DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS
Institute of Engineering & Technology

Lab Manual

Subject Name & Code: OPERATING SYSTEM LAB (BCAC 0818)

Course: BCA

Faculty's Name: **Dr. Anuj Mangal**
Dr. Anil Kumar Chanchal
Ms. Paromita Goswami
Mr. Anuj Kumar
Mr. Santosh Swarnkar
Mr. Jayati Krishna Goswami
Ms. Cheshtaa Bhardwaj
Mr. Ankit Prakash

Year: II **Semester: III** **Session: 2024-25**

OPERATING SYSTEM LAB (BCAC 0818)

Prerequisite: *Familiarity with programming constructs*

Credits: 02 **Semester III** **L–T–P : 0–0–2**

Case Study on Open Standard and Software:

Introduction to Linux operating system and Preparing for Installation – Installation Checklist, Hardware Requirements, Partitioning, Installation problems, Working with the System, Shells and Utilities, Linux commands, File Handling using vi editor, Getting familiar with shell scripts.

1. Introduction to Linux Operating System

Linux is an open-source operating system that follows the principles of open standards. It provides users and developers with the freedom to modify, distribute, and use the software without proprietary restrictions. Linux is widely adopted in various domains such as servers, desktops, and embedded systems due to its flexibility, security, and cost-effectiveness.

2. Preparing for Installation

To ensure a smooth installation process for Linux, careful preparation is required. The preparation steps involve checking hardware compatibility, understanding partitioning schemes, and following a checklist to avoid potential problems.

3. Installation Checklist

Before installing Linux, ensure that the following items are in place:

- **Backup of important data**: Ensure that all important data on the computer is backed up.
- **Linux distribution**: Select and download the appropriate Linux distribution (e.g., Ubuntu, Fedora, CentOS).
- **Create a bootable USB**: Use tools like Rufus or UNetbootin to create a bootable USB drive for installation.
- **Check hardware compatibility**: Ensure that the system's hardware supports the selected Linux distribution.
- **Partitioning scheme**: Plan how to partition the hard disk. Linux generally needs at least a root ("/") partition and a swap partition.

4. Hardware Requirements

While Linux can run on a wide range of hardware, it is important to verify the minimum system requirements for the chosen distribution.

- **Minimum hardware requirements**:
    - Processor: 1 GHz (x86 or x86_64)
    - RAM: 1 GB for lightweight distributions, 2-4 GB for mainstream ones (e.g., Ubuntu, Fedora)
    - Hard Disk: Minimum 20 GB
    - Display: VGA capable of 1024x768 screen resolution

5. Partitioning

Linux allows for flexible partitioning schemes. It is recommended to use the following partitions:

- **Root ("/")**: This is where the operating system files are stored.
- **Swap**: This is used for virtual memory. Its size is typically 1.5 times the system's RAM.
- **Home ("/home")**: This partition stores user data and settings.
- **Optional partitions**: You may choose to have separate partitions for /boot or /var for better system management.

6. Installation Problems

Common installation problems include:

- **Boot issues**: The system might fail to boot if the bootloader (GRUB) is not properly configured.
- **Partitioning errors**: Incorrect partitioning can result in data loss or installation failure.
- **Driver issues**: Some hardware components (e.g., Wi-Fi, graphics cards) may not be supported out of the box.

To address these issues, ensure that you:

- **Check documentation**: Review the distribution's installation guide.

- **Test live environments**: Use a live USB session to test the hardware before installation.
- **Update drivers**: Download the necessary drivers from the manufacturer if required.

7. Working with the System

Once Linux is installed, familiarize yourself with the desktop environment (e.g., GNOME, KDE, or Xfce) and package management tools (e.g., apt, yum, or dnf) to install software.

- **Package management**: Use commands like sudo apt install <package> (Debian-based) or sudo dnf install <package> (Fedora-based) to install software.

8. Shells and Utilities

Linux provides powerful shell environments like Bash, Zsh, and others. Shells allow users to interact with the system through command-line utilities. Important shell utilities include:

- **Navigating directories**: cd, ls
- **Copying and moving files**: cp, mv
- **Displaying file contents**: cat, less

9. Linux Commands

Some essential Linux commands for system management include:

- **System management**:
    - sudo: Execute commands with superuser privileges.
    - df -h: Show disk space usage.
    - free -m: Display memory usage.
- **File management**:
    - touch <filename>: Create a new file.
    - rm <filename>: Remove a file.
    - chmod: Change file permissions.

10. File Handling using vi Editor

The **vi editor** is a powerful text editor in Linux. Basic vi commands include:

- **Entering vi**: vi <filename>

- **Insert mode**: Press i to start editing the file.
- **Save and exit**: Press Esc followed by :wq to save and exit.

11. Getting Familiar with Shell Scripts

Shell scripting automates tasks in Linux. A simple shell script example:

```
#!/bin/bash

echo "Hello, World!"
```

To run the script, save it as script.sh, then execute the following:

```
chmod +x script.sh

./script.sh
```

Shell scripts can handle loops, conditions, and functions, making them ideal for automating tasks like backups, file management, or system monitoring.

Conclusion

Linux, with its open standards and vast capabilities, is a robust operating system suited for diverse computing needs. Understanding the installation process, commands, and scripting enhances productivity and efficiency in managing Linux systems.

EXPERIMENT NO: 2

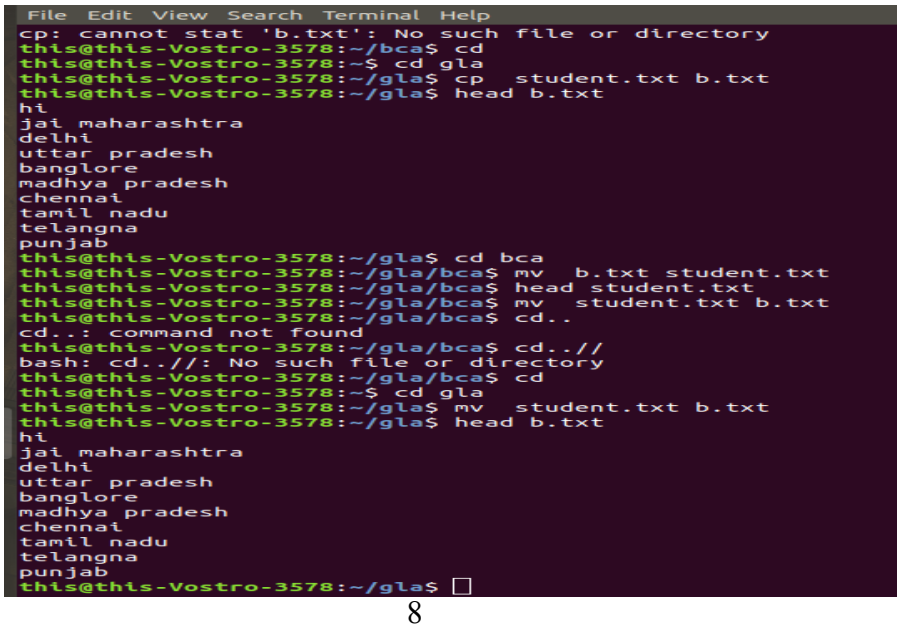
Environment: Ubuntu

Objective: Implement the following basic commands (with options) used in UNIX/LINUX OS: a) cp b) mv c) sort d) cut e) who f) whoami g) ps h) kill i) bc j) top k) grep l) chmod

Layout :

Procedure:

1. cp



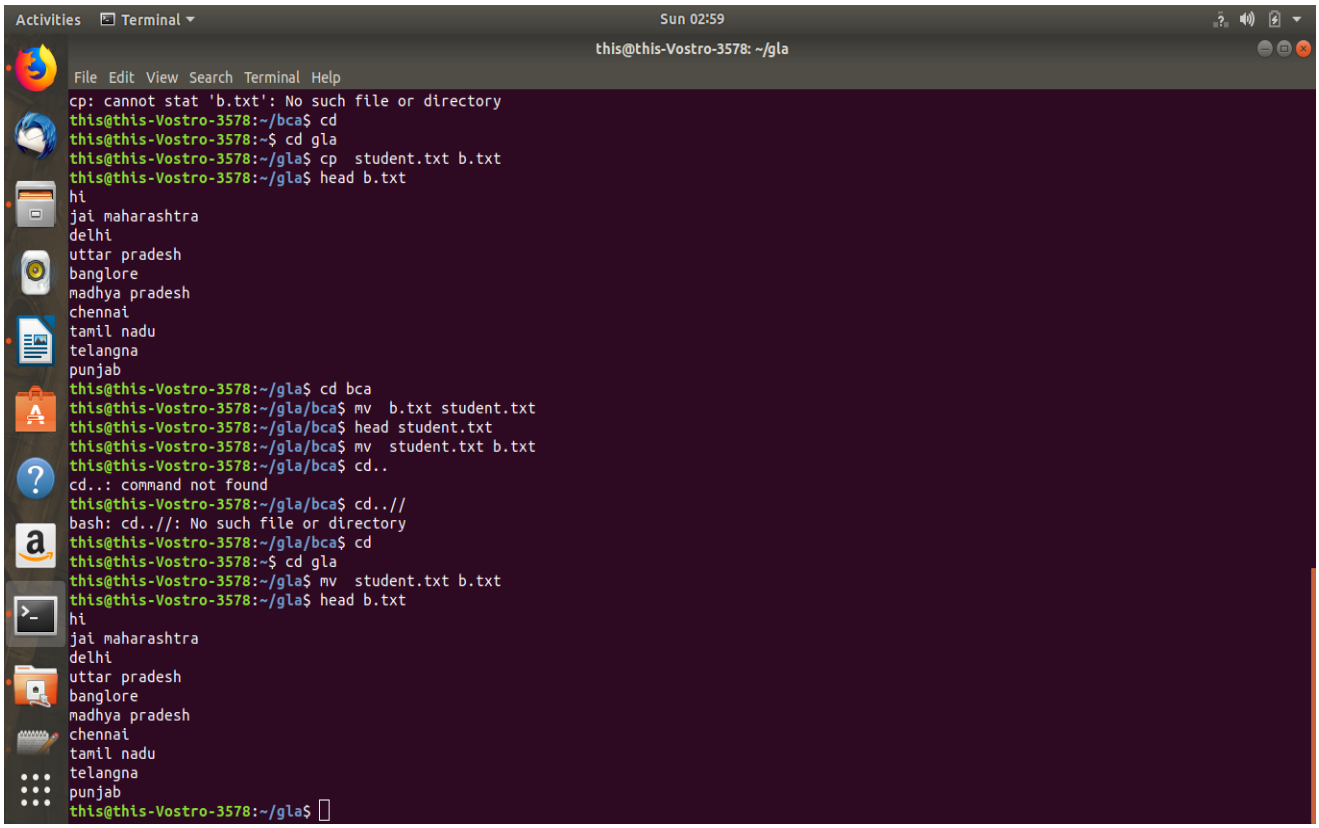
DESCRIPTION-
cp - copy files and directories
SYNTAX-

- cp <first file name><second file name>

2. mv

DESCRIPTION-
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.
SYNTAX-

- mv <file_name1><file_name2>



3. sort

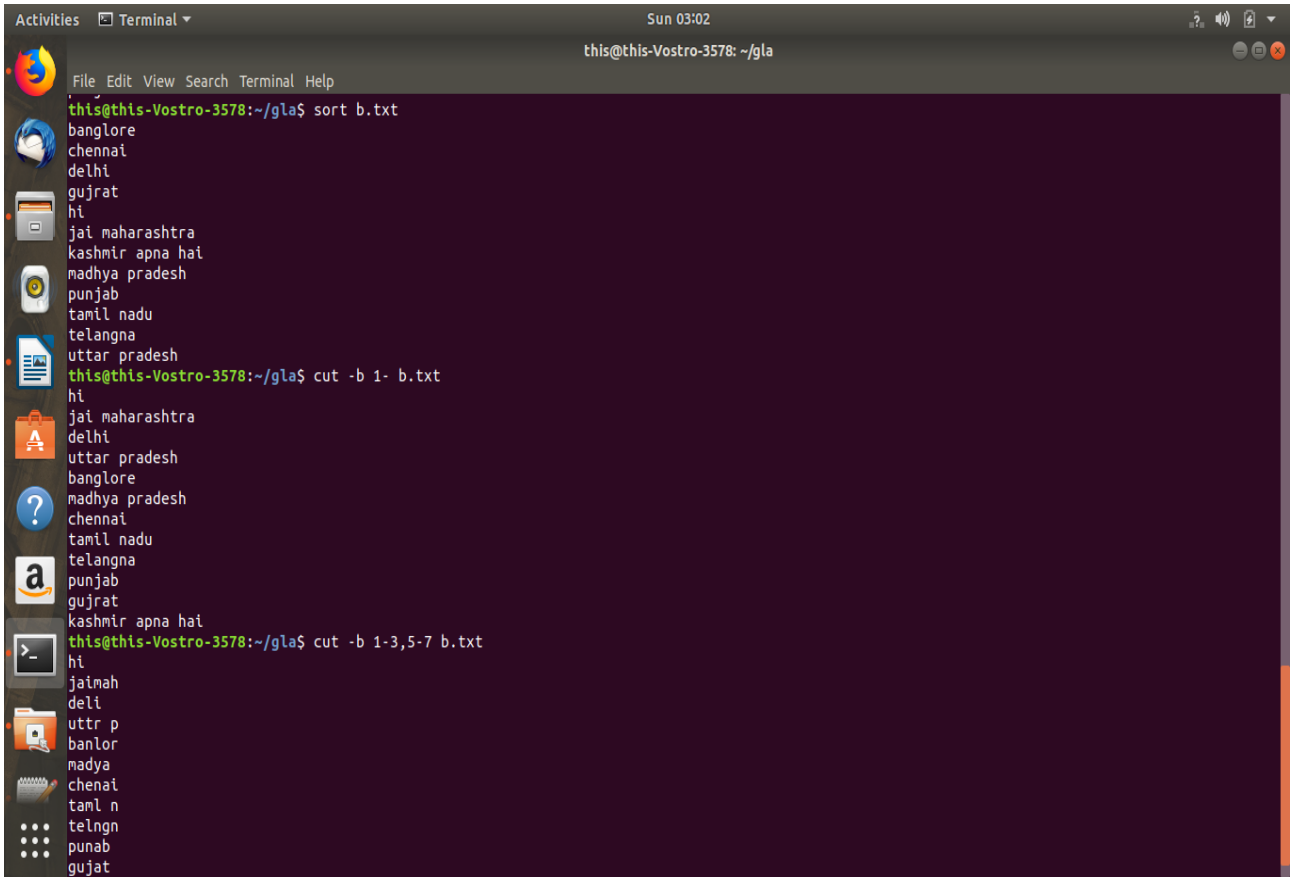
DESCRIPTION-
Write sorted concatenation of all FILE(s) to standard output.
SYNTAX-

- sort <file_name>
- sort -n <file_name>
- sort -r <file_name>

4. cut

DESCRIPTION-
cut - remove sections from each line of files
SYNTAX-

- cut -b <file_name>
- cut -c <file_name>kill - send a signal to a process



5. who

DESCRIPTION-
show who is logged on
SYNTAX-

- who

6. whoami

DESCRIPTION-
print effective userid
SYNTAX-

- whoami

7. ps

DESCRIPTION-
report a snapshot of the current processes.
SYNTAX-

- ps

8. kill

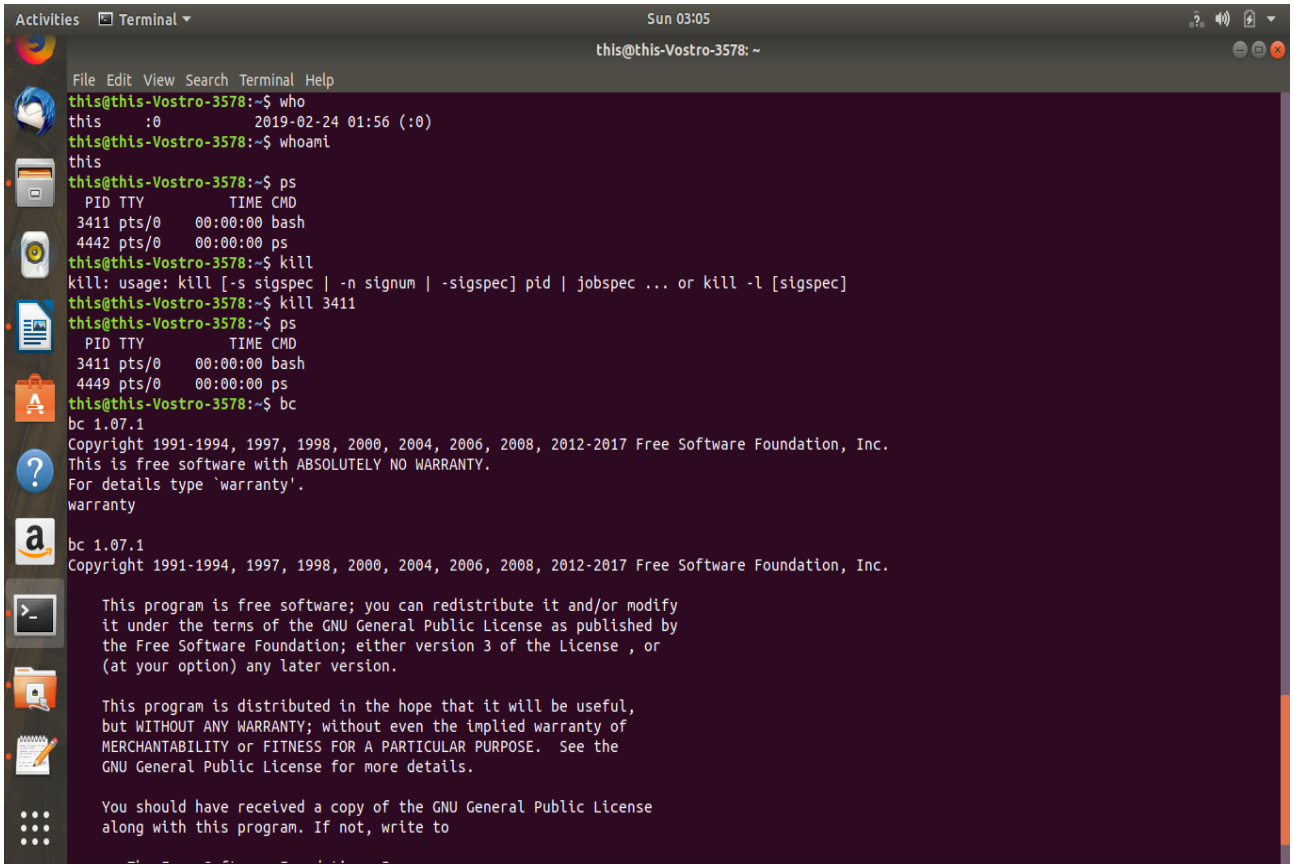
DESCRIPTION-
kill - send a signal to a process to kill.
SYNTAX-

- kill <process_id>

9. bc

DESCRIPTION-
An arbitrary precision calculator language.
SYNTAX-

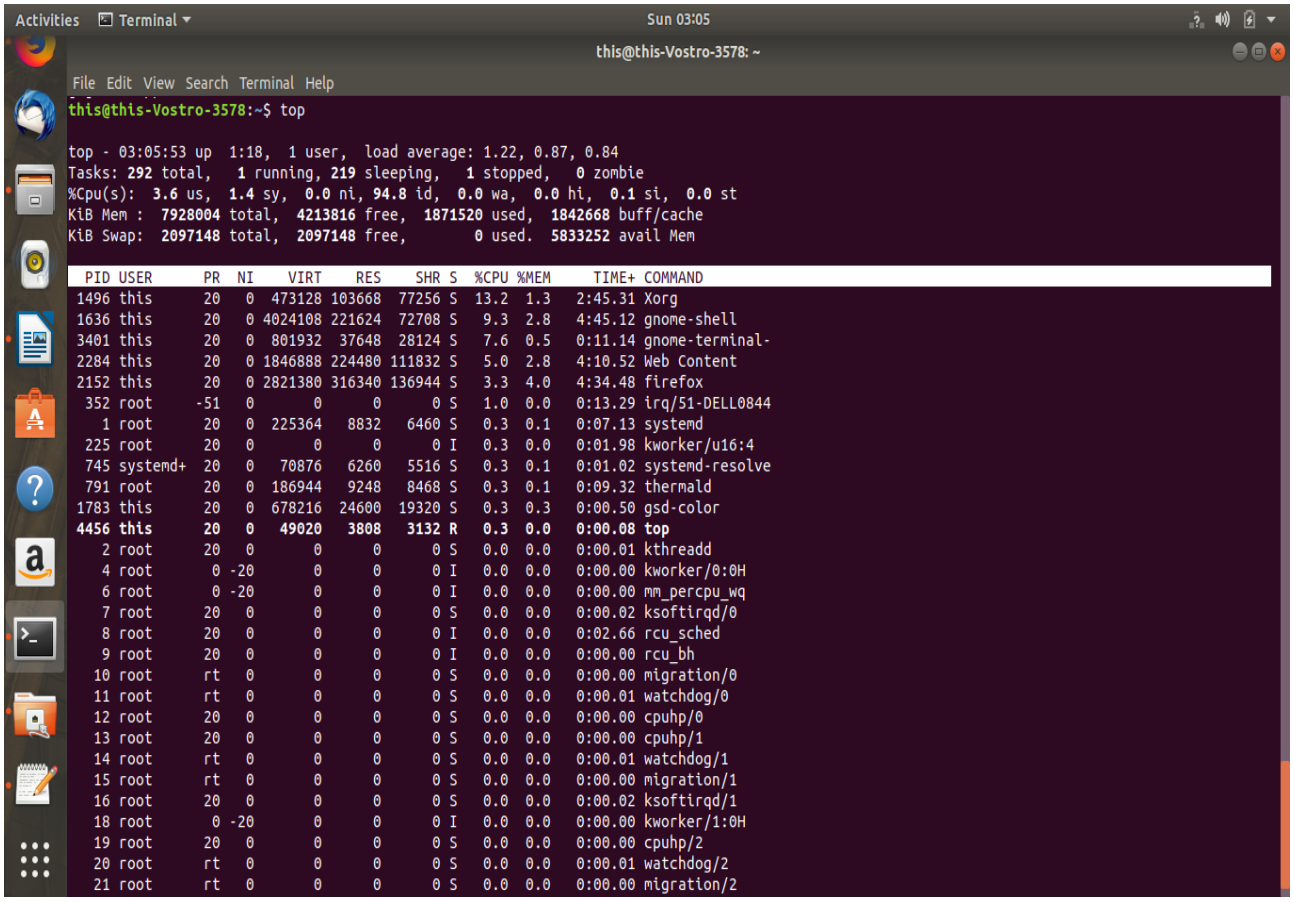
- bc



10. top

DESCRIPTION-
top - display Linux processes
SYNTAX-

- top



11. grep

DESCRIPTION-
print lines matching a pattern
SYNTAX-

- grep

12. chmod

DESCRIPTION-
change file mode bits.
SYNTAX-

- chmod -x <file_name>

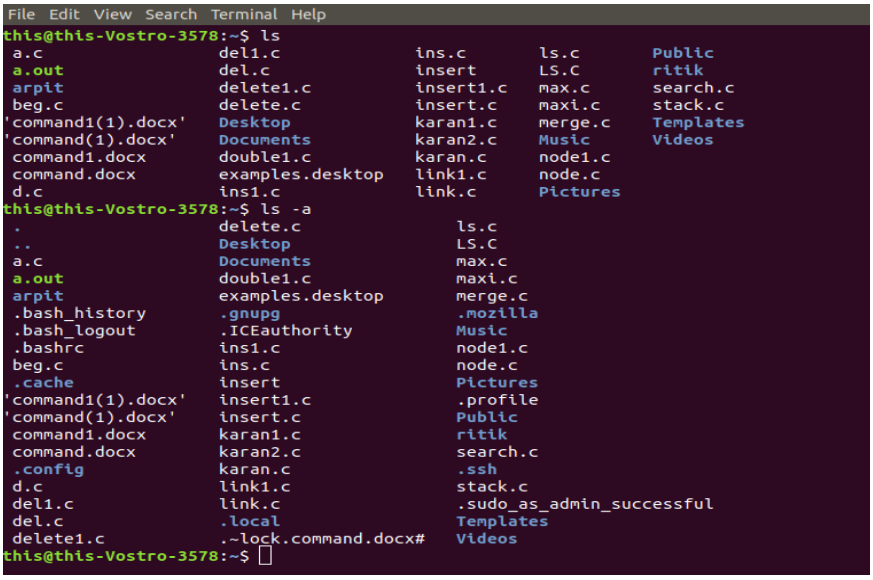
chmod -w <file_name>

EXPERIMENT NO: 3

Environment: Ubuntu

Objective: Implement the following basic commands (with options) used in UNIX/LINUX OS: a) ls b) mkdir c) cd d) cat e) man f) date g) cal h) rm i) rmdir j) head k) tail l) pwd

Layout:



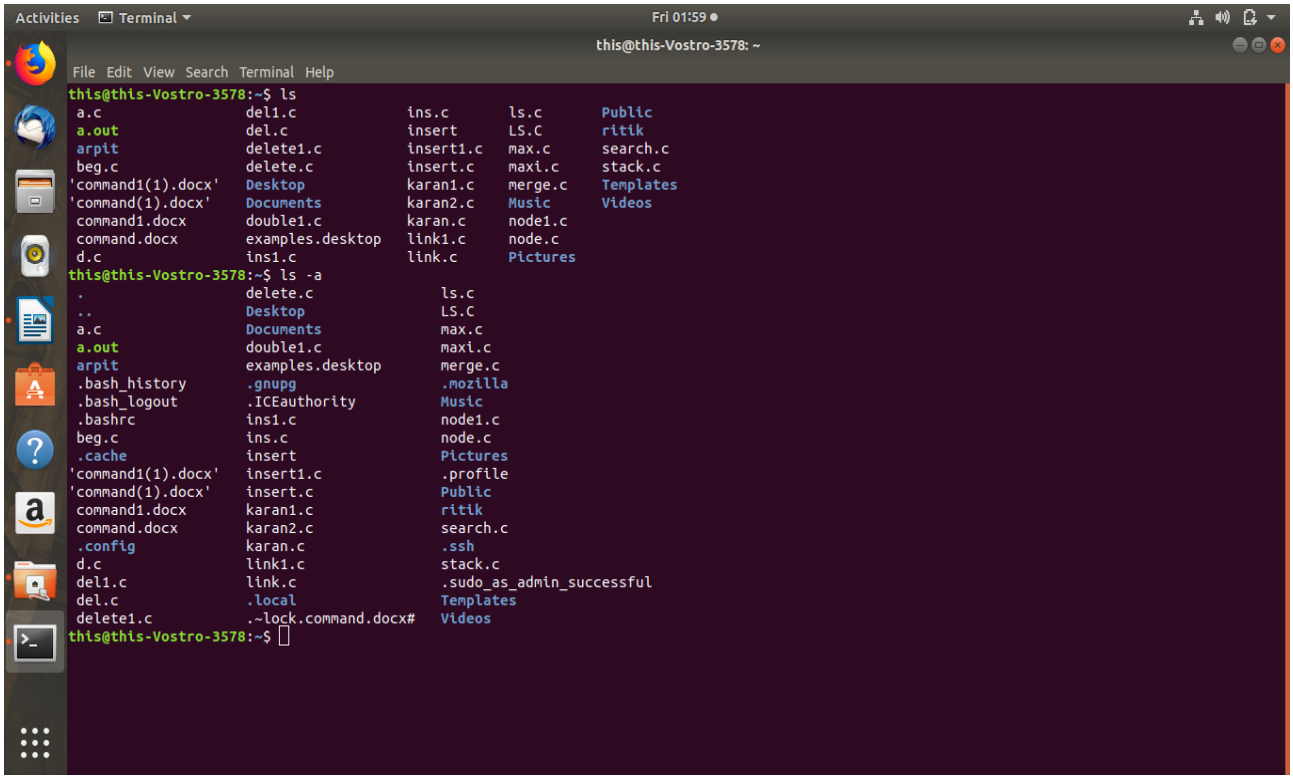
Procedure:

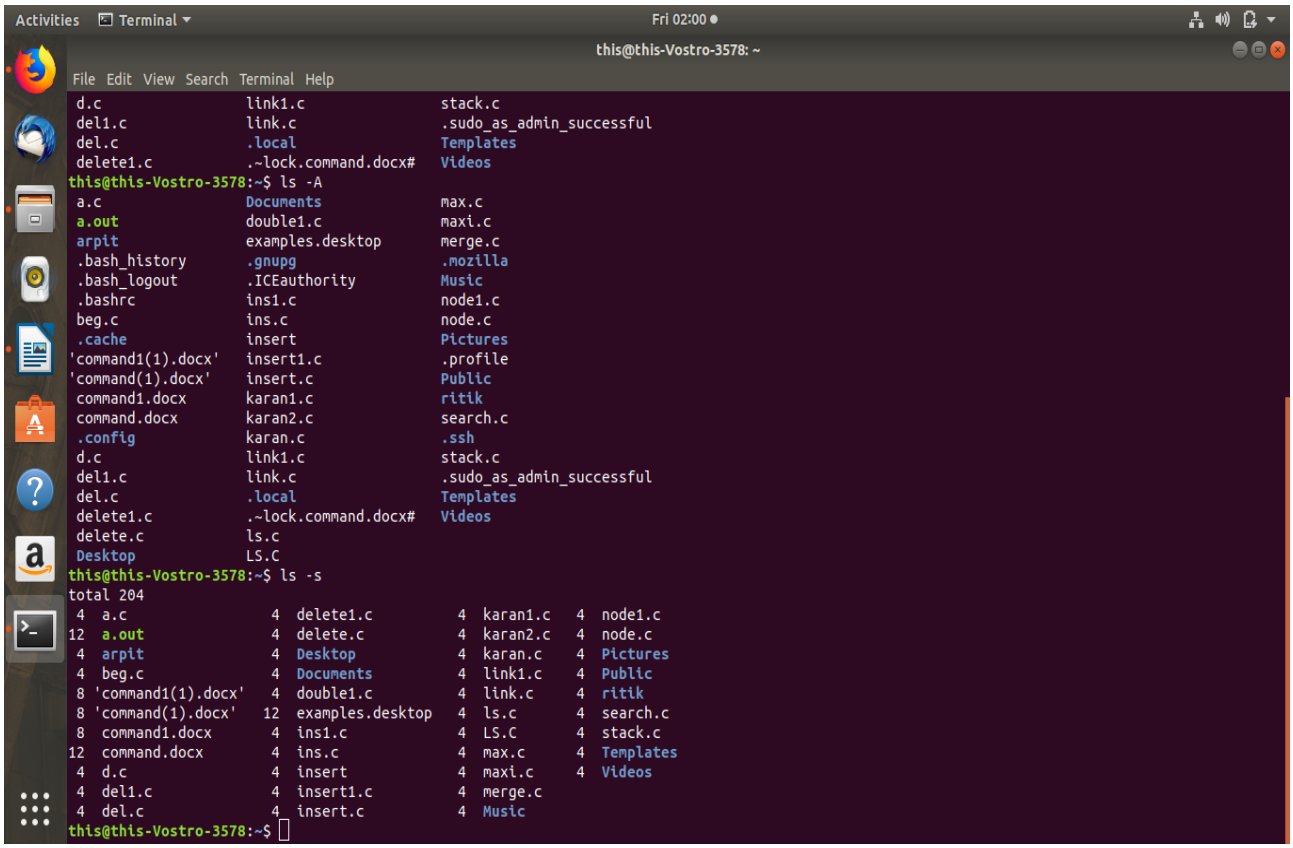
1. ls

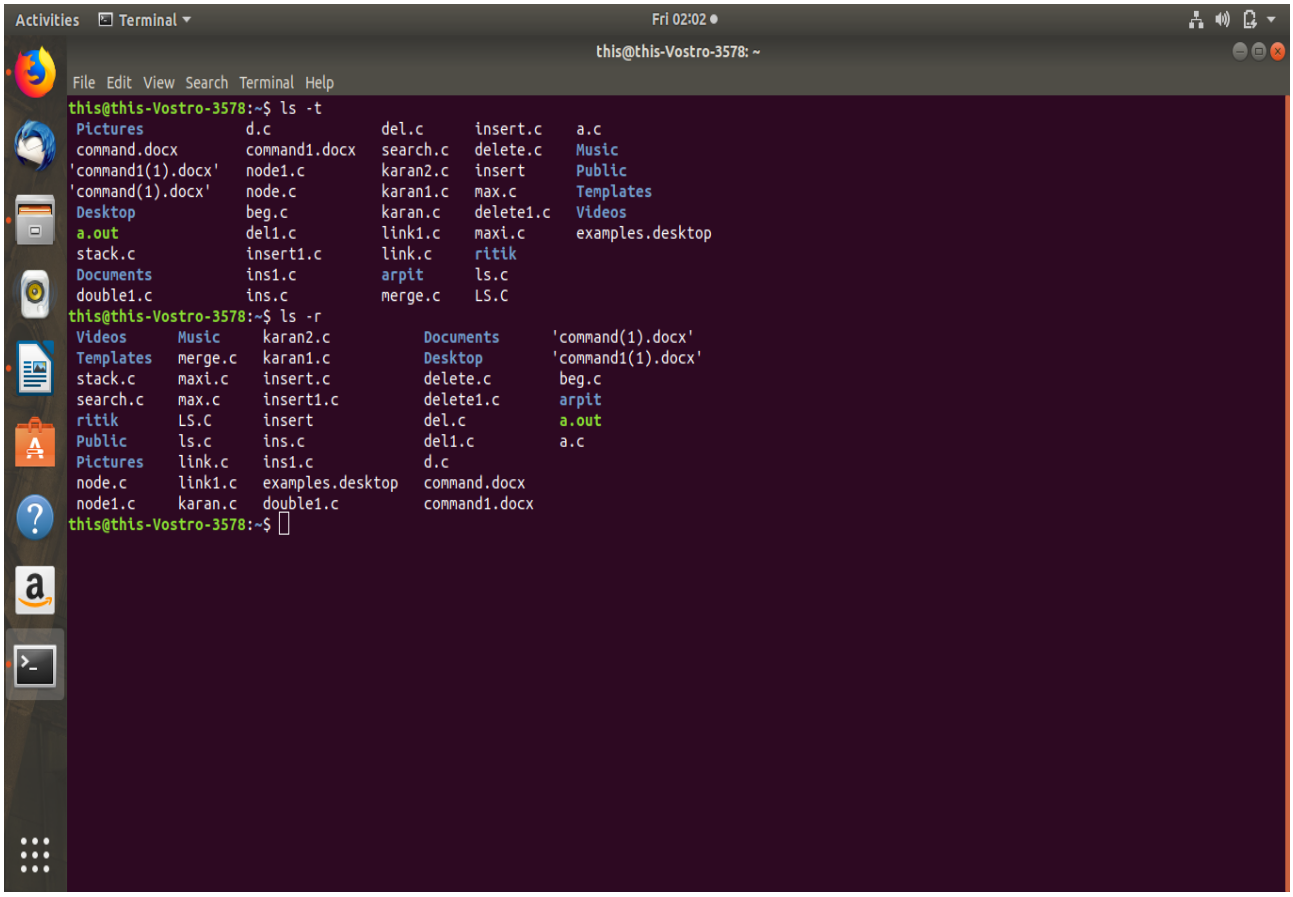
DESCRIPTION-
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
SYNTAX-

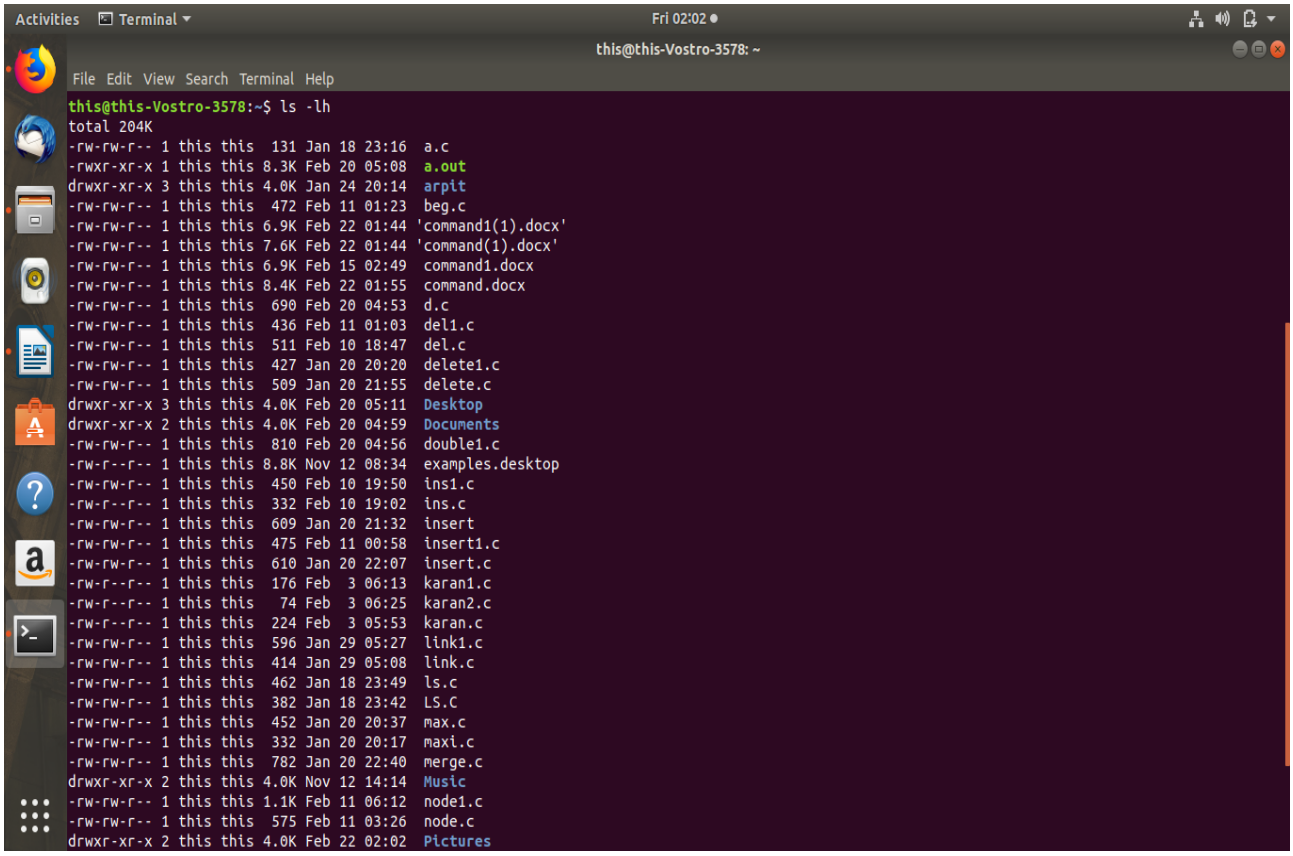
- ls
- ls -a
- ls -A
- ls -s

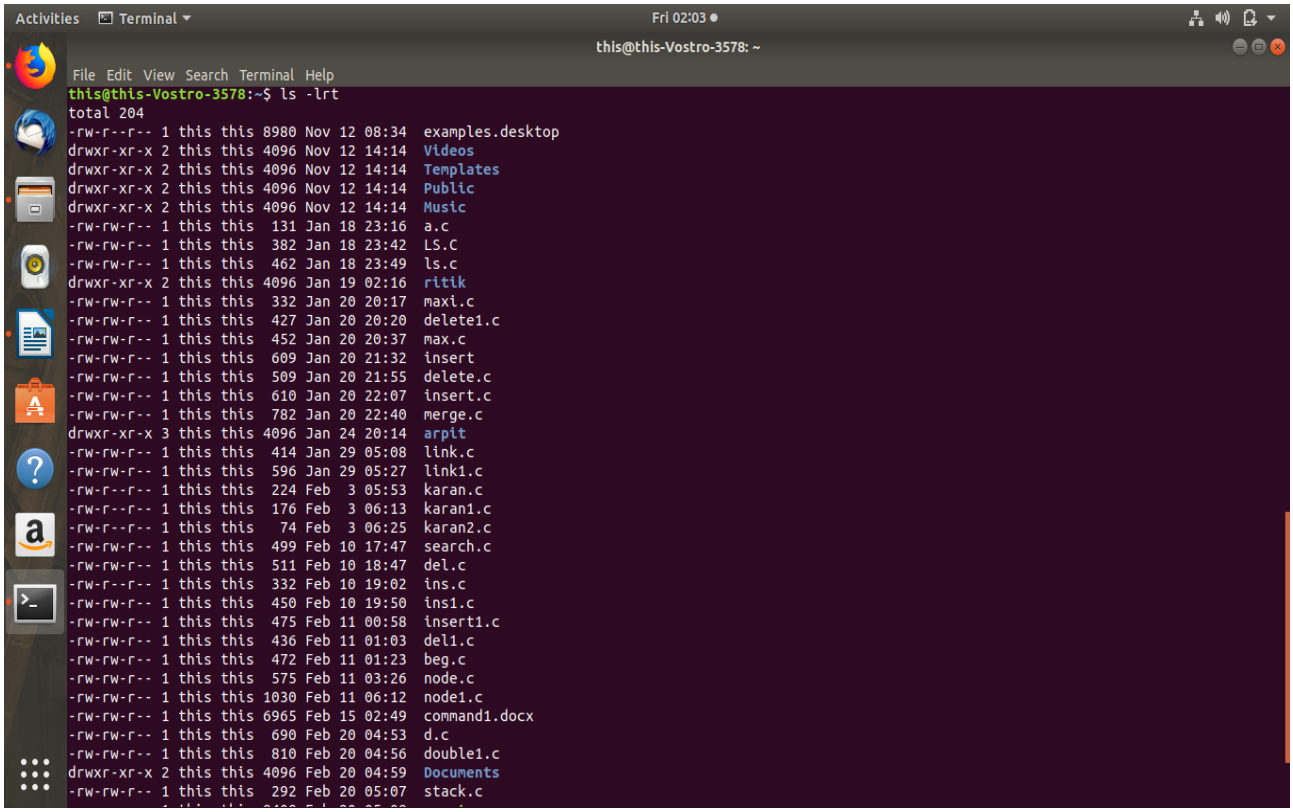
- ls -t
- ls -r
- ls -lh
- ls -lrt






Activities
Terminal
Fri 02:02
this@this-Vostro-3578: ~
File Edit View Search Terminal Help
this@this-Vostro-3578:~$ ls -t
Pictures d.c del.c insert.c a.c
command.docx command1.docx search.c delete.c Music
'command1(1).docx' node1.c karan2.c insert Public
'command(1).docx' node.c karan1.c max.c Templates
Desktop beg.c karan.c delete1.c Videos
a.out del1.c link1.c maxi.c examples.desktop
stack.c insert1.c link.c ritik
Documents ins1.c arpit ls.c
double1.c ins.c merge.c LS.C
this@this-Vostro-3578:~$ ls -r
Videos Music karan2.c Documents 'command(1).docx'
Templates merge.c karan1.c Desktop 'command1(1).docx'
stack.c maxi.c insert.c delete.c beg.c
search.c max.c insert1.c delete1.c arpit
ritik LS.C insert del.c a.out
Public ls.c ins.c del1.c a.c
Pictures link.c ins1.c d.c
node.c link1.c examples.desktop command.docx
node1.c karan.c double1.c command1.docx
this@this-Vostro-3578:~$


Activities
Terminal
Fri 02:02
this@this-Vostro-3578: ~
File Edit View Search Terminal Help
this@this-Vostro-3578:~$ ls -lh
total 204K
-rw-rw-r-- 1 this this 131 Jan 18 23:16 a.c
-rwxr-xr-x 1 this this 8.3K Feb 20 05:08 a.out
drwxr-xr-x 3 this this 4.0K Jan 24 20:14 arpit
-rw-rw-r-- 1 this this 472 Feb 11 01:23 beg.c
-rw-rw-r-- 1 this this 6.9K Feb 22 01:44 'command1(1).docx'
-rw-rw-r-- 1 this this 7.6K Feb 22 01:44 'command(1).docx'
-rw-rw-r-- 1 this this 6.9K Feb 15 02:49 command1.docx
-rw-rw-r-- 1 this this 8.4K Feb 22 01:55 command.docx
-rw-rw-r-- 1 this this 690 Feb 20 04:53 d.c
-rw-rw-r-- 1 this this 436 Feb 11 01:03 del1.c
-rw-rw-r-- 1 this this 511 Feb 10 18:47 del.c
-rw-rw-r-- 1 this this 427 Jan 20 20:20 delete1.c
-rw-rw-r-- 1 this this 509 Jan 20 21:55 delete.c
drwxr-xr-x 3 this this 4.0K Feb 20 05:11 Desktop
drwxr-xr-x 2 this this 4.0K Feb 20 04:59 Documents
-rw-rw-r-- 1 this this 810 Feb 20 04:56 double1.c
-rw-r--r-- 1 this this 8.8K Nov 12 08:34 examples.desktop
-rw-rw-r-- 1 this this 450 Feb 10 19:50 ins1.c
-rw-r--r-- 1 this this 332 Feb 10 19:02 ins.c
-rw-rw-r-- 1 this this 609 Jan 20 21:32 insert
-rw-rw-r-- 1 this this 475 Feb 11 00:58 insert1.c
-rw-rw-r-- 1 this this 610 Jan 20 22:07 insert.c
-rw-r--r-- 1 this this 176 Feb 3 06:13 karan1.c
-rw-r--r-- 1 this this 74 Feb 3 06:25 karan2.c
-rw-r--r-- 1 this this 224 Feb 3 05:53 karan.c
-rw-rw-r-- 1 this this 596 Jan 29 05:27 link1.c
-rw-rw-r-- 1 this this 414 Jan 29 05:08 link.c
-rw-rw-r-- 1 this this 462 Jan 18 23:49 ls.c
-rw-rw-r-- 1 this this 382 Jan 18 23:42 LS.C
-rw-rw-r-- 1 this this 452 Jan 20 20:37 max.c
-rw-rw-r-- 1 this this 332 Jan 20 20:17 maxi.c
-rw-rw-r-- 1 this this 782 Jan 20 22:40 merge.c
drwxr-xr-x 2 this this 4.0K Nov 12 14:14 Music
-rw-rw-r-- 1 this this 1.1K Feb 11 06:12 node1.c
-rw-rw-r-- 1 this this 575 Feb 11 03:26 node.c
drwxr-xr-x 2 this this 4.0K Feb 22 02:02 Pictures



2. mkdir

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

- mkdir <directory_name>

3. cd

DESCRIPTION-
It changes your working directory. Use it to move around within the hierarchy of your file system.
SYNTAX-

- cd
- cd <directory_name>
- cd..
- cd../..

4. cat

DESCRIPTION-
Concatenate FILE(s) to standard output.

SYNTAX-

- cat <file_name>
- cat ><file_name>
- cat <file_name1>><file_name2>

5. man

DESCRIPTION-
man - an interface to the on-line reference manuals
SYNTAX-

- man <command>

6. date

DESCRIPTION-
date - print or set the system date and time
SYNTAX-

- date
- date -d

7. cal

DESCRIPTION-
cal, ncal — displays a calendar and the date of Easter
SYNTAX-

- cal

8. rm

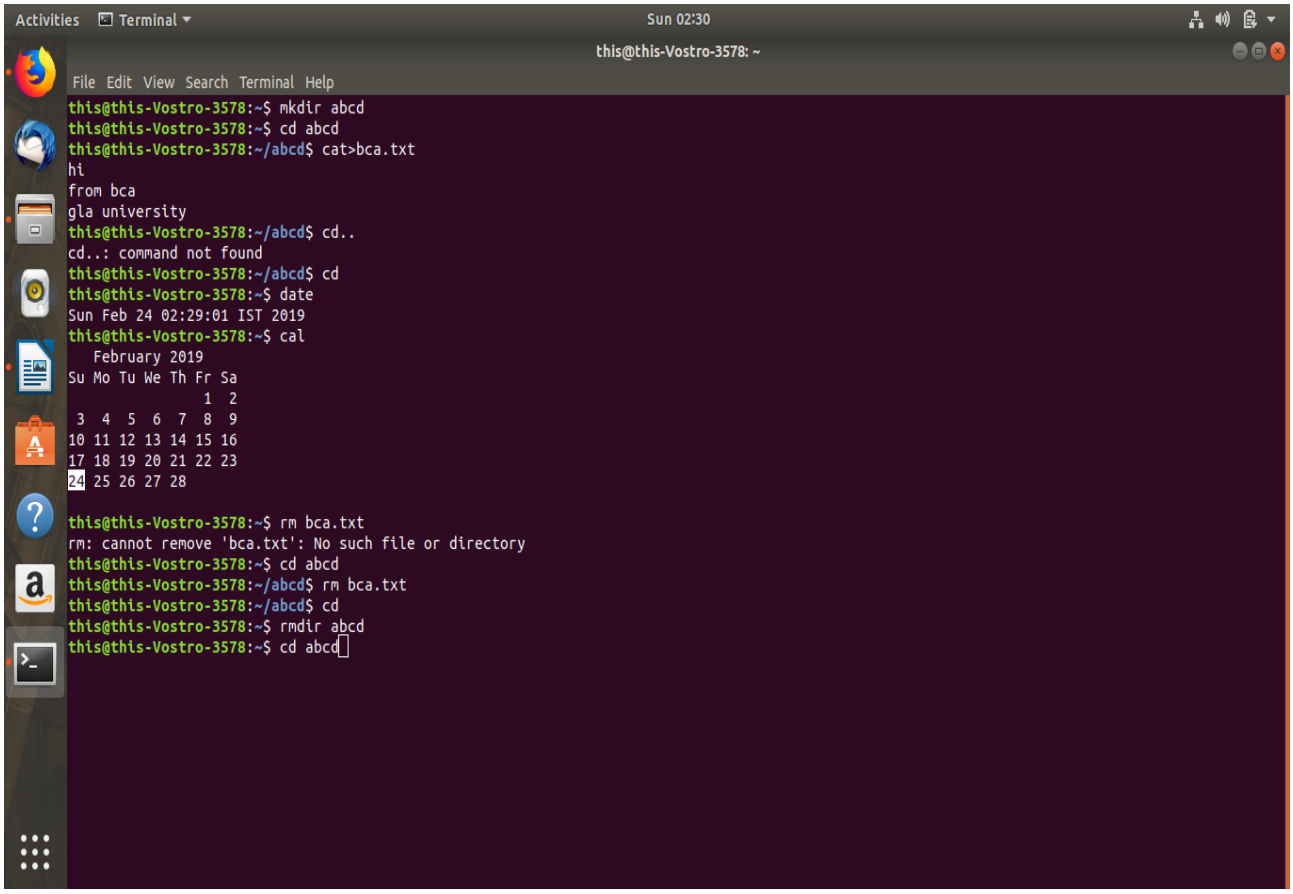
DESCRIPTION-
rm - remove files or directories
SYNTAX-

- rm <file_name>
- rm <directory_name>

9. rmdir

DESCRIPTION-
rmdir - remove empty directories
SYNTAX-

- rmdir <directory_name>



10. head

DESCRIPTION-
head - output the first part of files
SYNTAX-

- head <file_name>

11. tail

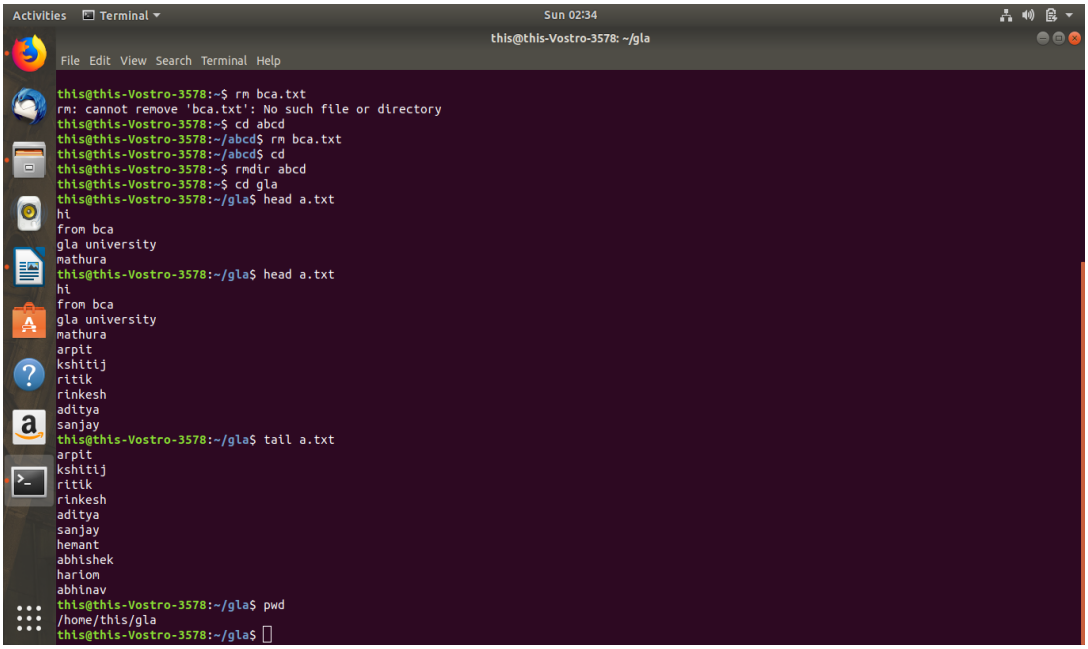
DESCRIPTION-
tail - output the last part of files
SYNTAX-

- tail <file_name>

12. pwd

DESCRIPTION-
pwd - print name of current/working directory
SYNTAX-

- pwd


Activities
Terminal
Sun 02:34
this@this-Vostro-3578: ~/gla
File Edit View Search Terminal Help
this@this-Vostro-3578:~$ rm bca.txt
rm: cannot remove 'bca.txt': No such file or directory
this@this-Vostro-3578:~$ cd abcd
this@this-Vostro-3578:~/abcd$ rm bca.txt
this@this-Vostro-3578:~/abcd$ cd
this@this-Vostro-3578:~$ rmdir abcd
this@this-Vostro-3578:~$ cd gla
this@this-Vostro-3578:~/gla$ head a.txt
hi
from bca
gla university
mathura
this@this-Vostro-3578:~/gla$ head a.txt
hi
from bca
gla university
mathura
arpit
kshitij
ritik
rinkesh
aditya
sanjay
this@this-Vostro-3578:~/gla$ tail a.txt
arpit
kshitij
ritik
rinkesh
aditya
sanjay
hemant
abhishek
hariom
abhinav
this@this-Vostro-3578:~/gla$ pwd
/home/this/gla
this@this-Vostro-3578:~/gla$

EXPERIMENT NO: 4

Environment: Ubuntu

Objective: Write a shell script to find whether a number is even or odd.

Layout:

```
harsh@Ubuntu:~/Desktop/lab3$ sh evenodd.sh
Hello user, Enter a number to check whether it is odd or even: 55
55 is odd
harsh@Ubuntu:~/Desktop/lab3$ sh evenodd.sh
Hello user, Enter a number to check whether it is odd or even: 44
44 is even
```

Procedure:

```
#!/bin/bash

read -p "Hello user, Enter a number to check whether it is odd or even: " num
if [ "$(expr $num % 2)" -eq 0 ]
then
        echo "$num is even"
else
        echo "$num is odd"
fi
```

EXPERIMENT NO: 5

Environment: Ubuntu

Objective: Write a shell script to check if the number entered at the command line is prime or not.

Layout:

```
harsh@Ubuntu:~/Desktop/lab3$ sh prime.sh
Enter a number: 12
12 is composite (not prime)
harsh@Ubuntu:~/Desktop/lab3$ sh prime.sh
Enter a number: 13
13 is Prime
```

Procedure:

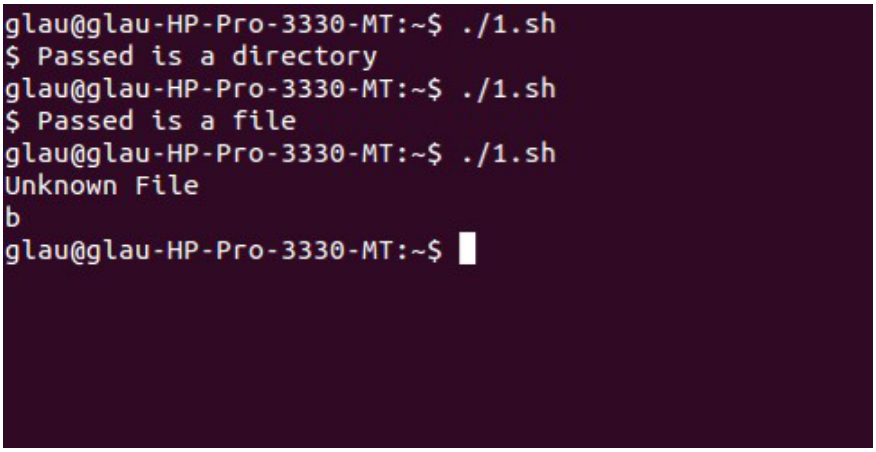
```
#!/bin/bash
read -p "Enter a number: " num
i=2
f=0
while [ $i -le `expr $num / 2` ]
do
if [ `expr $num % $i` -eq 0 ]
then
f=1
fi
i=`expr $i + 1`
done
if [ $f -eq 1 ]
then
echo "$num is composite (not prime)"
else
echo "$num is Prime"
fi
```

EXPERIMENT NO: 6

Environment: Uuntu

Objective: Write a shell script to input the name of a file as command line argument and display whether it is a file, a directory or anything else.

Layout:



Procedure:

```
#!/bin/bash

Passed=$"abc.txt"
if [ -d "${Passed}" ]
then
echo "$ Passed is a directory";
else
if [ -f "${Passed}" ]
then
echo "$ Passed is a file";
else
echo "Unknown File"
```

```
echo $Passed
exit $[Passed]
fi
fi
```

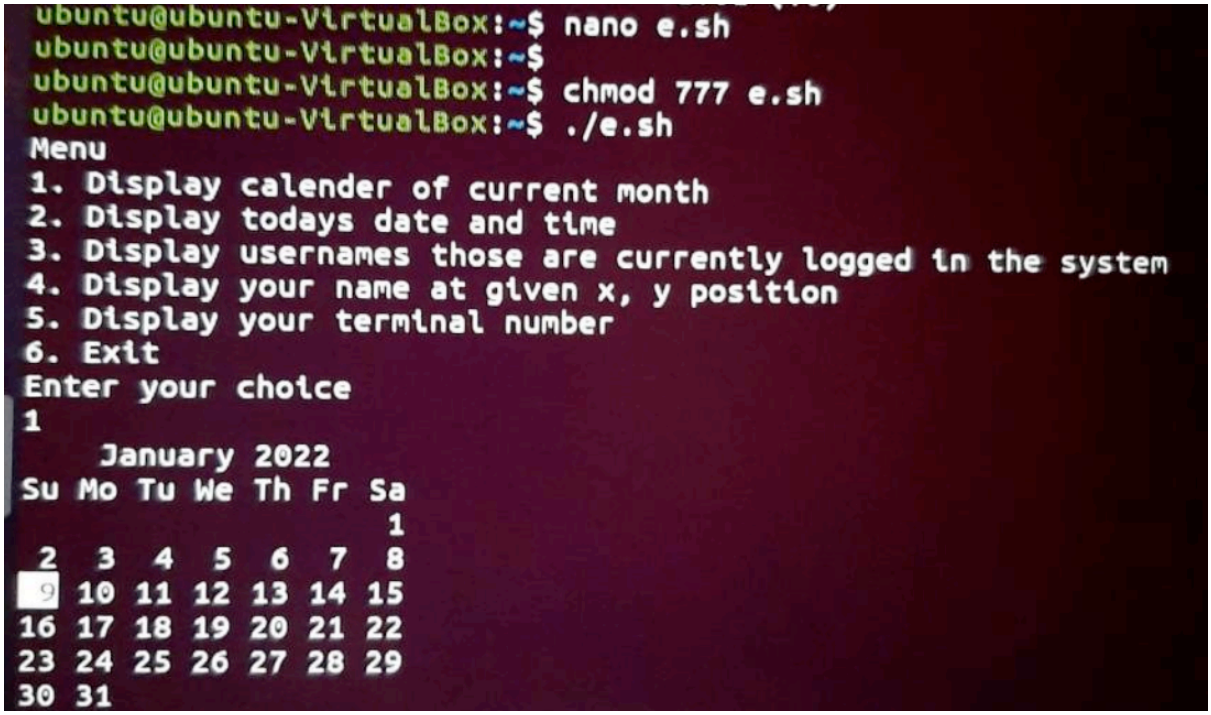
EXPERIMENT NO: 7

Environment: Ubuntu

Objective: Write a menu driven shell script, which will print the following menu and execute the given task.

- Display a calendar of current month
- Display today's date and time
- Display username those are currently logged in the system
- Display your name at the given x,y position.
- Display your terminal number.

Layout:



Procedure:

```
#!/bin/bash
echo "Menu"
echo "1. Display calender of current month " echo "2. Display
todays date and time"
echo "3. Display usernames those are currently logged in the system"
echo "4. Display your name at given x, y position" echo "5. Display your
terminal number"
echo "6. Exit"
echo "Enter your choice" read c
case $c in
1) cal;;2) date;;
3) who;;
4) clear
echo "Enter x, y position"
read x
read y
tput cup $x $y
whoami;;
```