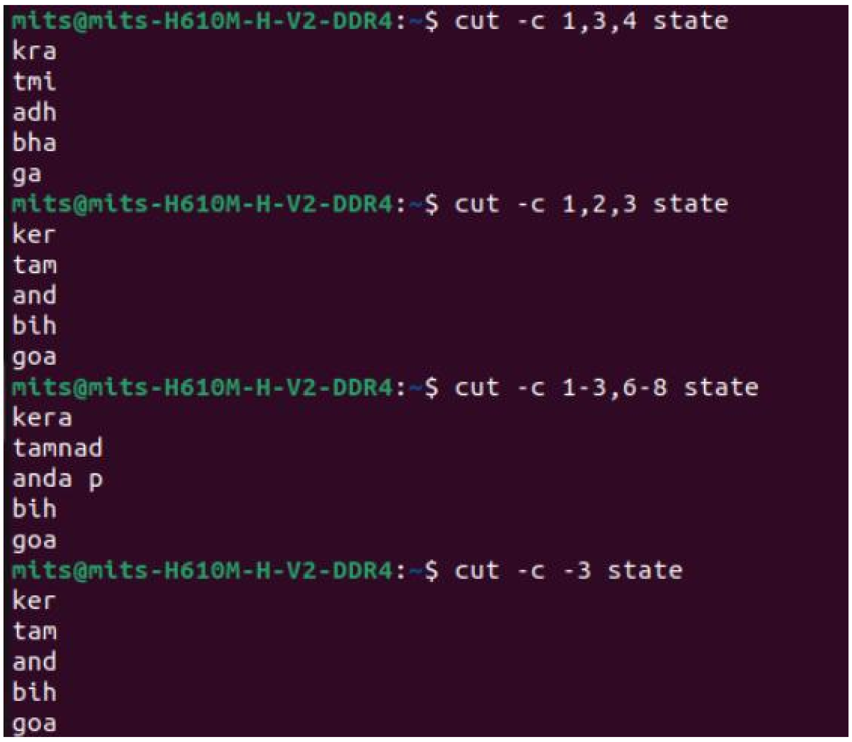
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ more -2 capitals
trivandrum
hydrabad
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ more -3 states
kerala
andra pradhesh
assam
```

16. less: The 'less' command is same as 'more' command but include some more features. It automatically adjusts with the width and height of the terminal window, while 'more' command cuts the content as the width of the terminal window get shorter.



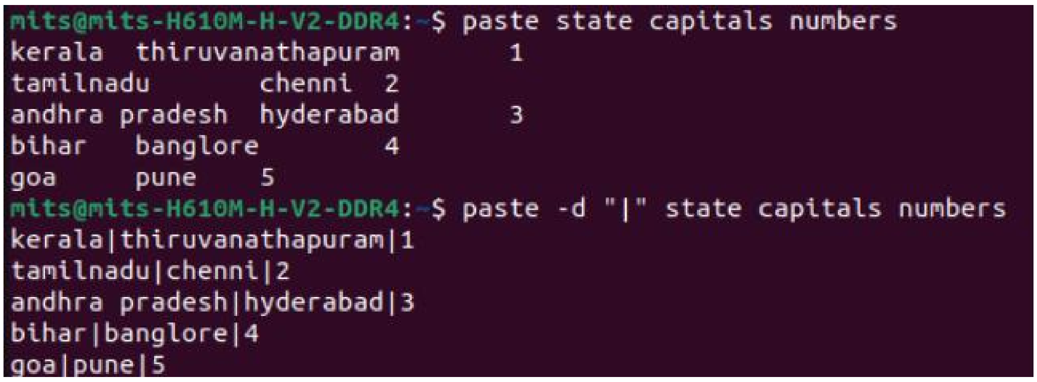
```
Linux is a family of open-source Unix-like operating systems based on the Linux kernel,[12] an operating system kernel first released on Septe
mber 17, 1991, by Linus Torvalds.[13][14][15] Linux is typically packaged as a Linux distribution (distro), which includes the kernel and supp
orting system software and libraries, many of which are provided by the GNU Project. Many Linux distributions use the word "Linux" in their na
me, but the Free Software Foundation uses and recommends the name "GNU/Linux" to emphasize the use and importance of GNU software in many dist
ributions, causing some controversy.[16][17]

Popular Linux distributions[18][19][20] include Debian, Fedora Linux, Arch Linux, and Ubuntu. Commercial distributions include Red Hat Enterpr
ise Linux and SUSE Linux Enterprise. Desktop Linux distributions include a windowing system such as X11 or Wayland and a desktop environment s
uch as GNOME or KDE Plasma. Distributions intended for servers may not have a graphical user interface at all, or include a solution stack suc
h as LAMP. Because Linux is freely redistributable, anyone may create a distribution for any purpose.[21]
```

17.cut : The cut command is used for cutting out the sections from each line offiles and writing the result to standard output. It can be used to cut parts of aline by byte position, character and file.



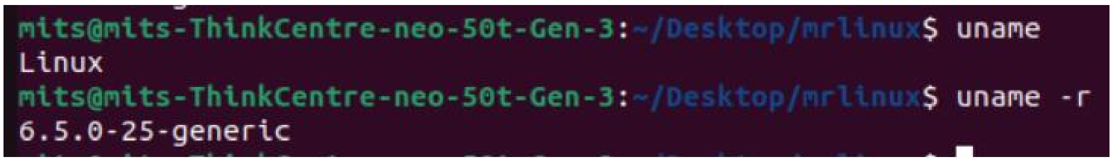
```
mits@mits-H610M-H-V2-DDR4:~$ cut -c 1,3,4 state
kra
tmi
adh
bha
ga
mits@mits-H610M-H-V2-DDR4:~$ cut -c 1,2,3 state
ker
tam
and
bih
goa
mits@mits-H610M-H-V2-DDR4:~$ cut -c 1-3,6-8 state
kera
tamnad
anda p
bih
goa
mits@mits-H610M-H-V2-DDR4:~$ cut -c -3 state
ker
tam
and
bih
goa
```

18.paste : It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output. paste [OPTION]... [FILES]...$ paste state.txt capital.txt.



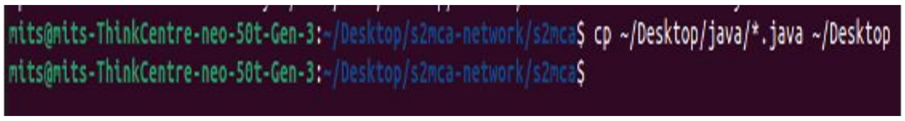
```
mits@mits-H610M-H-V2-DDR4:~$ paste state capitals numbers
kerala  thiruvanathapuram       1
tamilnadu       chenni  2
andhra pradesh  hyderabad       3
bihar   banglore        4
goa     pune    5
mits@mits-H610M-H-V2-DDR4:~$ paste -d "|" state capitals numbers
kerala|thiruvanathapuram|1
tamilnadu|chenni|2
andhra pradesh|hyderabad|3
bihar|banglore|4
goa|pune|5
```

19.uname : The uname command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on $uname, $uname-r



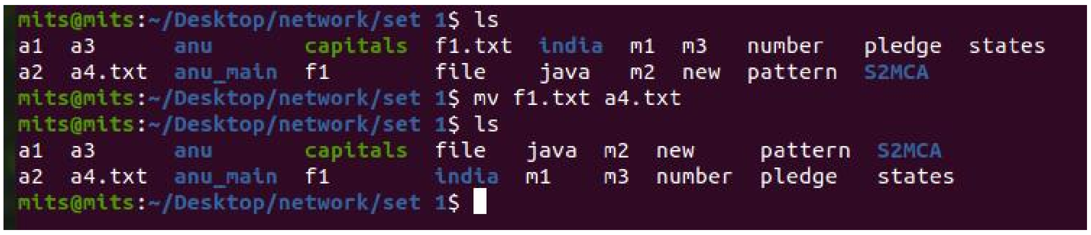
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ uname
Linux
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ uname -r
6.5.0-25-generic
```

20.cp : cp command issued to copy files from the current directory to a different directory. For instance, the command cp scenery.jpg /home/username/Pictures would create a copy of scenery.jpg (from your current directory) into the Pictures directory. cp -i will ask for user's consent in case of a potential file overwrite. cp -p will preserve source files'mode, ownership and timestamp. cp -r will copy directories recursively. cp -u copies files only if the destination file is not existing or thesource file is newer than the destination file.



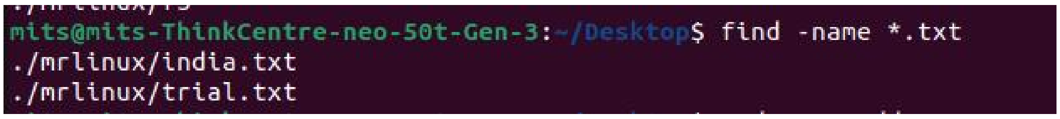
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca$ cp ~/Desktop/java/*.java ~/Desktop
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/s2mca-network/s2mca$
```

21.mv : The primary use of the mv command is to move files, it can also be used to rename files.The arguments in mv are similar to the cp command. You need to type mv, the file's name, and the destination's directory. mv file.txt /home/username/Documents .To rename files, the Linux is mv oldname.ext newname.ext.



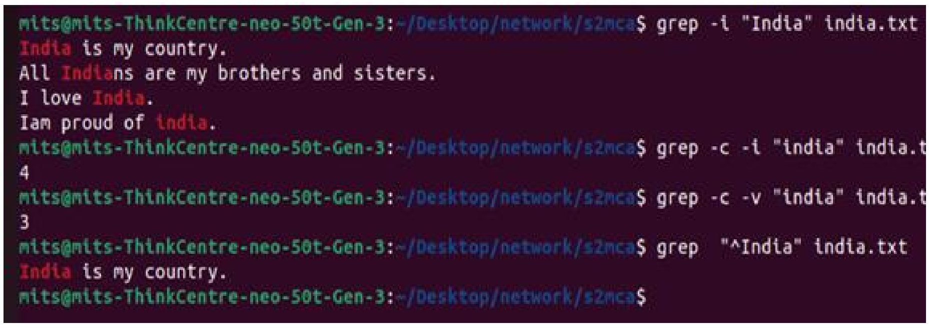
```
mits@mits:~/Desktop/network/set 1$ ls
a1  a3      anu       capitals  f1.txt  india  m1  m3   number   pledge  states
a2  a4.txt  anu_main  f1        file    java   m2  new  pattern  S2MCA
mits@mits:~/Desktop/network/set 1$ mv f1.txt a4.txt
mits@mits:~/Desktop/network/set 1$ ls
a1  a3      anu       capitals  file   java  m2  new     pattern  S2MCA
a2  a4.txt  anu_main  f1        india  m1    m3  number  pledge   states
mits@mits:~/Desktop/network/set 1$ 
```

22.Find : Similar to the locate command,using find also searches for files and directories. The difference is, you use the find command to locate files within a given directory. As an example, find /home/ -name notes.txt command will search for a file called notes.txt within the home directory and its subdirectories. Other variations when using the find are: To find files in the current directory use, find . -name notes.txt .To look for directorie suse, / -type d -name notes. Txt.



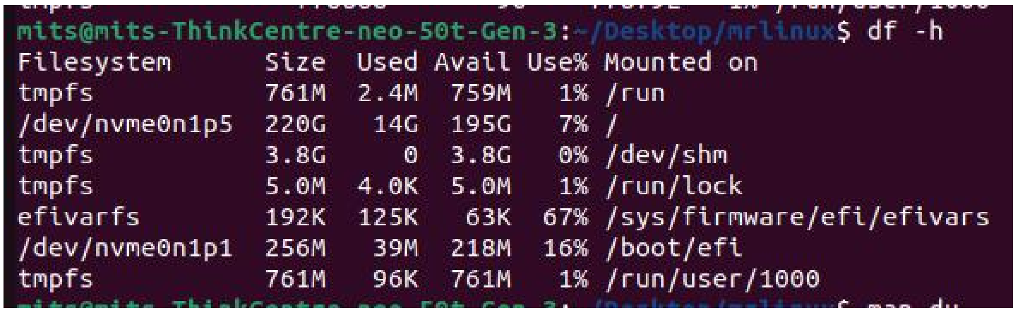
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop$ find -name *.txt
./mrlinux/india.txt
./mrlinux/trial.txt
```

23.grep : Another basic Linux command that is undoubtedly helpful for everyday use is grep. It lets you search through all the text in a given file. To illustrate, grep blue notepad.txt will search for the word blue in the notepadfile. Lines that contain the searched word will be displayed fully. Usually output of a previous command is piped intothe grep command. For example, ls -l |grep “kernel”.



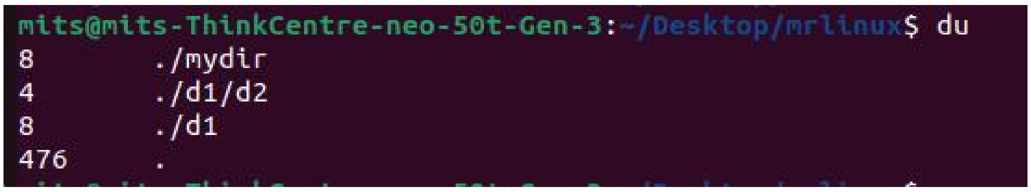
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ grep -i "India" india.txt
India is my country.
All Indians are my brothers and sisters.
I love India.
Iam proud of india.
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ grep -c -i "india" india.t
4
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ grep -c -v "india" india.t
3
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ grep  "^India" india.txt
India is my country.
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$
```

24.df : Use df command to get a report on the system’s disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type df – m.



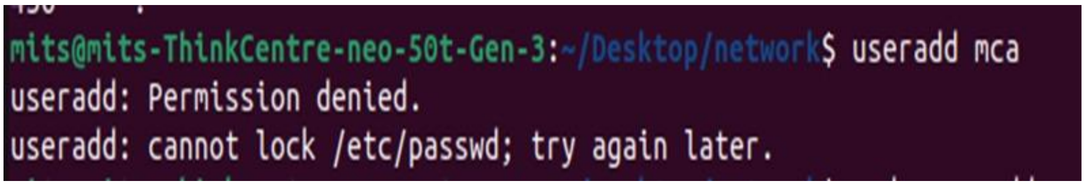
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           761M  2.4M  759M   1% /run
/dev/nvme0n1p5  220G   14G  195G   7% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
efivarfs        192K  125K   63K  67% /sys/firmware/efi/efivars
/dev/nvme0n1p1  256M   39M  218M  16% /boot/efi
tmpfs           761M   96K  761M   1% /run/user/1000
```

25.du : If you want to check how much space a file or a directory takes, the du (Disk Usage)command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the -h argument to the command line.



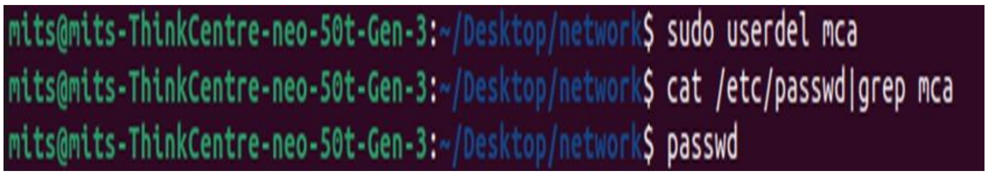
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ du
8       ./mydir
4       ./d1/d2
8       ./d1
476     .
```

26.useradd : This is available only to system admins .Since Linux is a multi- user system, this means more than one person can interact with the same system at the same time. useradd is used to create a new user, while passwd is adding a password to that user's account. To add a new person named John type, useradd John and then to add his password type, passwd 123456789.



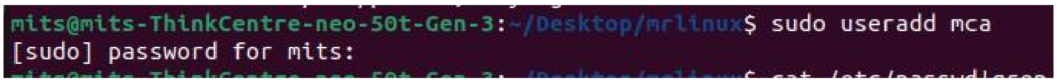
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network$ useradd mca
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
```

27.userdel : Remove a user is very similar to adding a new user. To delete the users accounttype, userdel UserName.



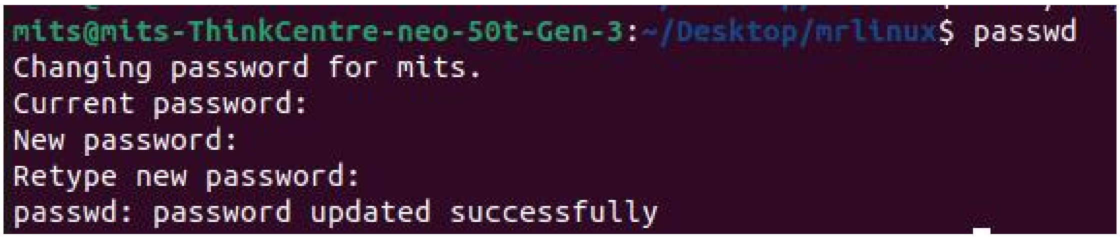
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network$ sudo userdel mca
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network$ cat /etc/passwd|grep mca
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network$ passwd
```

28sudo: Short for "SuperUser Do", this command enables you to perform tasks that require administrative or root permissions. You must have sufficient permissions to use this command



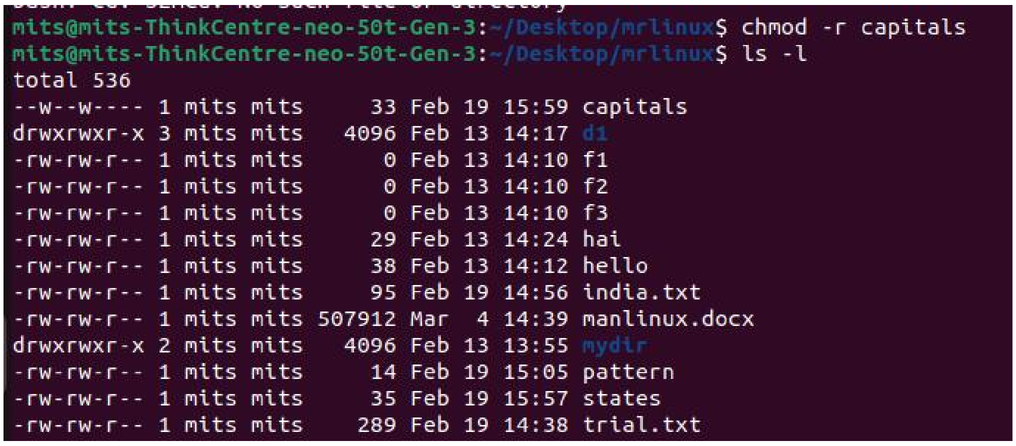
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ sudo useradd mca
[sudo] password for mits:
```

29.passwd : Changes passwords for user accounts. A normal user may only change the password for their own account, while the superuser may change the password for anyaccount.



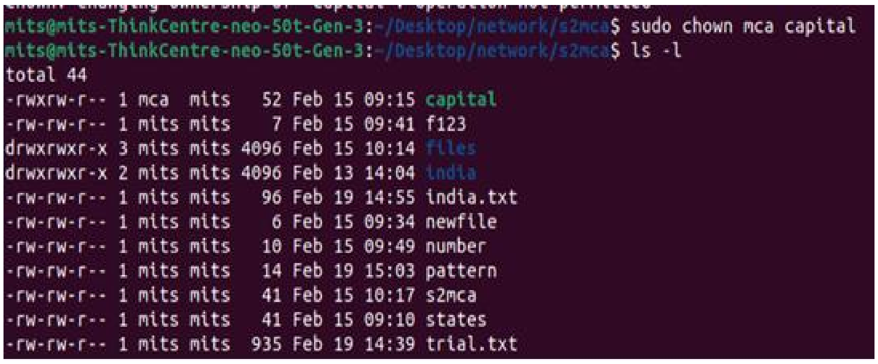
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ passwd
Changing password for mits.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

30.chmod : To change directory permissions of file/ Directory in Linux. #chmod who whatwhich file/directory chmod +rwx filename to add permissions. chmod -rwx directory name to remove permissions. chmod +x filename to allow executable permissions. chmod -wx filename to take out write and executable permissions. #chmod u+x test #chmod g- rwx test #chmod o-r test 4



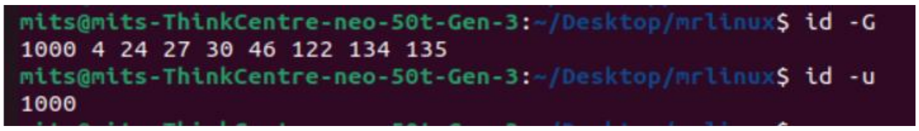
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ chmod -r capitals
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ ls -l
total 536
--w--w---- 1 mits mits     33 Feb 19 15:59 capitals
drwxrwxr-x 3 mits mits   4096 Feb 13 14:17 d1
-rw-rw-r-- 1 mits mits      0 Feb 13 14:10 f1
-rw-rw-r-- 1 mits mits      0 Feb 13 14:10 f2
-rw-rw-r-- 1 mits mits      0 Feb 13 14:10 f3
-rw-rw-r-- 1 mits mits     29 Feb 13 14:24 hai
-rw-rw-r-- 1 mits mits     38 Feb 13 14:12 hello
-rw-rw-r-- 1 mits mits     95 Feb 19 14:56 india.txt
-rw-rw-r-- 1 mits mits 507912 Mar  4 14:39 manlinux.docx
drwxrwxr-x 2 mits mits   4096 Feb 13 13:55 mydir
-rw-rw-r-- 1 mits mits     14 Feb 19 15:05 pattern
-rw-rw-r-- 1 mits mits     35 Feb 19 15:57 states
-rw-rw-r-- 1 mits mits    289 Feb 19 14:38 trial.txt
```

31. chown : The chown command allows you to change the user and/or group ownership of a given file, directory. #chownTom Test



```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ sudo chown mca capital
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ ls -l
total 44
-rwxrw-r-- 1 mca  mits   52 Feb 15 09:15 capital
-rw-rw-r-- 1 mits mits    7 Feb 15 09:41 f123
drwxrwxr-x 3 mits mits 4096 Feb 15 10:14 files
drwxrwxr-x 2 mits mits 4096 Feb 13 14:04 india
-rw-rw-r-- 1 mits mits   96 Feb 19 14:55 india.txt
-rw-rw-r-- 1 mits mits    6 Feb 15 09:34 newfile
-rw-rw-r-- 1 mits mits   10 Feb 15 09:49 number
-rw-rw-r-- 1 mits mits   14 Feb 19 15:03 pattern
-rw-rw-r-- 1 mits mits   41 Feb 15 10:17 s2mca
-rw-rw-r-- 1 mits mits   41 Feb 15 09:10 states
-rw-rw-r-- 1 mits mits  935 Feb 19 14:39 trial.txt
```

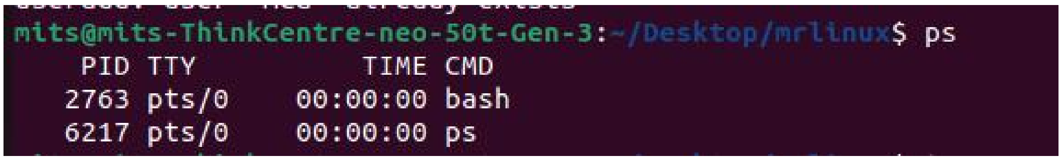
32.id : id command in Linux is used to find out user and group names and numeric ID's UID or group ID) of the current user.



```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ id -G
1000 4 24 27 30 46 122 134 135
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ id -u
1000
```

33.ps : The ps command, short for Process Status, is a command line utility that is used to display or view information related to the processes running in a Linux system. PID – This is the unique process ID TTY– This is the type of terminal that the user is logged in to . TIME – This is the time in minutes and seconds that the process has been running .CMD – The command that launched the process

Syntax:

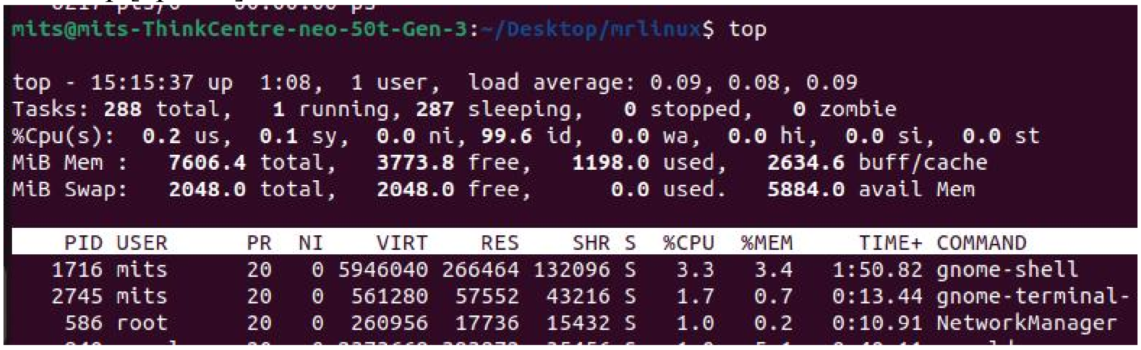


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ ps
    PID TTY          TIME CMD
   2763 pts/0    00:00:00 bash
   6217 pts/0    00:00:00 ps
```

34.top : top command is used to show the Linux processes. It provides a dynamic real-time view of the running system

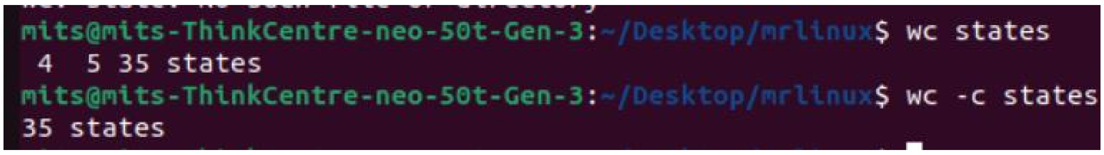
Syntax:

top [options]



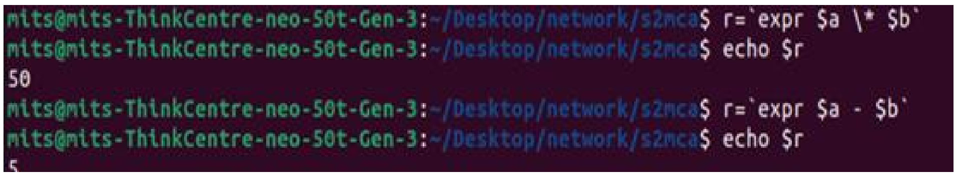
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ top

top - 15:15:37 up  1:08,  1 user,  load average: 0.09, 0.08, 0.09
Tasks: 288 total,   1 running, 287 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.1 sy,  0.0 ni, 99.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7606.4 total,   3773.8 free,   1198.0 used,   2634.6 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   5884.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1716 mits      20   0 5946040 266464 132096 S   3.3   3.4   1:50.82 gnome-shell
   2745 mits      20   0  561280  57552  43216 S   1.7   0.7   0:13.44 gnome-terminal-
    586 root      20   0  260956  17736  15432 S   1.0   0.2   0:10.91 NetworkManager
```

35. wc: wc stands for word count. Used for counting purpose. It is used to find out numberof lines, word count, byte and characters count in the files specified in the file arguments. #wc state.txt 6 8 54 state.tx . #wc state.txt capital.txt wc -l state.txt wc -w state.txt capital.txt wc -c state.txt .wc -m state.txt



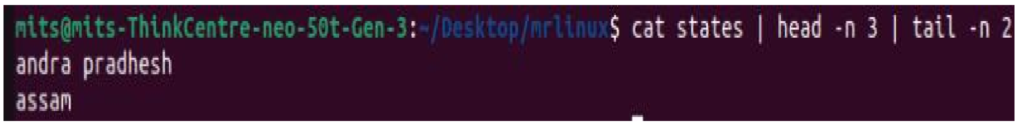
```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ wc states
 4  5 35 states
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ wc -c states
35 states
```

36.expr : The expr command evaluates a given expression and displays its corresponding output. It is used for: . Basic operations like addition, subtraction, multiplication, division, and modulus on integers. Evaluating regular expressions,string operations like substring, length of strings etc. Performing operations on variables inside a shell script.



```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ r=`expr $a \* $b`
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ echo $r
50
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ r=`expr $a - $b`
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ echo $r
```

37.Redirections & Piping : A pipe is a form of redirection to send the output of one command/program/process to another command/program/process for further processing. Pipe is used to combine two or more commands, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on.



```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ cat states | head -n 3 | tail -n 2
andra pradhesh
assam
```

SHELL SCRIPT

AIM

1.write a progranm to read 2 numbers and find sum,difference,product,quotient

PROGRAM

```
echo "Enter any two numbers:"
read a
read b
c=`expr $a + $b`
d=`expr $a - $b`
g=`expr $a \* $b`
h=`expr $a / $b`

echo -e "sum:"$c
echo -e "difference:"$d
echo -e "product:"$g
echo -e "quotient:"$h
```

OUTPUT

```
Enter any two numbers:
2
3
sum:5
difference:-1
product:6
quotient:0
```

AIM

2. check whether a number is divisible by 3

PROGRAM

```
echo "enter any numbers:"
read num
if [ `expr $num % 3` -eq 0 ]
then
echo "$num divisibe by 3"
else
echo "$num not divisible by 3"
Fi
```

OUTPUT

```
enter any numbers:
9
9 divisibe by 3

enter any numbers:
5
5  not divisibe by 3
```

AIM

3. write a program to check whether a person is eligible to vote

PROGRAM

```
echo "enter age of person:"
read age
if [  $age -ge 18 ]
then
echo "$age  is eligible to vote"
else
echo "$age  is  not eligible to vote"
fi
```

OUTPUT

```
enter age of person:
33
33  is eligible to vote
```

AIM
4. write a program largest of two numbers

PROGRAM

```
echo "enter any  two numbers:"
read a
read b
If [ $a -gt $b ]
then
echo "$a is larger"
elif [ $b -gt $a ]
then
echo "$b is larger"
else
echo ""both equal
fi
```

OUTPUT

```
enter any  two numbers:
3
4
4 is larger
```

AIM
5. write a program to find largest of three numbers

PROGRAM

```
echo "enter any   three numbers:"
read a
read b
read c
If [ $a -gt $b -a $a -gt $c ]
then
echo "$a is larger"
elif [ $b -gt $a -a $b -gt $c ]
then
echo "$b is larger"
elif [ $c -gt $a -a $c -gt $b ]
then
echo "$c is larger"
else "all are"
fi
```

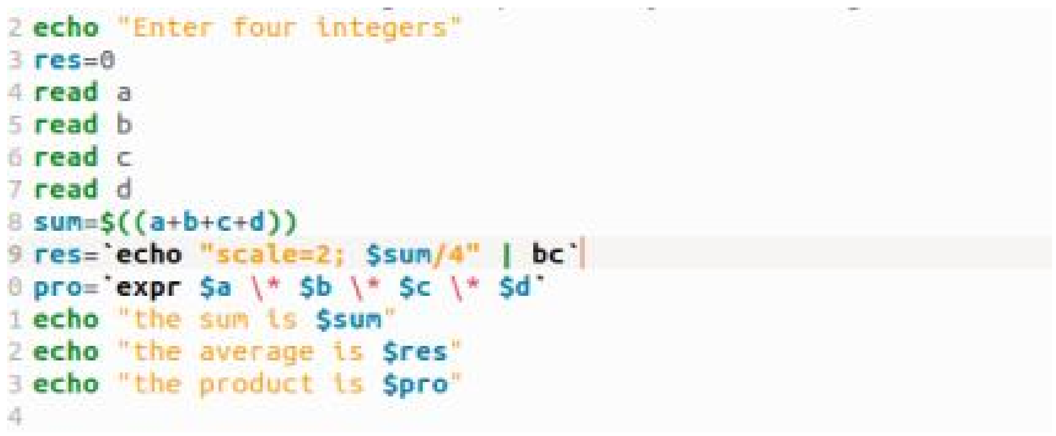
OUTPUT

```
4
5
6
6 is larger
```

AIM

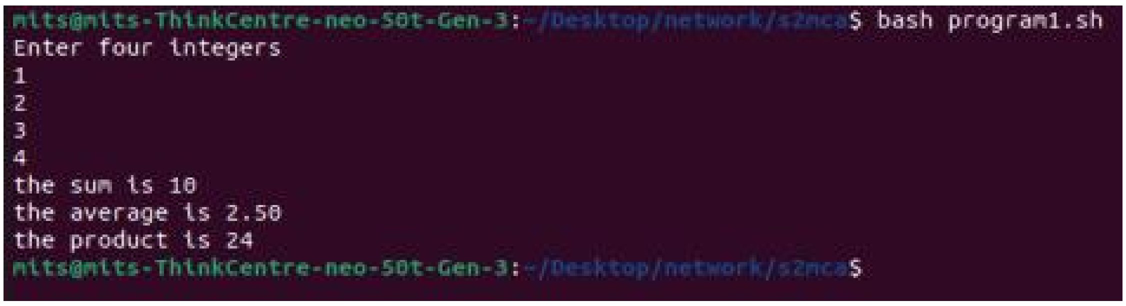
6.Find the sum,average and product of the 4 integers entered

PROGRAM



```
echo "Enter four integers"
res=0
read a
read b
read c
read d
sum=$((a+b+c+d))
res=`echo "scale=2; $sum/4" | bc`
pro=`expr $a \* $b \* $c \* $d`
echo "the sum is $sum"
echo "the average is $res"
echo "the product is $pro"
```

OUTPUT

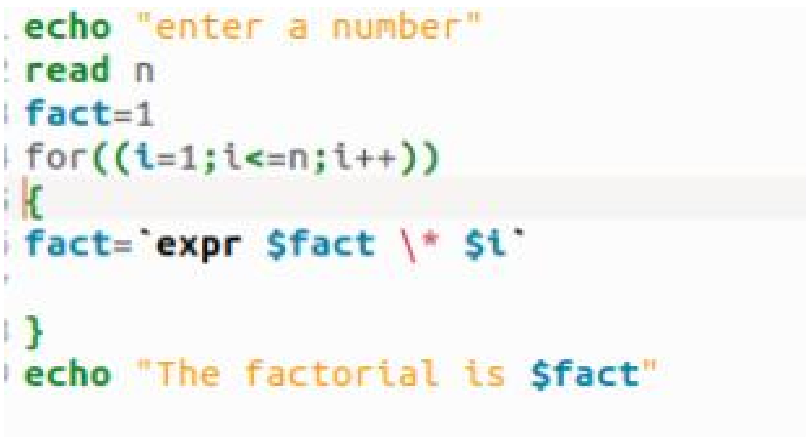


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ bash program1.sh
Enter four integers
1
2
3
4
the sum is 10
the average is 2.50
the product is 24
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$
```

AIM

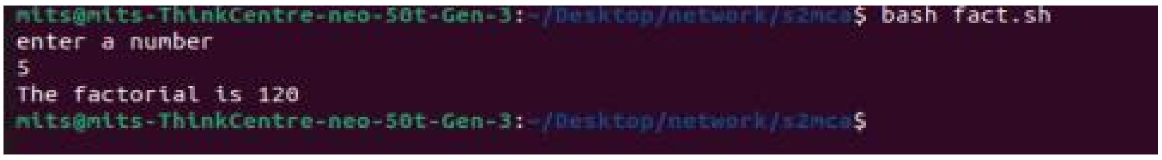
7.Write a shell script to find the factorial of a given number

PROGRAM



```
echo "enter a number"
read n
fact=1
for((i=1;i<=n;i++))
{
fact=`expr $fact \* $i`

}
echo "The factorial is $fact"
```

OUTPUT

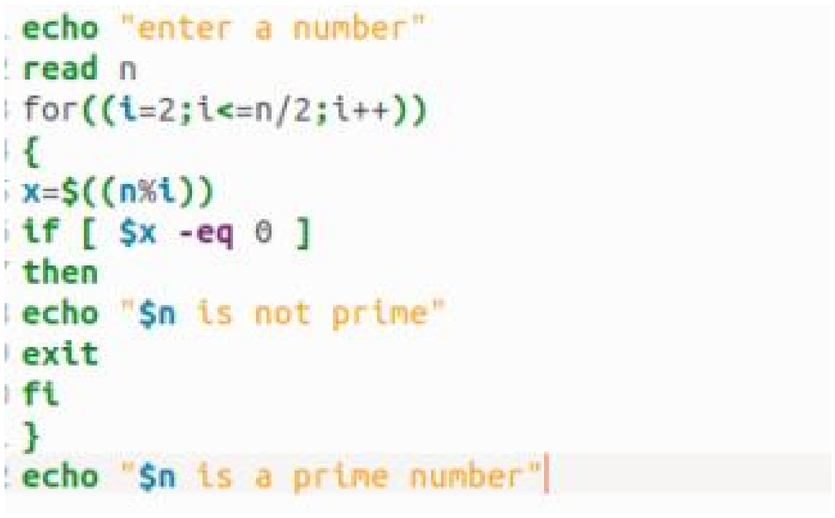


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ bash fact.sh
enter a number
5
The factorial is 120
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$
```

AIM

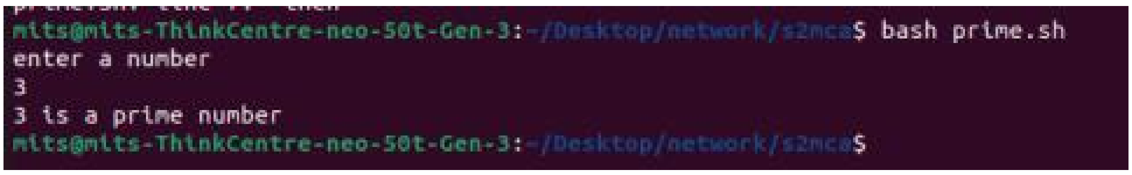
8.Write a shell script to check whether a given number is prime or not

PROGRAM



```
echo "enter a number"
read n
for((i=2;i<=n/2;i++))
{
x=$((n%i))
if [ $x -eq 0 ]
then
echo "$n is not prime"
exit
fi
}
echo "$n is a prime number"
```

OUTPUT

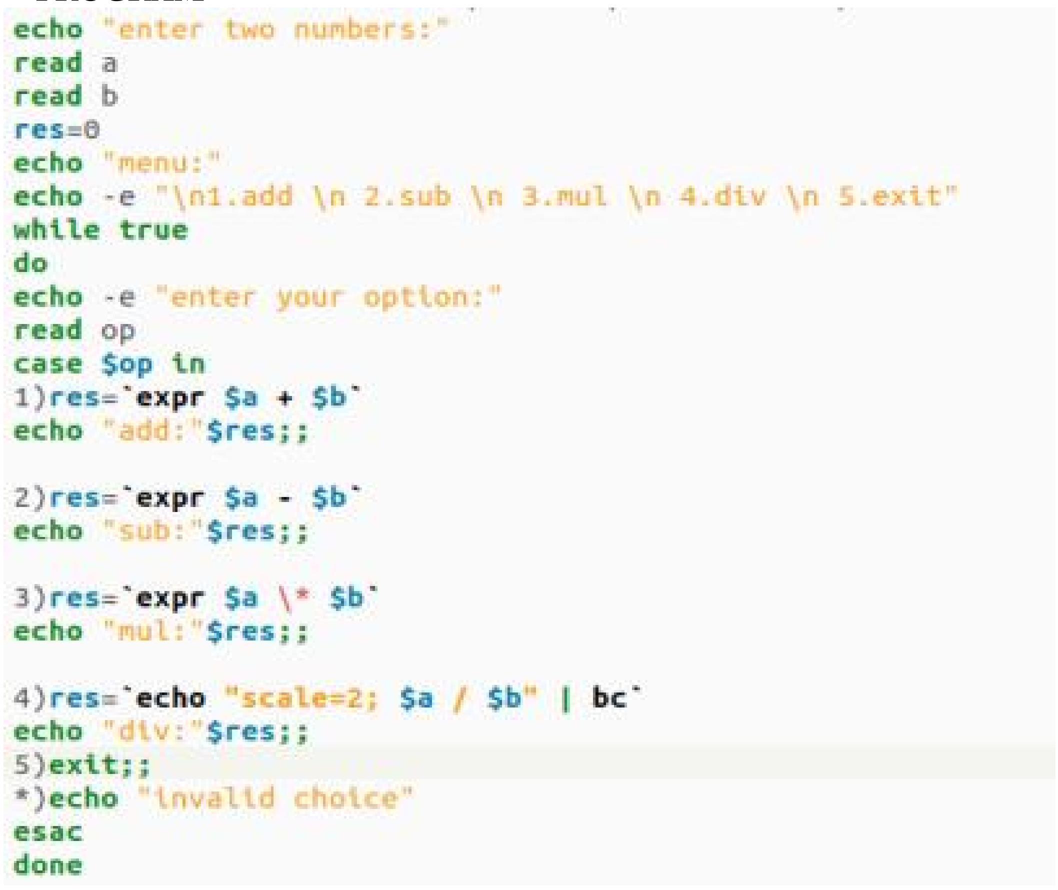


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$ bash prime.sh
enter a number
3
3 is a prime number
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/network/s2mca$
```

AIM

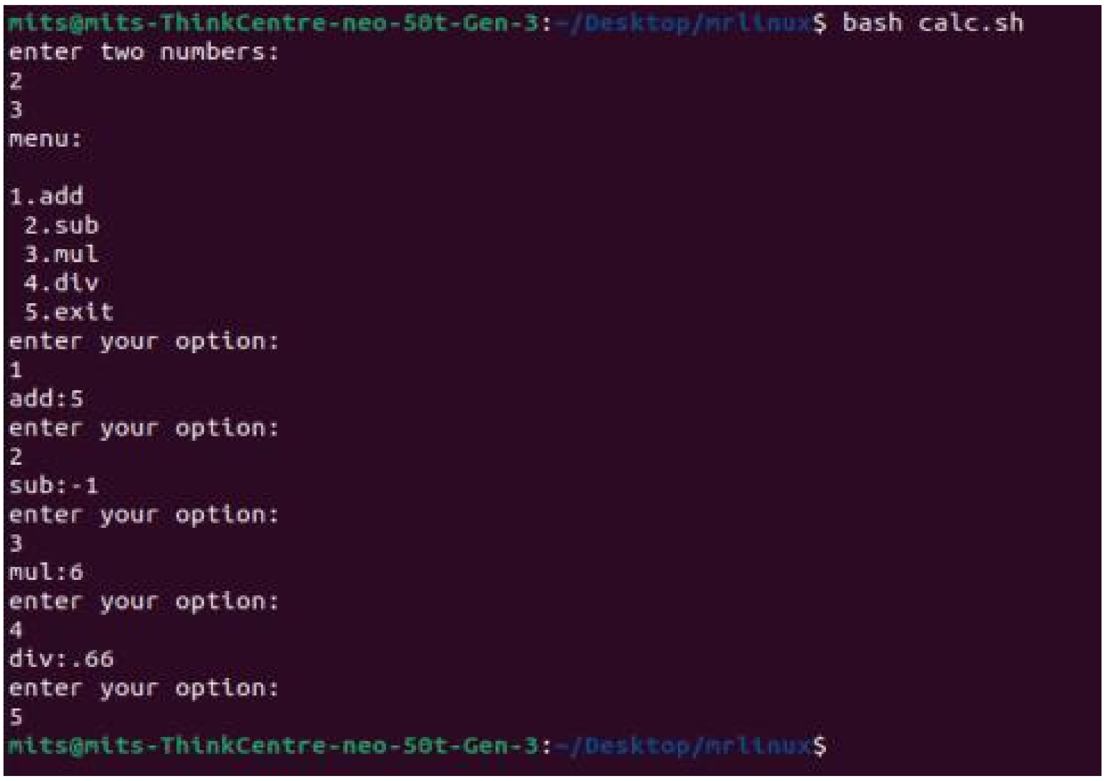
9.Write a shell script to implement a simple calculator

PROGRAM



```
echo "enter two numbers:"
read a
read b
res=0
echo "menu:"
echo -e "\n1.add \n 2.sub \n 3.mul \n 4.div \n 5.exit"
while true
do
echo -e "enter your option:"
read op
case $op in
1)res=`expr $a + $b`
echo "add:"$res;;

2)res=`expr $a - $b`
echo "sub:"$res;;

3)res=`expr $a \* $b`
echo "mul:"$res;;

4)res=`echo "scale=2; $a / $b" | bc`
echo "div:"$res;;
5)exit;;
*)echo "invalid choice"
esac
done
```

OUTPUT

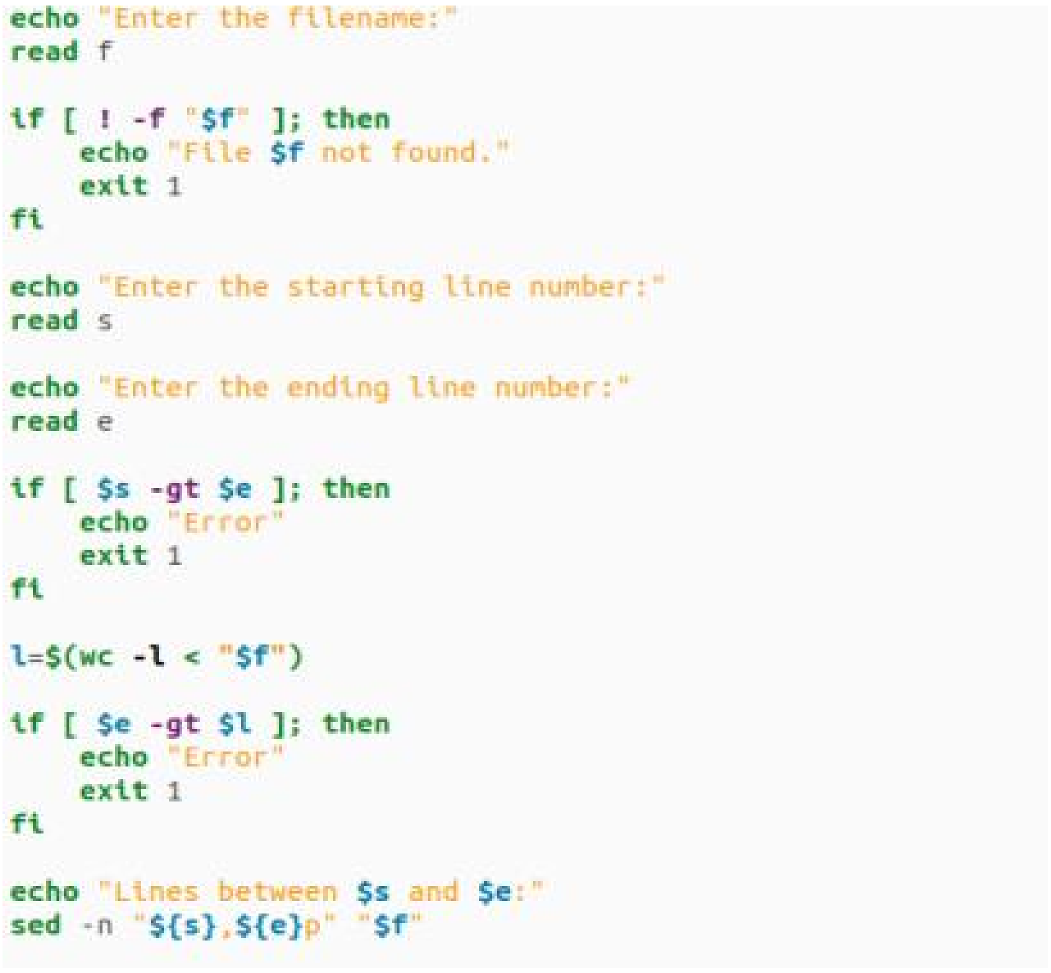


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ bash calc.sh
enter two numbers:
2
3
menu:

1.add
 2.sub
 3.mul
 4.div
 5.exit
enter your option:
1
add:5
enter your option:
2
sub:-1
enter your option:
3
mul:6
enter your option:
4
div:.66
enter your option:
5
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$
```

AIM

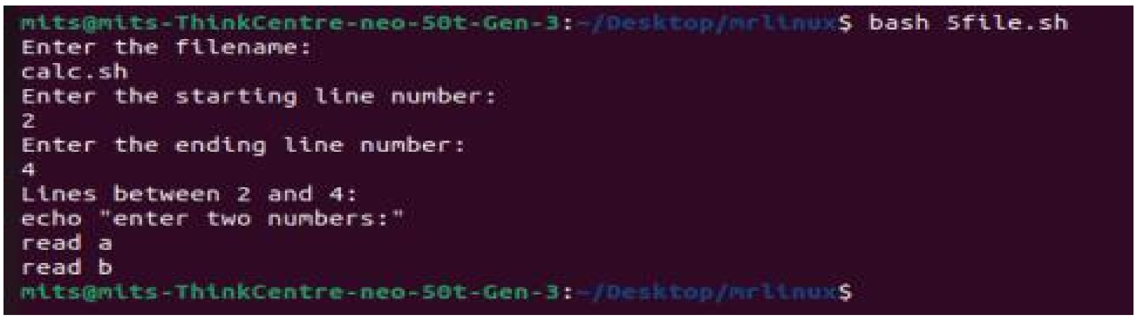
10.Write a shell script that accept a filename,starting and ending line numbers and display all the lines between the given line numbers.

PROGRAM



```
echo "Enter the filename:"
read f

if [ ! -f "$f" ]; then
    echo "File $f not found."
    exit 1
fi

echo "Enter the starting line number:"
read s

echo "Enter the ending line number:"
read e

if [ $s -gt $e ]; then
    echo "Error"
    exit 1
fi

l=$(wc -l < "$f")

if [ $e -gt $l ]; then
    echo "Error"
    exit 1
fi

echo "Lines between $s and $e:"
sed -n "${s},${e}p" "$f"
```

OUTPUT

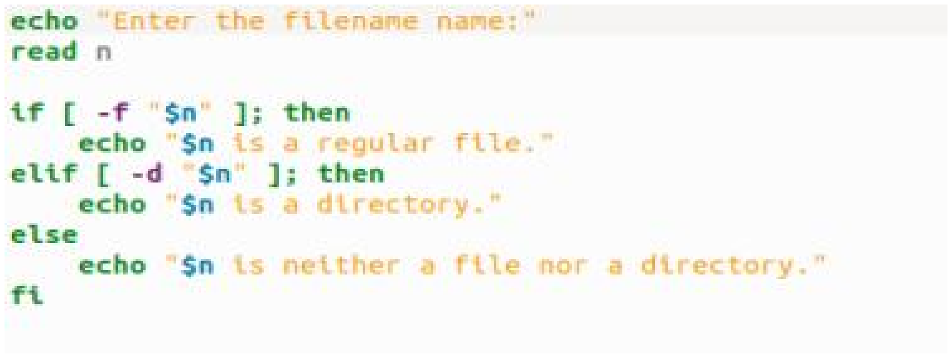


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ bash 5file.sh
Enter the filename:
calc.sh
Enter the starting line number:
2
Enter the ending line number:
4
Lines between 2 and 4:
echo "enter two numbers:"
read a
read b
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$
```

AIM

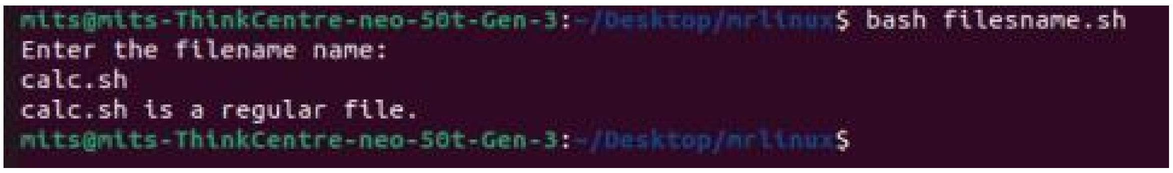
11.Write a shell script that receives file names or directory names as input and check if the input provided is a file or a directory

PROGRAM



```
echo "Enter the filename name:"
read n

if [ -f "$n" ]; then
    echo "$n is a regular file."
elif [ -d "$n" ]; then
    echo "$n is a directory."
else
    echo "$n is neither a file nor a directory."
fi
```

OUTPUT

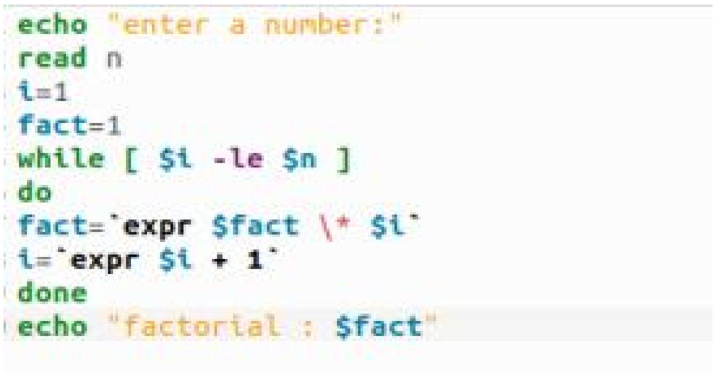


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ bash filesname.sh
Enter the filename name:
calc.sh
calc.sh is a regular file.
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$
```

AIM

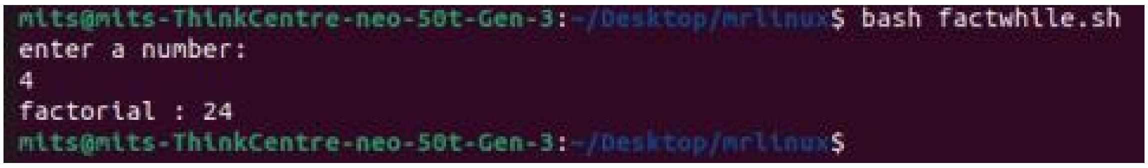
12.Write a shell program to find the factorial of a given number using a while loop.

PROGRAM



```
echo "enter a number:"
read n
i=1
fact=1
while [ $i -le $n ]
do
fact=`expr $fact \* $i`
i=`expr $i + 1`
done
echo "factorial : $fact"
```

OUTPUT

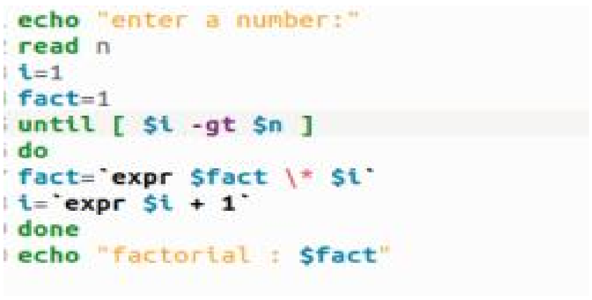


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ bash factwhile.sh
enter a number:
4
factorial : 24
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$
```

AIM

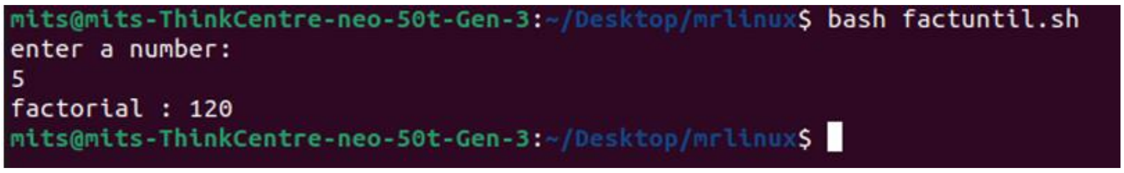
13.Write a shell program to find the factorial of a given number using do until

PROGRAM



```
echo "enter a number:"
read n
i=1
fact=1
until [ $i -gt $n ]
do
fact=`expr $fact \* $i`
i=`expr $i + 1`
done
echo "factorial : $fact"
```

OUTPUT

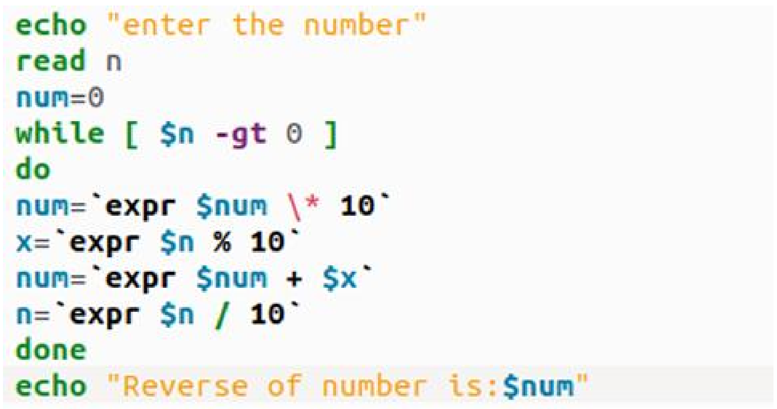


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ bash factuntil.sh
enter a number:
5
factorial : 120
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ 
```

AIM

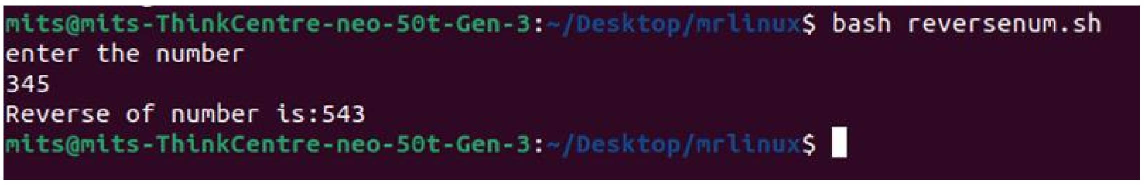
14.Write a program to find reverse of a given number

PROGRAM



```
echo "enter the number"
read n
num=0
while [ $n -gt 0 ]
do
num=`expr $num \* 10`
x=`expr $n % 10`
num=`expr $num + $x`
n=`expr $n / 10`
done
echo "Reverse of number is:$num"
```

OUTPUT

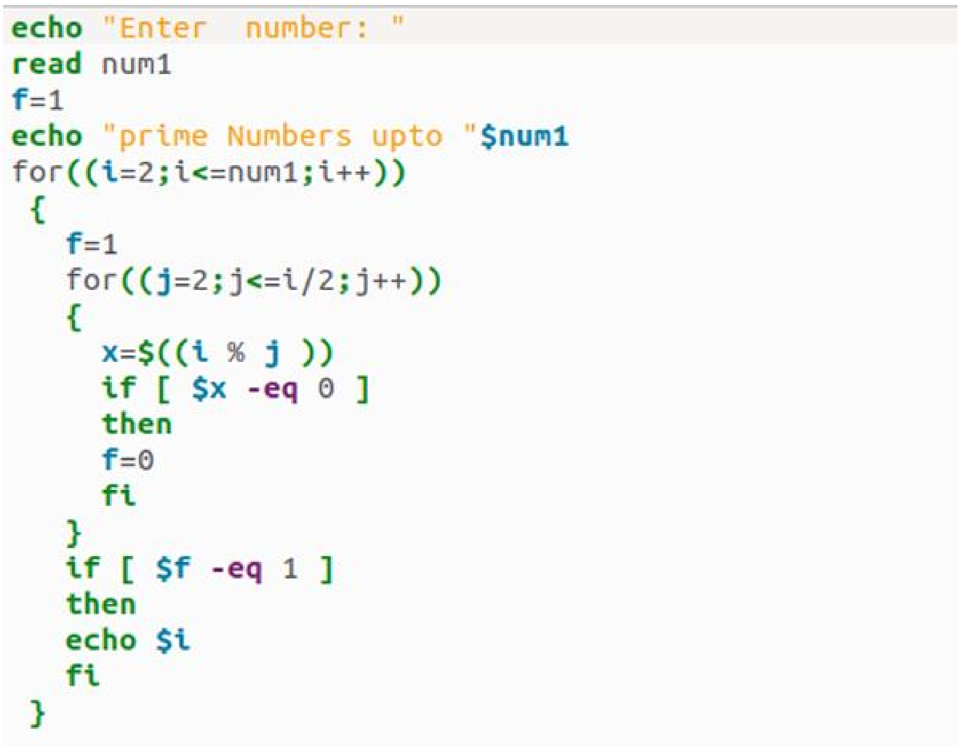


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ bash reversenum.sh
enter the number
345
Reverse of number is:543
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ 
```

AIM

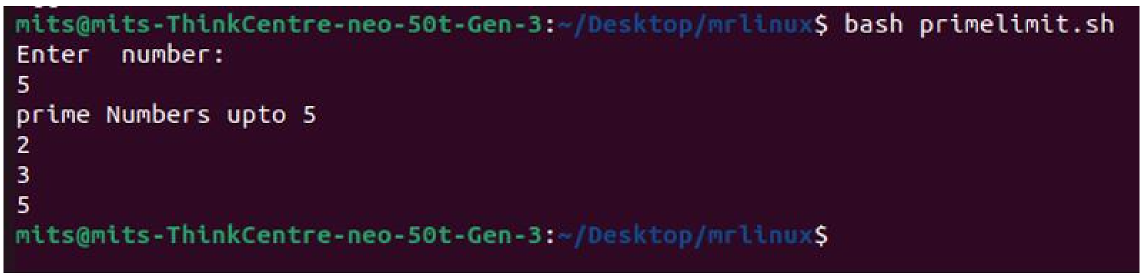
15.Write a shell script to display prime numbers within a specific limit.

PROGRAM



```
echo "Enter  number: "
read num1
f=1
echo "prime Numbers upto "$num1
for((i=2;i<=num1;i++))
 {
   f=1
   for((j=2;j<=i/2;j++))
   {
     x=$((i % j ))
     if [ $x -eq 0 ]
     then
     f=0
     fi
   }
   if [ $f -eq 1 ]
   then
   echo $i
   fi
 }
```

OUTPUT

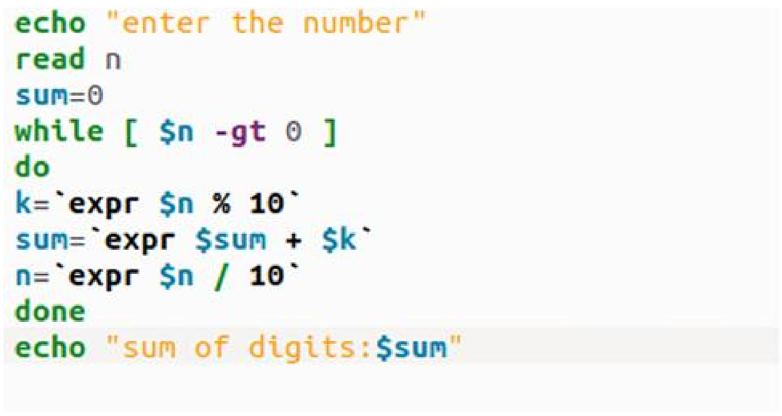


```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ bash primelimit.sh
Enter  number:
5
prime Numbers upto 5
2
3
5
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$
```

AIM

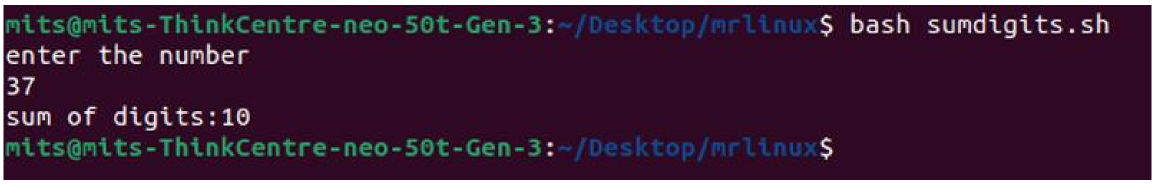
16.Write a shell script to find the sum of digits of a number using any loop

PROGRAM



```
echo "enter the number"
read n
sum=0
while [ $n -gt 0 ]
do
k=`expr $n % 10`
sum=`expr $sum + $k`
n=`expr $n / 10`
done
echo "sum of digits:$sum"
```

OUTPUT



```
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$ bash sumdigits.sh
enter the number
37
sum of digits:10
mits@mits-ThinkCentre-neo-50t-Gen-3:~/Desktop/mrlinux$
```