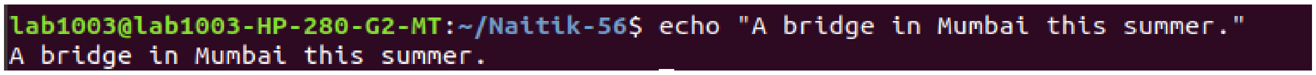
# 1. Basic Unix Commands

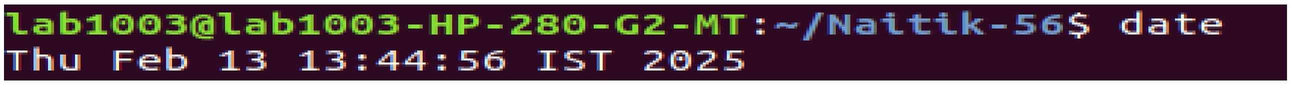
**AIM:** To implement Basic Commands in unix

1. echo – echo "Hello, Unix!" → Prints text to the terminal.



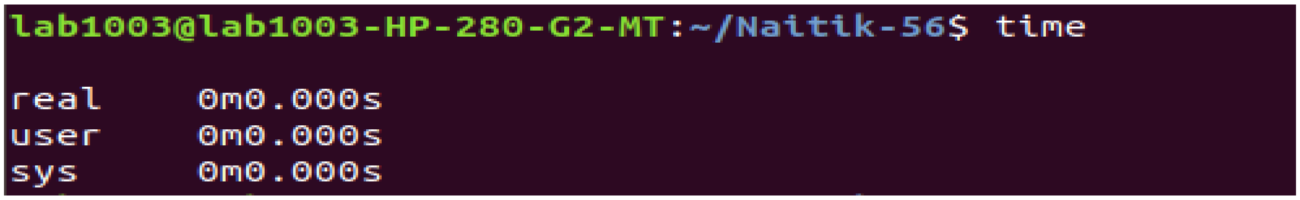
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ echo "A bridge in Mumbai this summer."
A bridge in Mumbai this summer.
```

2. clear – clear → Clears the terminal screen.
3. exit – exit → Exits the terminal session.
4. date – date → Displays the current date and time.



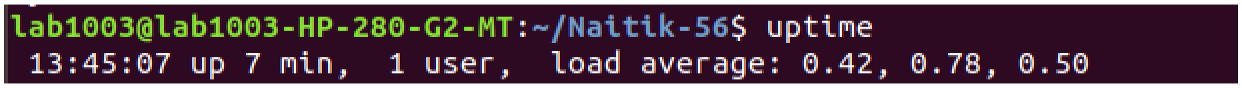
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ date
Thu Feb 13 13:44:56 IST 2025
```

5. time – time ls → Measures execution time of a command.



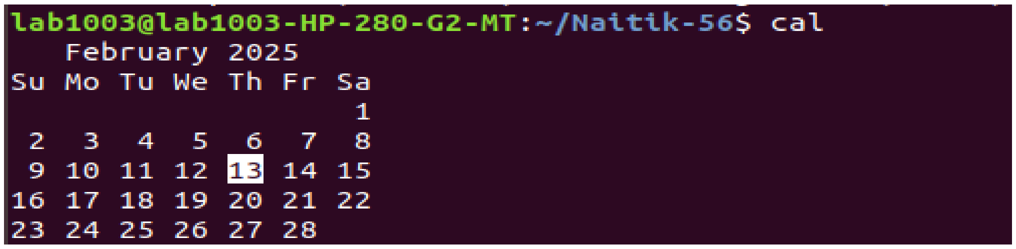
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ time

real    0m0.000s
user    0m0.000s
sys     0m0.000s
```

6. uptime – uptime → Shows system uptime and load average.



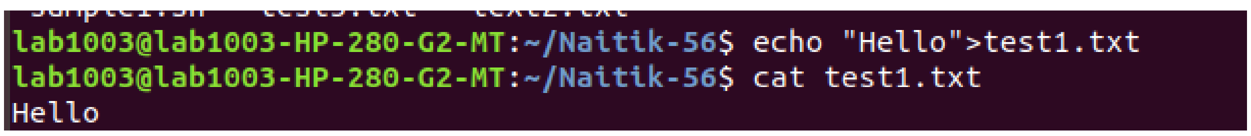
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ uptime
 13:45:07 up 7 min,  1 user,  load average: 0.42, 0.78, 0.50
```

7. cal – cal 2025 → Displays the calendar for the year 2025.



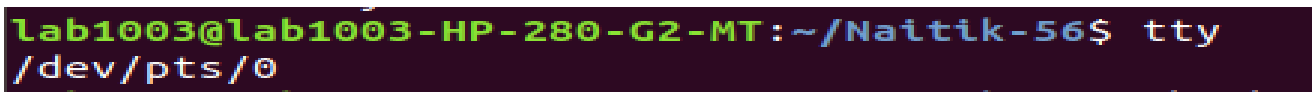
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ cal
   February 2025
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28
```

8. cat – cat file.txt → Displays the content of a file.



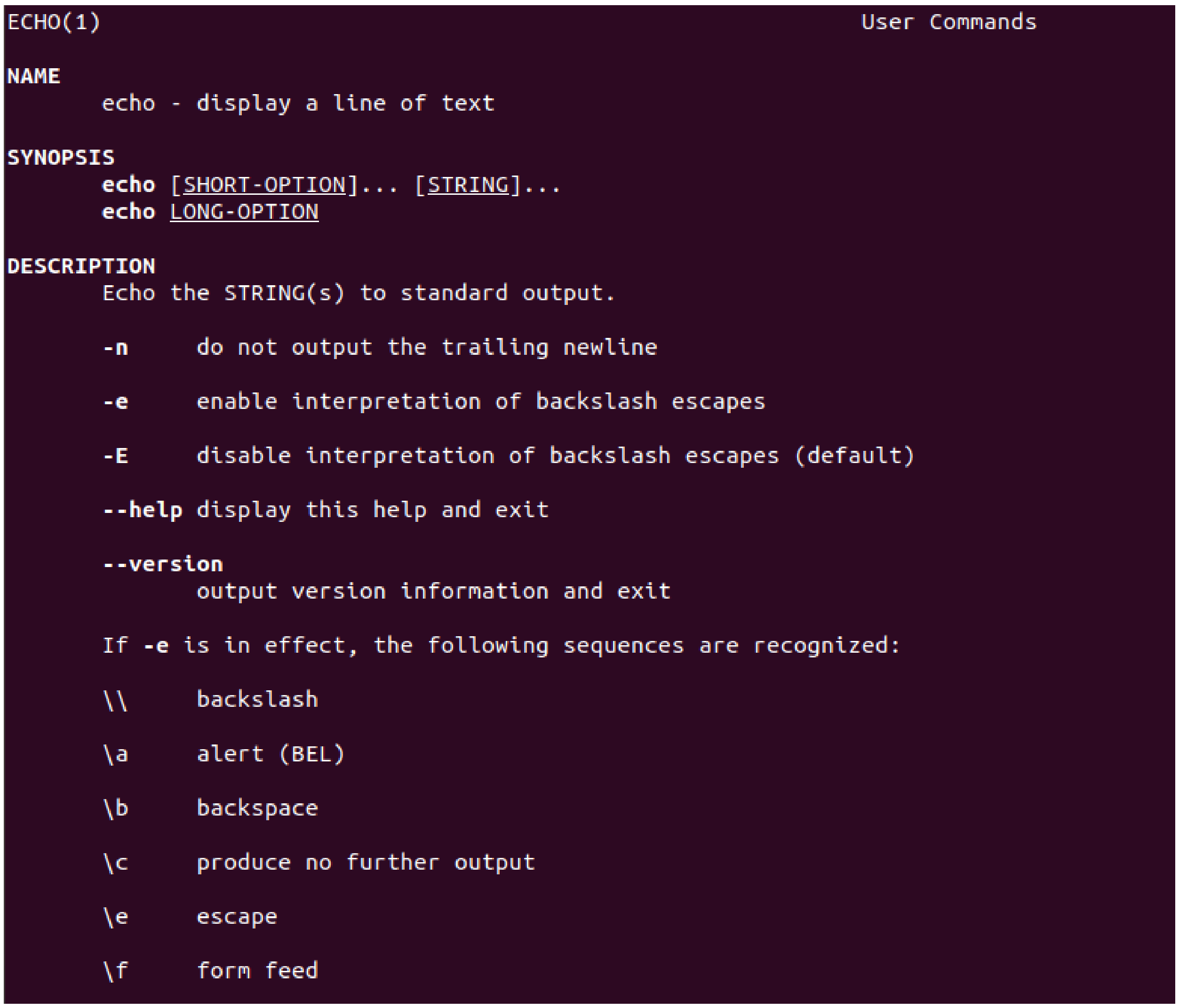
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ echo "Hello">test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ cat test1.txt
Hello
```

9. tty – tty → Shows the terminal's file name.



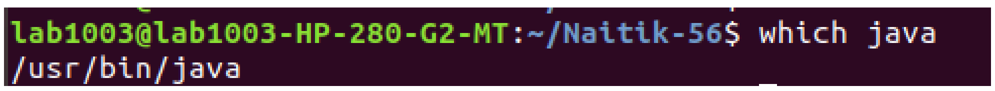
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ tty
/dev/pts/0
```

10. man – man ls → Displays the manual page for a command.



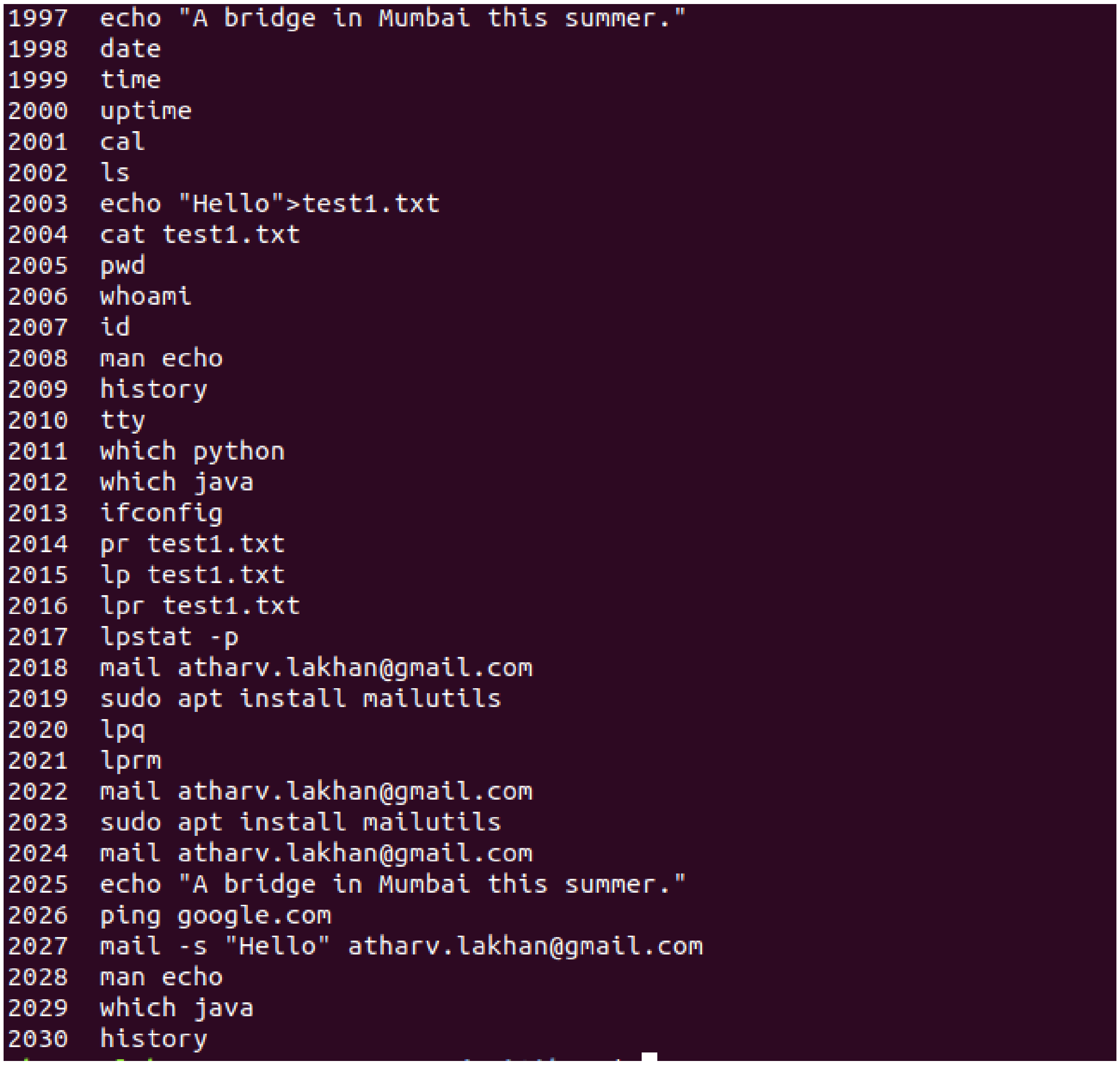
```
ECHO(1)                                                          User Commands

NAME
       echo - display a line of text

SYNOPSIS
       echo [SHORT-OPTION]... [STRING]...
       echo LONG-OPTION

DESCRIPTION
       Echo the STRING(s) to standard output.

       -n     do not output the trailing newline

       -e     enable interpretation of backslash escapes

       -E     disable interpretation of backslash escapes (default)

       --help display this help and exit

       --version
              output version information and exit

       If -e is in effect, the following sequences are recognized:

       \\     backslash

       \a     alert (BEL)

       \b     backspace

       \c     produce no further output

       \e     escape

       \f     form feed
```

11. which – which java → Shows the path of an executable command.



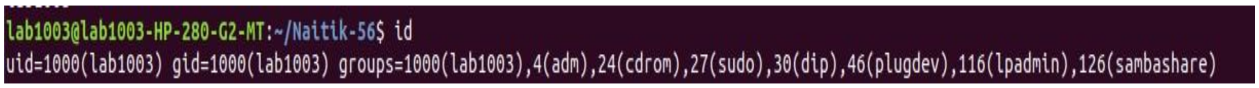
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ which java
/usr/bin/java
```

12. history – history → Lists previously executed commands.



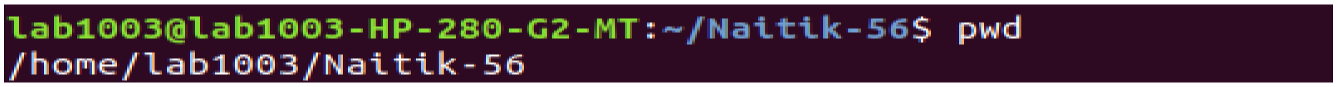
```
1997  echo "A bridge in Mumbai this summer."
1998  date
1999  time
2000  uptime
2001  cal
2002  ls
2003  echo "Hello">test1.txt
2004  cat test1.txt
2005  pwd
2006  whoami
2007  id
2008  man echo
2009  history
2010  tty
2011  which python
2012  which java
2013  ifconfig
2014  pr test1.txt
2015  lp test1.txt
2016  lpr test1.txt
2017  lpstat -p
2018  mail atharv.lakhan@gmail.com
2019  sudo apt install mailutils
2020  lpq
2021  lprm
2022  mail atharv.lakhan@gmail.com
2023  sudo apt install mailutils
2024  mail atharv.lakhan@gmail.com
2025  echo "A bridge in Mumbai this summer."
2026  ping google.com
2027  mail -s "Hello" atharv.lakhan@gmail.com
2028  man echo
2029  which java
2030  history
```

13. id – id → Displays user and group IDs.



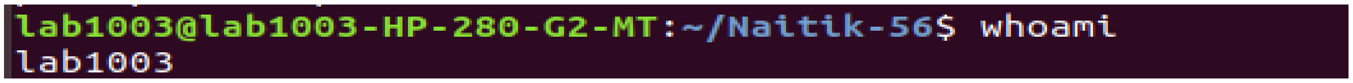
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ id
uid=1000(lab1003) gid=1000(lab1003) groups=1000(lab1003),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lpadmin),126(sambashare)
```

14. pwd – pwd → Prints the current working directory.



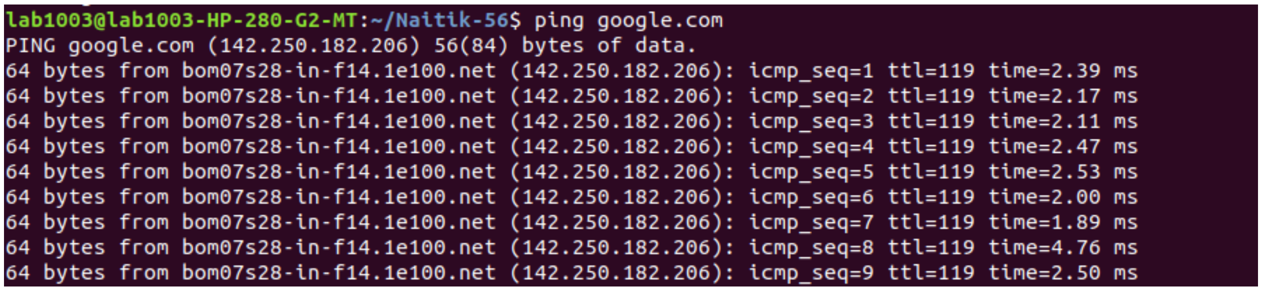
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ pwd
/home/lab1003/Naitik-56
```

15. whoami – whoami → Displays the current logged-in username.



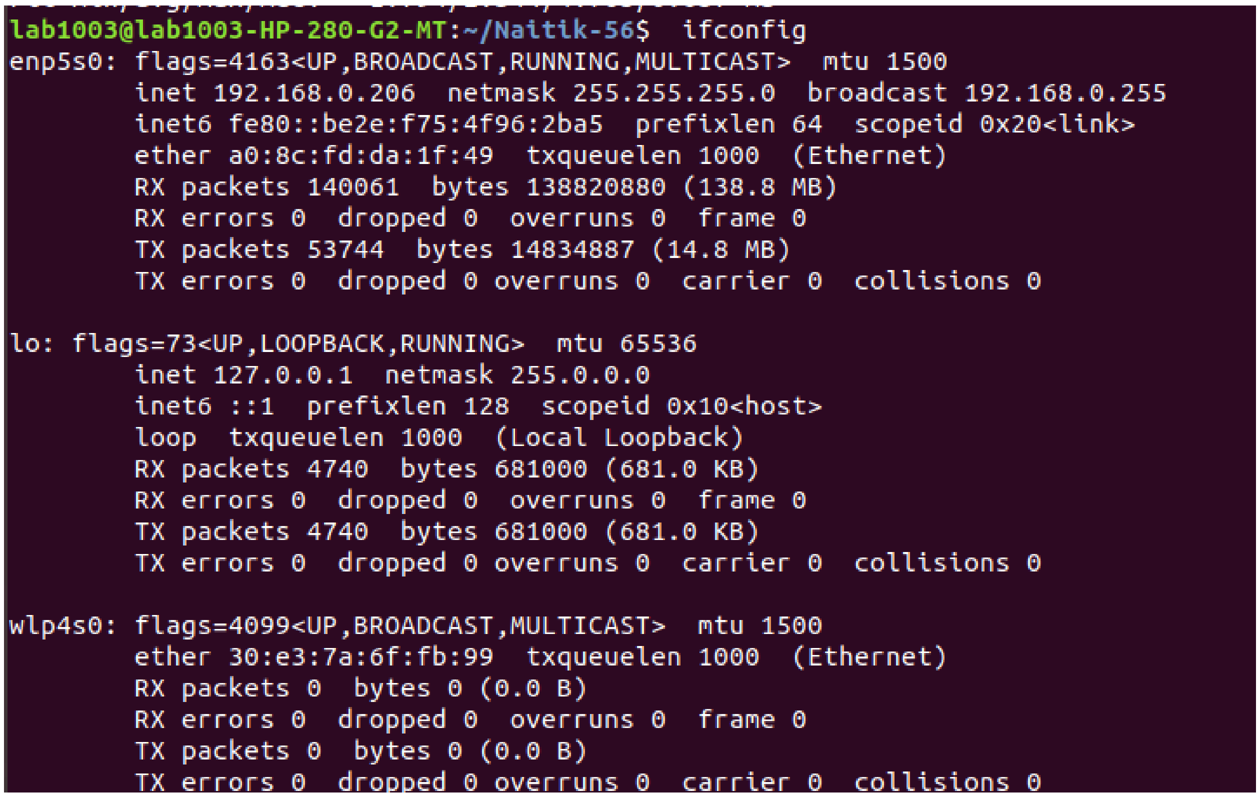
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ whoami
lab1003
```

16. ping – ping google.com → Checks network connectivity.



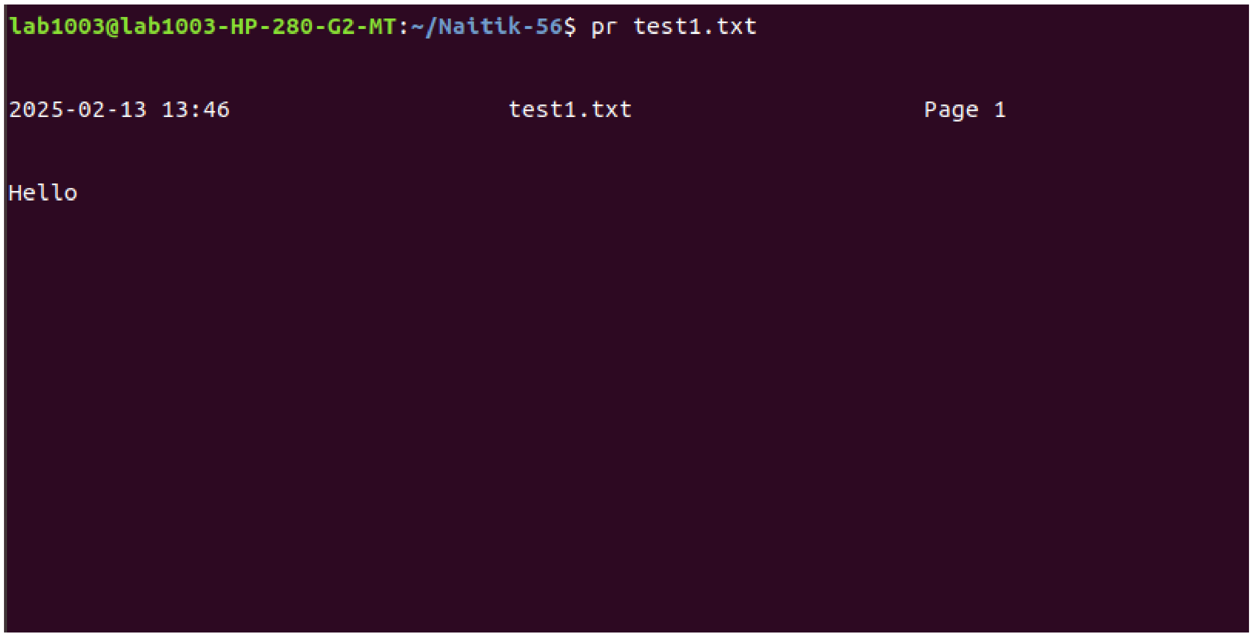
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ ping google.com
PING google.com (142.250.182.206) 56(84) bytes of data.
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=1 ttl=119 time=2.39 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=2 ttl=119 time=2.17 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=3 ttl=119 time=2.11 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=4 ttl=119 time=2.47 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=5 ttl=119 time=2.53 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=6 ttl=119 time=2.00 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=7 ttl=119 time=1.89 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=8 ttl=119 time=4.76 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=9 ttl=119 time=2.50 ms
```

17. ifconfig – ifconfig → Displays network interface details (deprecated in favor of ip a).



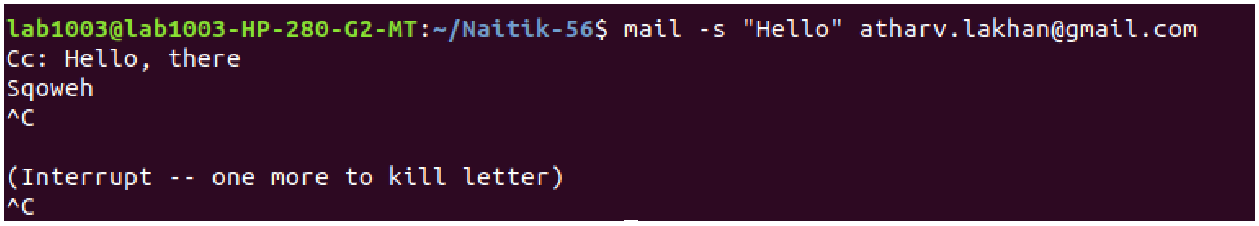
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$  ifconfig
enp5s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.206  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::be2e:f75:4f96:2ba5  prefixlen 64  scopeid 0x20<link>
        ether a0:8c:fd:da:1f:49  txqueuelen 1000  (Ethernet)
        RX packets 140061  bytes 138820880 (138.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 53744  bytes 14834887 (14.8 MB)
        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 4740  bytes 681000 (681.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4740  bytes 681000 (681.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp4s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 30:e3:7a:6f:fb:99  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
```

18. pr – pr file.txt → Formats a file for printing.



```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ pr test1.txt


2025-02-13 13:46                    test1.txt                      Page 1


Hello
```

19. lp – lp file.txt → Sends a file to the default printer.
20. lpr – lpr file.txt → Prints a file using the line printer daemon.
21. lpstat – lpstat -p → Shows the status of printers.
22. lpq – lpq → Displays the print queue.
23. lprm – lprm 2 → Removes job 2 from the print queue.
24. cancel – cancel jobID → Cancels a print job.
25. mail – mail user@example.com → Sends an email via the terminal.



```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ mail -s "Hello" atharv.lakhan@gmail.com
Cc: Hello, there
Sqoweh
^C

(Interrupt -- one more to kill letter)
^C
```

## CONCLUSION

So, here we saw some basic unix cmmands to start up the unix journey .

### LO2 mapped

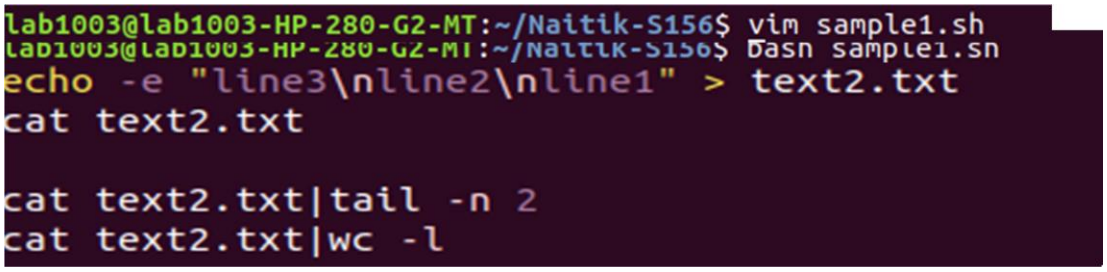
# 2. VI Editor

The **Vi editor** or **vim**(short for *Visual Editor*) is one of the oldest and most powerful text editors in Linux/Unix systems. It is lightweight, fast, and available by default on almost every Linux system.

**cat** command is used to display the contents of the file.
A "**.sh**" file is createdand edited in vi editor with the text being redirected to a text file "**text2.txt**".

To insert/edit in the file:

Press Esc -> i or Esc -> a



```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ vim sample1.sh
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ bash sample1.sh
echo -e "line3\nline2\nline1" > text2.txt
cat text2.txt

cat text2.txt|tail -n 2
cat text2.txt|wc -l
```

To save and exit the file:

Press Esc -> :wq.

To quit without saving:

Press Esc->:q!.

To go to the end of the file,

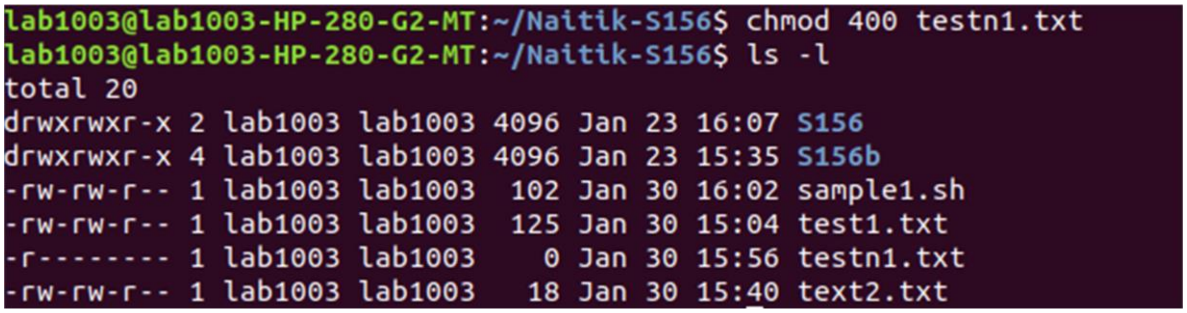
Press G

The **tail** command followed by a number indicates to the respective number of lines from the bottom.
The **wc** command displays the word count in the file.

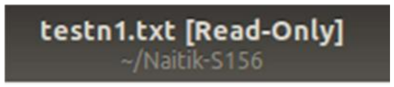
**wc –l** command is used for count of lines

Likely **wc –p** is used to count paragraphs

The **bash** command runs the ".sh file" which executes all the commands in it.
The **chmod** commands changes the access conditions of a file

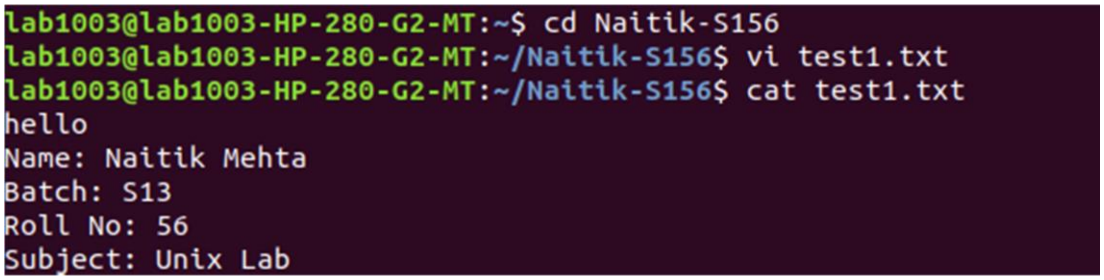


```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ chmod 400 testn1.txt
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ ls -l
total 20
drwxrwxr-x 2 lab1003 lab1003 4096 Jan 23 16:07 S156
drwxrwxr-x 4 lab1003 lab1003 4096 Jan 23 15:35 S156b
-rw-rw-r-- 1 lab1003 lab1003  102 Jan 30 16:02 sample1.sh
-rw-rw-r-- 1 lab1003 lab1003  125 Jan 30 15:04 test1.txt
-r-------- 1 lab1003 lab1003    0 Jan 30 15:56 testn1.txt
-rw-rw-r-- 1 lab1003 lab1003   18 Jan 30 15:40 text2.txt
```



testn1.txt [Read-Only]
~/Naitik-S156

b.



```
lab1003@lab1003-HP-280-G2-MT:~$ cd Naitik-S156
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ vi test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ cat test1.txt
hello
Name: Naitik Mehta
Batch: S13
Roll No: 56
Subject: Unix Lab
```

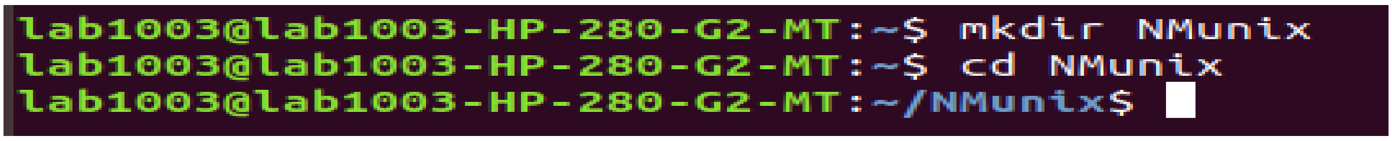
**CONCLUSION:** So, we saw the working of vi editor in Linux and how it is a powerful text editor.

Also, we studied changing file access modes through chmod
**LO2 mapped**

# 3. File Management Commands

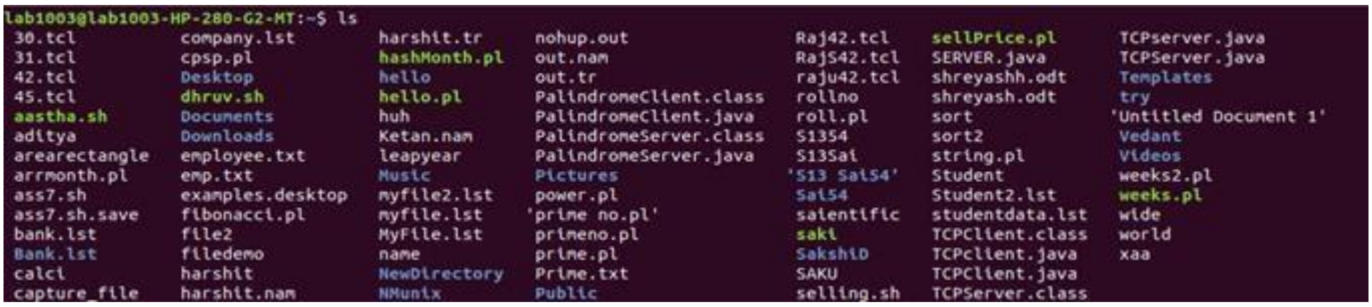
**Aim:** to implement and show file management command in linux.
**Theory and Output:**

**mkdir:** used to create a directory inside the current directory or any specified location.
**cd:** used to change directory for terminal



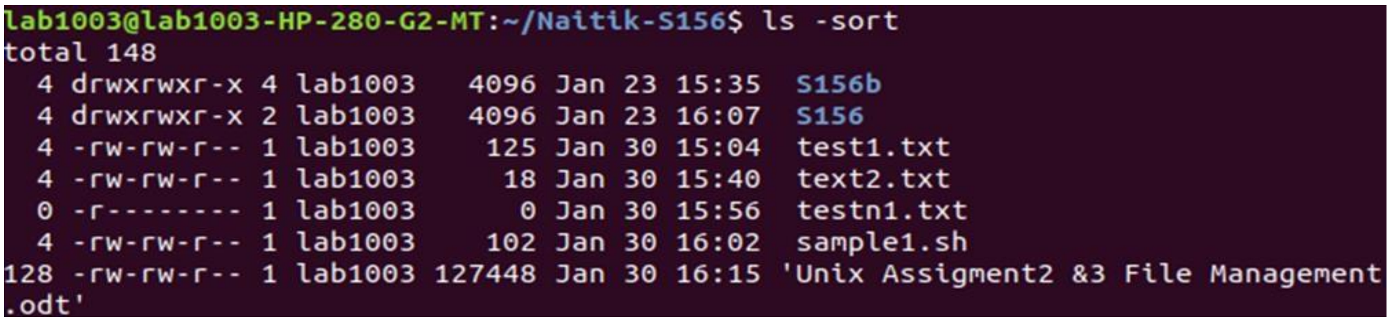
```
lab1003@lab1003-HP-280-G2-MT:~$ mkdir NMunix
lab1003@lab1003-HP-280-G2-MT:~$ cd NMunix
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ 
```

**ls:** the ls command in linux is used to list directory contents. it is one of the most commonly used commands for viewing the files and directories in your current working directory or in a specified location.



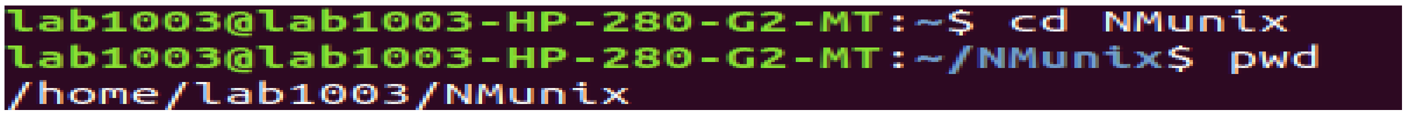
```
lab1003@lab1003-HP-280-G2-MT:~$ ls
 30.tcl          company.lst       harshit.tr      nohup.out                 Raj42.tcl     sellPrice.pl      TCPserver.java
 31.tcl          cpsp.pl           hashMonth.pl    out.nam                   RajS42.tcl    SERVER.java       TCPServer.java
 42.tcl          Desktop           hello           out.tr                    raju42.tcl    shreyashh.odt     Templates
 45.tcl          dhruv.sh          hello.pl        PalindromeClient.class    rollno        shreyash.odt      try
 aastha.sh       Documents         huh             PalindromeClient.java     roll.pl       sort             'Untitled Document 1'
 aditya          Downloads         Ketan.nam       PalindromeServer.class    S1354         sort2             Vedant
 arearectangle   employee.txt      leapyear        PalindromeServer.java     S13Sai        string.pl         Videos
 arrmonth.pl     emp.txt           Music           Pictures                 'S13 Sai54'    Student           weeks2.pl
 ass7.sh         examples.desktop  myfile2.lst     power.pl                  Sai54         Student2.lst      weeks.pl
 ass7.sh.save    fibonacci.pl      myfile.lst     'prime no.pl'              saientific    studentdata.lst   wide
 bank.lst        file2             MyFile.lst      primeno.pl                saki          TCPClient.class   world
 Bank.lst        filedemo          name            prime.pl                  SakshiD       TCPclient.java    xaa
 calci           harshit           NewDirectory    Prime.txt                 SAKU          TCPClient.java
 capture_file    harshit.nam       NMunix          Public                    selling.sh    TCPServer.class
```

**ls –sort:** Sort all the files in the directory as default by name. First subdirectories then name.



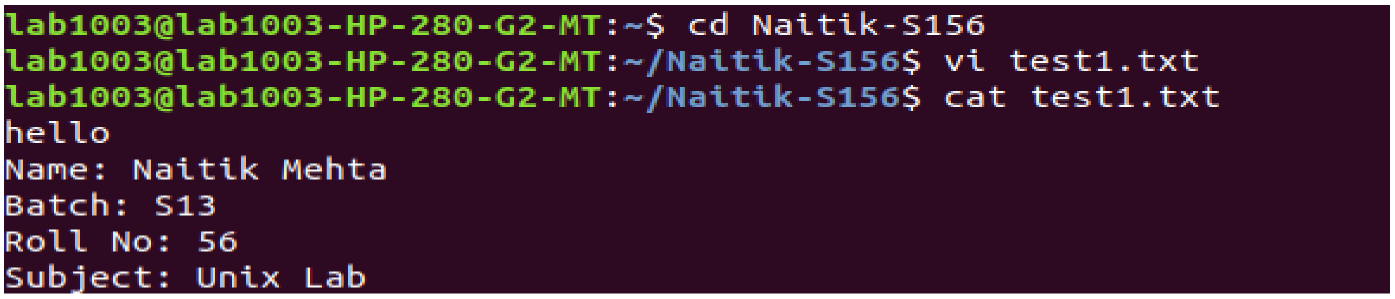
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ ls -sort
total 148
  4 drwxrwxr-x 4 lab1003   4096 Jan 23 15:35  S156b
  4 drwxrwxr-x 2 lab1003   4096 Jan 23 16:07  S156
  4 -rw-rw-r-- 1 lab1003    125 Jan 30 15:04  test1.txt
  4 -rw-rw-r-- 1 lab1003     18 Jan 30 15:40  text2.txt
  0 -r-------- 1 lab1003      0 Jan 30 15:56  testn1.txt
  4 -rw-rw-r-- 1 lab1003    102 Jan 30 16:02  sample1.sh
128 -rw-rw-r-- 1 lab1003 127448 Jan 30 16:15 'Unix Assigment2 &3 File Management
.odt'
```

**pwd:** the pwd command in linux stands for **"print working directory"**. it displays the full absolute path to the current directory you are in.

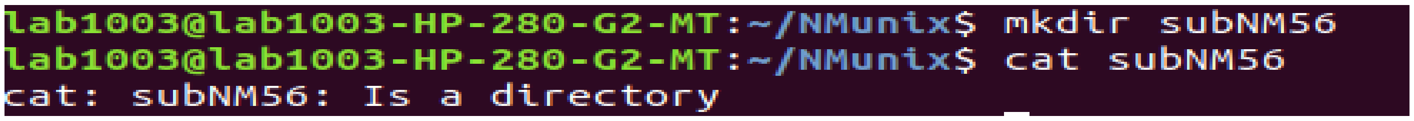


```
lab1003@lab1003-HP-280-G2-MT:~$ cd NMunix
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ pwd
/home/lab1003/NMunix
```

**cat:** the cat command in linux is primarily used to **display the contents of files**, but it can also be used for various other tasks like creating files, appending text, and concatenating multiple files.

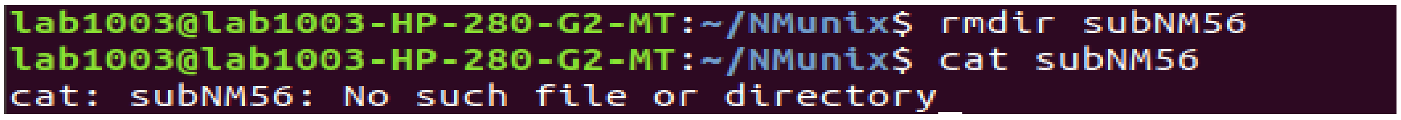


```
lab1003@lab1003-HP-280-G2-MT:~$ cd Naitik-S156
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ vi test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ cat test1.txt
hello
Name: Naitik Mehta
Batch: S13
Roll No: 56
Subject: Unix Lab
```



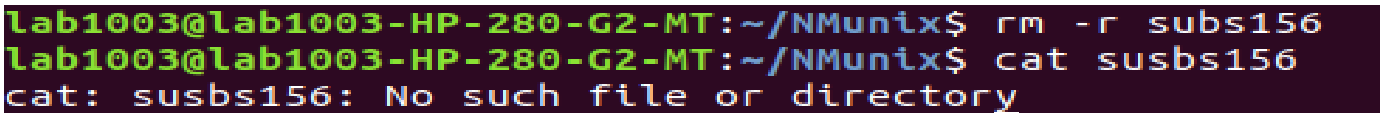
```
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ mkdir subNM56
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ cat subNM56
cat: subNM56: Is a directory
```

**rmdir:** used to **remove empty directories**. it is a simple and effective way to delete directories that do not contain any files or subdirectories.

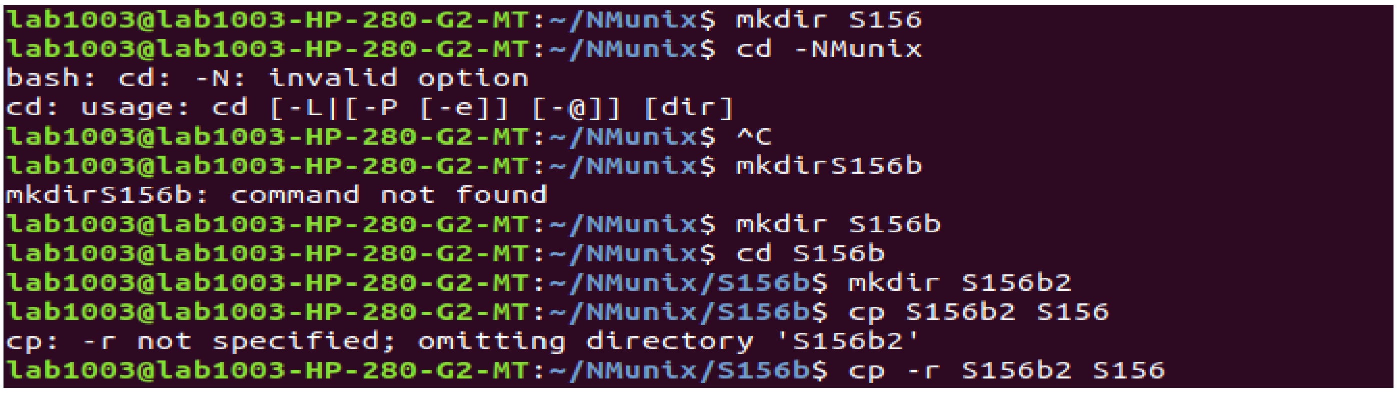


```
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ rmdir subNM56
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ cat subNM56
cat: subNM56: No such file or directory
```

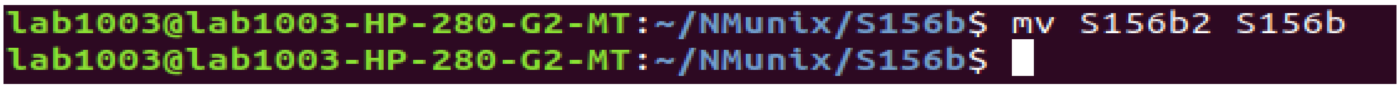
**rm:** used to **delete files and directories**. unlike rmdir, it just does not delete empty files or directories.


```
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ rm -r subs156
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ cat susbs156
cat: susbs156: No such file or directory
```

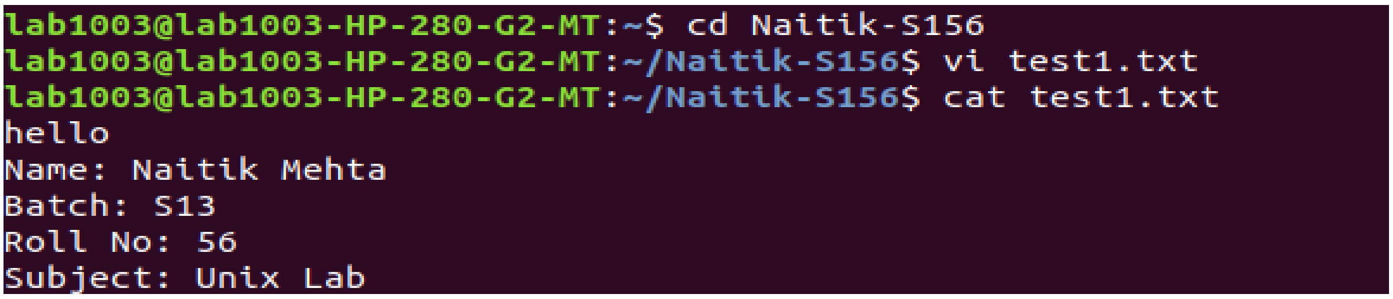
**cp:** used to copy files and directories from one location to another. it's one of the most common commands for managing files on the system.


```
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ mkdir S156
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ cd -NMunix
bash: cd: -N: invalid option
cd: usage: cd [-L|[-P [-e]] [-@]] [dir]
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ ^C
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ mkdirS156b
mkdirS156b: command not found
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ mkdir S156b
lab1003@lab1003-HP-280-G2-MT:~/NMunix$ cd S156b
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b$ mkdir S156b2
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b$ cp S156b2 S156
cp: -r not specified; omitting directory 'S156b2'
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b$ cp -r S156b2 S156
```

**mv:** used to **move or rename files and directories**.

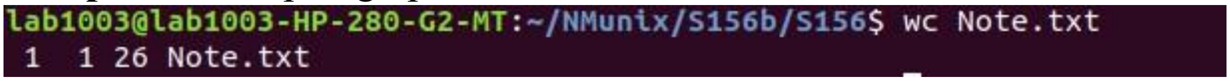

```
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b$ mv S156b2 S156b
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b$ 
```

**chmod:** change mode of files or directory


```
lab1003@lab1003-HP-280-G2-MT:~$ cd Naitik-S156
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ vi test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ cat test1.txt
hello
Name: Naitik Mehta
Batch: S13
Roll No: 56
Subject: Unix Lab
```

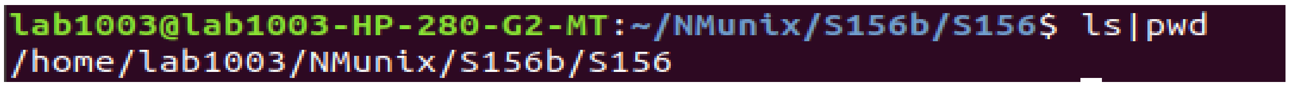
**wc:** used for word count of the file
**wc –l**: used for line count.

**wc –p**: used for paragraph count



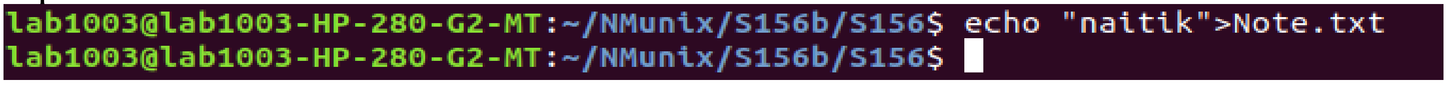
```
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b/S156$ wc Note.txt
 1  1 26 Note.txt
```

**piping:** allows you to **pass the output** of one command as the **input** to another command

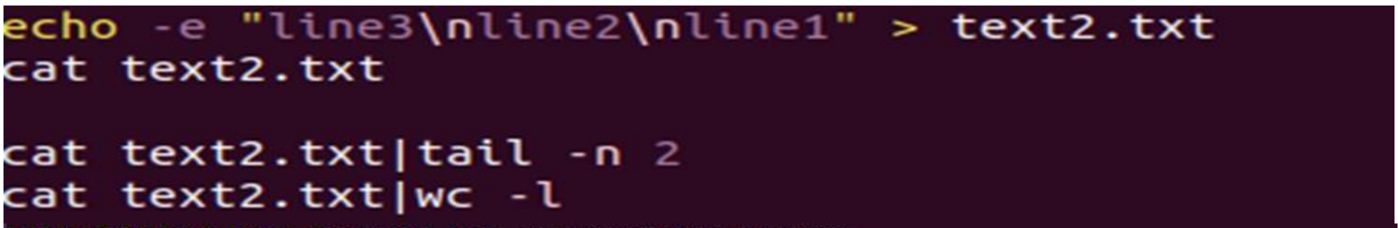


```
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b/S156$ ls|pwd
/home/lab1003/NMunix/S156b/S156
```

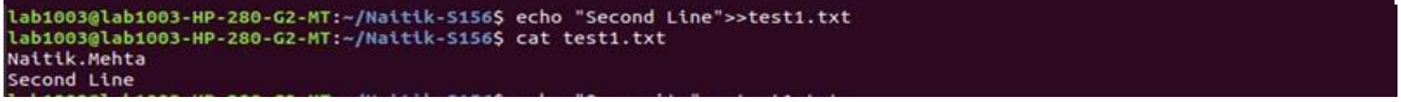
**redirection:** is a way to control where the output of a command goes and where input comes from.



```
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b/S156$ echo "naitik">Note.txt
lab1003@lab1003-HP-280-G2-MT:~/NMunix/S156b/S156$ 
```

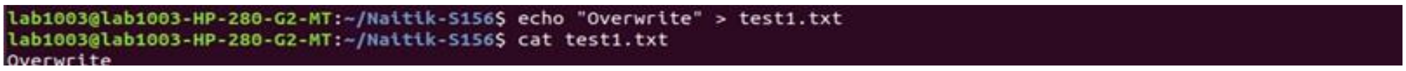


```
echo -e "line3\nline2\nline1" > text2.txt
cat text2.txt

cat text2.txt|tail -n 2
cat text2.txt|wc -l
```



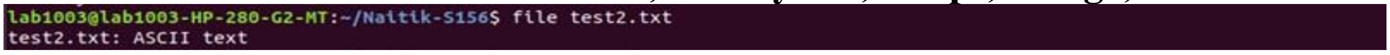
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ echo "Second Line">>test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ cat test1.txt
Naitik.Mehta
Second Line
```

**echo**: is used to display text or output to the terminal.



```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ echo "Overwrite" > test1.txt
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ cat test1.txt
Overwrite
```

**file : the file command is used to determine the type of a file. it analyzes the file and tells whether it is a text file, binary file, script, image, etc.**



```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-S156$ file test2.txt
test2.txt: ASCII text
```

**CONCLUSION:**

Studied File Management Commands

**LO2 mapped**

# 4. User Management Commands(3c.) - LO3

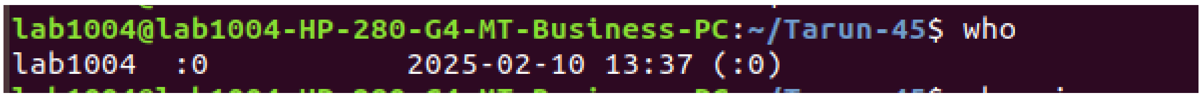
**AIM:** To implement user management commands in unix

**THEORY**: User management in Unix is basically about **creating**, **modifying**, **deleting**, and **controlling user accounts** and **groups** on a system.

- **Security**: Not everyone should have the power to delete system files (duh!).
- **Organization**: Managing access and resources for many users becomes easy.
- **Multi-user environment**: Unix is built for multiple users — so proper user management is **essential**.

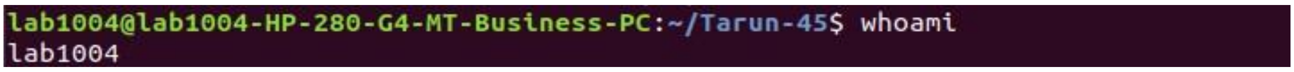
**COMMANDS & THEORY:**

## 1. who command:-



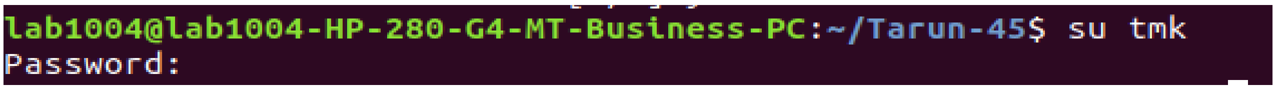
```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ who
lab1004  :0           2025-02-10 13:37 (:0)
```

## 2. whoami command:-



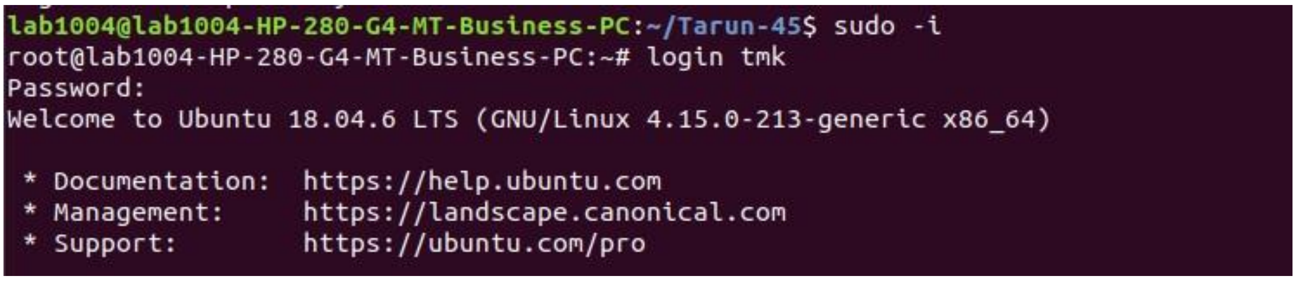
```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ whoami
lab1004
```

## 3. su command:-



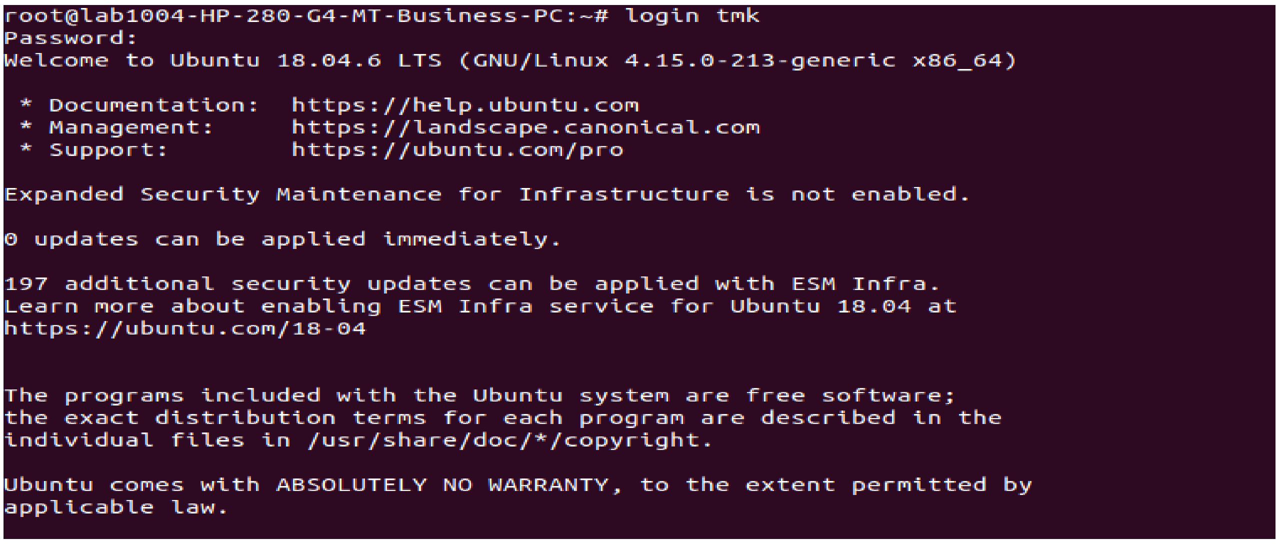
```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ su tmk
Password:
```

## 4. sudo command:-



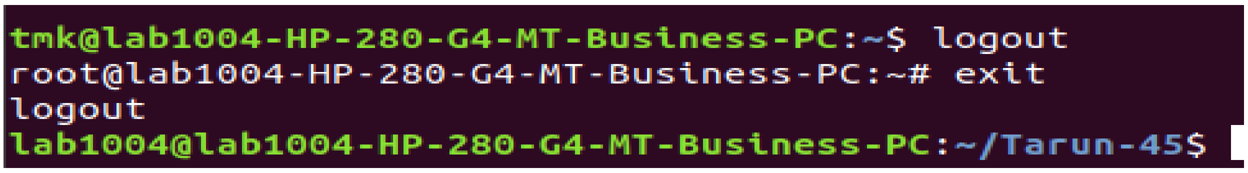
```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ sudo -i
root@lab1004-HP-280-G4-MT-Business-PC:~# login tmk
Password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-213-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro
```

## 5. login command:-



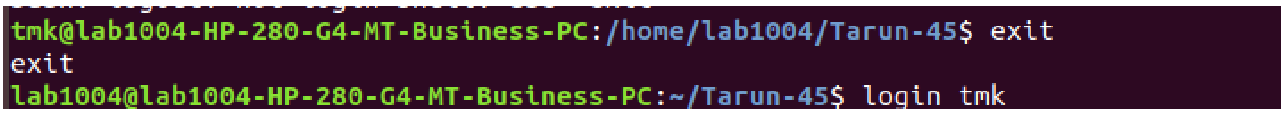
```
root@lab1004-HP-280-G4-MT-Business-PC:~# login tmk
Password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-213-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

197 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

## 6. logout command:-



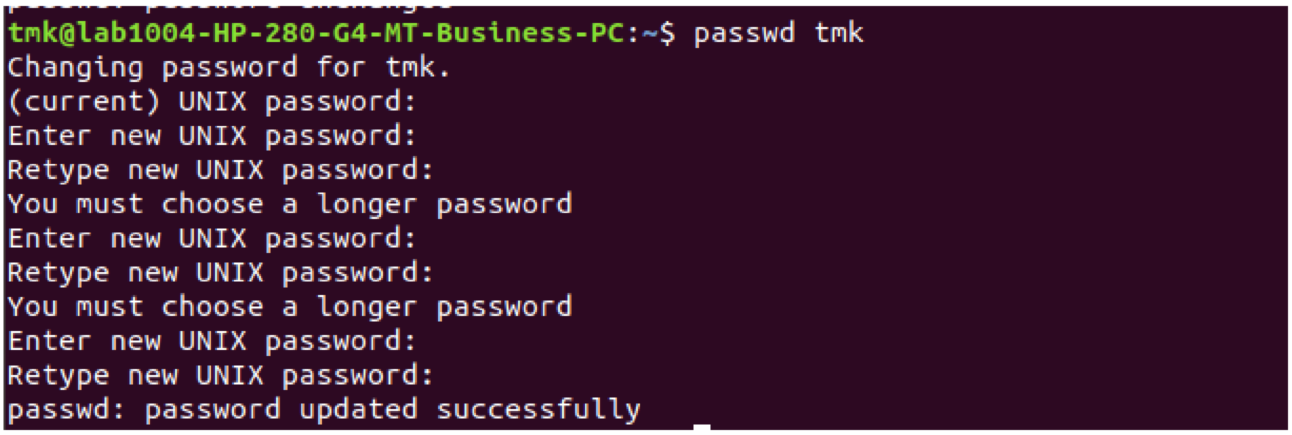
```
tmk@lab1004-HP-280-G4-MT-Business-PC:~$ logout
root@lab1004-HP-280-G4-MT-Business-PC:~# exit
logout
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$
```

## 7. exit command:-



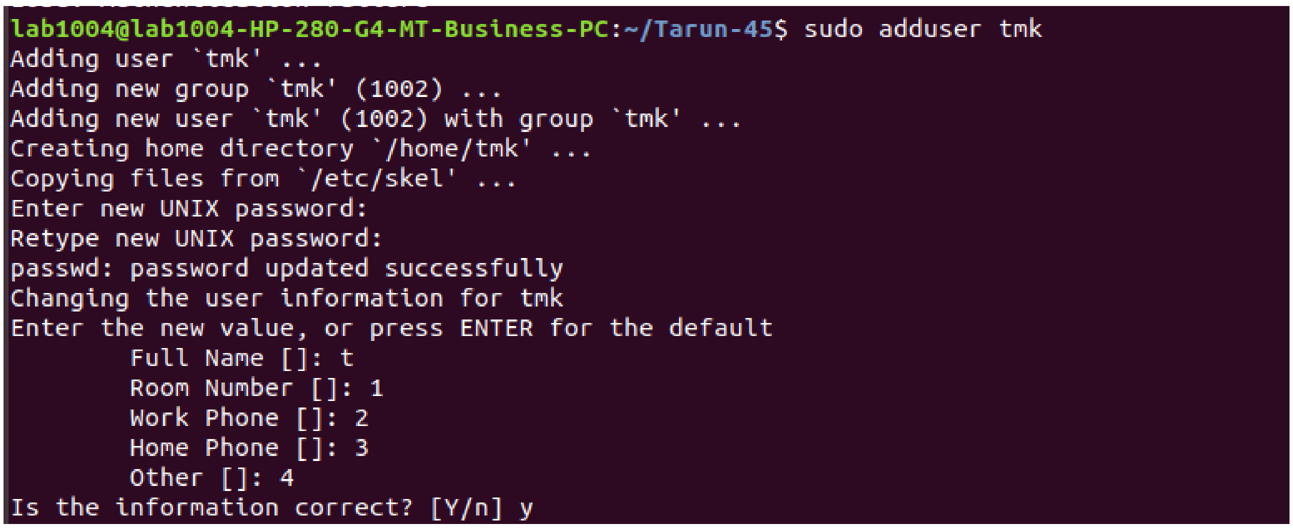
```
tmk@lab1004-HP-280-G4-MT-Business-PC:/home/lab1004/Tarun-45$ exit
exit
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ login tmk
```

## 8. passwd command:-



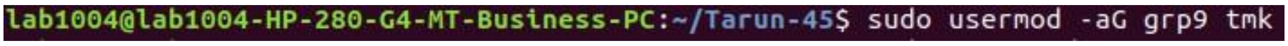
```
tmk@lab1004-HP-280-G4-MT-Business-PC:~$ passwd tmk
Changing password for tmk.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
You must choose a longer password
Enter new UNIX password:
Retype new UNIX password:
You must choose a longer password
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

## 9. adduser/useradd command:-



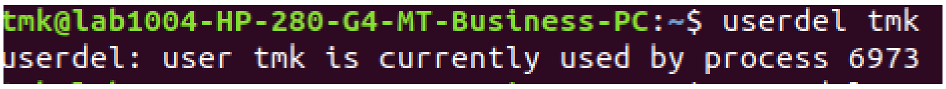
```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ sudo adduser tmk
Adding user `tmk' ...
Adding new group `tmk' (1002) ...
Adding new user `tmk' (1002) with group `tmk' ...
Creating home directory `/home/tmk' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for tmk
Enter the new value, or press ENTER for the default
        Full Name []: t
        Room Number []: 1
        Work Phone []: 2
        Home Phone []: 3
        Other []: 4
Is the information correct? [Y/n] y
```

## 10. usermod command:-



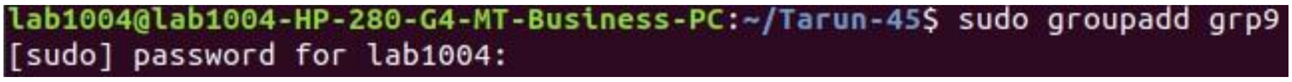
```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ sudo usermod -aG grp9 tmk
```

## 11. userdel command:-



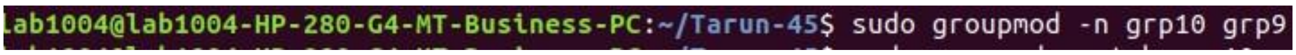
```
tmk@lab1004-HP-280-G4-MT-Business-PC:~$ userdel tmk
userdel: user tmk is currently used by process 6973
```

## 12. groupadd command:-



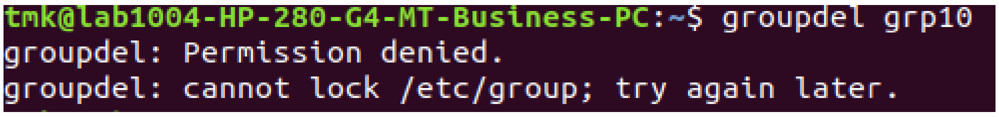
```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ sudo groupadd grp9
[sudo] password for lab1004:
```

## 13. groupmod command:-



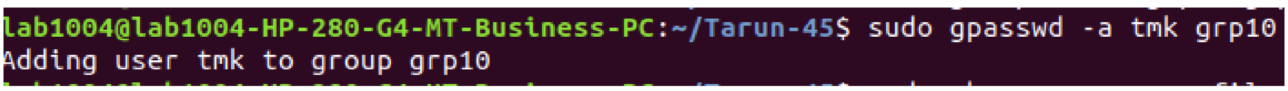
```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ sudo groupmod -n grp10 grp9
```

## 14. groupdel command:-



```
tmk@lab1004-HP-280-G4-MT-Business-PC:~$ groupdel grp10
groupdel: Permission denied.
groupdel: cannot lock /etc/group; try again later.
```

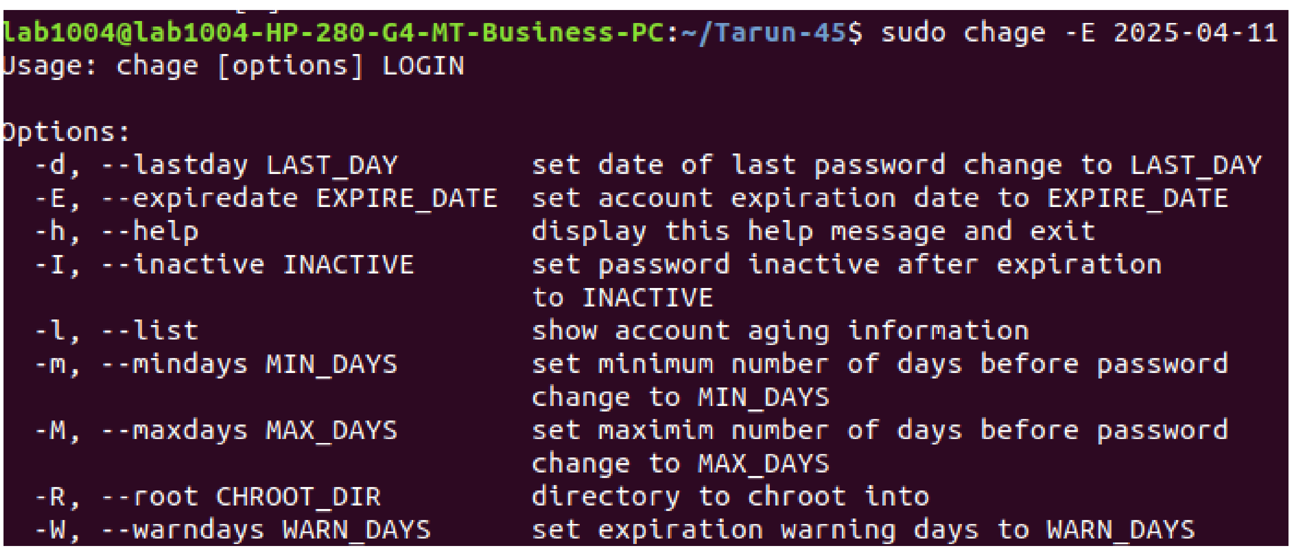
## 15. gpasswd command:-



```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ sudo gpasswd -a tmk grp10
Adding user tmk to group grp10
```

## 16. chown command:-

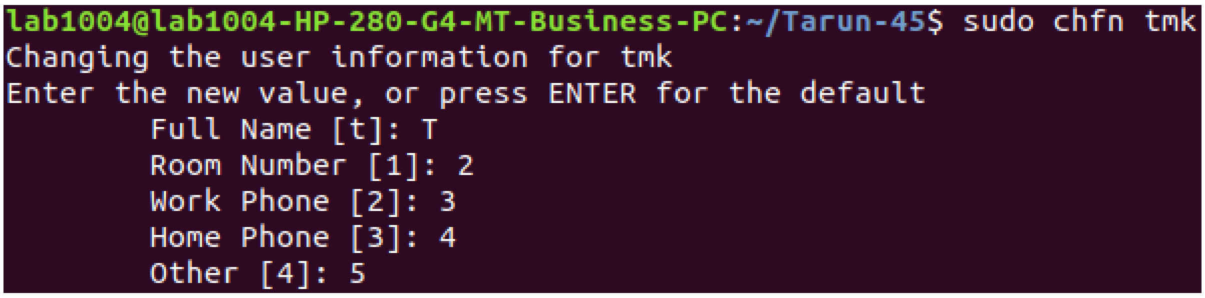
## 17. chage command:-



```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ sudo chage -E 2025-04-11
Usage: chage [options] LOGIN

Options:
  -d, --lastday LAST_DAY        set date of last password change to LAST_DAY
  -E, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
  -h, --help                    display this help message and exit
  -I, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -l, --list                    show account aging information
  -m, --mindays MIN_DAYS        set minimum number of days before password
                                change to MIN_DAYS
  -M, --maxdays MAX_DAYS        set maximim number of days before password
                                change to MAX_DAYS
  -R, --root CHROOT_DIR         directory to chroot into
  -W, --warndays WARN_DAYS      set expiration warning days to WARN_DAYS
```

## 18. chgrp command:-

## 19. chfn command:-



```
lab1004@lab1004-HP-280-G4-MT-Business-PC:~/Tarun-45$ sudo chfn tmk
Changing the user information for tmk
Enter the new value, or press ENTER for the default
        Full Name [t]: T
        Room Number [1]: 2
        Work Phone [2]: 3
        Home Phone [3]: 4
        Other [4]: 5
```

**CONCLUSION:**

- Everyone gets the access they need (and *only* that much).
- The system stays safe, organized, and efficient.
- Chaos is avoided — because giving root access to every user is like handing your house keys to every random person on the street.

**LO3 mapped**

# 5. Process Management Commands(4c) - LO4

**AIM:** To implement process management commands in unix

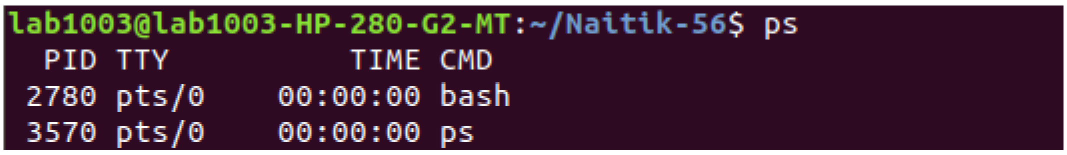
**THEORY**: In Unix-like operating systems, **process management** is crucial for controlling the execution of processes (programs that are being executed) on a system. A process can be anything from a running program to a system task. The operating system manages processes to ensure proper resource allocation, process scheduling, and handling of inputs/outputs.

Key concepts in process management include:

1. **Processes**: Every executing program is considered a process.
2. **Process IDs (PID)**: Each process is identified by a unique number called a Process ID (PID).
3. **Parent and Child Processes**: Processes can create other processes, called child processes.
4. **Process States**: Processes can be in different states like running, waiting, or stopped.

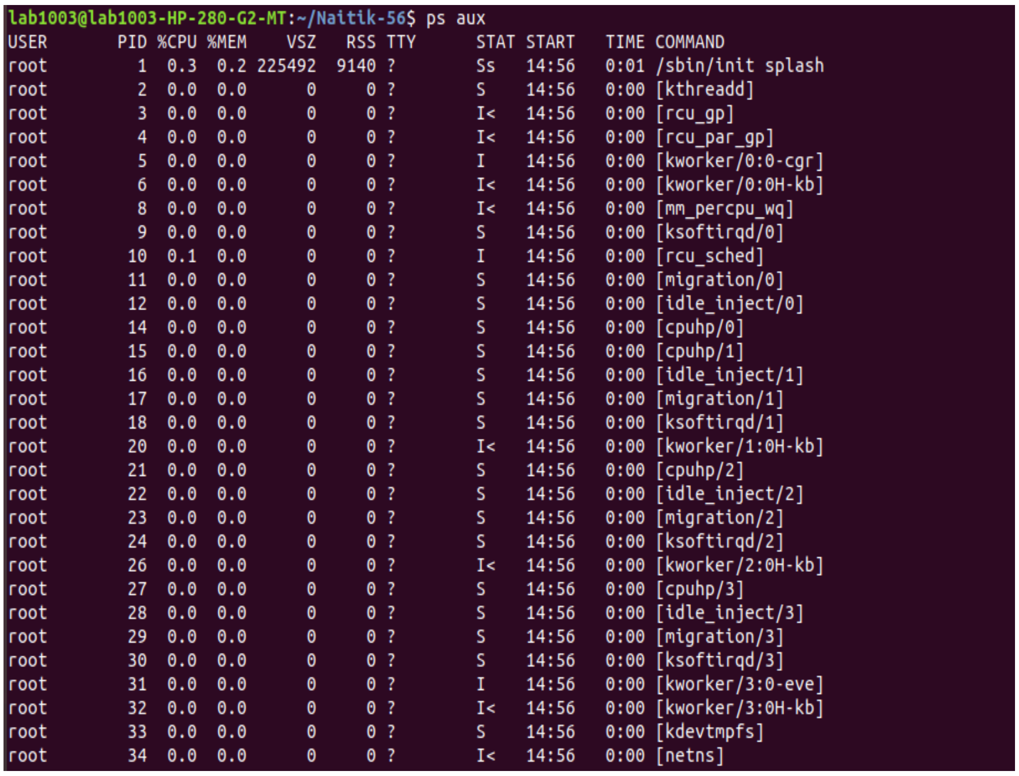
## COMMANDS:

**Ps:** Displays information about running processes.



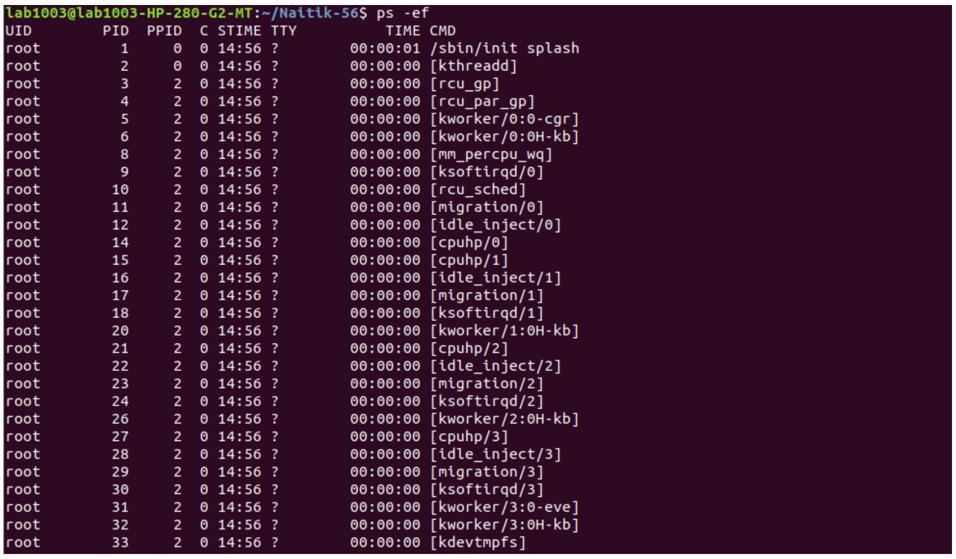
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ ps
  PID TTY          TIME CMD
 2780 pts/0    00:00:00 bash
 3570 pts/0    00:00:00 ps
```

**Ps aux:** Shows all processes with detailed information



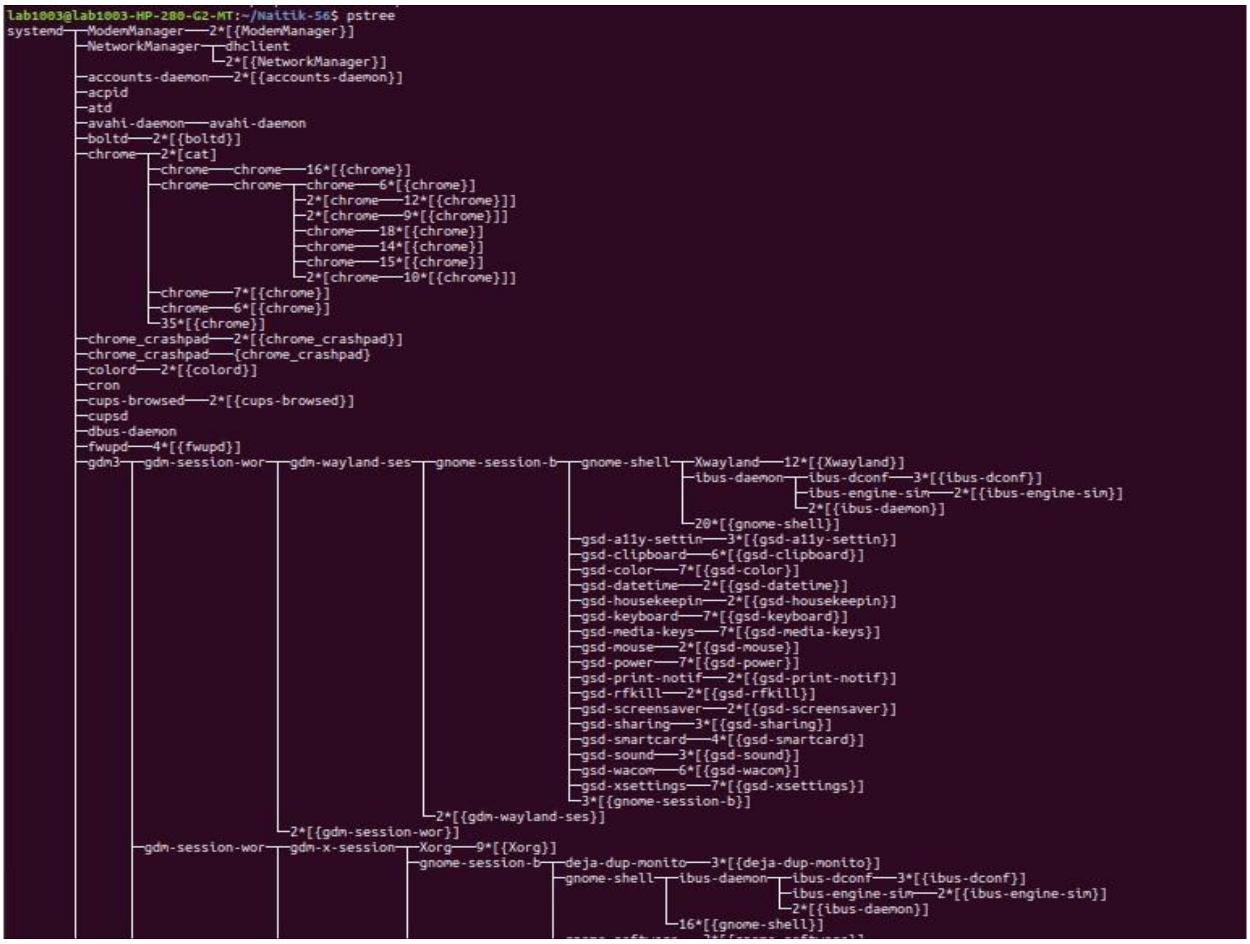
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.3  0.2 225492  9140 ?        Ss   14:56   0:01 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    14:56   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   14:56   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   14:56   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I    14:56   0:00 [kworker/0:0-cgr]
root           6  0.0  0.0      0     0 ?        I<   14:56   0:00 [kworker/0:0H-kb]
root           8  0.0  0.0      0     0 ?        I<   14:56   0:00 [mm_percpu_wq]
root           9  0.0  0.0      0     0 ?        S    14:56   0:00 [ksoftirqd/0]
root          10  0.1  0.0      0     0 ?        I    14:56   0:00 [rcu_sched]
root          11  0.0  0.0      0     0 ?        S    14:56   0:00 [migration/0]
root          12  0.0  0.0      0     0 ?        S    14:56   0:00 [idle_inject/0]
root          14  0.0  0.0      0     0 ?        S    14:56   0:00 [cpuhp/0]
root          15  0.0  0.0      0     0 ?        S    14:56   0:00 [cpuhp/1]
root          16  0.0  0.0      0     0 ?        S    14:56   0:00 [idle_inject/1]
root          17  0.0  0.0      0     0 ?        S    14:56   0:00 [migration/1]
root          18  0.0  0.0      0     0 ?        S    14:56   0:00 [ksoftirqd/1]
root          20  0.0  0.0      0     0 ?        I<   14:56   0:00 [kworker/1:0H-kb]
root          21  0.0  0.0      0     0 ?        S    14:56   0:00 [cpuhp/2]
root          22  0.0  0.0      0     0 ?        S    14:56   0:00 [idle_inject/2]
root          23  0.0  0.0      0     0 ?        S    14:56   0:00 [migration/2]
root          24  0.0  0.0      0     0 ?        S    14:56   0:00 [ksoftirqd/2]
root          26  0.0  0.0      0     0 ?        I<   14:56   0:00 [kworker/2:0H-kb]
root          27  0.0  0.0      0     0 ?        S    14:56   0:00 [cpuhp/3]
root          28  0.0  0.0      0     0 ?        S    14:56   0:00 [idle_inject/3]
root          29  0.0  0.0      0     0 ?        S    14:56   0:00 [migration/3]
root          30  0.0  0.0      0     0 ?        S    14:56   0:00 [ksoftirqd/3]
root          31  0.0  0.0      0     0 ?        I    14:56   0:00 [kworker/3:0-eve]
root          32  0.0  0.0      0     0 ?        I<   14:56   0:00 [kworker/3:0H-kb]
root          33  0.0  0.0      0     0 ?        S    14:56   0:00 [kdevtmpfs]
root          34  0.0  0.0      0     0 ?        I<   14:56   0:00 [netns]
```

**Ps -ef**: Another common format for listing processes



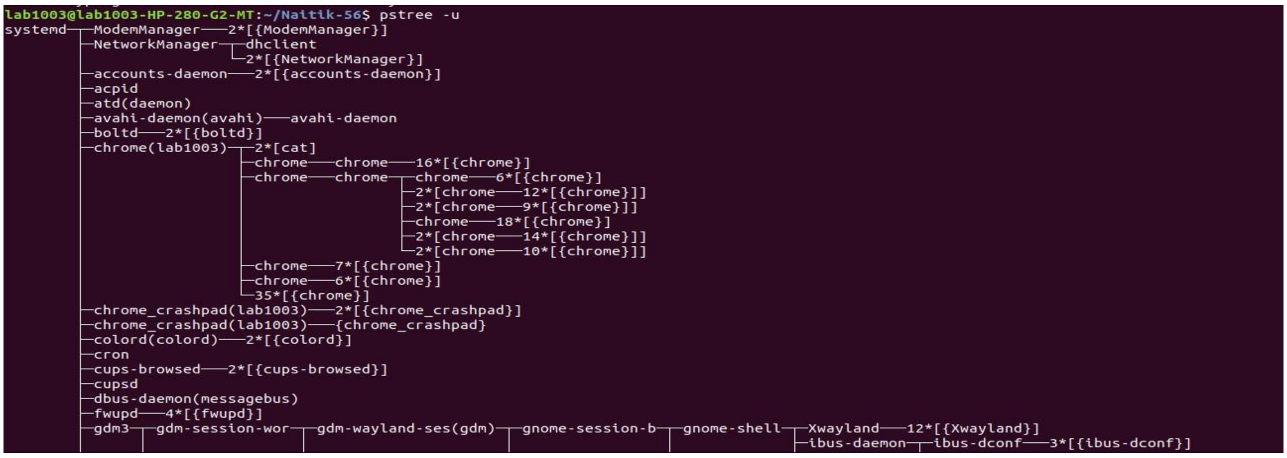
```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 14:56 ?        00:00:01 /sbin/init splash
root           2       0  0 14:56 ?        00:00:00 [kthreadd]
root           3       2  0 14:56 ?        00:00:00 [rcu_gp]
root           4       2  0 14:56 ?        00:00:00 [rcu_par_gp]
root           5       2  0 14:56 ?        00:00:00 [kworker/0:0-cgr]
root           6       2  0 14:56 ?        00:00:00 [kworker/0:0H-kb]
root           8       2  0 14:56 ?        00:00:00 [mm_percpu_wq]
root           9       2  0 14:56 ?        00:00:00 [ksoftirqd/0]
root          10       2  0 14:56 ?        00:00:00 [rcu_sched]
root          11       2  0 14:56 ?        00:00:00 [migration/0]
root          12       2  0 14:56 ?        00:00:00 [idle_inject/0]
root          14       2  0 14:56 ?        00:00:00 [cpuhp/0]
root          15       2  0 14:56 ?        00:00:00 [cpuhp/1]
root          16       2  0 14:56 ?        00:00:00 [idle_inject/1]
root          17       2  0 14:56 ?        00:00:00 [migration/1]
root          18       2  0 14:56 ?        00:00:00 [ksoftirqd/1]
root          20       2  0 14:56 ?        00:00:00 [kworker/1:0H-kb]
root          21       2  0 14:56 ?        00:00:00 [cpuhp/2]
root          22       2  0 14:56 ?        00:00:00 [idle_inject/2]
root          23       2  0 14:56 ?        00:00:00 [migration/2]
root          24       2  0 14:56 ?        00:00:00 [ksoftirqd/2]
root          26       2  0 14:56 ?        00:00:00 [kworker/2:0H-kb]
root          27       2  0 14:56 ?        00:00:00 [cpuhp/3]
root          28       2  0 14:56 ?        00:00:00 [idle_inject/3]
root          29       2  0 14:56 ?        00:00:00 [migration/3]
root          30       2  0 14:56 ?        00:00:00 [ksoftirqd/3]
root          31       2  0 14:56 ?        00:00:00 [kworker/3:0-eve]
root          32       2  0 14:56 ?        00:00:00 [kworker/3:0H-kb]
root          33       2  0 14:56 ?        00:00:00 [kdevtmpfs]
```

**Pstree:** Displays processes in a tree format, showing the parent-child relationships between processes.



```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager─┬─dhclient
        │                └─2*[{NetworkManager}]
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─acpid
        ├─atd
        ├─avahi-daemon───avahi-daemon
        ├─boltd───2*[{boltd}]
        ├─chrome─┬─2*[cat]
        │        ├─chrome───chrome───16*[{chrome}]
        │        ├─chrome───chrome─┬─chrome───6*[{chrome}]
        │        │                 ├─2*[chrome───12*[{chrome}]]
        │        │                 ├─2*[chrome───9*[{chrome}]]
        │        │                 ├─chrome───18*[{chrome}]
        │        │                 ├─chrome───14*[{chrome}]
        │        │                 ├─chrome───15*[{chrome}]
        │        │                 └─2*[chrome───10*[{chrome}]]
        │        ├─chrome───7*[{chrome}]
        │        ├─chrome───6*[{chrome}]
        │        └─35*[{chrome}]
        ├─chrome_crashpad───2*[{chrome_crashpad}]
        ├─chrome_crashpad───{chrome_crashpad}
        ├─colord───2*[{colord}]
        ├─cron
        ├─cups-browsed───2*[{cups-browsed}]
        ├─cupsd
        ├─dbus-daemon
        ├─fwupd───4*[{fwupd}]
        ├─gdm3─┬─gdm-session-wor─┬─gdm-wayland-ses─┬─gnome-session-b─┬─gnome-shell─┬─Xwayland───12*[{Xwayland}]
        │      │                 │                 │                 │             ├─ibus-daemon─┬─ibus-dconf───3*[{ibus-dconf}]
        │      │                 │                 │                 │             │             ├─ibus-engine-sim───2*[{ibus-engine-sim}]
        │      │                 │                 │                 │             │             └─2*[{ibus-daemon}]
        │      │                 │                 │                 │             └─20*[{gnome-shell}]
        │      │                 │                 │                 ├─gsd-a11y-settin───3*[{gsd-a11y-settin}]
        │      │                 │                 │                 ├─gsd-clipboard───6*[{gsd-clipboard}]
        │      │                 │                 │                 ├─gsd-color───7*[{gsd-color}]
        │      │                 │                 │                 ├─gsd-datetime───2*[{gsd-datetime}]
        │      │                 │                 │                 ├─gsd-housekeepin───2*[{gsd-housekeepin}]
        │      │                 │                 │                 ├─gsd-keyboard───7*[{gsd-keyboard}]
        │      │                 │                 │                 ├─gsd-media-keys───7*[{gsd-media-keys}]
        │      │                 │                 │                 ├─gsd-mouse───2*[{gsd-mouse}]
        │      │                 │                 │                 ├─gsd-power───7*[{gsd-power}]
        │      │                 │                 │                 ├─gsd-print-notif───2*[{gsd-print-notif}]
        │      │                 │                 │                 ├─gsd-rfkill───2*[{gsd-rfkill}]
        │      │                 │                 │                 ├─gsd-screensaver───2*[{gsd-screensaver}]
        │      │                 │                 │                 ├─gsd-sharing───3*[{gsd-sharing}]
        │      │                 │                 │                 ├─gsd-smartcard───4*[{gsd-smartcard}]
        │      │                 │                 │                 ├─gsd-sound───3*[{gsd-sound}]
        │      │                 │                 │                 ├─gsd-wacom───6*[{gsd-wacom}]
        │      │                 │                 │                 ├─gsd-xsettings───7*[{gsd-xsettings}]
        │      │                 │                 │                 └─3*[{gnome-session-b}]
        │      │                 │                 └─2*[{gdm-wayland-ses}]
        │      │                 └─2*[{gdm-session-wor}]
        │      ├─gdm-session-wor─┬─gdm-x-session─┬─Xorg───9*[{Xorg}]
        │      │                 │               ├─gnome-session-b─┬─deja-dup-monito───3*[{deja-dup-monito}]
        │      │                 │               │                 ├─gnome-shell─┬─ibus-daemon─┬─ibus-dconf───3*[{ibus-dconf}]
        │      │                 │               │                 │             │             ├─ibus-engine-sim───2*[{ibus-engine-sim}]
        │      │                 │               │                 │             │             └─2*[{ibus-daemon}]
        │      │                 │               │                 │             └─16*[{gnome-shell}]
```

**Pstree -u:** Shows the user processes for each command

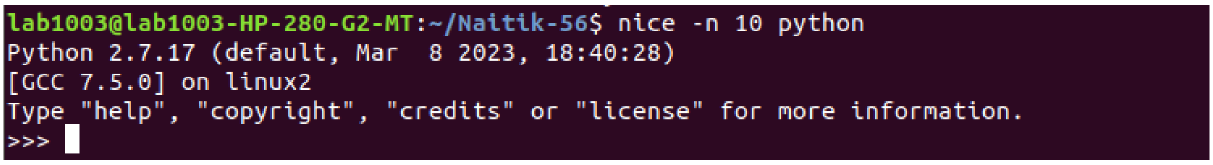


```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ pstree -u
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager─┬─dhclient
        │                └─2*[{NetworkManager}]
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─acpid
        ├─atd(daemon)
        ├─avahi-daemon(avahi)───avahi-daemon
        ├─boltd───2*[{boltd}]
        ├─chrome(lab1003)─┬─2*[cat]
        │                 ├─chrome───chrome───16*[{chrome}]
        │                 ├─chrome───chrome─┬─chrome───6*[{chrome}]
        │                 │                 ├─2*[chrome───12*[{chrome}]]
        │                 │                 ├─2*[chrome───9*[{chrome}]]
        │                 │                 ├─chrome───18*[{chrome}]
        │                 │                 ├─2*[chrome───14*[{chrome}]]
        │                 │                 └─2*[chrome───10*[{chrome}]]
        │                 ├─chrome───7*[{chrome}]
        │                 ├─chrome───6*[{chrome}]
        │                 └─35*[{chrome}]
        ├─chrome_crashpad(lab1003)───2*[{chrome_crashpad}]
        ├─chrome_crashpad(lab1003)───{chrome_crashpad}
        ├─colord(colord)───2*[{colord}]
        ├─cron
        ├─cups-browsed───2*[{cups-browsed}]
        ├─cupsd
        ├─dbus-daemon(messagebus)
        ├─fwupd───4*[{fwupd}]
        ├─gdm3─┬─gdm-session-wor─┬─gdm-wayland-ses(gdm)─┬─gnome-session-b─┬─gnome-shell─┬─Xwayland───12*[{Xwayland}]
        │      │                 │                      │                 │             ├─ibus-daemon─┬─ibus-dconf───3*[{ibus-dconf}]
```

# 3.

**Nice**: starts a process with a specified scheduling priority (nice value).

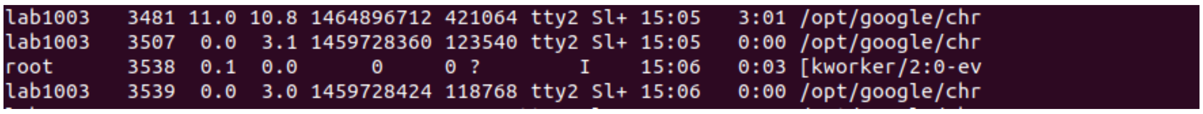
**nice -n 10 program :**Starts 'program' with a lower priority



```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ nice -n 10 python
Python 2.7.17 (default, Mar  8 2023, 18:40:28)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

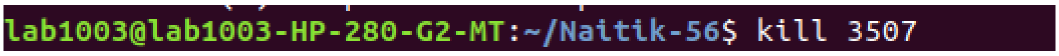
## 4. kill

**Description:** Sends a signal to a process, usually to terminate it. By default, it sends the SIGTERM signal.

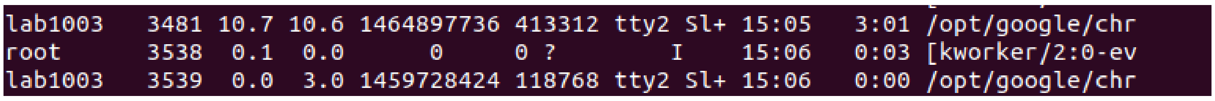


```
lab1003   3481 11.0 10.8 1464896712 421064 tty2 Sl+ 15:05   3:01 /opt/google/chr
lab1003   3507  0.0  3.1 1459728360 123540 tty2 Sl+ 15:05   0:00 /opt/google/chr
root      3538  0.1  0.0          0      0 ?       I    15:06   0:03 [kworker/2:0-ev
lab1003   3539  0.0  3.0 1459728424 118768 tty2 Sl+ 15:06   0:00 /opt/google/chr
```

Kill 3507



```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ kill 3507
```

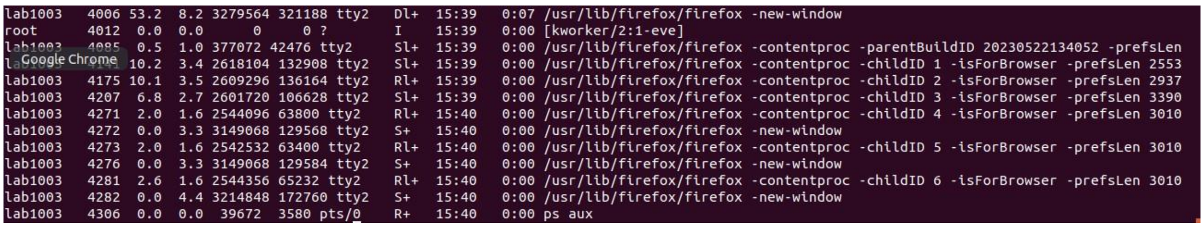


```
lab1003   3481 10.7 10.6 1464897736 413312 tty2 Sl+ 15:05   3:01 /opt/google/chr
root      3538  0.1  0.0          0      0 ?       I    15:06   0:03 [kworker/2:0-ev
lab1003   3539  0.0  3.0 1459728424 118768 tty2 Sl+ 15:06   0:00 /opt/google/chr
```

## 5. pkill

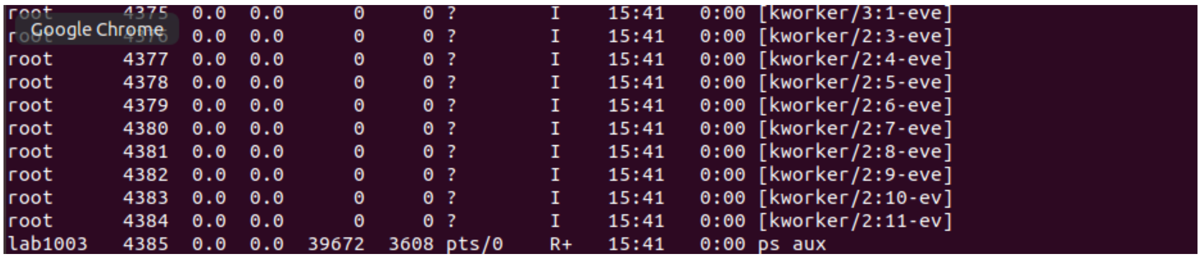
**Description:** Kills processes by name instead of PID.

pkill firefox  : Kills all processes named 'firefox'



```
lab1003   4006 53.2  8.2 3279564 321188 tty2   Dl+  15:39   0:07 /usr/lib/firefox/firefox -new-window
root      4012  0.0  0.0       0      0 ?      I    15:39   0:00 [kworker/2:1-eve]
lab1003   4085  0.5  1.0 377072 42476 tty2     Sl+  15:39   0:00 /usr/lib/firefox/firefox -contentproc -parentBuildID 20230522134052 -prefsLen
lab1003   4141 10.2  3.4 2618104 132908 tty2   Sl+  15:39   0:00 /usr/lib/firefox/firefox -contentproc -childID 1 -isForBrowser -prefsLen 2553
lab1003   4175 10.1  3.5 2609296 136164 tty2   Rl+  15:39   0:00 /usr/lib/firefox/firefox -contentproc -childID 2 -isForBrowser -prefsLen 2937
lab1003   4207  6.8  2.7 2601720 106628 tty2   Sl+  15:39   0:00 /usr/lib/firefox/firefox -contentproc -childID 3 -isForBrowser -prefsLen 3390
lab1003   4271  2.0  1.6 2544096 63800 tty2    Rl+  15:40   0:00 /usr/lib/firefox/firefox -contentproc -childID 4 -isForBrowser -prefsLen 3010
lab1003   4272  0.0  3.3 3149068 129568 tty2   S+   15:40   0:00 /usr/lib/firefox/firefox -new-window
lab1003   4273  2.0  1.6 2542532 63400 tty2    Rl+  15:40   0:00 /usr/lib/firefox/firefox -contentproc -childID 5 -isForBrowser -prefsLen 3010
lab1003   4276  0.0  3.3 3149068 129584 tty2   S+   15:40   0:00 /usr/lib/firefox/firefox -new-window
lab1003   4281  2.6  1.6 2544356 65232 tty2    Rl+  15:40   0:00 /usr/lib/firefox/firefox -contentproc -childID 6 -isForBrowser -prefsLen 3010
lab1003   4282  0.0  4.4 3214848 172760 tty2   S+   15:40   0:00 /usr/lib/firefox/firefox -new-window
lab1003   4306  0.0  0.0  39672  3580 pts/0    R+   15:40   0:00 ps aux
```

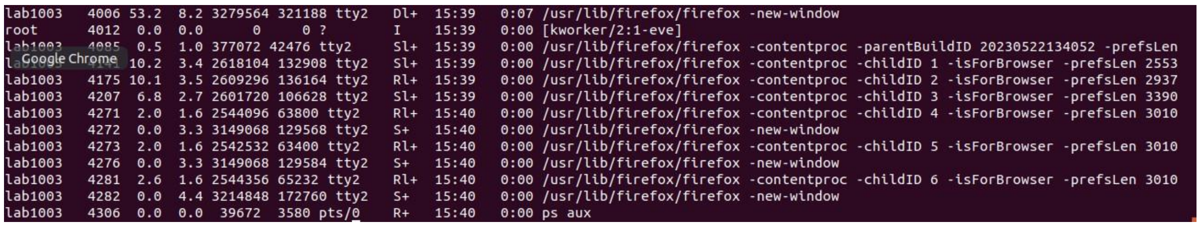
After pkill firefox



```
root      4375  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/3:1-eve]
root      4376  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:3-eve]
root      4377  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:4-eve]
root      4378  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:5-eve]
root      4379  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:6-eve]
root      4380  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:7-eve]
root      4381  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:8-eve]
root      4382  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:9-eve]
root      4383  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:10-ev]
root      4384  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:11-ev]
lab1003   4385  0.0  0.0  39672  3608 pts/0    R+   15:41   0:00 ps aux
```

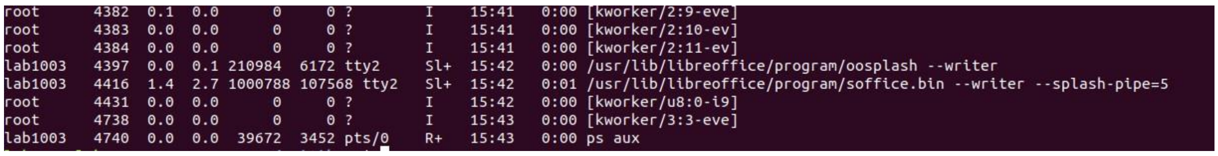
## 6. killall

**Description:** Terminates all processes with the specified name.



```
lab1003   4006 53.2  8.2 3279564 321188 tty2   Dl+  15:39   0:07 /usr/lib/firefox/firefox -new-window
root      4012  0.0  0.0      0      0 ?       I    15:39   0:00 [kworker/2:1-eve]
lab1003   4085  0.5  1.0 377072 42476 tty2     Sl+  15:39   0:00 /usr/lib/firefox/firefox -contentproc -parentBuildID 20230522134052 -prefsLen
lab1003   4141 10.2  3.4 2618104 132908 tty2   Sl+  15:39   0:00 /usr/lib/firefox/firefox -contentproc -childID 1 -isForBrowser -prefsLen 2553
lab1003   4175 10.1  3.5 2609296 136164 tty2   Rl+  15:39   0:00 /usr/lib/firefox/firefox -contentproc -childID 2 -isForBrowser -prefsLen 2937
lab1003   4207  6.8  2.7 2601720 106628 tty2   Sl+  15:39   0:00 /usr/lib/firefox/firefox -contentproc -childID 3 -isForBrowser -prefsLen 3390
lab1003   4271  2.0  1.6 2544096 63800 tty2    Rl+  15:40   0:00 /usr/lib/firefox/firefox -contentproc -childID 4 -isForBrowser -prefsLen 3010
lab1003   4272  0.0  3.3 3149068 129568 tty2   S+   15:40   0:00 /usr/lib/firefox/firefox -new-window
lab1003   4273  2.0  1.6 2542532 63400 tty2    Rl+  15:40   0:00 /usr/lib/firefox/firefox -contentproc -childID 5 -isForBrowser -prefsLen 3010
lab1003   4276  0.0  3.3 3149068 129584 tty2   S+   15:40   0:00 /usr/lib/firefox/firefox -new-window
lab1003   4281  2.6  1.6 2544356 65232 tty2    Rl+  15:40   0:00 /usr/lib/firefox/firefox -contentproc -childID 6 -isForBrowser -prefsLen 3010
lab1003   4282  0.0  4.4 3214848 172760 tty2   S+   15:40   0:00 /usr/lib/firefox/firefox -new-window
lab1003   4306  0.0  0.0  39672  3580 pts/0    R+   15:40   0:00 ps aux
```

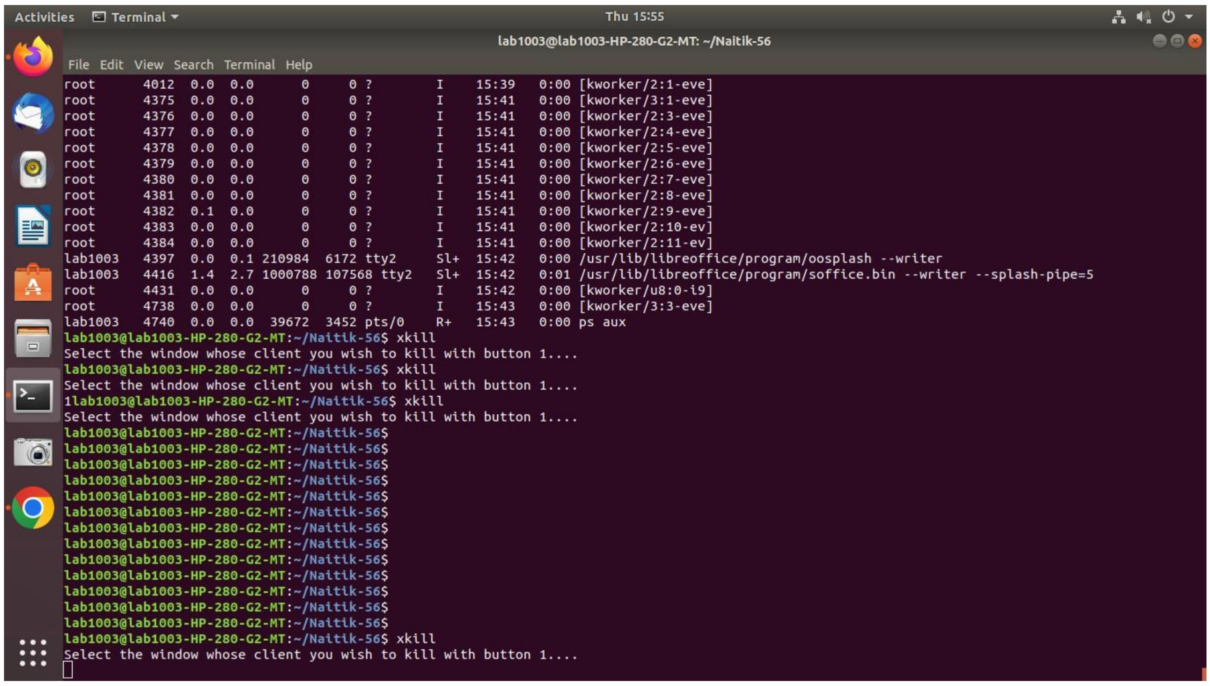
After killall firefox



```
root      4382  0.1  0.0      0     0 ?        I    15:41   0:00 [kworker/2:9-eve]
root      4383  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:10-ev]
root      4384  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:11-ev]
lab1003   4397  0.0  0.1 210984  6172 tty2     Sl+  15:42   0:00 /usr/lib/libreoffice/program/oosplash --writer
lab1003   4416  1.4  2.7 1000788 107568 tty2   Sl+  15:42   0:01 /usr/lib/libreoffice/program/soffice.bin --writer --splash-pipe=5
root      4431  0.0  0.0      0     0 ?        I    15:42   0:00 [kworker/u8:0-i9]
root      4738  0.0  0.0      0     0 ?        I    15:43   0:00 [kworker/3:3-eve]
lab1003   4740  0.0  0.0  39672  3452 pts/0    R+   15:43   0:00 ps aux
```

## 7. xkill

**Description:** Allows the user to click on a window to kill the associated process.

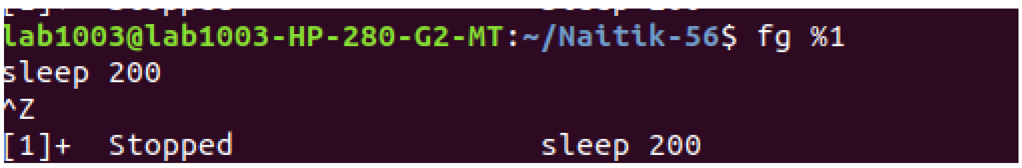


Activities
Terminal
Thu 15:55
lab1003@lab1003-HP-280-G2-MT: ~/Naitik-56
File Edit View Search Terminal Help

```
root      4012  0.0  0.0      0     0 ?        I    15:39   0:00 [kworker/2:1-eve]
root      4375  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/3:1-eve]
root      4376  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:3-eve]
root      4377  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:4-eve]
root      4378  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:5-eve]
root      4379  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:6-eve]
root      4380  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:7-eve]
root      4381  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:8-eve]
root      4382  0.1  0.0      0     0 ?        I    15:41   0:00 [kworker/2:9-eve]
root      4383  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:10-ev]
root      4384  0.0  0.0      0     0 ?        I    15:41   0:00 [kworker/2:11-ev]
lab1003   4397  0.0  0.1 210984  6172 tty2     Sl+  15:42   0:00 /usr/lib/libreoffice/program/oosplash --writer
lab1003   4416  1.4  2.7 1000788 107568 tty2   Sl+  15:42   0:01 /usr/lib/libreoffice/program/soffice.bin --writer --splash-pipe=5
root      4431  0.0  0.0      0     0 ?        I    15:42   0:00 [kworker/u8:0-i9]
root      4738  0.0  0.0      0     0 ?        I    15:43   0:00 [kworker/3:3-eve]
lab1003   4740  0.0  0.0  39672  3452 pts/0    R+   15:43   0:00 ps aux
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ xkill
Select the window whose client you wish to kill with button 1....
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ xkill
Select the window whose client you wish to kill with button 1....
1lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ xkill
Select the window whose client you wish to kill with button 1....
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ xkill
Select the window whose client you wish to kill with button 1....
```

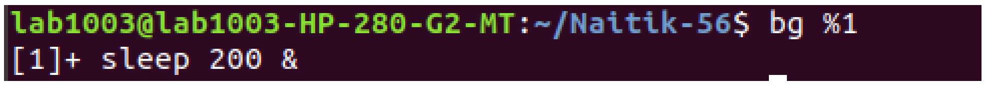
## 8. fg

**Description:** Brings a background process to the foreground.


```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ fg %1
sleep 200
^Z
[1]+  Stopped                 sleep 200
```

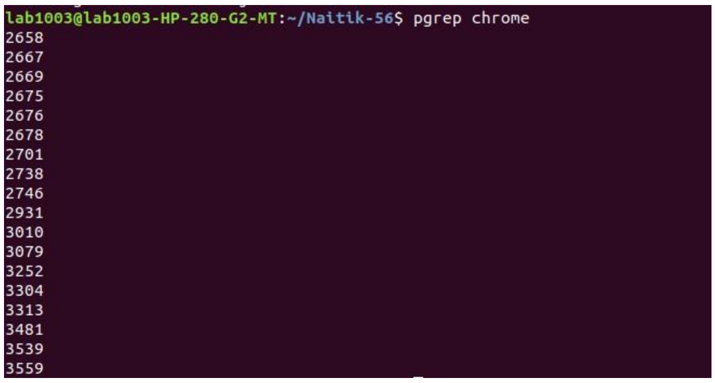
### G. bg

**Description:** Resumes a paused job in the background.


```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ bg %1
[1]+ sleep 200 &
```

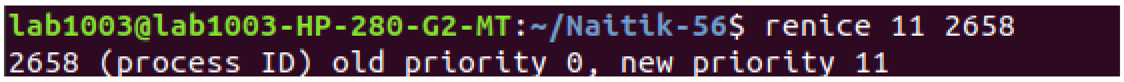
## 10. pgrep

**Description:** Searches for processes based on name and other attributes, and outputs their PID.


```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ pgrep chrome
2658
2667
2669
2675
2676
2678
2701
2738
2746
2931
3010
3079
3252
3304
3313
3481
3539
3559
```

## 11. renice

**Description:** Changes the priority (nice value) of running processes.


```
lab1003@lab1003-HP-280-G2-MT:~/Naitik-56$ renice 11 2658
2658 (process ID) old priority 0, new priority 11
```

**CONCLUSION :** Thus, we saw all process management commands in unix helpful for handling, indexing, killing, scheduling, etc. Of processes

## LO4 mapped

# UL 6. GREP, AWK and SED commands

**AIM:** To implement grep, awk and sed commands on a given database.

**THEORY:**

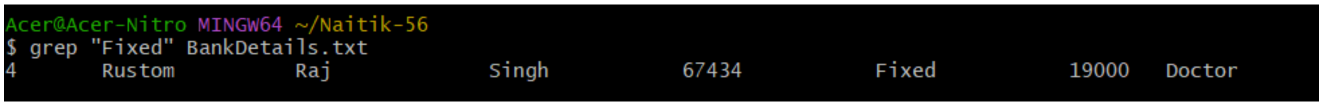
Bank Details of following people are given. We will do the required data manipulation and handling using grep, awk and ed commands

| ID | First Name | Middle Name | LastName | Account No. | Account type | Balance | Occupation |
|---|---|---|---|---|---|---|---|
| 1 | Anish | Shah | Rao | 12345 | Current | 100000 | Engineer |
| 2 | Harsh | Akash | Shah | 54321 | Current | 51000 | Doctor |
| 3 | Nadeem | Jerry | Shah | 23541 | Savings | 32690 | Teacher |
| 4 | Rustom | Raj | Singh | 67434 | Fixed | 19000 | Doctor |
| 5 | Shah | Pratap | Rathod | 62222 | Savings | 48790 | Baker |

**GREP :** Global Regular Expression Print

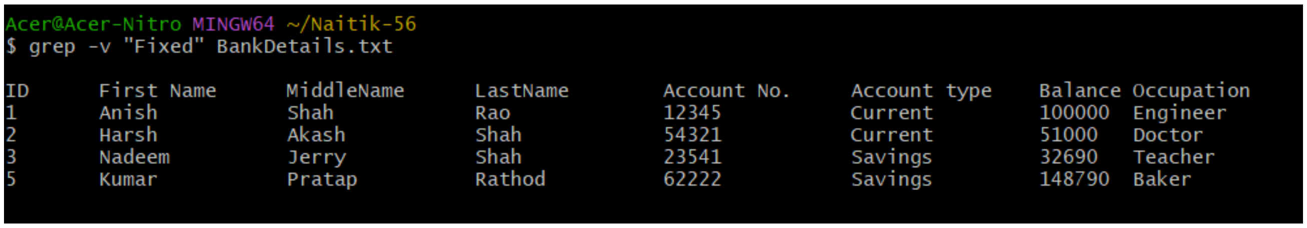
- Used to **search and filter** lines in a file based on patterns or regular expressions.
- It prints **only the matching lines**, making it great for finding keywords or values in logs or data files.

1. i) Display all the records of fixed account.



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ grep "Fixed" BankDetails.txt
4       Rustom          Raj             Singh           67434           Fixed           19000   Doctor
```

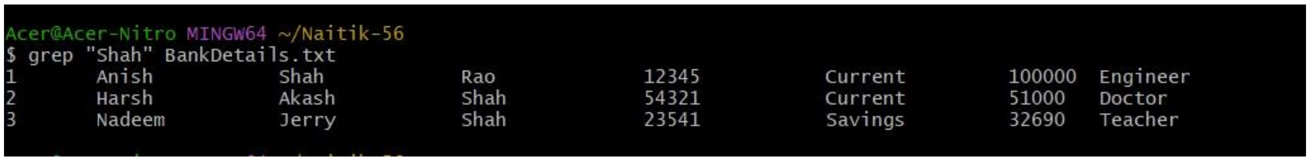
1. ii) Display all the records without fixed account.



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ grep -v "Fixed" BankDetails.txt

ID      First Name      MiddleName      LastName        Account No.     Account type    Balance Occupation
1       Anish           Shah            Rao             12345           Current         100000  Engineer
2       Harsh           Akash           Shah            54321           Current         51000   Doctor
3       Nadeem          Jerry           Shah            23541           Savings         32690   Teacher
5       Kumar           Pratap          Rathod          62222           Savings         148790  Baker
```

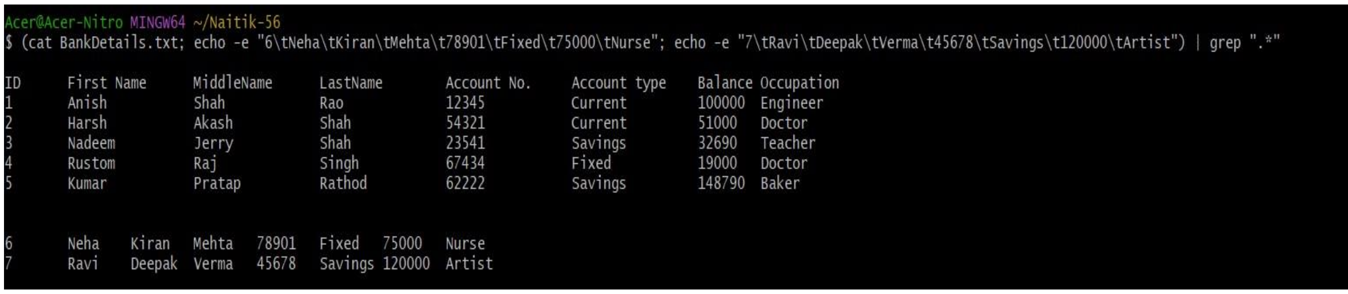
2. Display records with Shah in it

Grep, awk, sed



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ grep "Shah" BankDetails.txt
1       Anish           Shah            Rao             12345           Current         100000  Engineer
2       Harsh           Akash           Shah            54321           Current         51000   Doctor
3       Nadeem          Jerry           Shah            23541           Savings         32690   Teacher
```

3. Append 2 records and display the whole table

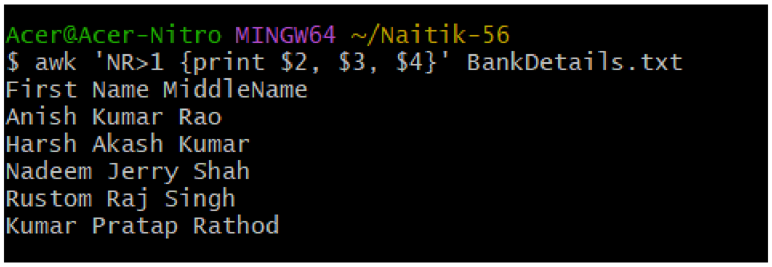


```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ (cat BankDetails.txt; echo -e "6\tNeha\tKiran\tMehta\t78901\tFixed\t75000\tNurse"; echo -e "7\tRavi\tDeepak\tVerma\t45678\tSavings\t120000\tArtist") | grep ".*"

ID      First Name      MiddleName      LastName        Account No.     Account type    Balance Occupation
1       Anish           Shah            Rao             12345           Current         100000  Engineer
2       Harsh           Akash           Shah            54321           Current         51000   Doctor
3       Nadeem          Jerry           Shah            23541           Savings         32690   Teacher
4       Rustom          Raj             Singh           67434           Fixed           19000   Doctor
5       Kumar           Pratap          Rathod          62222           Savings         148790  Baker


6       Neha    Kiran   Mehta   78901   Fixed   75000   Nurse
7       Ravi    Deepak  Verma   45678   Savings 120000  Artist
```

**AWK:** Aho, Weinberger, and Kernighan

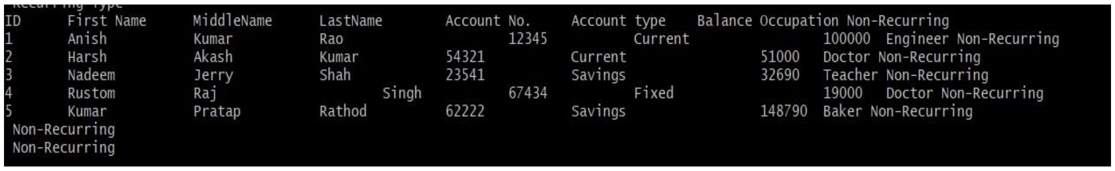
- A powerful text-processing tool for **pattern scanning, data extraction**, and **report generation.**
- Works by dividing lines into **fields** and allows **column-wise operations**, perfect for structured data like CSV or tabular logs.

1. Display full name of account holders



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ awk 'NR>1 {print $2, $3, $4}' BankDetails.txt
First Name MiddleName
Anish Kumar Rao
Harsh Akash Kumar
Nadeem Jerry Shah
Rustom Raj Singh
Kumar Pratap Rathod
```

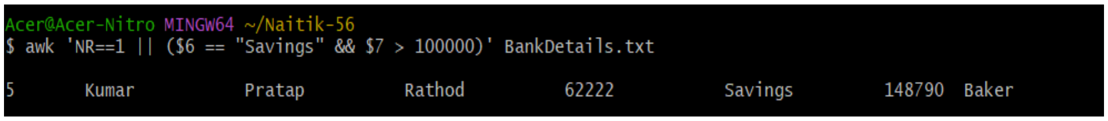
2. Display recurring accounts



```
ID      First Name      MiddleName      LastName        Account No.     Account type    Balance Occupation Non-Recurring
1       Anish           Kumar           Rao                     12345           Current                 100000  Engineer Non-Recurring
2       Harsh           Akash           Kumar           54321           Current                 51000   Doctor Non-Recurring
3       Nadeem          Jerry           Shah            23541           Savings                 32690   Teacher Non-Recurring
4       Rustom          Raj                     Singh           67434           Fixed           19000   Doctor Non-Recurring
5       Kumar           Pratap          Rathod          62222           Savings                 148790  Baker Non-Recurring
 Non-Recurring
 Non-Recurring
```

3. Display all records where saving account balance is greater than 100000

Grep, awk, sed

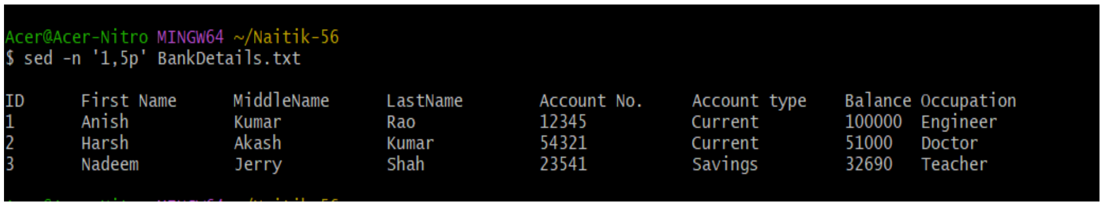


```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ awk 'NR==1 || ($6 == "Savings" && $7 > 100000)' BankDetails.txt

5       Kumar           Pratap          Rathod          62222           Savings         148790  Baker
```

**SED:** Stream Editor

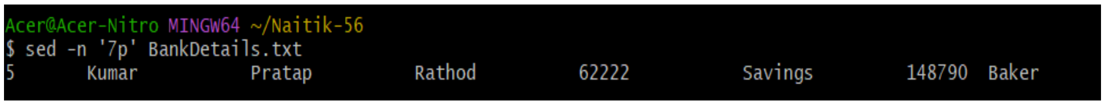
- A line-oriented tool used to **modify text on the fly**, like search C replace or delete operations.
- It processes input **line by line**, making it perfect for automating edits in files without opening them.

1. **Display first 3 records**



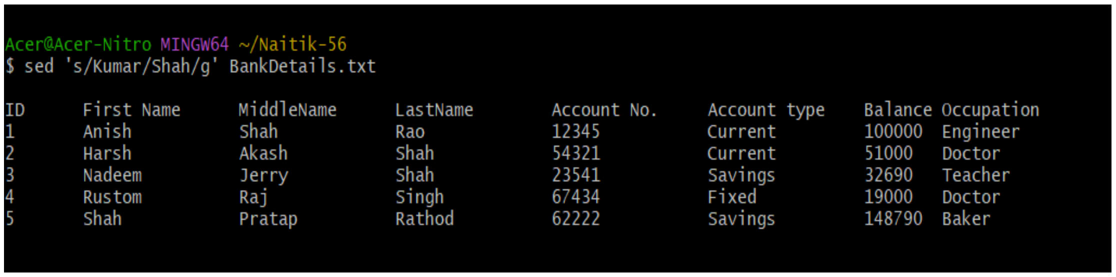
```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ sed -n '1,5p' BankDetails.txt

ID      First Name      MiddleName      LastName        Account No.     Account type    Balance Occupation
1       Anish           Kumar           Rao             12345           Current         100000  Engineer
2       Harsh           Akash           Kumar           54321           Current         51000   Doctor
3       Nadeem          Jerry           Shah            23541           Savings         32690   Teacher
```

2. **Display last record**



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ sed -n '7p' BankDetails.txt
5       Kumar           Pratap          Rathod          62222           Savings         148790  Baker
```

3. **Convert all names with Kumar to Shah**



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ sed 's/Kumar/Shah/g' BankDetails.txt

ID      First Name      MiddleName      LastName        Account No.     Account type    Balance Occupation
1       Anish           Shah            Rao             12345           Current         100000  Engineer
2       Harsh           Akash           Shah            54321           Current         51000   Doctor
3       Nadeem          Jerry           Shah            23541           Savings         32690   Teacher
4       Rustom          Raj             Singh           67434           Fixed           19000   Doctor
5       Shah            Pratap          Rathod          62222           Savings         148790  Baker
```

## CONCLUSION:

Thus we performed required data manipulation using Grep, awk and sed command on the given "Bank Details Database".

Grep, awk, sed

SEIT-S13-56

**LO4 mapped.**

SEIT-S13-56

Grep, awk, sed

## 7.Basic Shell Scripting

7a)

AIM: Write a shell script with file name as your 'firstame.sh' to do the following:

1. Assign a variable to your first name and output of date command to another variable.

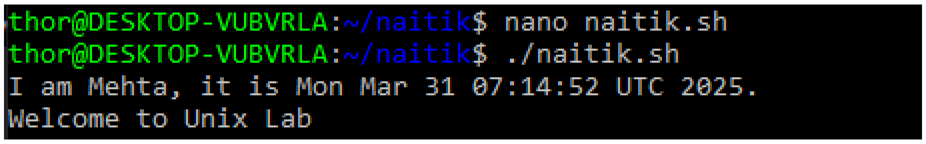
2. Output message:

"I am 'last name', it is 'output of date's.

Welcome to Unix Lab.

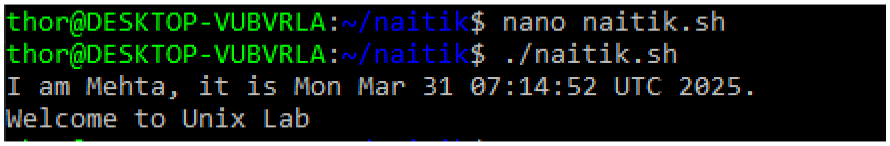
**Theory:** A **shell script** is a program written in a shell (command-line interpreter) to automate tasks. It contains a series of **commands** that the shell executes sequentially. Shell scripting is widely used in system administration, automation, data processing, and software development.

**CODE:**



```
thor@DESKTOP-VUBVRLA:~/naitik$ nano naitik.sh
thor@DESKTOP-VUBVRLA:~/naitik$ ./naitik.sh
I am Mehta, it is Mon Mar 31 07:14:52 UTC 2025.
Welcome to Unix Lab
```

**OUTPUT:**

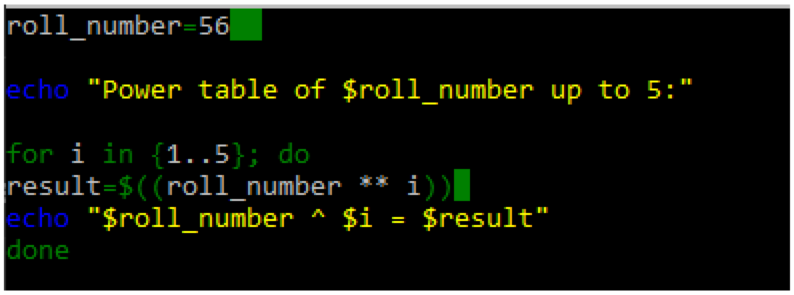


```
thor@DESKTOP-VUBVRLA:~/naitik$ nano naitik.sh
thor@DESKTOP-VUBVRLA:~/naitik$ ./naitik.sh
I am Mehta, it is Mon Mar 31 07:14:52 UTC 2025.
Welcome to Unix Lab
```

7b)

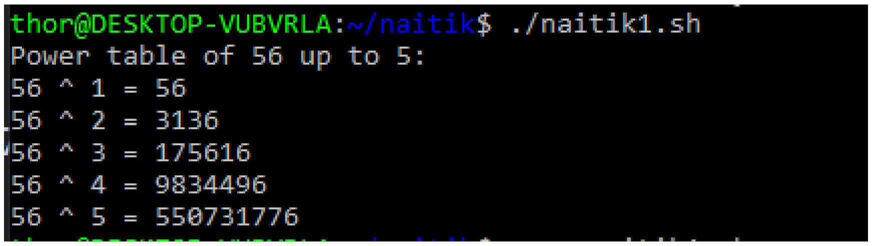
**AIM:** Display power table of your roll number upto 5 using loop in a shell script named as your 'firstname.sh'.

**CODE:**



```
roll_number=56

echo "Power table of $roll_number up to 5:"

for i in {1..5}; do
result=$((roll_number ** i))
echo "$roll_number ^ $i = $result"
done
```

**OUTPUT:**

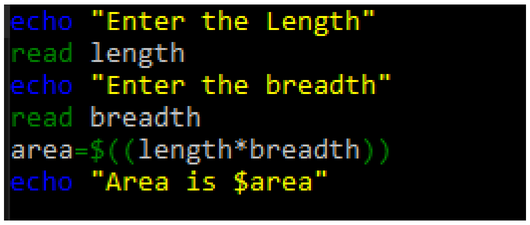


```
thor@DESKTOP-VUBVRLA:~/naitik$ ./naitik1.sh
Power table of 56 up to 5:
56 ^ 1 = 56
56 ^ 2 = 3136
56 ^ 3 = 175616
56 ^ 4 = 9834496
56 ^ 5 = 550731776
```

7c)

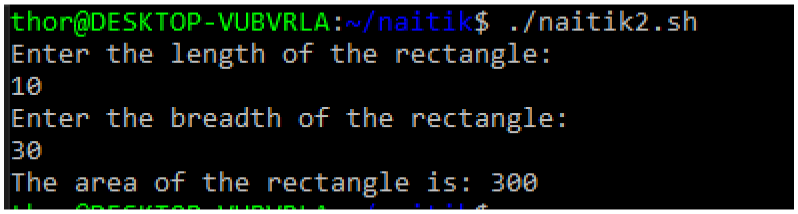
**AIM:** Calculate area of a rectangle using shell script. Use user input for length and breadth.

**CODE:**



```
echo "Enter the Length"
read length
echo "Enter the breadth"
read breadth
area=$((length*breadth))
echo "Area is $area"
```

**OUTPUT:**



```
thor@DESKTOP-VUBVRLA:~/naitik$ ./naitik2.sh
Enter the length of the rectangle:
10
Enter the breadth of the rectangle:
30
The area of the rectangle is: 300
```

**Conclusion** : Shell scripting is a powerful tool for automating tasks and performing calculations efficiently. The power table script demonstrates the use of loops and arithmetic operations to compute exponential values, making it useful for understanding iteration and variable handling in shell scripts.

**LO 2,3 mapped**

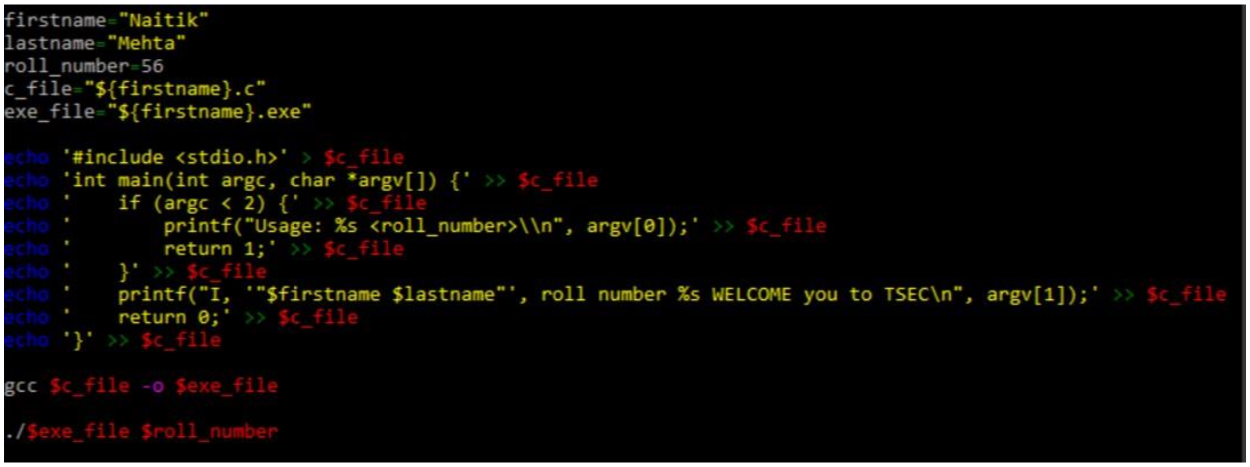
# 8. Advanced Shell Script

## 8a)

**AIM:** Write a shell script to create a c program with name as your 'firstname.c', compile it as an executive file 'firstname.exe', pass parameter as your roll number and run it to display:

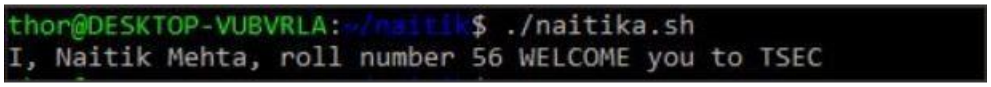
"I, firstname lastname, roll number 'num' WELCOME you to TSEC"

**CODE:**



```
firstname="Naitik"
lastname="Mehta"
roll_number=56
c_file="${firstname}.c"
exe_file="${firstname}.exe"

echo '#include <stdio.h>' > $c_file
echo 'int main(int argc, char *argv[]) {' >> $c_file
echo '    if (argc < 2) {' >> $c_file
echo '        printf("Usage: %s <roll_number>\\n", argv[0]);' >> $c_file
echo '        return 1;' >> $c_file
echo '    }' >> $c_file
echo '    printf("I, '"$firstname $lastname"', roll number %s WELCOME you to TSEC\n", argv[1]);' >> $c_file
echo '    return 0;' >> $c_file
echo '}' >> $c_file

gcc $c_file -o $exe_file

./$exe_file $roll_number
```

**OUTPUT:**

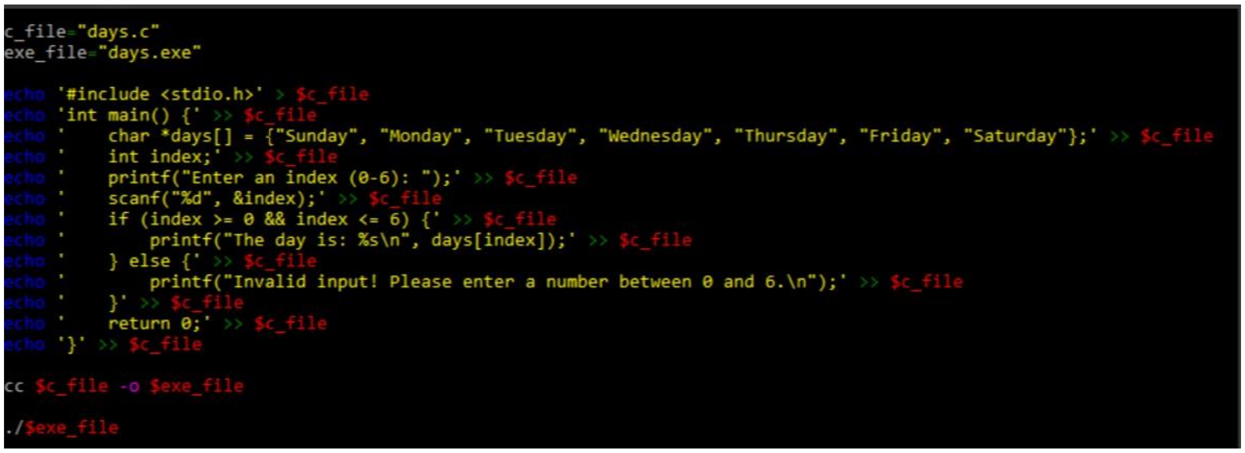


```
thor@DESKTOP-VUBVRLA:~/naitik$ ./naitika.sh
I, Naitik Mehta, roll number 56 WELCOME you to TSEC
```

## 8b)

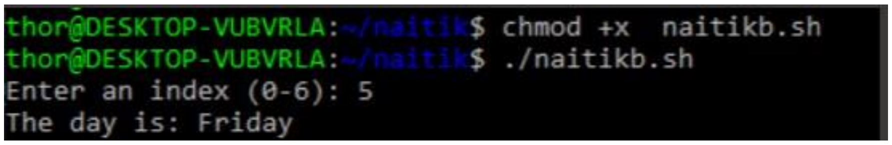
**AIM:** Write a shell script to create an array 'days' of week and display the day of a week based on user inputs index position.

**CODE:**



```
c_file="days.c"
exe_file="days.exe"

echo '#include <stdio.h>' > $c_file
echo 'int main() {' >> $c_file
echo '    char *days[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};' >> $c_file
echo '    int index;' >> $c_file
echo '    printf("Enter an index (0-6): ");' >> $c_file
echo '    scanf("%d", &index);' >> $c_file
echo '    if (index >= 0 && index <= 6) {' >> $c_file
echo '        printf("The day is: %s\n", days[index]);' >> $c_file
echo '    } else {' >> $c_file
echo '        printf("Invalid input! Please enter a number between 0 and 6.\n");' >> $c_file
echo '    }' >> $c_file
echo '    return 0;' >> $c_file
echo '}' >> $c_file

cc $c_file -o $exe_file

./$exe_file
```

**OUTPUT:**

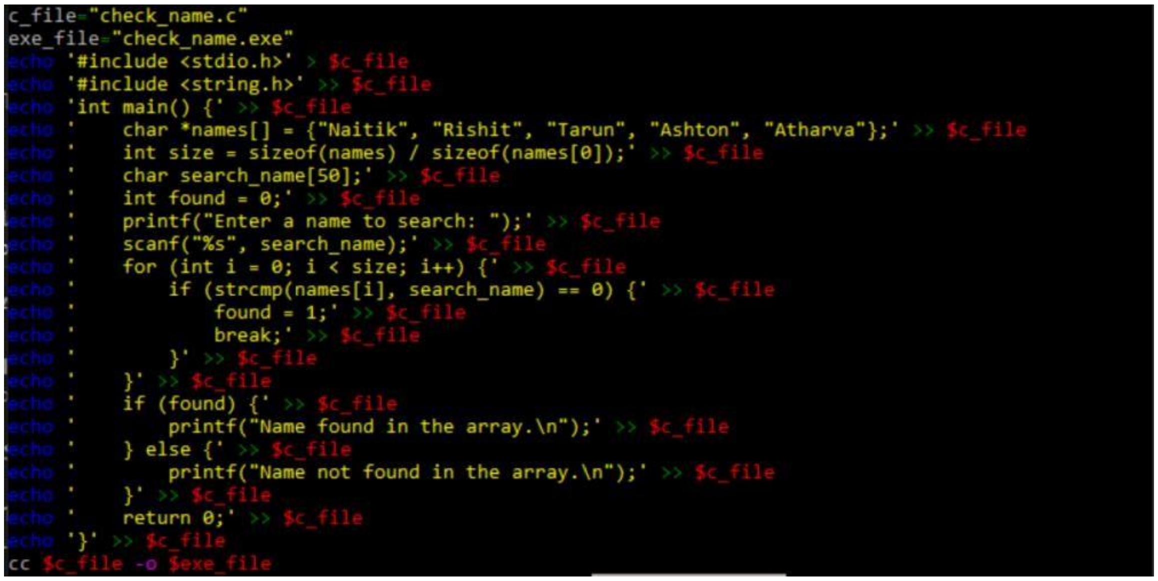


```
thor@DESKTOP-VUBVRLA:~/naitik$ chmod +x  naitikb.sh
thor@DESKTOP-VUBVRLA:~/naitik$ ./naitikb.sh
Enter an index (0-6): 5
The day is: Friday
```

**8c)**

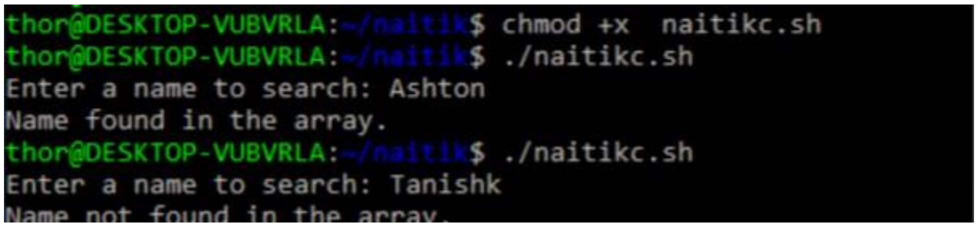
**AIM:** Write a shell script to check if a name is present in an statically assigned array of names.

**CODE:**



```
c_file="check_name.c"
exe_file="check_name.exe"
echo '#include <stdio.h>' > $c_file
echo '#include <string.h>' >> $c_file
echo 'int main() {' >> $c_file
echo '    char *names[] = {"Naitik", "Rishit", "Tarun", "Ashton", "Atharva"};' >> $c_file
echo '    int size = sizeof(names) / sizeof(names[0]);' >> $c_file
echo '    char search_name[50];' >> $c_file
echo '    int found = 0;' >> $c_file
echo '    printf("Enter a name to search: ");' >> $c_file
echo '    scanf("%s", search_name);' >> $c_file
echo '    for (int i = 0; i < size; i++) {' >> $c_file
echo '        if (strcmp(names[i], search_name) == 0) {' >> $c_file
echo '            found = 1;' >> $c_file
echo '            break;' >> $c_file
echo '        }' >> $c_file
echo '    }' >> $c_file
echo '    if (found) {' >> $c_file
echo '        printf("Name found in the array.\n");' >> $c_file
echo '    } else {' >> $c_file
echo '        printf("Name not found in the array.\n");' >> $c_file
echo '    }' >> $c_file
echo '    return 0;' >> $c_file
echo '}' >> $c_file
cc $c_file -o $exe_file
```

**OUTPUT:**



```
thor@DESKTOP-VUBVRLA:~/naitik$ chmod +x  naitikc.sh
thor@DESKTOP-VUBVRLA:~/naitik$ ./naitikc.sh
Enter a name to search: Ashton
Name found in the array.
thor@DESKTOP-VUBVRLA:~/naitik$ ./naitikc.sh
Enter a name to search: Tanishk
Name not found in the array.
```

**Conclusion:** The above programs demonstrate the automation of C program creation, compilation, and execution using shell scripting. They cover fundamental concepts like arrays, user input handling, and string comparisons.

**Learning outcome:** LO2,3,6 mapped

# G. Basic Perl Scrpting

## Ga.)

**AIM:** To implement arrays in a basic perl script

**THEORY:**

- Arrays in Perl are ordered lists of scalars that can store multiple values. They are prefixed with `@`.

Declaration and Access:

- Declared using @ symbol, e.g., `@months = ("Jan", "Feb", "Mar");`.
- Access elements using zero-based indexing, e.g., `$months[0]` gives `"Jan"`.

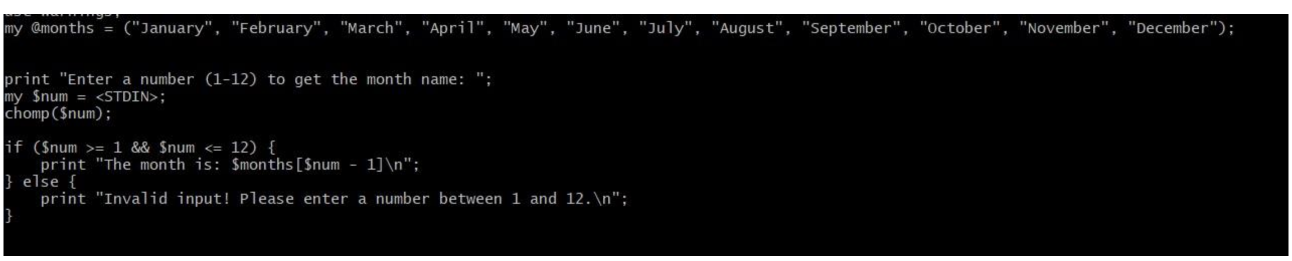
User Input Handling:

- `STDIN` is used to take user input.
- `chomp($variable)` removes the trailing newline from input.

Condition Checking:

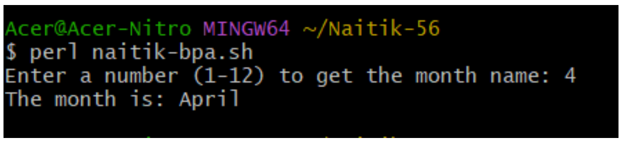
- The `if` statement ensures valid input.
- Perl arrays allow easy retrieval of values based on indices.

**CODE:**



```
my @months = ("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December");


print "Enter a number (1-12) to get the month name: ";
my $num = <STDIN>;
chomp($num);

if ($num >= 1 && $num <= 12) {
    print "The month is: $months[$num - 1]\n";
} else {
    print "Invalid input! Please enter a number between 1 and 12.\n";
}
```

**OUTPUT:**



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ perl naitik-bpa.sh
Enter a number (1-12) to get the month name: 4
The month is: April
```

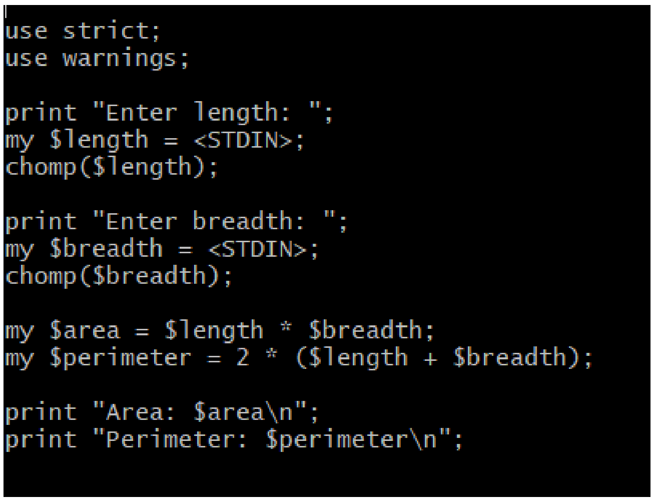
## Gb.)

**AIM:** To calculate area and perimeter of rectangle using basic perl script.

**THEORY:**

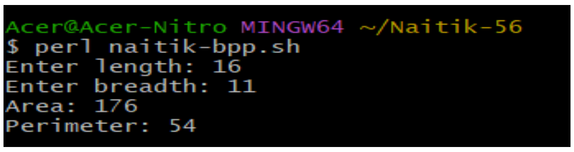
- Perl is a high-level scripting language used for text processing, automation, and system administration. A Perl script starts with #!/usr/bin/perl, followed by necessary modules like use strict; and use warnings; for error checking.
- **User Input Handling:** The STDIN function is used to take input from the user, and chomp() removes the newline character.
- **Arithmetic Operations:** Perl supports mathematical calculations using standard operators like * (multiplication) and + (addition). Variables are prefixed with $ and assigned values dynamically.

## CODE:



```perl
use strict;
use warnings;

print "Enter length: ";
my $length = <STDIN>;
chomp($length);

print "Enter breadth: ";
my $breadth = <STDIN>;
chomp($breadth);

my $area = $length * $breadth;
my $perimeter = 2 * ($length + $breadth);

print "Area: $area\n";
print "Perimeter: $perimeter\n";
```

## OUTPUT:



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ perl naitik-bpp.sh
Enter length: 16
Enter breadth: 11
Area: 176
Perimeter: 54
```

## CONCLUSION:

Perl is a powerful scripting language that simplifies text processing, user input handling, and arithmetic operations.

## LO2, 3, 6 mapped

# 10. Advanced Perl Scripting

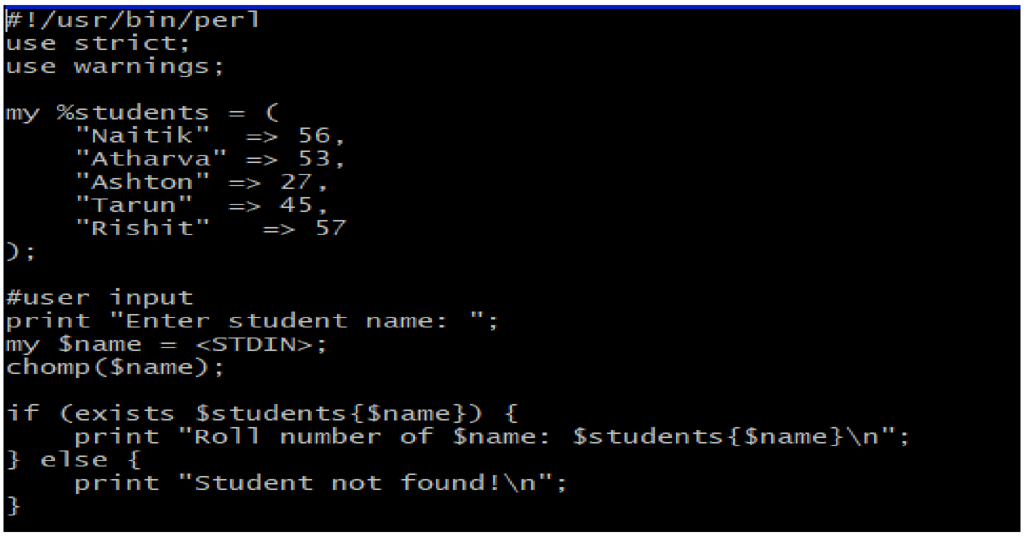
## 10a.)

**AIM:** To implement hashes to write names and roll numbers in Perl

**THEORY:** Hashes in Perl are key-value pairs, where keys are unique and associated with specific values. They are declared using % (e.g., `%students = ("Naitik" => 56, "Atharva" => 53);`).

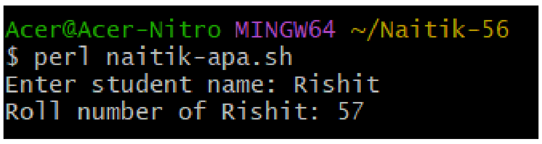
- <u>Accessing Hash Elements</u>: Use `$hash{key}` to retrieve the corresponding value.
  Example: `$roll = $students{"Atharva"};  retrieves Atharva's roll number`
- <u>Checking Key Existence</u>: The `exists` function checks whether a key exists in the hash, preventing errors when accessing non-existent keys.
- <u>Use Cases</u>: Hashes are useful for lookup tables, database-like storage, and fast data retrieval, making them ideal for name-roll number mappings in this script.

**PROGRAM:**



```
#!/usr/bin/perl
use strict;
use warnings;

my %students = (
    "Naitik"   => 56,
    "Atharva"  => 53,
    "Ashton"  => 27,
    "Tarun"   => 45,
    "Rishit"    => 57
);

#user input
print "Enter student name:  ";
my $name = <STDIN>;
chomp($name);

if (exists $students{$name}) {
    print "Roll number of $name: $students{$name}\n";
} else {
    print "Student not found!\n";
}
```

**OUTPUT:**



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ perl naitik-apa.sh
Enter student name: Rishit
Roll number of Rishit: 57
```

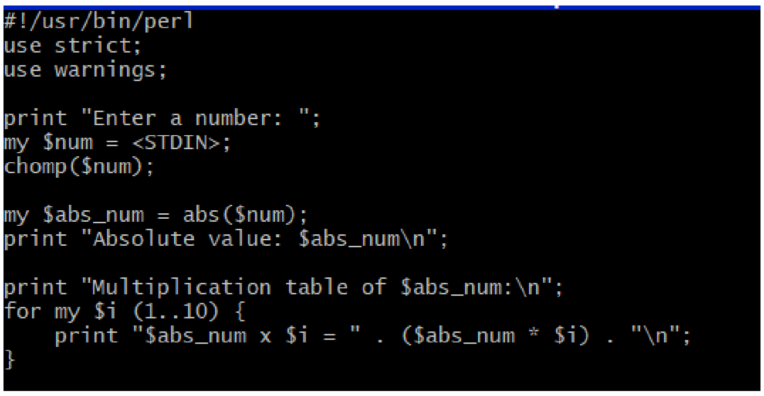
## 10b.)

**AIM:** To generate an absolute value table of a number using Perl

**THEORY:**

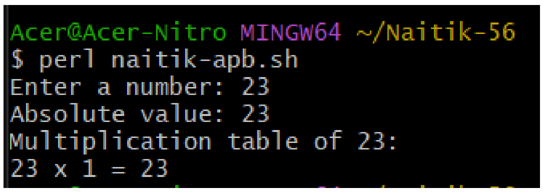
1. Absolute Value:
    a. The `abs()` function in Perl returns the absolute value of a given number, converting negative numbers to positive.
2. User Input Handling:
    a. `STDIN` reads input from the user, and `chomp()` removes any newline characters.
3. Loops in Perl:
    a. The `for` loop (`for my $i (1..10)`) generates and prints the multiplication table from 1 to 10.
4. String Interpolation C Arithmetic:
    a. Perl allows embedding expressions within strings and performing arithmetic operations directly.

**PROGRAM:**



```
#!/usr/bin/perl
use strict;
use warnings;

print "Enter a number: ";
my $num = <STDIN>;
chomp($num);

my $abs_num = abs($num);
print "Absolute value: $abs_num\n";

print "Multiplication table of $abs_num:\n";
for my $i (1..10) {
    print "$abs_num x $i = " . ($abs_num * $i) . "\n";
}
```

**OUTPUT:**



```
Acer@Acer-Nitro MINGW64 ~/Naitik-56
$ perl naitik-apb.sh
Enter a number: 23
Absolute value: 23
Multiplication table of 23:
23 x 1 = 23
```

**CONCLUSION:**

Perl provides a simple and efficient way to perform mathematical operations, including absolute value computation and table generation. Using abs(), for loops, and hash-based lookups, Perl scripts handle tasks such as user input validation, data storage, and retrieval. The flexibility of Perl makes it ideal for automation and quick computations in various real-world applications.

**LO 2, 3, 6 mapped**