RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

CS23431 **OPERATING SYSTEMS LAB**
Laboratory Observation NoteBook

Name : VISHWAK S
Year/Branch/Section : II/CSE/D
Register No. : 230701385
Semester : IV
Academic Year: 2024-25

INDEX

Ex. No. : 1a **Date : 25.01.2025**

Register No. : 230701385 **Name : VISHWAK S**

INSTALLATION AND CONFIGURATION OF LINUX

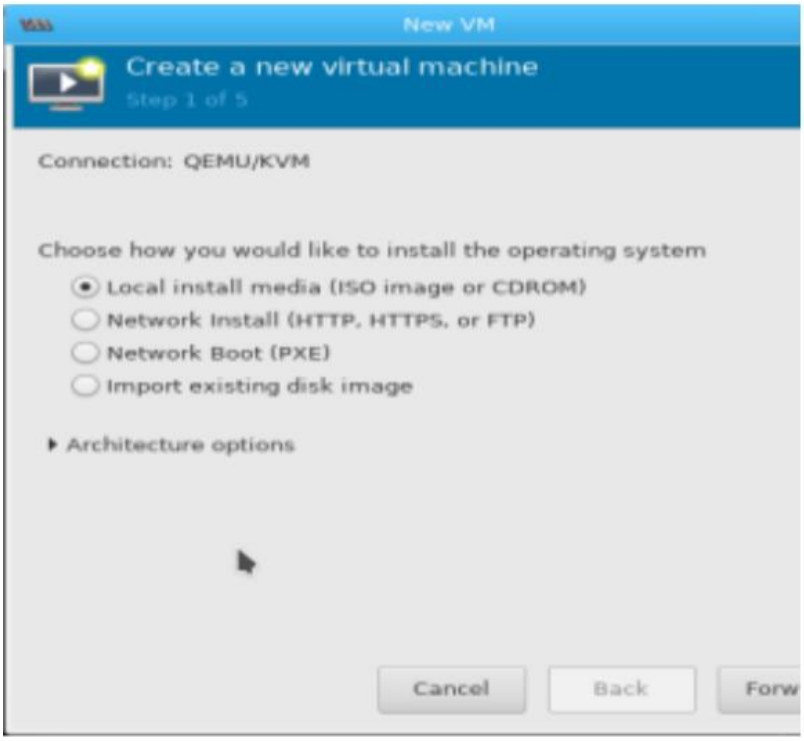
Aim:

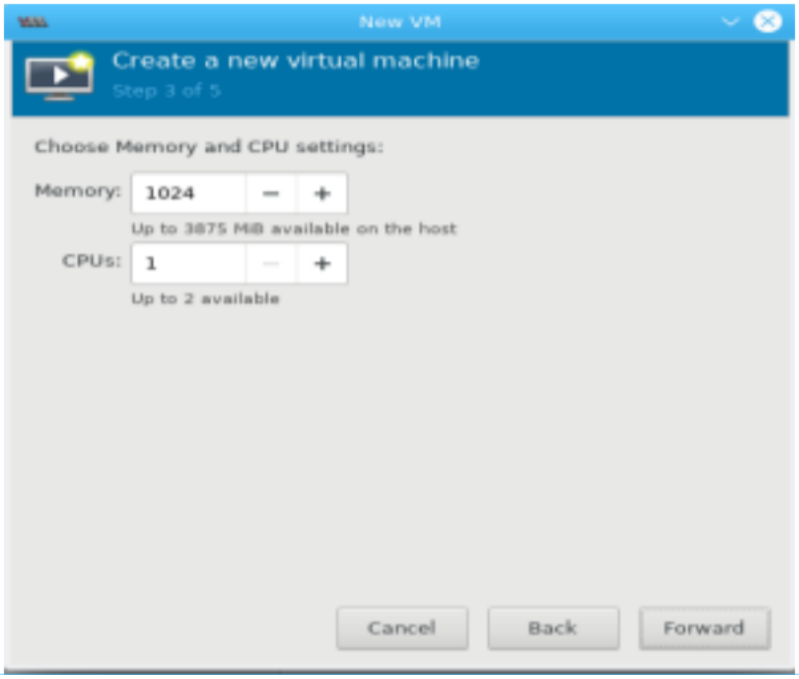
To install and configure Linux operating systems in a Virtual Machine.

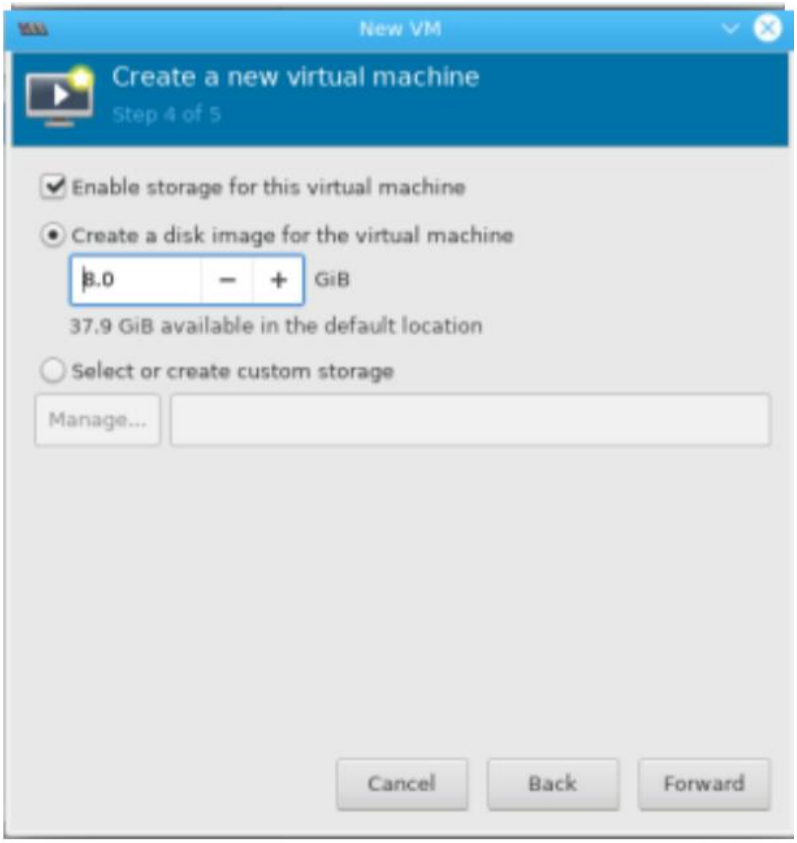
Installation/Configuration Steps:

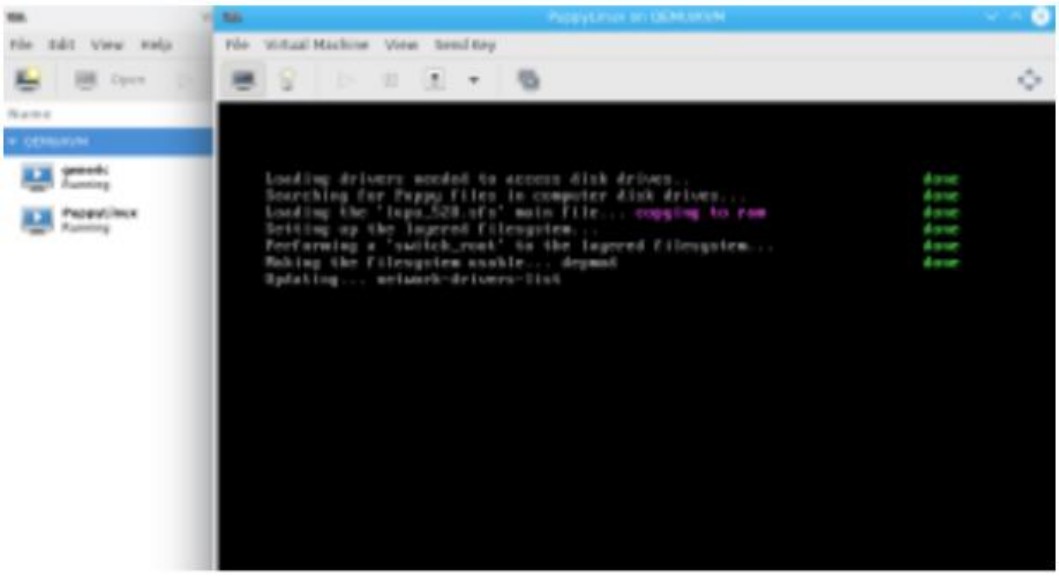
1. Install the required packages for virtualization
   dnf install xen virt-manager qemu libvirt
2. Configure xend to startup on boot
   systemctl enable virt-manager.service
3. Reboot the machine.
4. Create a virtual machine by first running virt-manager.
5. Click on File and then click to connect to localhost.
6. In the base menu, right click on to localhost(QEMU) to create new VM
7. Select Linux IOS image.
8. Choose puppy-linux ios and then kernel version
9. Select CPU and RAM limits
10. Create default disk to 8GB
11. Click create for the new VM with PuppyLinux

Output:


New VM
Create a new virtual machine
Step 1 of 5
Connection: QEMU/KVM
Choose how you would like to install the operating system
Local install media (ISO image or CDROM)
Network Install (HTTP, HTTPS, or FTP)
Network Boot (PXE)
Import existing disk image
Architecture options
Cancel
Back
Forw


New VM
Create a new virtual machine
Step 3 of 5
Choose Memory and CPU settings:
Memory:
1024
Up to 3875 MiB available on the host
CPUs:
1
Up to 2 available
Cancel
Back
Forward





Result:

Hence the Linux installation has been studied successfully.

Ex. No. : 1b **Date : 01.02.2025**

Register No. : 230701385 **Name : VISHWAK S**

BASIC LINUX COMMANDS

1.1 GENERAL PURPOSE COMMANDS

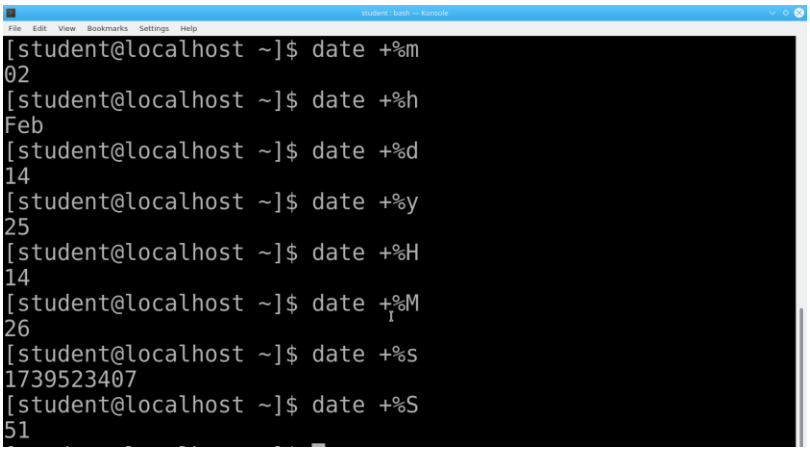
1. The 'date' command:

The date command displays the current date with day of week, month, day, time (24 hours clock) and the year.

SYNTAX: $ date

The date command can also be used with the following format:

Format	Purpose	Example
+ %m	To display only month	$ date + %m
+ %h	To display month name	$ date + %h
+ %d	To display day of month	$ date + %d
+ %y	To display last two digits of the year	$ date + %y
+ %H	To display Hours	$ date + %H
+ %M	To display Minutes	$ date + %M
+ %S	To display Seconds	$ date + %S



2. The echo’command:

The echo command is used to print the message on the screen. SYNTAX: $ echo

EXAMPLE: $ echo “God is Great”

3. The ‘cal’ command:

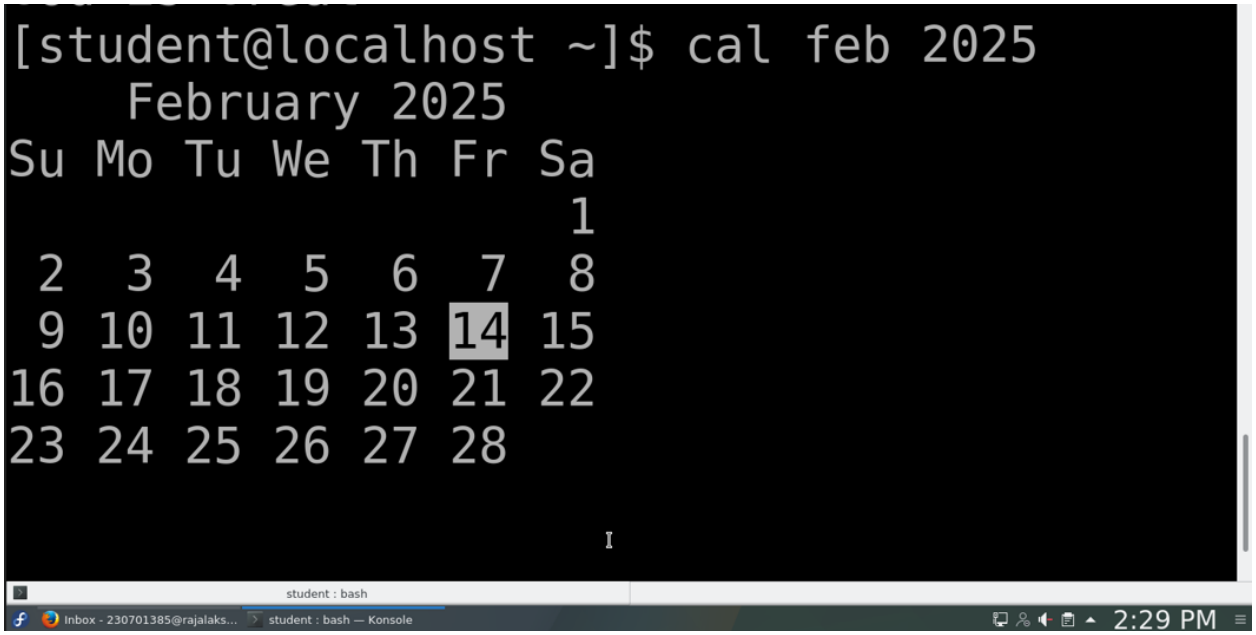
The cal command displays the specified month or year calendar.

SYNTAX: $ cal

[month] [year]

EXAMPLE: $ cal Jan

2012



4. The ‘bc’ command:

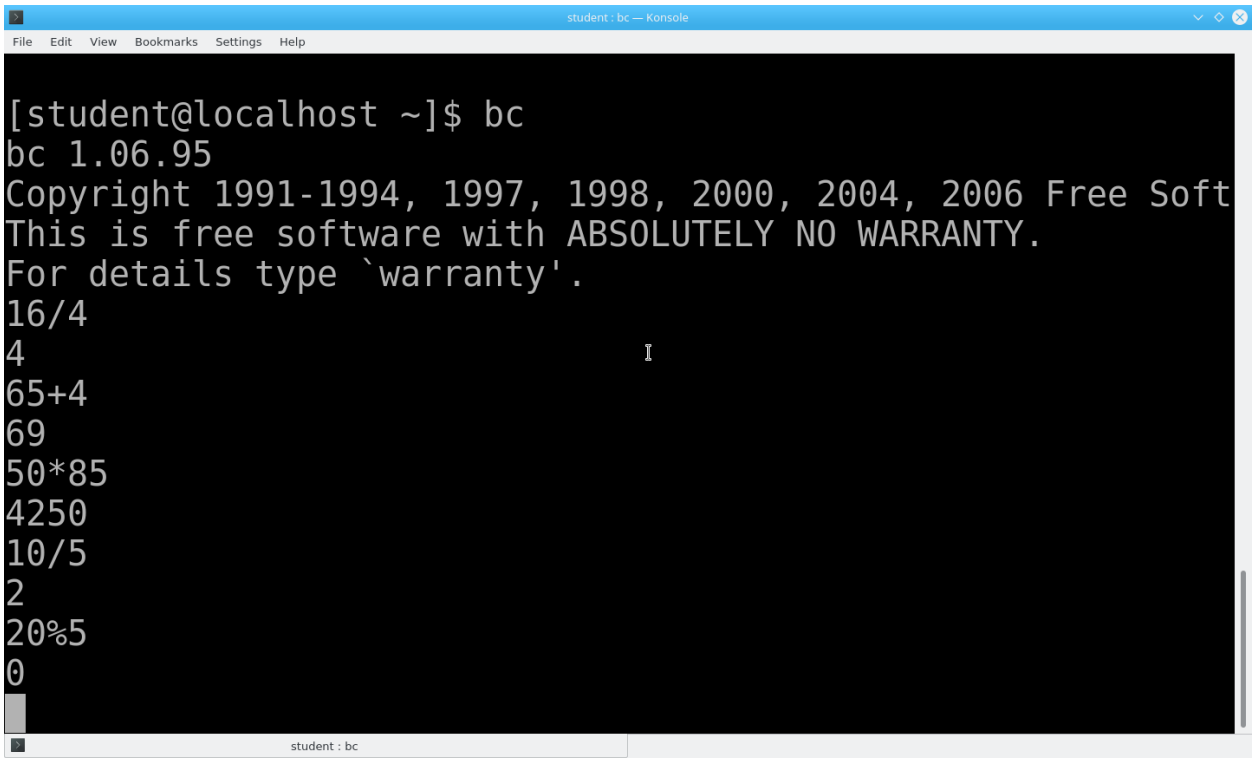
Unix offers an online calculator and can be invoked by the command bc.

SYNTAX: $ bc

EXAMPLE: bc –l

16/4

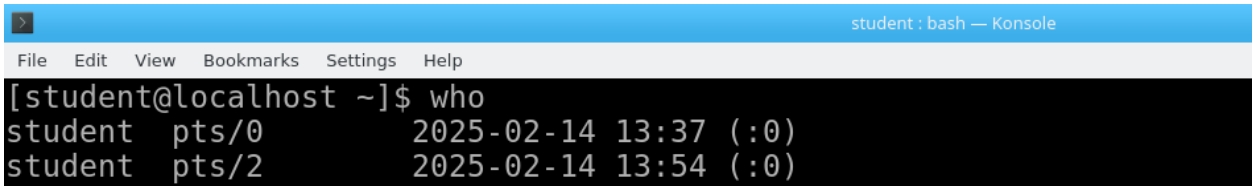
5/2



5. The ‘who’ command

The who command is used to display the data about all the users who are currently logged into the system.

SYNTAX: $ who



6. The ‘who am i’ command

The who am i command displays data about login details of the user.

SYNTAX: $ who am i

```
[student@localhost ~]$ who am i
student  pts/2        2025-02-14 13:54 (:0)
```

7. The ‘id’ command

The id command displays the numerical value corresponding to your login.

SYNTAX: $ id

```
[student@localhost ~]$ id
uid=1000(student) gid=1000(student) groups=1000(student) context=unconfined_u:unconfined_r:unconfined
_t:s0-s0:c0.c1023
```

8. The ‘tty’ command

The tty (teletype) command is used to know the terminal name that we are using.

SYNTAX: $ tty

```
[student@localhost ~]$ tty
/dev/pts/2
```

9. The ‘ps’ command

The ps command is used to the process currently alive in the machine with the 'ps' (process status) command, which displays information about process that are alive when you run the command. 'ps;' produces a snapshot of machine activity.

SYNTAX: $ ps

EXAMPLE: $ ps $ ps –e $ps -aux

```
[student@localhost ~]$ ps
  PID TTY          TIME CMD
 1711 pts/2    00:00:00 bash
 2004 pts/2    00:00:00 ps
```

10. The ‘uname’ command

The uname command is used to display relevant details about the operating system on the standard output.

-m -> Displays the machine id (i.e., name of the system hardware)

-n -> Displays the name of the network node. (host name)

-r -> Displays the release number of the operating system.

-s -> Displays the name of the operating system (i.e.. system name)

-v -> Displays the version of the operating system.

-a -> Displays the details of all the above five options.

SYNTAX: $ uname
[option] EXAMPLE:
$ uname -a

```
[student@localhost ~]$ uname
Linux
```

1.2 DIRECTORY COMMANDS

1. The ‘pwd’ command:

The pwd (print working directory) command displays the current working directory.

SYNTAX: $ pwd

2. The ‘mkdir’ command:

The mkdir is used to create an empty directory in a disk.

SYNTAX: $ mkdir
dirname

EXAMPLE: $ mkdir receee

3. The ‘rmdir’ command:

The rmdir is used to remove a directory from the disk. Before removing a directory, the directory must be empty (no files and directories).

SYNTAX: $ rmdir dirname

EXAMPLE: $ rmdir receee

4. The ‘cd’ command:

The cd command is used to move from one directory to another.

SYNTAX: $ cd dirname

EXAMPLE: $ cd receee

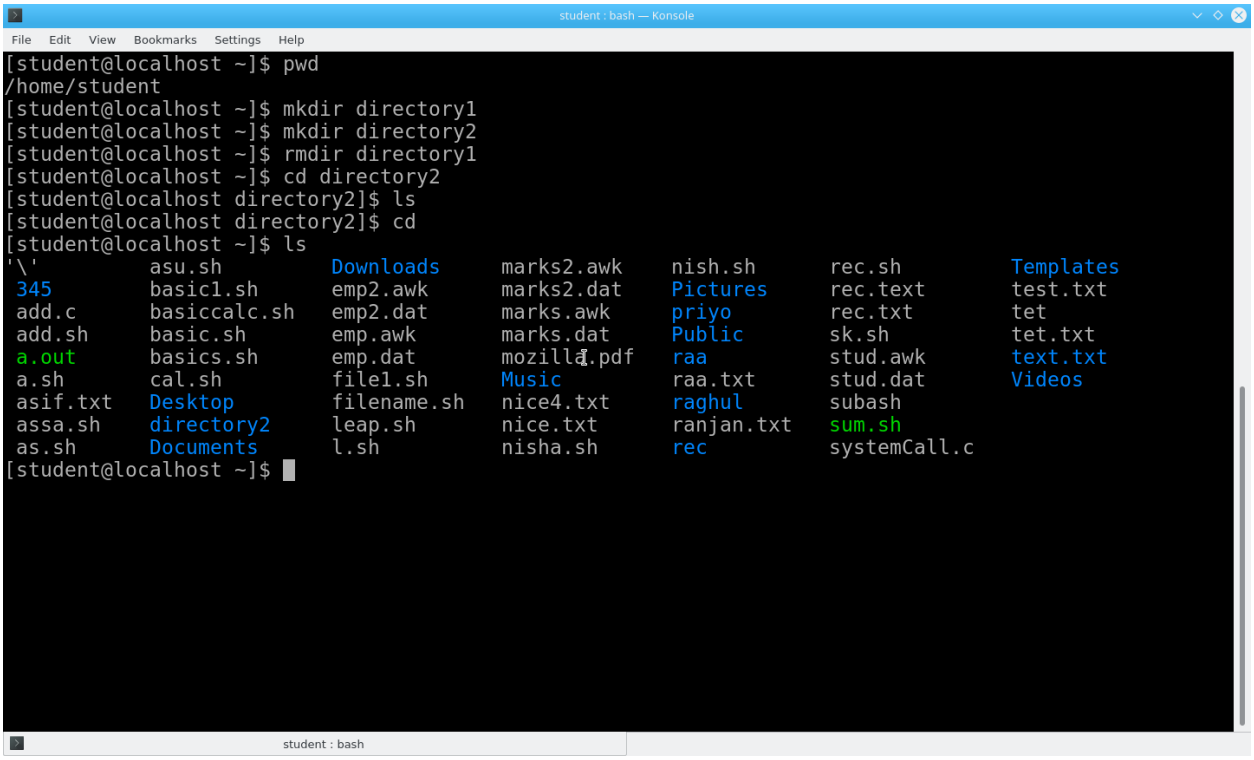
5. The ‘ls’ command:

The ls command displays the list of files in the current working directory.

SYNTAX: $ ls EXAMPLE: $ ls

$ ls –l

$ ls –a



1.3 FILE HANDLING COMMANDS

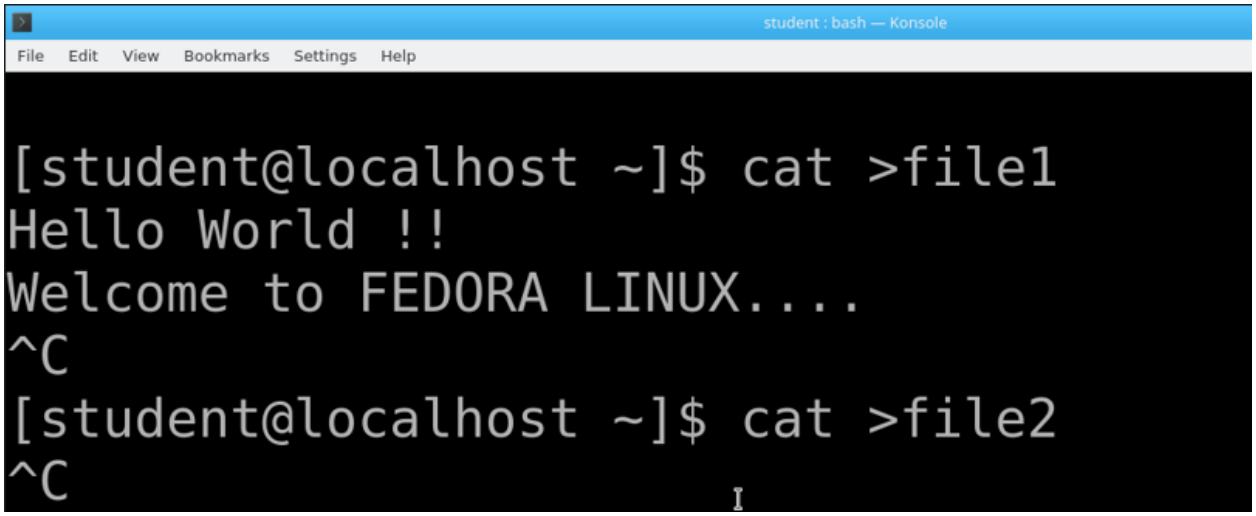
1. The ‘cat’ command:

The cat command is used to create a file.

SYNTAX: $ cat >

filename EXAMPLE:

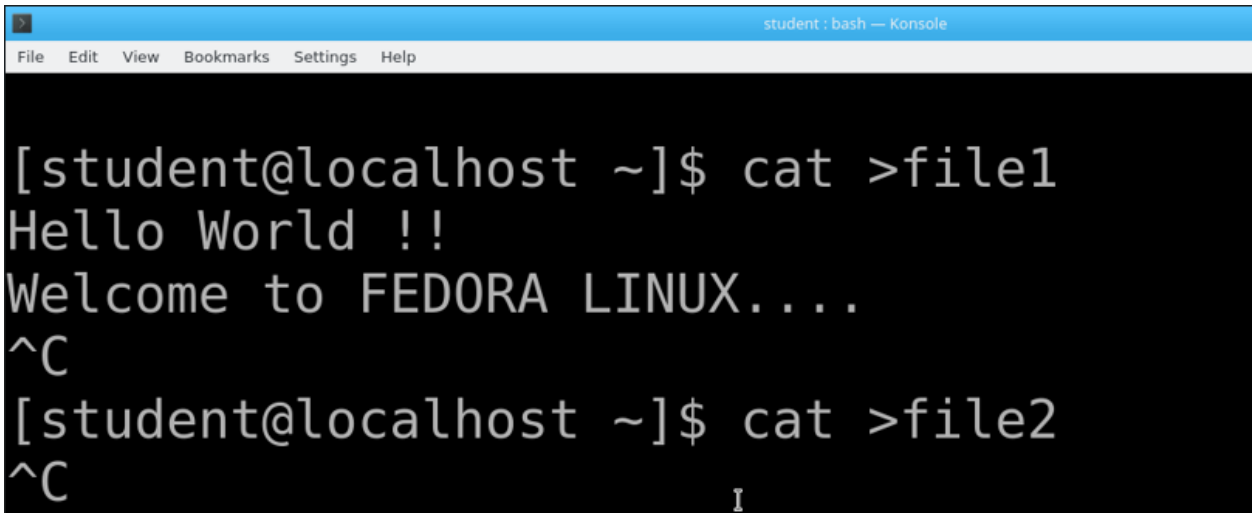
$ cat > rec



2. The ‘Display contents of a file’ command:

The cat command is also used to view the contents of a specified file.

SYNTAX: $ cat filename



3. The ‘cp’ command:

The cp command is used to copy the contents of one file to another and copies the file from one place to another.

SYNTAX: $ cp oldfile

newfile EXAMPLE: $

cp cse ece

```
[student@localhost ~]$ cp file1 file2
[student@localhost ~]$ cat file2
Hello World !!
Welcome to FEDORA LINUX....
```

4. The ‘rm’ command:

The rm command is used to remove or erase an existing file

SYNTAX: $ rm filename

EXAMPLE: $ rm rec

$ rm –f rec

Use option –fr to delete recursively the contents of the directory and its subdirectories.

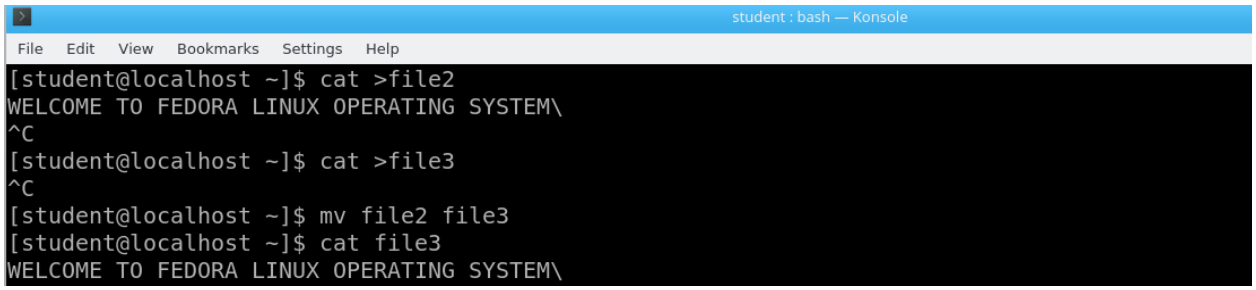
```
[student@localhost ~]$ rm file1
[student@localhost ~]$ cat file1
cat: file1: No such file or directory
```

5. The ‘mv’ command:

The mv command is used to move a file from one place to another. It removes a specified file from its original location and places it in a specified location.

SYNTAX: $ mv oldfile newfile

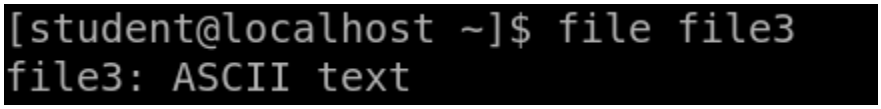
EXAMPLE: $ mv cse eee



6. The ‘file’ command:

The file command is used to determine the type of file.

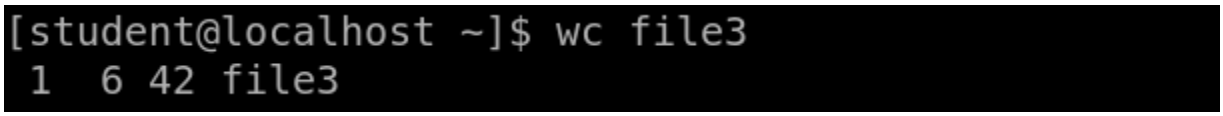
SYNTAX: $ file
filename EXAMPLE:
$ file receee



7. The ‘wc’ command:

The wc command is used to count the number of words, lines and characters in a file.

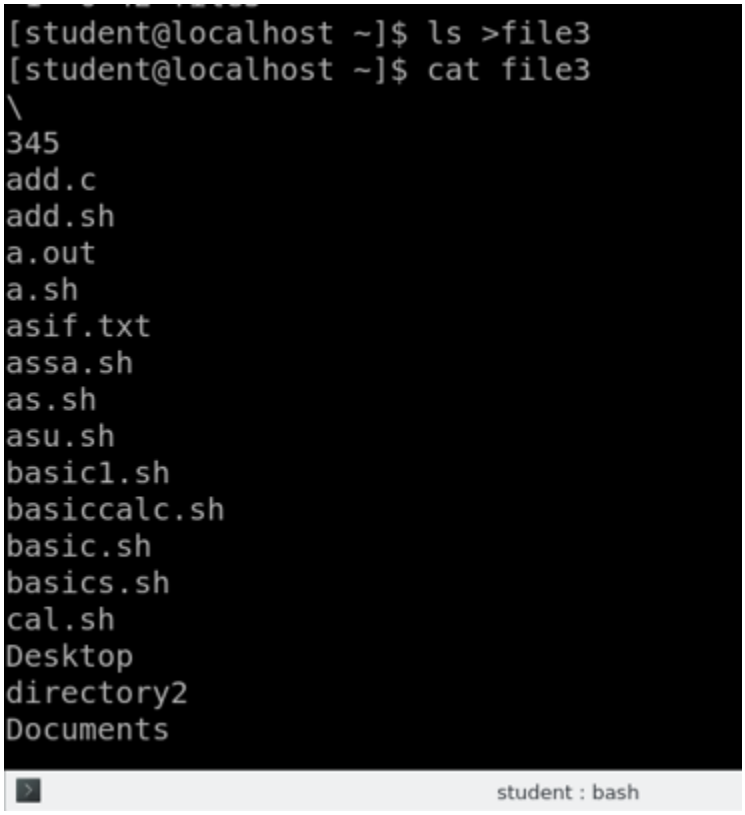
SYNTAX: $ wc
filename EXAMPLE:
$ wc receee



8. The ‘Directing output to a file’ command:

The ls command lists the files on the terminal (screen). Using the redirection operator ‘>’ we can send the output to the file instead of showing it on the screen.

SYNTAX: $ ls >
filename EXAMPLE:
$ ls > cseeee



9. The ‘pipes’ command:

The Unix allows us to connect two commands together using these pipes. A pipe (|) is a mechanism by which the output of one command can be channeled into the input of another command.

SYNTAX: $
command1 |
command2
EXAMPLE: $ who |
wc -l

10. The ‘tee’ command:

While using pipes, we have not seen any output from a command that gets piped into another command. To save the output, which is produced in the middle of a pipe, the tee command is very useful. SYNTAX: $ command | tee filename
EXAMPLE: $ who | tee sample | wc -l

11. The ‘Metacharacters of unix’ command:

Metacharacters are special characters that are at a higher and abstract level compared to most other characters in Unix. The shell understands and interprets these metacharacters in a special way.

* - Specifies number of characters

?- Specifies a single character

[ ]- used to match a whole set of file names at a command line.

! – Used to Specify Not

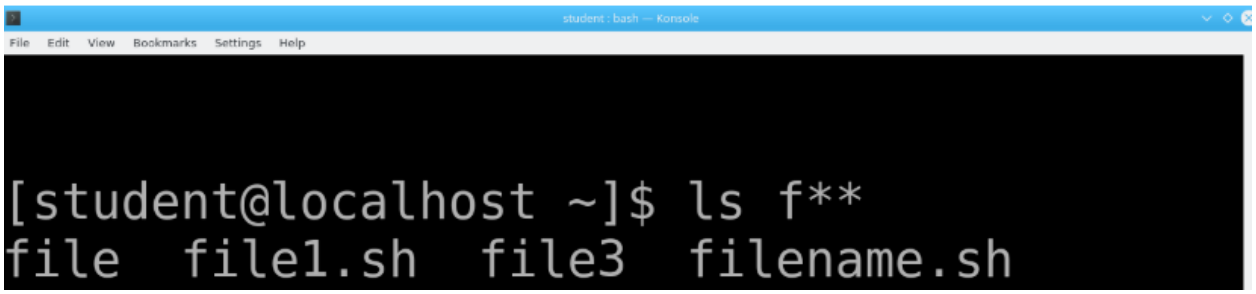
EXAMPLE:

$ ls r** - Displays all the files whose name begins with ‘r’

$ ls ?kkk - Displays the files which are having ‘kkk’, from the second character irrespective of the first character.

$ ls [a-m] – Lists the files whose names begins alphabets from ‘a’ to ‘m’

$ ls [!a-m] – Lists all files other than files whose names begin with alphabets from ‘a’ to ‘m’ 12.



The ‘File permissions’ command:

File permission is the way of controlling the accessibility of files for each of three users namely Users, Groups and Others.

There are three types of file permissions are available, they are

r-read

w-write

x-execute

The permissions for each file can be divided into three parts of three bits each:

First three bits	Owner of the file
Next three bits	Group to which owner of the file belongs
Last three bits	Others

EXAMPLE: $ ls college

-rwxr-xr-- 1 Lak std 1525 jan10 12:10 college

-rwx The file is readable, writable and executable by the owner of the file.

Lak Specifies Owner of the file.

r-x Indicates the absence of the write permission by the Group owner of the file. Std Is the Group Owner of the file.

r-- Indicates read permissions for others.

```
[student@localhost ~]$ ls >file3
[student@localhost ~]$ cat file3
\
345
add.c
add.sh
a.out
a.sh
asif.txt
assa.sh
as.sh
asu.sh
basic1.sh
basiccalc.sh
```

13. The ‘chmod’ command:

The chmod command is used to set the read, write and execute permissions for all categories of users for file.

SYNTAX: $ chmod category operation permission file

EXAMPLE:

$ chmod u –wx college

Removes write & execute permission for users for ‘college’ file.

$ chmod u +rw, g+rw college

Assigns read & write permission for users and groups for ‘college’ file.

$ chmod g=wx college

Assigns absolute permission for groups of all read, write and execute permissions for ‘college’ file.

```
[student@localhost ~]$ chmod u-wx file3
[student@localhost ~]$ chmod g+wr file3
[student@localhost ~]$ chmod 761 file3
[student@localhost ~]$
```

14. The ‘Octal Notations’ command:

The file permissions can be changed using octal notations also. The octal notations for file permission are

EXAMPLE:

$ chmod 761 college

Assigns all permission to the owner, read and write permissions to the group and only executable permission to the others for the ‘college’ file.

1.4 GROUPING COMMANDS

1. The ‘semicolon’ command:

The semicolon(;) command is used to separate multiple commands at the command line.

SYNTAX:

$command1;command2;command3..... ;commandn

EXAMPLE: $ who;date

2. The ‘&&’ operator:

The ‘&&’ operator signifies the logical AND operation in between two or more valid Unix commands.It means that only if the first command is successfully executed, then the next command will executed.

SYNTAX:

$command && command && command3..... && commandn

EXAMPLE: $ who && date

3. The ‘||’ operator:

The ‘||’ operator signifies the logical OR operation in between two or more valid Unix commands.It means, that only if the first command will happen to be un successfully,it will continue to execute next commands.

SYNTAX:

$command1||command||command3..... ||commandn

EXAMPLE: $ who || date

```
student : bash — Konsole
File  Edit  View  Bookmarks  Settings  Help
[student@localhost ~]$ who && date
student  pts/0        2025-03-08 13:37 (:0)
student  pts/1        2025-03-08 13:38 (:0)
Sat Mar  8 13:59:56 IST 2025
[student@localhost ~]$ who am i; date +%m
student  pts/1        2025-03-08 13:38 (:0)
03
[student@localhost ~]$ date || who
Sat Mar  8 14:00:21 IST 2025
[student@localhost ~]$
```

1.5 FILTERS

1. The head filter

It displays the first ten lines of a file.

SYNTAX: $ head filename

EXAMPLE: $ head college Display the top ten lines.

$ head -5 college Display the top five lines.

```
student : bash — Konsole
File  Edit  View  Bookmarks  Settings  Help
[student@localhost ~]$ head company
TCS
Tech Mahindra
Infosys
IBM
Google
Microsoft
Oracle
Accenture
SAP
Cisco
```

2. The tail filter

It displays ten lines of a file from the end of the file.

SYNTAX: $ tail filename

EXAMPLE: $ tail college Display the last ten lines.

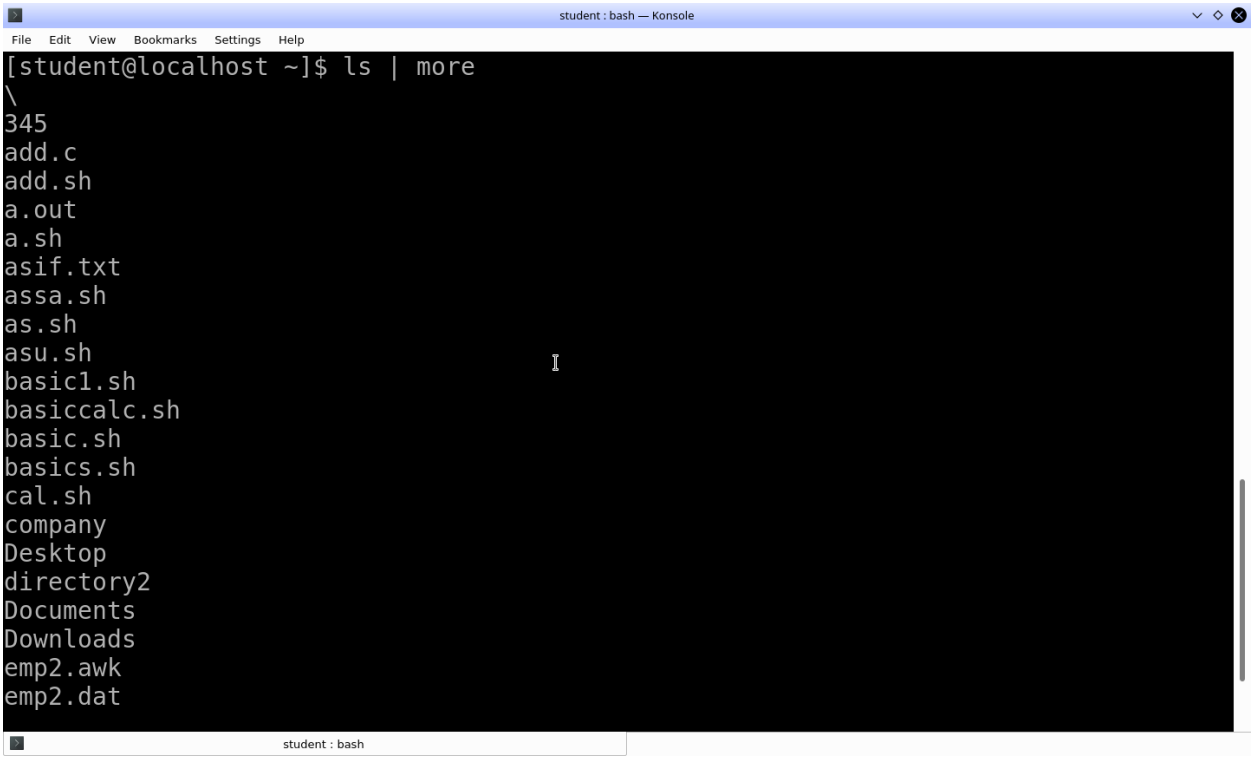
$tail -5 college Display the last five lines.

```
[student@localhost ~]$ tail company
Microsoft
Oracle
Accenture
SAP
Cisco
Adobe
Wipro
Zoho
Amazaon
Walmart
```

3. The more filter:

The pg command shows the file page by page.

SYNTAX: $ ls –l | more



4. The ‘grep’ command:

This command is used to search for a particular pattern from a file or from the standard input and display those lines on the standard output. “Grep” stands for “global search for regular expression.”

SYNTAX: $ grep [pattern] [file_name]

EXAMPLE: $ cat> student

Arun cse

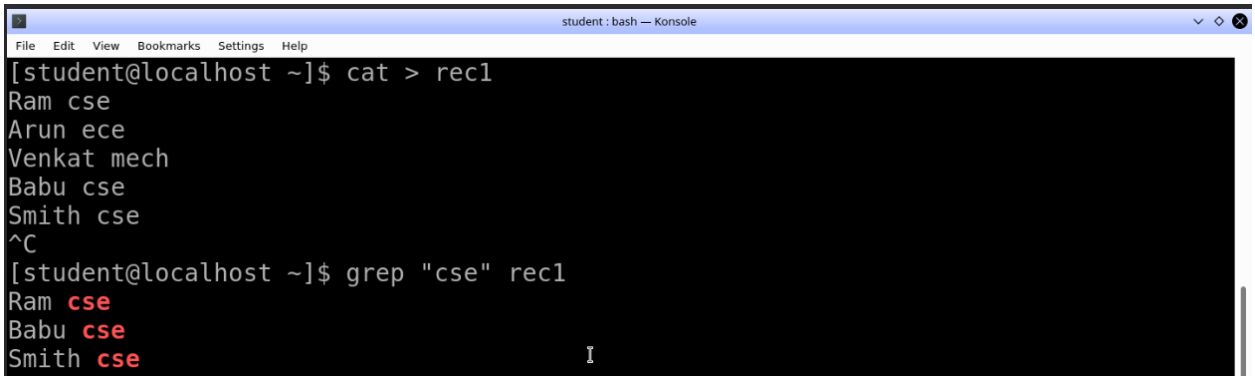
Ram ece

Kani cse

$ grep “cse” student

Arun cse

Kani cse



5. The ‘sort’ command:

The sort command is used to sort the contents of a file. The sort command reports only to the screen, the actual file remains unchanged.

SYNTAX: $ sort filename

```
[student@localhost ~]$ sort company
Accenture
Adobe
Amazaon
Cisco
Google
IBM
Infosys
Microsoft
Oracle
SAP
TCS
```

student : bash

6. The ‘nl’ command:

The nl filter adds line numbers to a file and it displays the file and not provides a to edit but simply displays the contents on the screen.

SYNTAX: $ nl filename

EXAMPLE: $ nl college

student : bash — Kon

File Edit View Bookmarks Settings Help

```
[student@localhost ~]$ nl company
     1  TCS
     2  Tech Mahindra
     3  Infosys
     4  IBM
     5  Google
     6  Microsoft
     7  Oracle
     8  Accenture
     9  SAP
    10  Cisco
    11  Adobe
    12  Wipro
    13  Zoho
    14  Amazaon
    15  Walmart
```

7. The ‘cut’ command:

We can select specific fields from a line of text using the cut command.

SYNTAX: $ cut -c filename

EXAMPLE: $ cut -c college

OPTION:

-c – Option cut on the specified character position from each line.

```
[student@localhost ~]$ cut -c -3 company
TCS
Tec
Inf
IBM
Goo
Mic
Ora
Acc
SAP
Cis
Ado
Wip
Zoh
Ama
Wal
```

1.6 OTHER ESSENTIAL COMMANDS

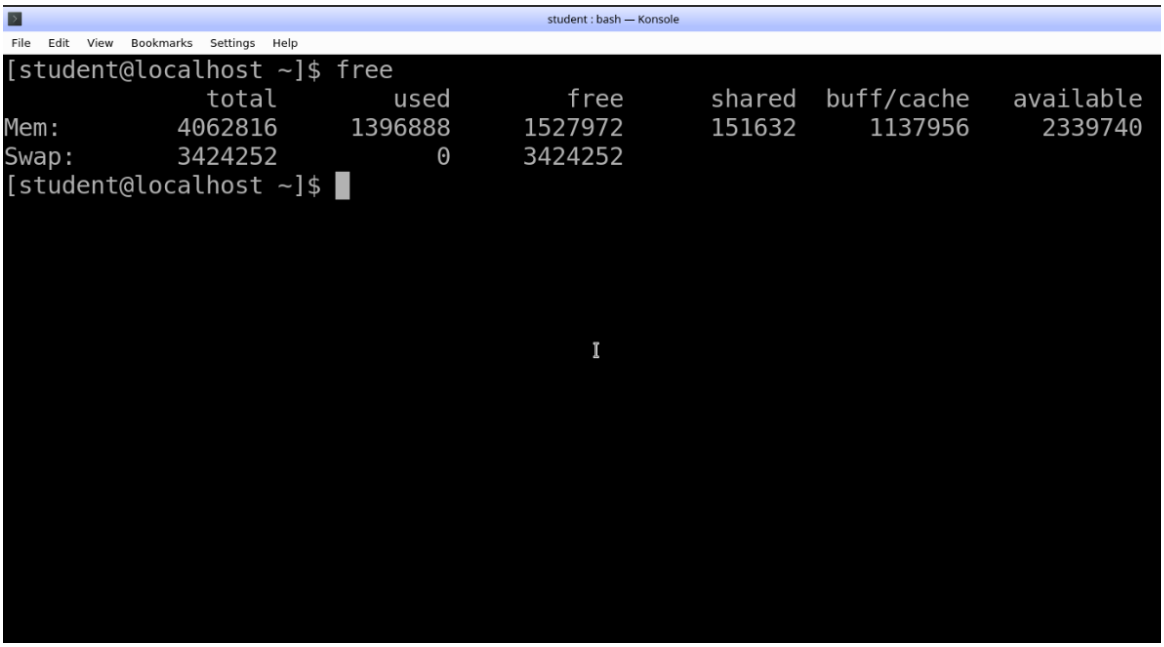
1. **free**

   Display amount of free and used physical and swapped memory system. syno

   free [options]

   <u>example</u>

   [root@localhost ~]# free -t



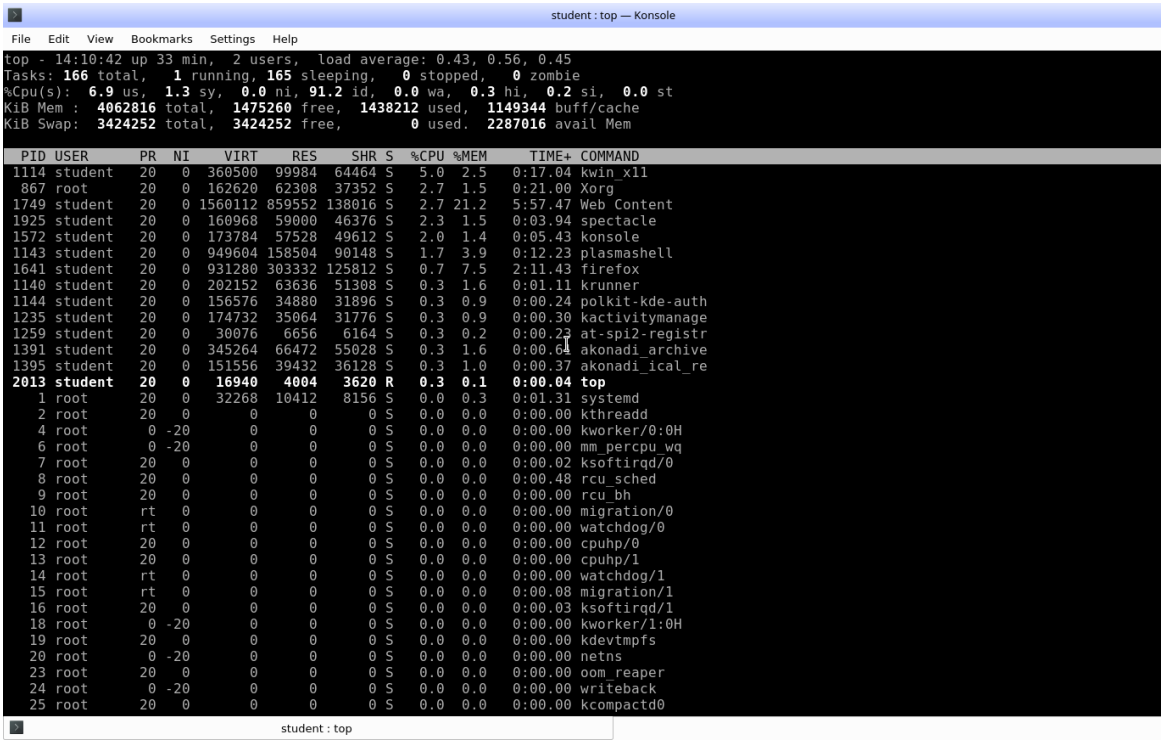
2. **top**

It provides a dynamic real-time view of processes in the system.

synopsis- top [options]

<u>example</u>

[root@localhost ~]# top



3. **ps**

It reports the snapshot of current processes.

synopsis- ps [options]

<u>example</u>

[root@localhost ~]# ps -e

```
student : bash — Konsole
File  Edit  View  Bookmarks  Settings  Help
[student@localhost ~]$ ps
  PID TTY          TIME CMD
 1576 pts/1    00:00:00 bash
 2018 pts/1    00:00:00 ps
```

4. **vmstat**

It reports virtual memory statistics.

synopsis- vmstat [options] <u>example</u>

[root@localhost ~]# vmstat

```
[student@localhost ~]$ vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 1481292  76780 1082632    0    0   194    66  630  961 14  2 84  0  0
```

5. **df**

It displays the amount of disk space available in file-system.

Synopsis- df [options]

<u>example</u> [root@localhost ~]# df

```
[student@localhost ~]$ df
Filesystem              1K-blocks    Used Available Use% Mounted on
devtmpfs                  2020420       0   2020420   0% /dev
tmpfs                     2031408   41176   1990232   3% /dev/shm
tmpfs                     2031408    1208   2030200   1% /run
tmpfs                     2031408       0   2031408   0% /sys/fs/cgroup
/dev/mapper/fedora-root  44186572 5164500  36747768  13% /
tmpfs                     2031408      12   2031396   1% /tmp
/dev/sda6                  999320  151196    779312  17% /boot
/dev/mapper/fedora-home  21567312  830380  19618316   5% /home
tmpfs                      406280      20    406260   1% /run/user/1000
```

6**. ping**

It is used to verify that a device can communicate with another on the networl

PING stands for Packet Internet Groper. synopsis- ping [options]

[root@localhost ~]# ping 172.16.4.1

```
[student@localhost ~]$ ping
Usage: ping [-aAbBdDfhLnOqrRUvV64] [-c count] [-i interval] [-I interface]
            [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos
            [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
            [-w deadline] [-W timeout] [hop1 ...] destination
Usage: ping -6 [-aAbBdDfhLnOqrRUvV] [-c count] [-i interval] [-I interface]
             [-l preload] [-m mark] [-M pmtudisc_option]
             [-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize]
             [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline]
             [-W timeout] destination
```

7**. ifconfig**

It is used to configure network interfaces.

synopsis- ifconfig [options]
Example: [root@localhost ~]# ifconfig

```
[student@localhost ~]$ ifconfig
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.9.2  netmask 255.255.252.0  broadcast 172.16.11.255
        inet6 fe80::ed62:ec73:ec9f:3ad9  prefixlen 64  scopeid 0x20<link>
        ether 00:27:0e:13:ea:36  txqueuelen 1000  (Ethernet)
        RX packets 181394  bytes 112770604 (107.5 MiB)
        RX errors 0  dropped 43  overruns 0  frame 0
        TX packets 39784  bytes 21829577 (20.8 MiB)
        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
```

student : bash

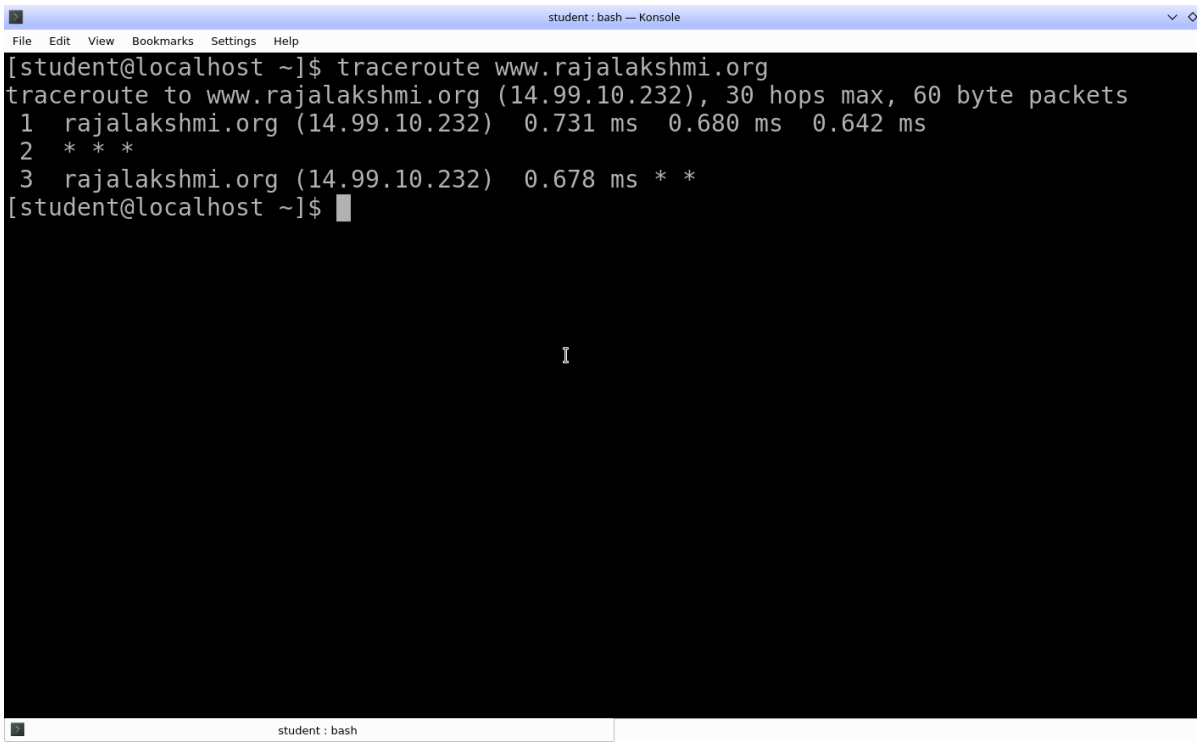
8.**traceroute**

It tracks the route the packet takes to reach the destination.

synopsis- traceroute [options]

Example

[root@localhost ~]# traceroute www.rajalakshmi.org



Result:

Hence the basic LINUX commands, directory commands, file handling commands, grouping commands, filters and other essential commands have been studied and executed successfully.