

S.Y.B.Sc. I.T.

Sem-III

LAB-MANUAL

OPERATING SYSTEM



“Special thanks to Nishant Biswal, Satish Prajapati, Akash Gupta, Soham Rane and Prince Chauhan for their co-operation to compile this document.”



Compiled by:
Asst. Prof. Megha Sharma

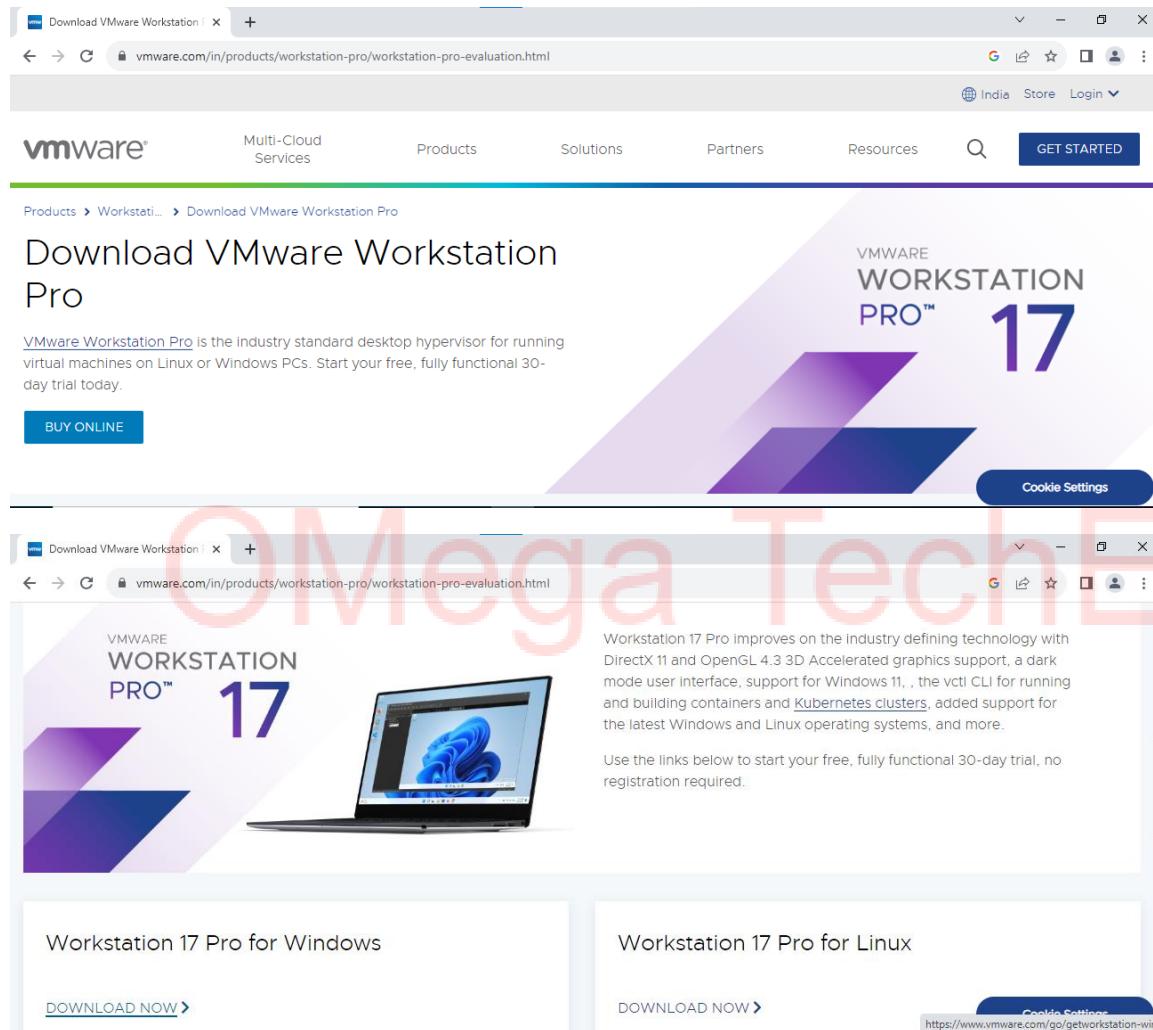
Practical 1

1. Installation and Configuration of virtual machine

Aim: Installation of **Virtual machine** software and **Linux OS** on Virtual Machine.

Procedure:

Step 1: Download the Virtual machine software from the Internet.



VMware Workstation Pro 17

VMware Workstation Pro 17 improves on the industry defining technology with DirectX 11 and OpenGL 4.3 3D Accelerated graphics support, a dark mode user interface, support for Windows 11, the vcli CLI for running and building containers and Kubernetes clusters, added support for the latest Windows and Linux operating systems, and more.

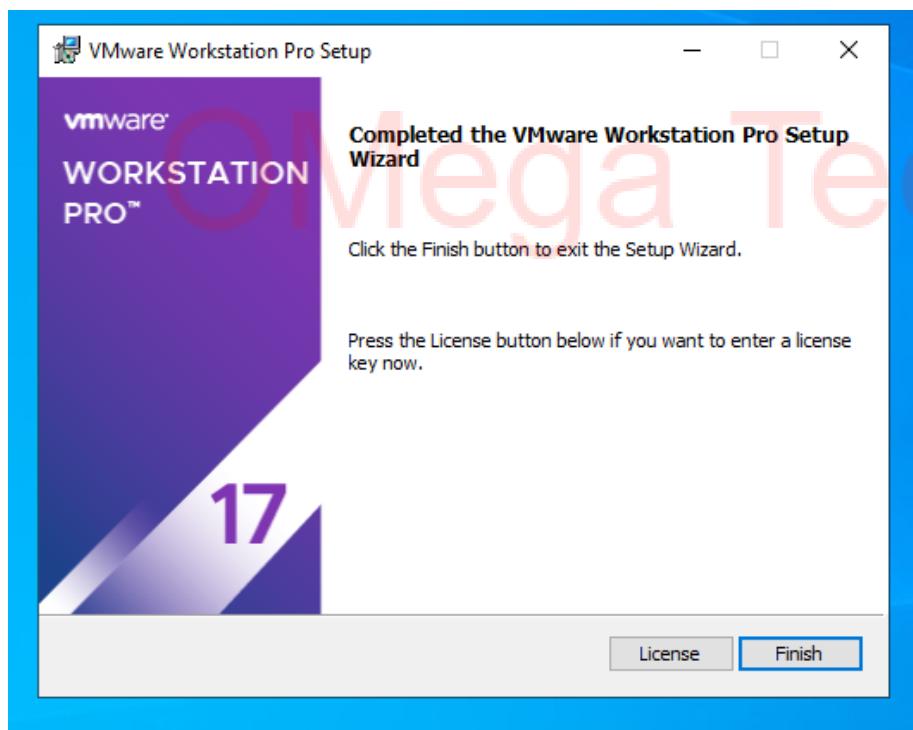
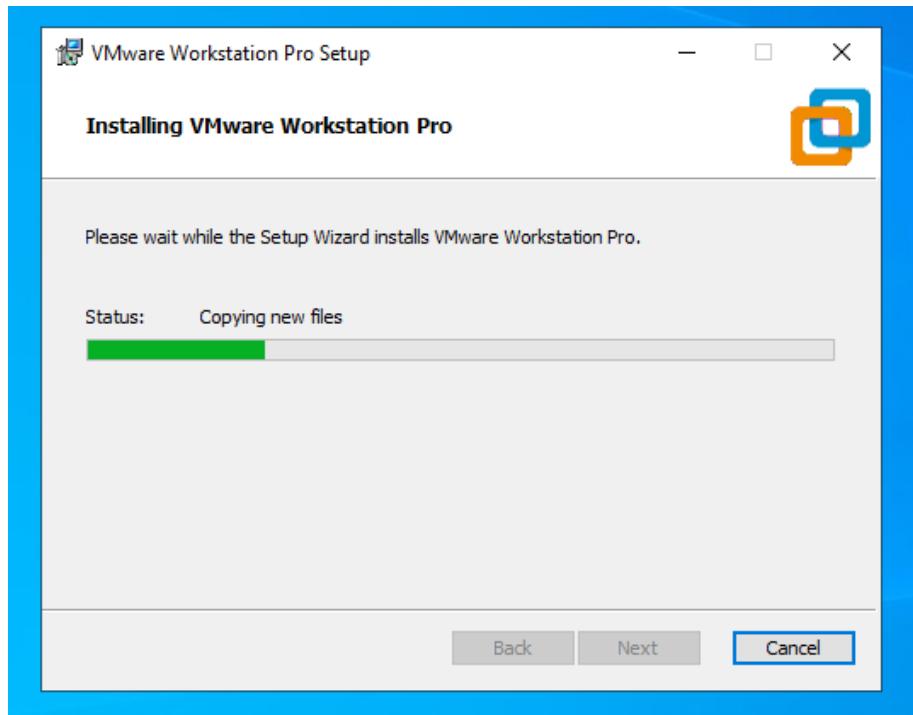
Workstation 17 Pro for Windows

Workstation 17 Pro for Linux

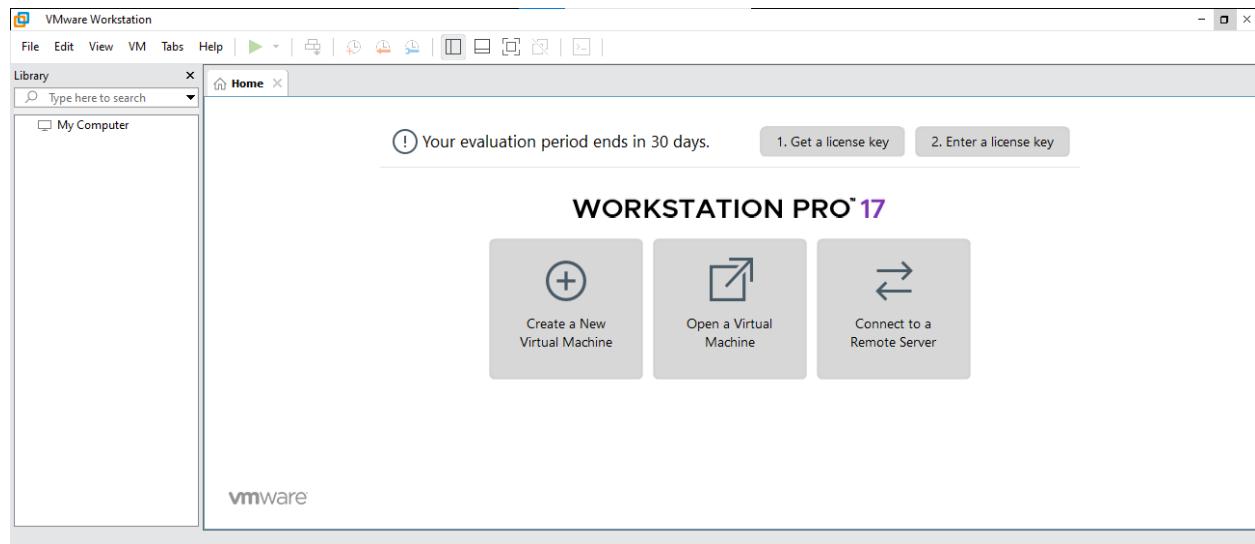
[DOWNLOAD NOW >](#)

[DOWNLOAD NOW >](#)

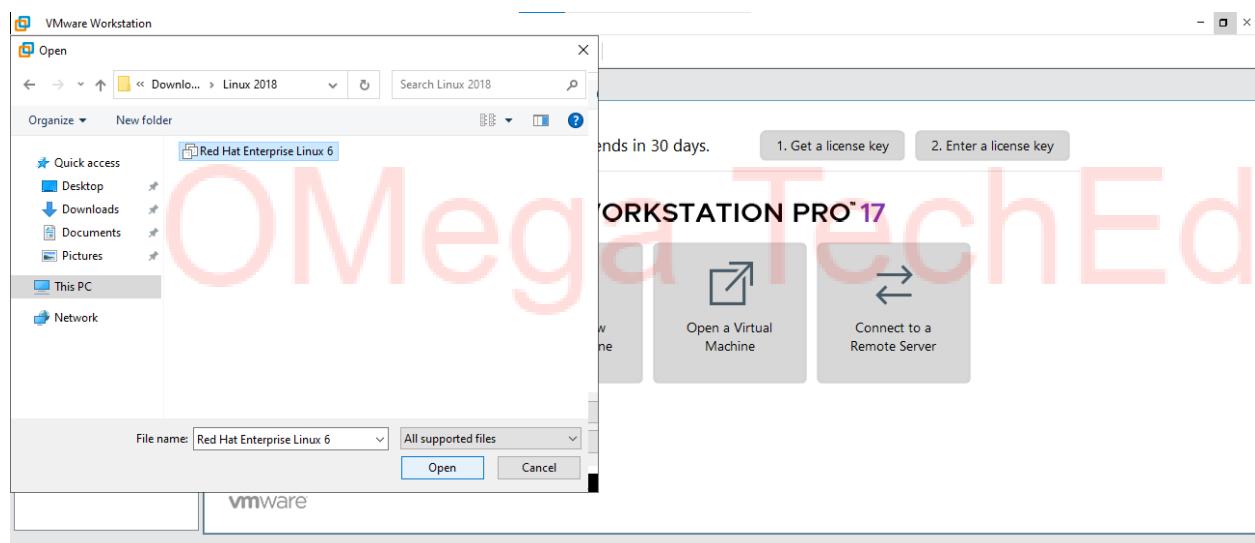
Step 2: Run VMware installation setup.



Step 3: Open virtual machine software installed on your operating system.

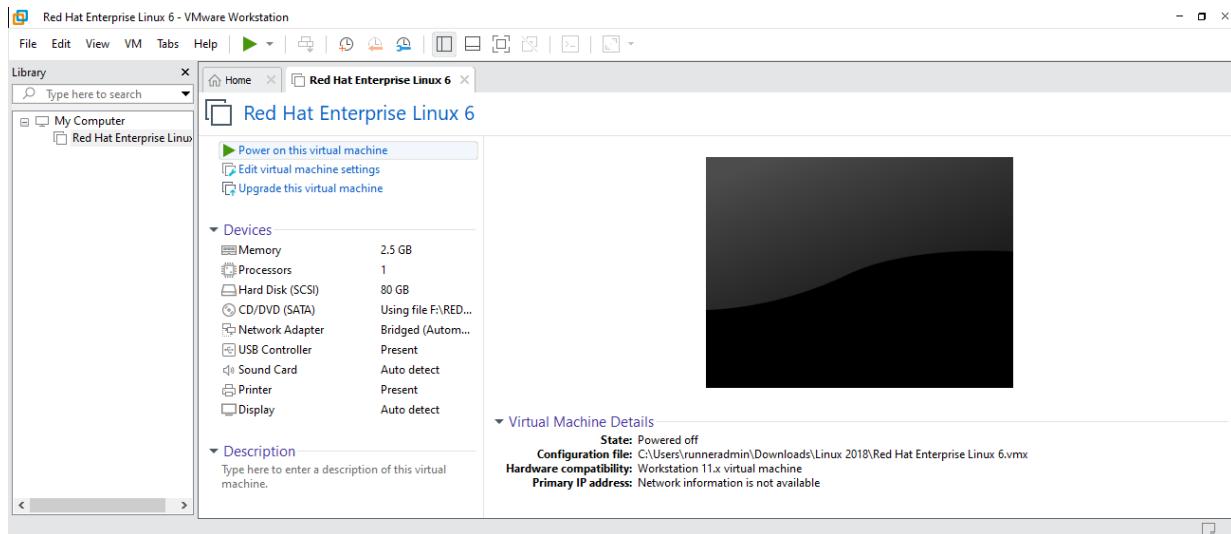


Step 4: Click on (Open a Virtual Machine) option in VMware software.

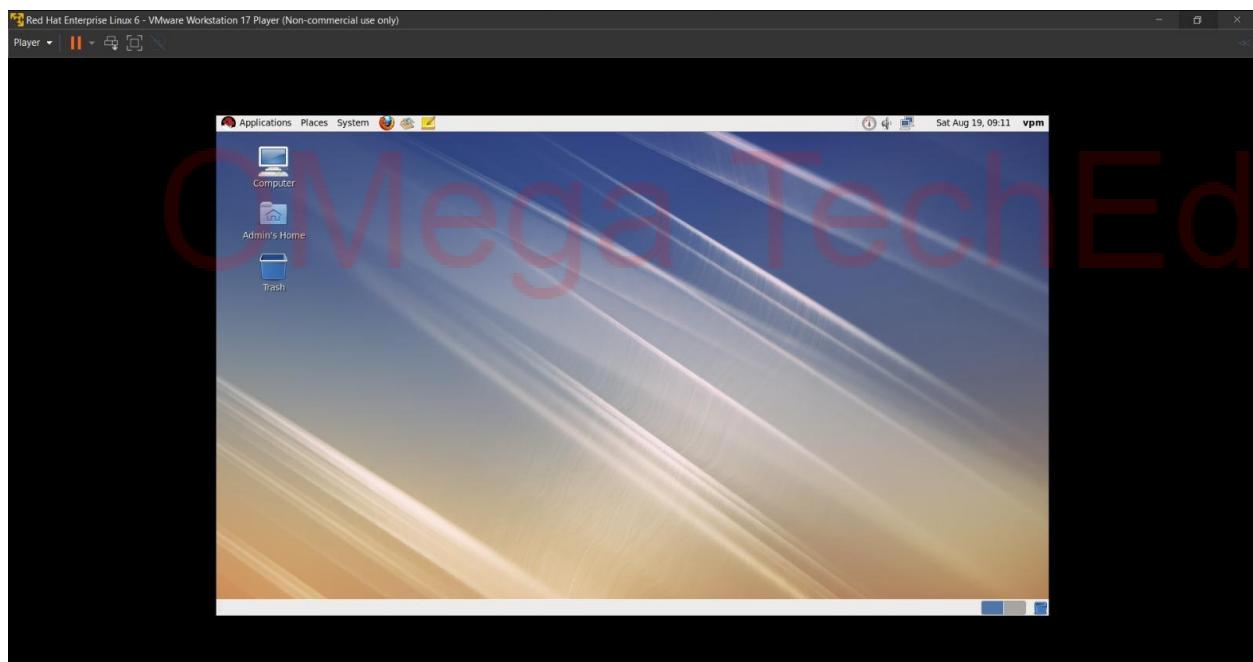


SYIT SEM-III OPERATING SYSTEM BY: MEGHA SHARMA

Step 5: Final step is to click on install and wait for the installation process to be finish.



Now our Red Hat Enterprise Linux 6 Operating system is installed.



Practical 2**2. Windows (DOS) Commands****2a Date, time, prompt**

Date: The date command can view or change the current date of the system clock.

Syntax: c:\> date

Time: The time command can view or change the current time of the system.

Syntax: c:\> date

Prompt: The prompt command allows us to change the MS-DOS prompt to display information.

Syntax: PROMPT[Text]

Where Text specifies a new command prompt. The prompt can be made up of normal characters and the below special code.

\$A	& Ampersand	\$H	Backspace
\$B	Pipe	\$L	< less than sign
\$C	(Left parenthesis	\$N	Current drive
\$D	Current Date	\$P	Current drive and path
\$E	Escape code	\$Q	= Equal sign
\$F) Right Parenthesis	\$S	space
\$G	> Greater than sign	\$T	Current time
\$V	Window version number	\$_	Carriage returns and linefeed

```
C:\>date
The current date is: 06-07-2023
Enter the new date: (dd-mm-yy)

C:\>time
The current time is: 2:49:58.58
Enter the new time:

C:\>prompt $d
06-07-2023prompt
C:\>prompt $t
2:50:26.88prompt
C:\>
```

2b) md, cd, rd, path, copy, xcopy

md: Allows us to create directories in MS-DOS.

Syntax: md [drive] path <directory name>

Example: C:\>md abc

cd: CD Change directory is a command used to switch directories in MS-DOS and the windows command line.

CD\ goes to the root of the drive.

CD.. goes back one directory.

Syntax: CD[D] [Drive:] [path]

Example: C:\Users\Omega>cd\

C:>

Example: C:\Users\Omega>cd..

C:\Users>

rd/rmdir: Removes an empty directory in MS-DOS.

Syntax: rd [drive:] path <directory name>

Example: C:\> rd test

Path: Path is used to specify the location where MS-DOS looks when using a command. Type path without parameters to display the current path.

Syntax: path [drive:]path[;...]

Path: used to clear all search path settings and directs windows to search only in current directory.

```
C:\>md abc
C:\>cd abc
C:\abc>md newdic
C:\abc>dir
Volume in drive C is Windows
Volume Serial Number is 3018-0C3D

Directory of C:\abc

06-07-2023 02:51    <DIR>      .
06-07-2023 02:51    <DIR>      newdic
      0 File(s)           0 bytes
      2 Dir(s)  100,545,101,824 bytes free

C:\abc>rd newdic
C:\abc>dir
Volume in drive C is Windows
Volume Serial Number is 3018-0C3D

Directory of C:\abc

06-07-2023 02:52    <DIR>      .
      0 File(s)           0 bytes
      1 Dir(s)  100,552,871,936 bytes free

C:\abc>path
PATH=C:\Program Files (x86)\VMware\Player\bin;C:\Program Files\Common Files\Oracle\Java\javapath;C:\oraclexe\app\oracle\product\11.2.0\server\bin;C:\windows\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\WindowsPowerShell\v1.0\;C:\windows\System32\OpenSSH\;C:\Program Files\nodejs\;C:\Program Files\Git\cmd;C:\Users\Akash Gupta\AppData\Local\Programs\Python\Python311\Scripts\;C:\Users\Akash Gupta\AppData\Local\Programs\Python\Python311\;C:\Users\Akash Gupta\AppData\Local\Microsoft\WindowsApps;C:\Users\Akash Gupta\AppData\Local\Programs\Microsoft VS Code\bin;C:\Users\Akash Gupta\AppData\Roaming\npm;C:\src\flutter\bin;

C:\abc>
```

Copy: Allows us to copy one or more files to an alternate location.

Syntax: copy [/D] [/V] [/N] [A|B] [+Source [/A|B] [destination[/A |B]]

Where:

Source	Specifies the file or files to be copied
/A	Indicates an ASCII text file.
/B	Indicates a binary file.
/D	Allows the destination file to be created decrypted.
destination	Specifies the directory or file name for the new file(s).
/V	Verifies the new files are written correctly.
/N	Uses short file name, if available when copying a file with a non-8dot3 name.

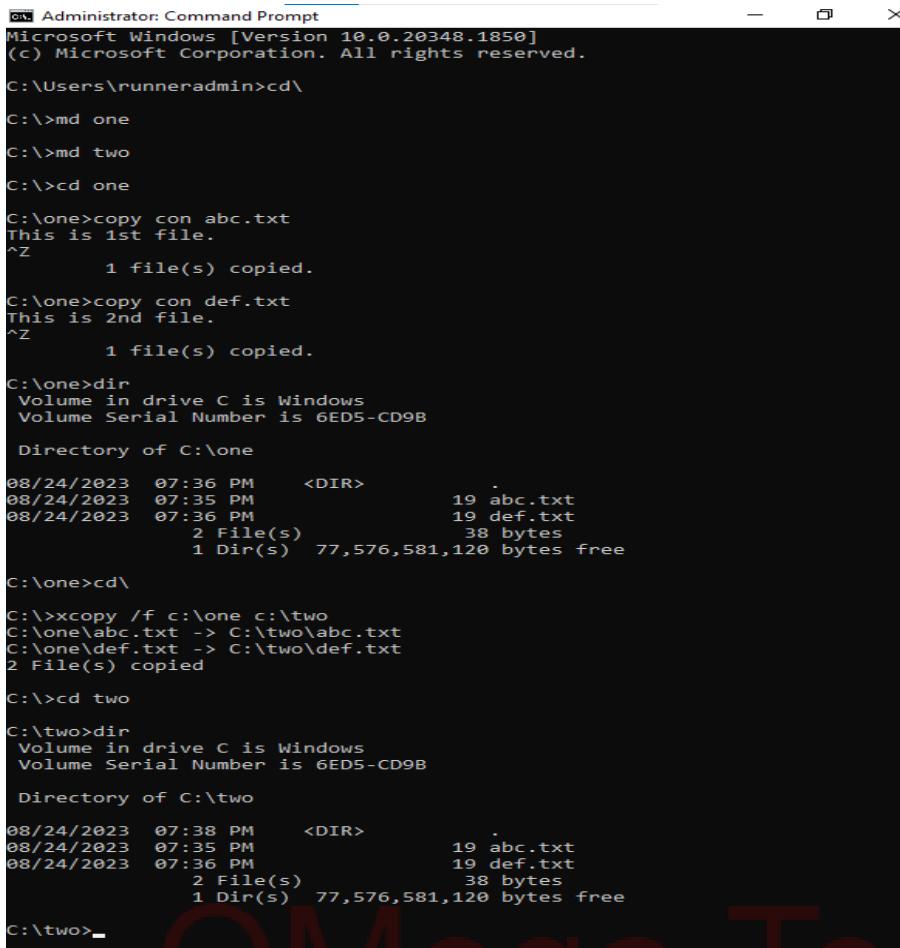
Example: C:\abc>copy*.txt D:\

Copy all text files from the current directory to the D:\ root directory.

Xcopy: xcopy is a powerful version of the copy command with additional features; has the capability of moving files, directories, and even whole drives from one location to another.

Example: C:\abc> xcopy *.* /h test

In the above example xcopy commands copy all files in the current directory including any hidden files into the test directory.



```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.20348.1850]
(c) Microsoft Corporation. All rights reserved.

c:\Users\runneradmin>cd\

c:>md one
c:>md two
c:>cd one
c:\one>copy con abc.txt
This is 1st file.
^Z
      1 file(s) copied.

c:\one>copy con def.txt
This is 2nd file.
^Z
      1 file(s) copied.

c:\one>dir
Volume in drive C is Windows
Volume Serial Number is 6ED5-CD9B

  Directory of C:\one

08/24/2023  07:36 PM    <DIR>
08/24/2023  07:35 PM           19 abc.txt
08/24/2023  07:36 PM           19 def.txt
               2 File(s)      38 bytes
               1 Dir(s)  77,576,581,120 bytes free

c:\one>cd\

c:>xcopy /f c:\one c:\two
c:\one\abc.txt -> C:\two\abc.txt
c:\one\def.txt -> C:\two\def.txt
2 File(s) copied

c:>cd two
c:\two>dir
Volume in drive C is Windows
Volume Serial Number is 6ED5-CD9B

  Directory of C:\two

08/24/2023  07:38 PM    <DIR>
08/24/2023  07:35 PM           19 abc.txt
08/24/2023  07:36 PM           19 def.txt
               2 File(s)      38 bytes
               1 Dir(s)  77,576,581,120 bytes free

c:\two>

```

2c) del, cls, move, echo.

del: del command is used to delete files from the computer.

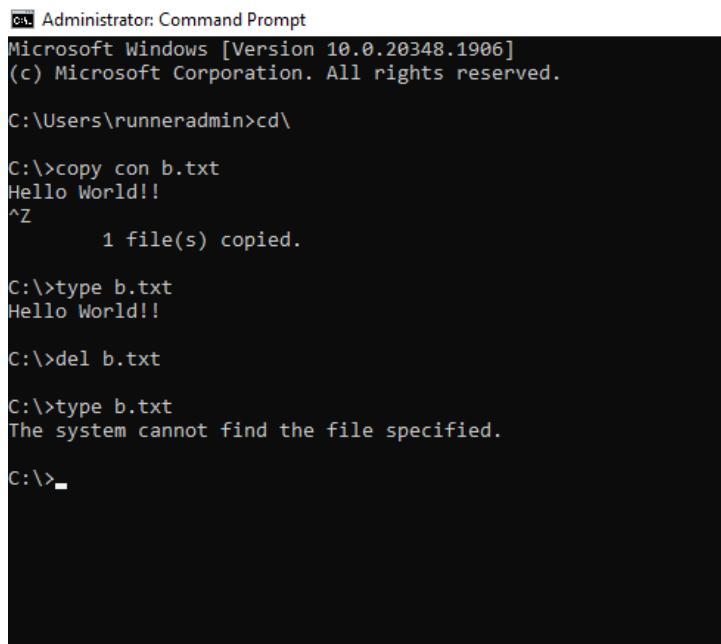
Syntax: del[/P] [/F] [/S] [/Q] [/A] [[: attributes]] names

Where

Names:	Specifies a list of one or more files or directories. Wildcard may be used to delete multiple files. If a directory is specified all files within the directory will be deleted.
/P	Prompts for confirmation before deleting each file.
/F	Force deleting a read only files.
/S	Delete specified files from all subdirectories.
/Q	Quite mode, do not ask if ok to delete on global wildcard.
/A	Select files to delete based on attributes.
Attributes:	R: Read only, S:System files, H:Hidden files

Example: C:\>del test.tmp

Delete the test.tmp in the current directory if the file exists.



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.20348.1906]
(c) Microsoft Corporation. All rights reserved.

C:\Users\runneradmin>cd\

C:\>copy con b.txt
Hello World!!
^Z
    1 file(s) copied.

C:\>type b.txt
Hello World!!

C:\>del b.txt
C:\>type b.txt
The system cannot find the file specified.

C:\>
```

cls: cls command is used to clear the screen.

move: Allows us to move files or directories from one folder to another or from one drive to another.

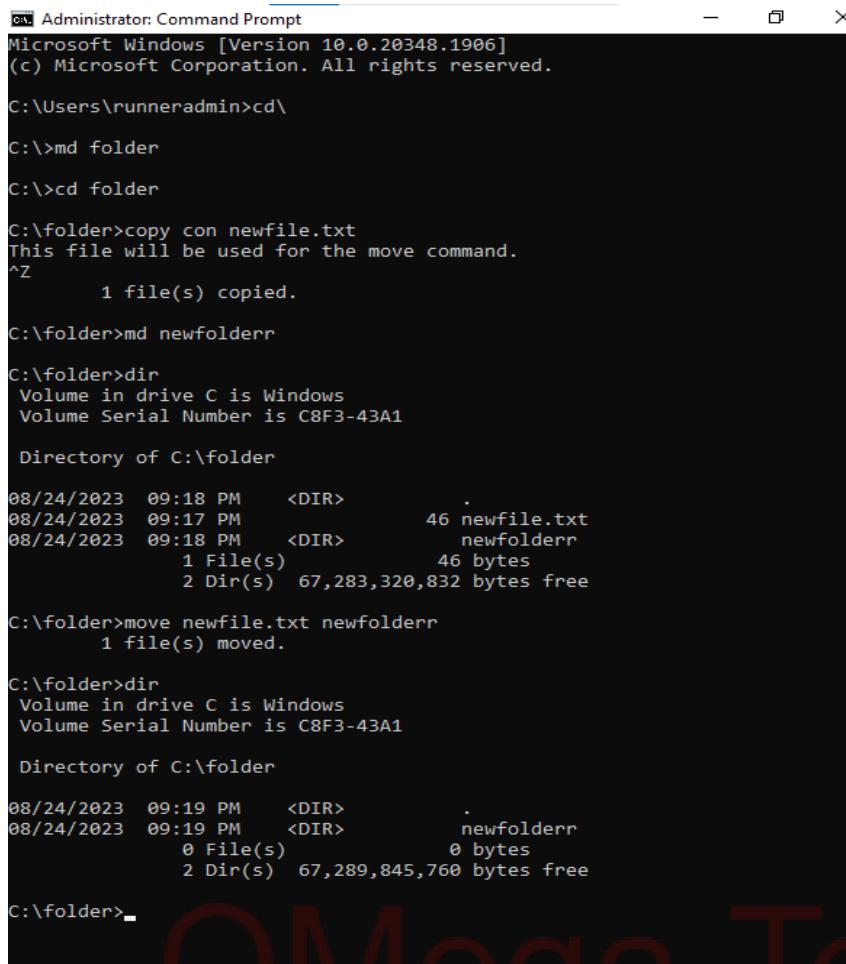
Syntax: move [/Y | -Y] [drive:] [path]filename [,...] destination.

Where:

[drive:] [path]filename	Specifies the location and name of file or files where you want to move.
Destination	Specifies the new location of the files.
/Y:	Suppresses prompting to confirm you want to overwrite an existing destination file.

Example: move c:\windows\temp*.* c:\temp

Move the files of c:\windows\temp to the temp directory in root. In this example, *.* is wildcard telling the computer every file with every extension.



```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.20348.1906]
(c) Microsoft Corporation. All rights reserved.

c:\Users\runneradmin>cd\

c:>md folder

c:>cd folder

c:\folder>copy con newfile.txt
This file will be used for the move command.
^Z
      1 file(s) copied.

c:\folder>md newfolderr

c:\folder>dir
Volume in drive C is Windows
Volume Serial Number is C8F3-43A1

Directory of C:\folder

08/24/2023  09:18 PM    <DIR>.
08/24/2023  09:17 PM           46 newfile.txt
08/24/2023  09:18 PM    <DIR>    newfolderr
      1 File(s)           46 bytes
      2 Dir(s)  67,283,320,832 bytes free

c:\folder>move newfile.txt newfolderr
      1 file(s) moved.

c:\folder>dir
Volume in drive C is Windows
Volume Serial Number is C8F3-43A1

Directory of C:\folder

08/24/2023  09:19 PM    <DIR>.
08/24/2023  09:19 PM    <DIR>    newfolderr
      0 File(s)           0 bytes
      2 Dir(s)  67,289,845,760 bytes free

c:\folder>_

```

echo: echo is one of the most commonly and widely used built-in commands that is typically used in scripting language and batch files to display a line of text/string on standard output or a file.

Syntax: ECHO [ON |OFF]

ECHO[message]

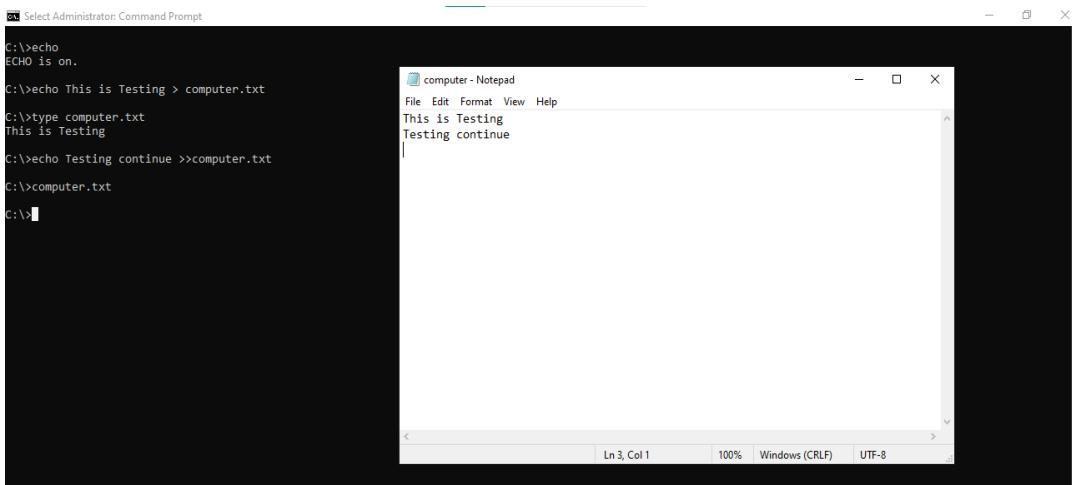
Type echo without parameters to display the current echo settings.

Example: C:\>echo "hi"

Hi

C:\>echo 100

100



```

Select Administrator: Command Prompt
C:\>echo
ECHO is on.
C:\>echo This is Testing > computer.txt
This is Testing
C:\>type computer.txt
This is Testing
Testing continue
C:\>echo Testing continue >>computer.txt
C:\>computer.txt
C:\>

```

2d) fc, find, rename, set, type, ver

fc: FC or file compare, is used to compare two files against each other. Once completed fc returns lines that differ between the two files. If no line differs, you will receive a message indicating no difference encountered.

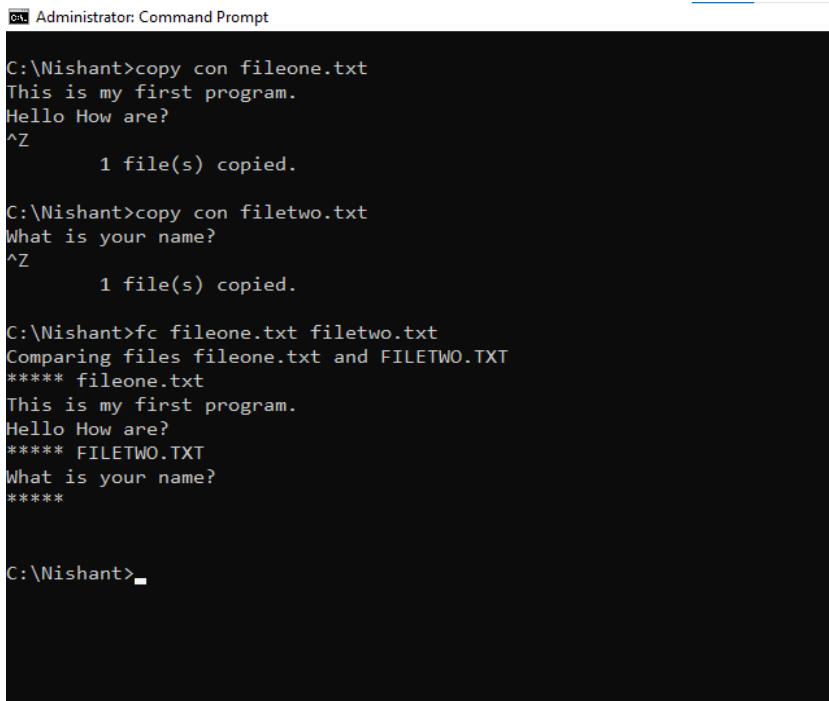
Syntax:

Fc [options] file1 file2

Where options:

/A	Displays only first and last lines for each set of difference.
/B	Performs a binary comparison.
/C	Disregards the case of letters.
/L	Compares files as ASCII text.
/N	Display the line numbers on an ASCII comparison
/T	Does not expand tabs to space
/W	Compresses white space for comparison.

Example: D:\>fc f1.txt f2.txt



```

Administrator: Command Prompt

C:\Nishant>copy con fileone.txt
This is my first program.
Hello How are?
^Z
      1 file(s) copied.

C:\Nishant>copy con filetwo.txt
What is your name?
^Z
      1 file(s) copied.

C:\Nishant>fc fileone.txt filetwo.txt
Comparing files fileone.txt and FILETWO.TXT
***** fileone.txt
This is my first program.
Hello How are?
***** FILETWO.TXT
What is your name?
*****


C:\Nishant>_

```

find: Allows us to search for text within a file.

Syntax: find [SWITCH] "String" [Pathname/s]

Where

/V	Display all lines not containing the specified string.
/C	Display only the count of lines containing the string.
/N	Display line numbers with the displayed lines.
/I	Ignores the case of characters when searching for the string.
"string"	Specifies the text string to find.
pathname	Specifies a file or files to search.

Example: D:\>find "frame" t1.txt

rename: Used to rename files and directories from the original name to a new name.

Syntax: ren [drive:] [path] [filename1] [filename2]

Example: C:\>ren a.txt b.txt

```
C:\>cd nishant
C:\nishant>copy con g.txt
this is my first program.^Z
    1 file(s) copied.

C:\nishant>findstr my g.txt
this is my first program.
C:\nishant>
```

```
C:\nishant>copy con s.txt
this is a file for rename command.
^Z
    1 file(s) copied.

C:\nishant>ren s.txt t.txt
C:\nishant>type s.txt
The system cannot find the file specified.

C:\nishant>type t.txt
this is a file for rename command.

C:\nishant>
```

set: Allows you to change one variable or string to another.

Syntax: [variable=[string]]

Where:

variable	Specifies the environment variable name.
string	Specifies a series of characters to assign to the variable.

Example: set path=c:\windows\command

type: Allows the user to see the contents of a file. To edit the files, the user would need to use either edit or copy con.

Syntax: Type[drive:] [path] filename

Example: C:\>type t1.txt

ver: Displays the version of MS-DOS or if running windows 95 or above the version of windows.

Syntax: ver

Example: C:\Users\abc>ver

SYIT SEM-III OPERATING SYSTEM BY: MEGHA SHARMA

Microsoft Windows [Version 10.0.22621.1992]

```
C:\>cd nishant
C:\nishant>copy con file1.txt
this is a file for type command.
^Z
1 file(s) copied.

C:\nishant>type file1.txt
this is a file for type command.

C:\nishant>set p
Path=C:\Program Files (x86)\VMware\VMware Player\bin;C:\Program Files\Common Files\Oracle\Java\javapath;C:\oracle\app\oracle\product\11.2.0\server\bin;C:\Windows\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\WindowsPowerShell\v1.0\;C:\windows\System32\OpenSSH\;C:\Program Files\nodejs\;C:\Program Files\Git\cmd;C:\Users\Akash Gupta\AppData\Local\Programs\Python\Python311\Scripts\;C:\Users\Akash Gupta\AppData\Local\Programs\Python\Python311\;C:\Users\Akash Gupta\AppData\Local\Microsoft\WindowsApps;C:\Users\Akash Gupta\AppData\Local\Programs\Microsoft VS Code\bin;C:\Users\Akash Gupta\AppData\Roaming\Npm;C:\src\flutter\bin;
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
platformcode=0
PROCESSOR_ARCHITECTURE=AMD64
PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 140 Stepping 2, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=8c02
ProgramData=C:\ProgramData
ProgramFiles=C:\Program Files
ProgramFiles(x86)=C:\Program Files (x86)
ProgramW6432=C:\Program Files
PROMPT=$P$G
PSModulePath=C:\Program Files\WindowsPowerShell\Modules;C:\windows\system32\WindowsPowerShell\v1.0\Modules
PTTHOME=C:\Program Files\Cisco Packet Tracer 7.1
PUBLIC=C:\Users\Public

C:\nishant>ver
Microsoft Windows [Version 10.0.22621.1992]
```

OMega TechEd

Practical 3

3a) pwd, cd, absolute and relative paths ls mkdir rmdir

pwd: This command displays the current working directory.

Syntax: \$ pwd

/home/Admin/Desktop

cd: This command is used to change the directory for working.

Syntax: cd [Directory]

Example:

Move to abc folder.

\$ cd /abc

Move up one folder.

\$ cd..

Going back to home folder

\$cd

Absolute and Relative path.

An absolute path is defined as specifying the location of a file or directory from the root directory (/). In other words, we can say absolute path is a complete path from start of actual filesystem from / directory.

Relative path is defined as path related to the present working directory.

ls: This command is used to list the details about the files.

Syntax: ls [Options] ...[Files]

mkdir: To create a directory 'mkdir' command is used.

Syntax: \$ mkdir folder name

rmdir: Remove directory, this command will only work if the folders are empty.

Syntax: rmdir folder name

```

rel.sh
Admin@vpm:~/Desktop/documents
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ pwd
/home/Admin/Desktop
[Admin@vpm Desktop]$ cd /home/Admin/Desktop/documents
[Admin@vpm documents]$ mkdir abc
[Admin@vpm documents]$ mkdir efg
[Admin@vpm documents]$ ls
abc efg
[Admin@vpm documents]$ cd abc
[Admin@vpm abc]$ cd ..
[Admin@vpm documents]$ ls
abc efg
[Admin@vpm documents]$ rmdir efg
[Admin@vpm documents]$ ls
abc
[Admin@vpm documents]$ 

```

3b file, touch, rm

file: The file utility determines the file type.

```
[Admin@vpm documents]$ file abc.jpeg
```

Abc.jpeg: JPEG image data, JFIF standard 1.01.

touch: touch is used to create an empty file.

```
[Admin@vpm documents]$ touch text1.txt
```

rm: The rm command is used to remove the file permanently from the machine.

```
[Admin@vpm documents]$ rm text1.txt
```

```

rel.sh
Admin@vpm:~/Desktop/documents
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ cd documents
[Admin@vpm documents]$ ls
abc Ss.png
[Admin@vpm documents]$ file Ss.png
Ss.png: PNG image data, 1280 x 768, 8-bit/color RGB, non-interlaced
[Admin@vpm documents]$ touch file1.txt
[Admin@vpm documents]$ touch file2.txt
[Admin@vpm documents]$ ls
abc file1.txt file2.txt Ss.png
[Admin@vpm documents]$ rm file2.txt
[Admin@vpm documents]$ ls
abc file1.txt Ss.png
[Admin@vpm documents]$ 

```

cp, mv, rename.

cp: The cp is a Linux shell command to copy files and directories.

Syntax: \$ cp [options] source destination

cp command options:

cp-a	archive files
cp-f	Force copy by removing the destination file if needed.
cp - i	Interactive – ask before overwriting.
cp -l	Link files instead of copy
cp -l	Follow symbolic links
cp-n	No file overwrites
cp -r	Recursive copy (including hidden files)
cp -u	Update -copy when source is newer than destination
cp -v	Verbose- print informative message

Copy single file abc.c to destination directory bak:

\$ cp abc.c bak

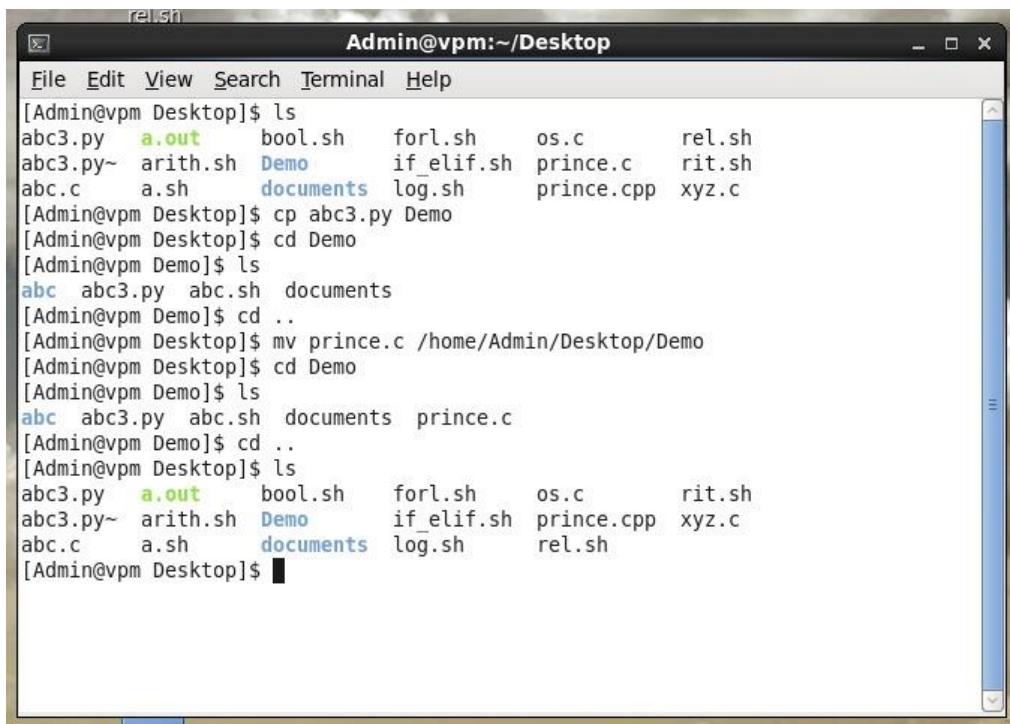
mv: The mv command is used to move files and directories.

Syntax: mv [options] source dest

Options:

mv -f	Force moves by overwriting destination file without prompt.
mv -i	Interactive prompt before overwrite.
mv - u	Update move when source is newer than destination.
mv -v	Verbose- print source and destination file.
man mv	Help manul

Example: \$mv abc.c /home/Admin/Desktop/demo



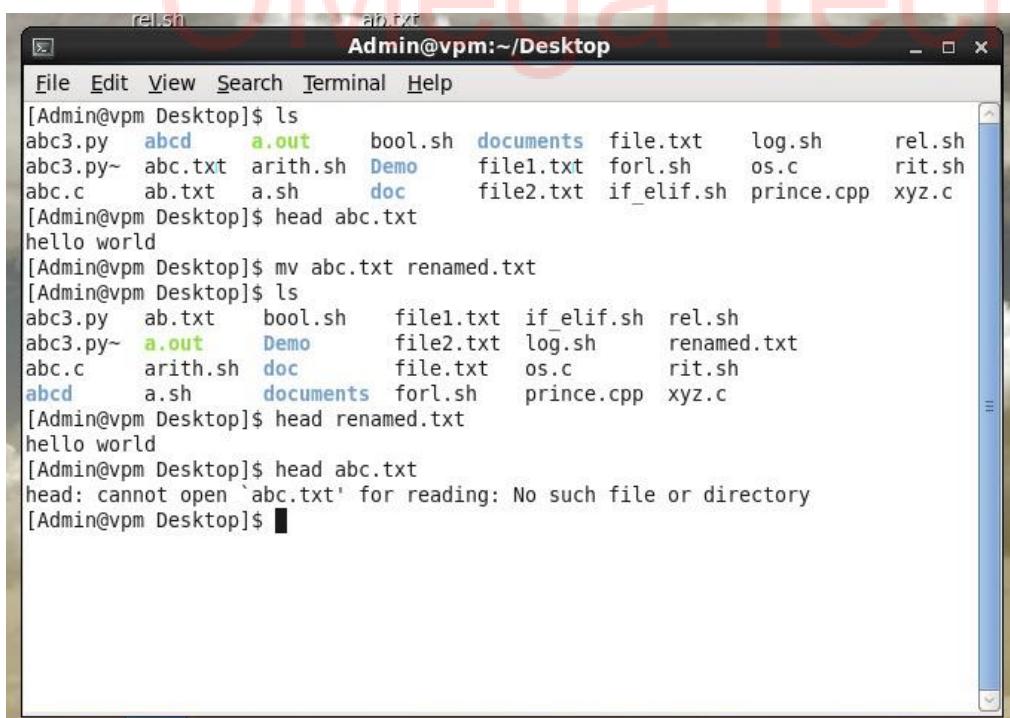
```

relish
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ ls
abc3.py  a.out  bool.sh  forl.sh  os.c  rel.sh
abc3.py~  arith.sh  Demo  if_elif.sh  prince.c  rit.sh
abc.c  a.sh  documents  log.sh  prince.cpp  xyz.c
[Admin@vpm Desktop]$ cp abc3.py Demo
[Admin@vpm Desktop]$ cd Demo
[Admin@vpm Demo]$ ls
abc  abc3.py  abc.sh  documents  prince.c
[Admin@vpm Demo]$ cd ..
[Admin@vpm Desktop]$ mv prince.c /home/Admin/Desktop/Demo
[Admin@vpm Desktop]$ cd Demo
[Admin@vpm Demo]$ ls
abc3.py  a.out  bool.sh  forl.sh  os.c  rit.sh
abc3.py~  arith.sh  Demo  if_elif.sh  prince.cpp  xyz.c
abc.c  a.sh  documents  log.sh  rel.sh
[Admin@vpm Desktop]$ 

```

rename: The rename command will rename the specified files by replacing the first occurrence of from in their name with to.

Syntax: rename from to file...



```

relish
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ ls
abc3.py  abcd  a.out  bool.sh  documents  file.txt  log.sh  rel.sh
abc3.py~  abc.txt  arith.sh  Demo  file1.txt  forl.sh  os.c  rit.sh
abc.c  ab.txt  a.sh  doc  file2.txt  if_elif.sh  prince.cpp  xyz.c
[Admin@vpm Desktop]$ head abc.txt
hello world
[Admin@vpm Desktop]$ mv abc.txt renamed.txt
[Admin@vpm Desktop]$ ls
abc3.py  ab.txt  bool.sh  file1.txt  if_elif.sh  rel.sh
abc3.py~  a.out  Demo  file2.txt  log.sh  renamed.txt
abc.c  arith.sh  doc  file.txt  os.c  rit.sh
abcd  a.sh  documents  forl.sh  prince.cpp  xyz.c
[Admin@vpm Desktop]$ head renamed.txt
hello world
[Admin@vpm Desktop]$ head abc.txt
head: cannot open `abc.txt' for reading: No such file or directory
[Admin@vpm Desktop]$ 

```

head, tail, cat, more, less,

head: This command used for output the first part of files, print the first 10 lines by default of each file.

Syntax: head [options]...[file]...

Example: \$head /etc/passwd

tail: This command outputs the last part of files, print the last 10 lines by default of each file.

Syntax: tail [options]...[file]...

Example: \$ tail /etc/passwd

cat: The cat command is used to display the content of text files and to combine several files to one file. The cat command does not accept directories.

Syntax: cat file1 file2

Example: \$ cat l1.txt l2.txt

more: This command display output one screen at a time.

Syntax: more [options] [file]

options	Description
-c	Clear screen before displaying
-e	Exit immediately after writing the last line of the last file in the argument list
-n	Specify how many lines are printed in the screen for a given file.
+n	Starts up the file from the given number.

Example: \$more -c index.php

less: This command is used to display text in the terminal screen. To display the text from the specified line, enter the line number followed by colon(:).

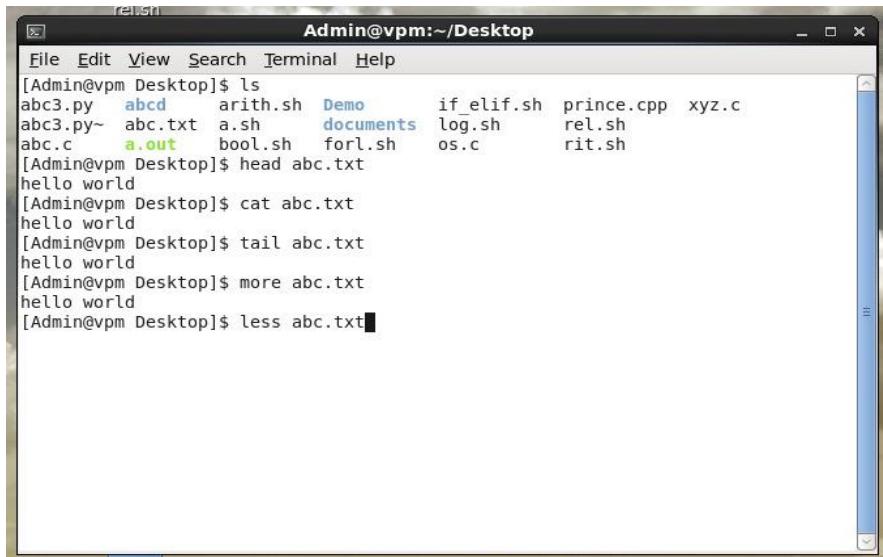
Syntax: less[options] filename

Options:

Option	Description
-c	Clear screen before displaying
+n	Starts up the file from the given number.
:p	Examine the previous file in the command list line
:d	Remove the current file from the list of files.

Example: To start printing from the third line of the file.

\$less +3 index.php



```

[Admin@vpm Desktop]$ ls
abc3.py  abcd  arith.sh  Demo  if elif.sh  prince.cpp  xyz.c
abc3.py~  abc.txt  a.sh  documents  log.sh  rel.sh
abc.c  a.out  bool.sh  forl.sh  os.c  rit.sh
[Admin@vpm Desktop]$ head abc.txt
hello world
[Admin@vpm Desktop]$ cat abc.txt
hello world
[Admin@vpm Desktop]$ tail abc.txt
hello world
[Admin@vpm Desktop]$ more abc.txt
hello world
[Admin@vpm Desktop]$ less abc.txt

```



```

[Admin@vpm Desktop]$ cat abc.txt
hello world
abc.txt (END)

```

OMega TechEd

3c ps, top, grep, locate, find, cal

ps: This command is used to report the process status.

top: The top command is used to find CPU usage. It provides dynamic real time view of running system.



```

[Admin@vpm Desktop]$ ps
  PID TTY      TIME CMD
 7502 pts/43  00:00:00 bash
 7519 pts/43  00:00:00 ps
[Admin@vpm Desktop]$ top

```

```

top - 15:02:04 up 8:15, 4 users, load average: 0.01, 0.02, 0.03
Tasks: 189 total, 1 running, 188 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.0%us, 1.3%sy, 0.0%ni, 97.3%id, 0.0%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 2402816k total, 926168k used, 1476648k free, 107948k buffers
Swap: 2539512k total, 0k used, 2539512k free, 391688k cached

PID USER      PR  NI  VIRT   RES   SHR   S %CPU %MEM   TIME+ COMMAND
2024 root      20   0 114m 1380  772 S  0.7  0.1  0:02.18 crond
2299 root      20   0 256m  37m 9708 S  0.7  1.6  2:37.36 Xorg
274 root      20   0     0     0 S  0.3  0.0  1:03.43 scsi_eh_4
1801 root      20   0 20216 1192 1024 S  0.3  0.0  0:45.70 hald-addon-stor
2867 Admin     20   0 299m 14m 10m S  0.3  0.6  0:30.16 gnome-terminal
9484 Admin     20   0 15032 1300 952 R  0.3  0.1  0:00.07 top
 1 root      20   0 19356 1536 1224 S  0.0  0.1  0:04.02 init
 2 root      20   0     0     0 S  0.0  0.0  0:00.04 kthreadd
 3 root      RT   0     0     0 S  0.0  0.0  0:00.00 migration/0
 4 root      20   0     0     0 S  0.0  0.0  0:00.38 ksoftirqd/0
 5 root      RT   0     0     0 S  0.0  0.0  0:00.00 migration/0
 6 root      RT   0     0     0 S  0.0  0.0  0:01.30 watchdog/0
 7 root      20   0     0     0 S  0.0  0.0  0:04.61 events/0
 8 root      20   0     0     0 S  0.0  0.0  0:00.00 cgroup
 9 root      20   0     0     0 S  0.0  0.0  0:00.01 khelper
10 root     20   0     0     0 S  0.0  0.0  0:00.00 netsn
11 root     20   0     0     0 S  0.0  0.0  0:00.00 async/mgr

```

grep: The grep command selects and prints the lines from a file which matches a given string or pattern.

Syntax: grep [options] pattern [file]

Options	Description
-c	Prints the count of line matched
-w	Match Whole word only
-A	Print num lines of text that occur after the matching line.
-i	Ignores changes in case
-n	Print line and line number

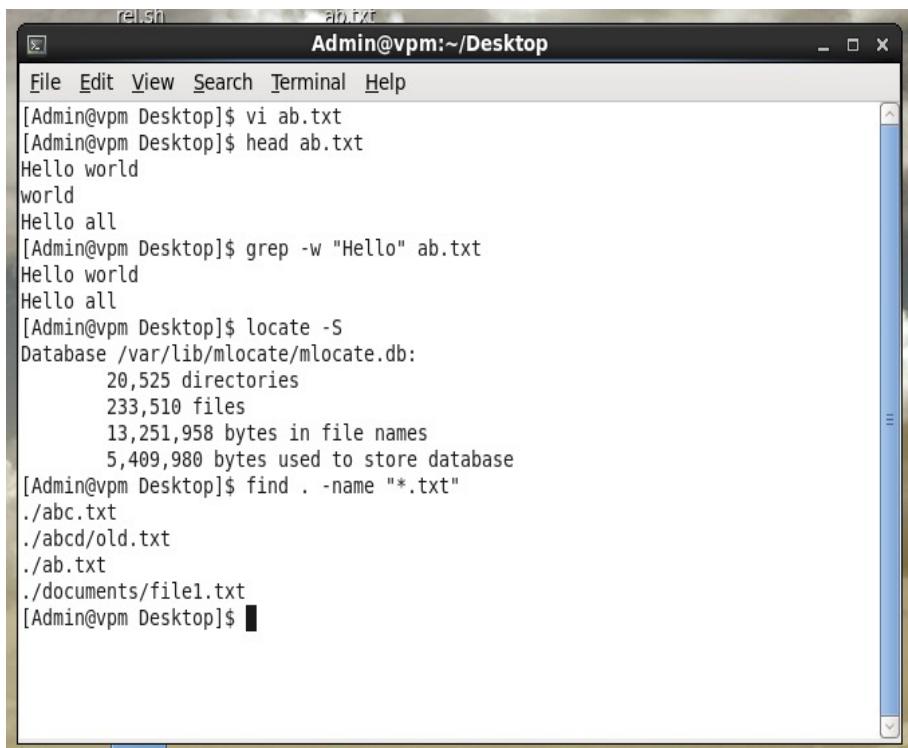
Example: grep -c "hi" file1.txt

locate: The locate command is used to find the location of files and directories in the system. It is used to find the location of list of files in the path, which is specified in the command.

Syntax: locate[options] pattern

Example: \$locate -b 'test'

Locates and returns the path of files where exactly file name test is stored.



```

relish          ab.txt
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ vi ab.txt
[Admin@vpm Desktop]$ head ab.txt
Hello world
world
Hello all
[Admin@vpm Desktop]$ grep -w "Hello" ab.txt
Hello world
Hello all
[Admin@vpm Desktop]$ locate -S
Database /var/lib/mlocate/mlocate.db:
  20,525 directories
  233,510 files
  13,251,958 bytes in file names
  5,409,980 bytes used to store database
[Admin@vpm Desktop]$ find . -name "*.txt"
./abc.txt
./abcd/old.txt
./ab.txt
./documents/file1.txt
[Admin@vpm Desktop]$ 

```

find: The find command finds one or more files if you know their approximate file name.

Syntax: find path [option]

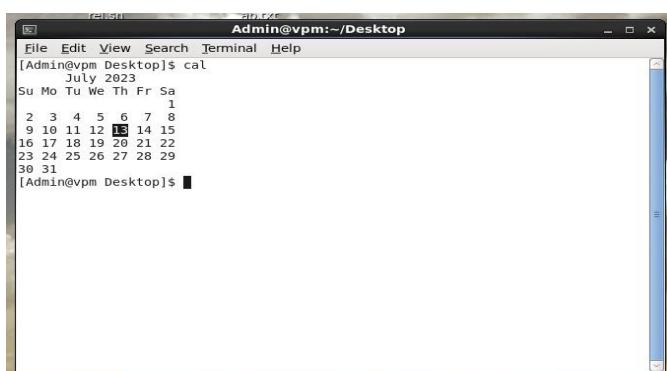
Example: For searching any file named 'sal.txt' in the current directory and any subdirectory in system.

\$ find -name 'sal.txt'

cal: The cal command is used to display the calendar.

Syntax: cal [options] [month] [year]

Example: \$cal



```

relish          ab.txt
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ cal
July 2023
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 29
30 31
[Admin@vpm Desktop]$ 

```

Practical 4

4a) The vi editor

Welcome to VI editor.

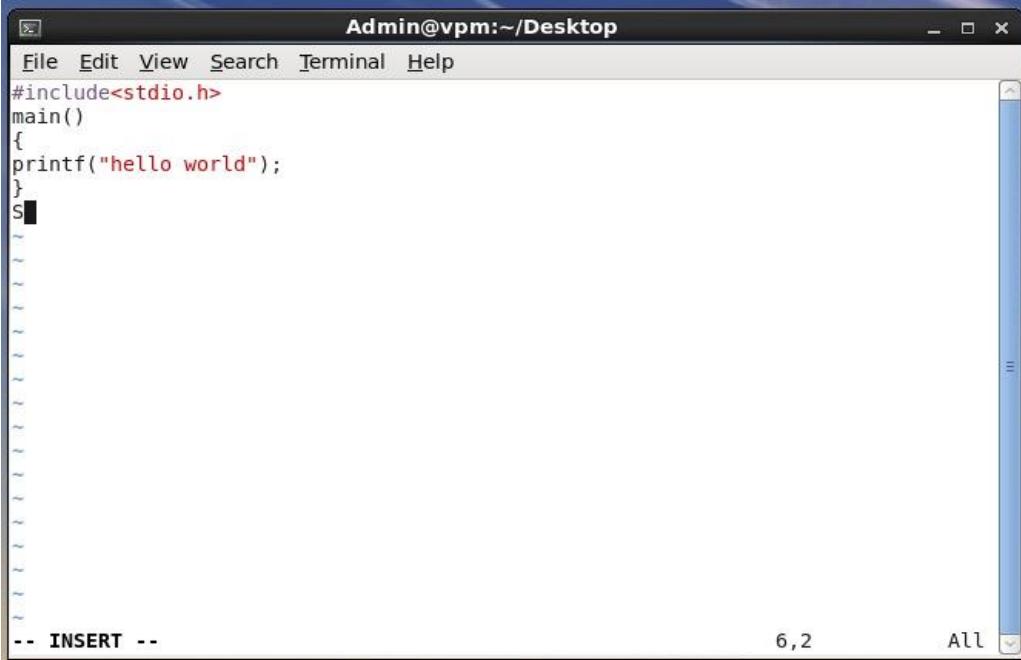
1. **VI editor** is available in almost all Linux distributions.
2. It works the same across different platforms and distributions.
3. It is user friendly.
4. Vi editor opens in command mode.
5. Commands are case sensitive.
6. Press 'i' on the keyboard for insert mode.
7. In **insert mode**, any key would be taken as an input.
8. Press **ESC key** to save changes and return to command mode. Press **SHIFT + zz** to save the file.



The screenshot shows a terminal window titled "Admin@vpm:~/Desktop". The window contains the following text:

```
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ vi os.c
[Admin@vpm Desktop]$ gcc os.c
[Admin@vpm Desktop]$ ./a.out
hello world[Admin@vpm Desktop]$ █
```

The terminal window has a standard Linux-style interface with a menu bar and a scroll bar on the right side. The text in the window represents the steps of writing a C program, compiling it, and running the resulting executable.



```

Admin@vpm:~/Desktop
File Edit View Search Terminal Help
#include<stdio.h>
main()
{
printf("hello world");
}
S
-- INSERT -- 6,2 All

```

The screenshot shows a terminal window titled 'Admin@vpm:~/Desktop'. The window contains a C program with a syntax error. The code is as follows:

```

#include<stdio.h>
main()
{
printf("hello world");
}
S

```

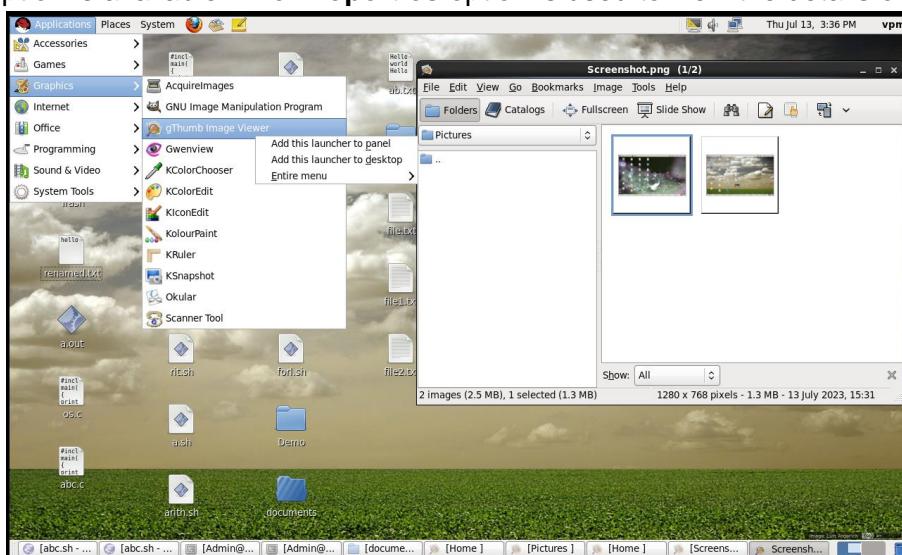
The terminal shows the message '-- INSERT --' at the bottom left, the line number '6,2' at the bottom center, and the word 'All' at the bottom right. The window has a standard Linux-style title bar and scroll bars.

4b) Graphical user interface

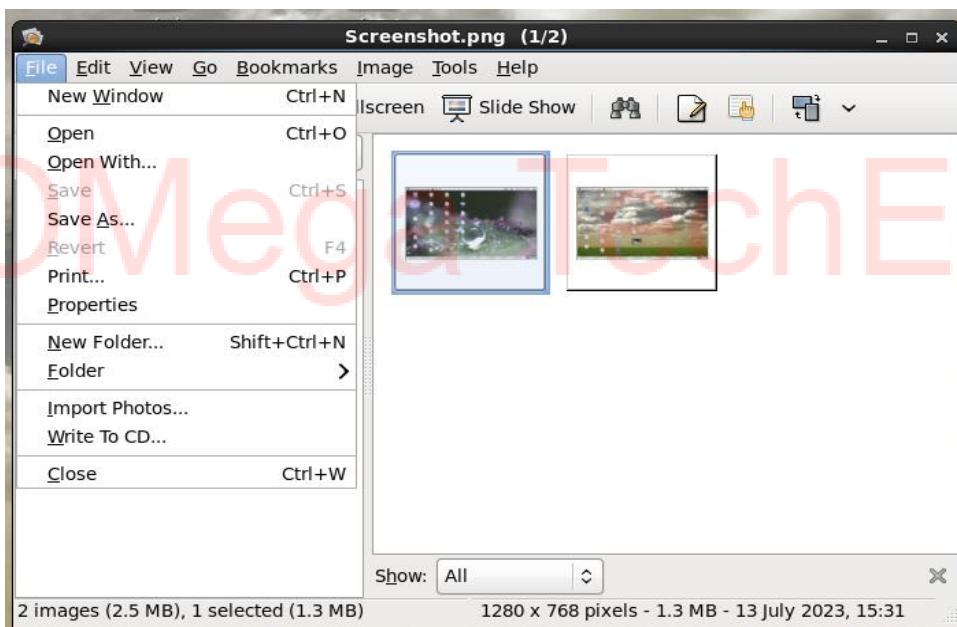
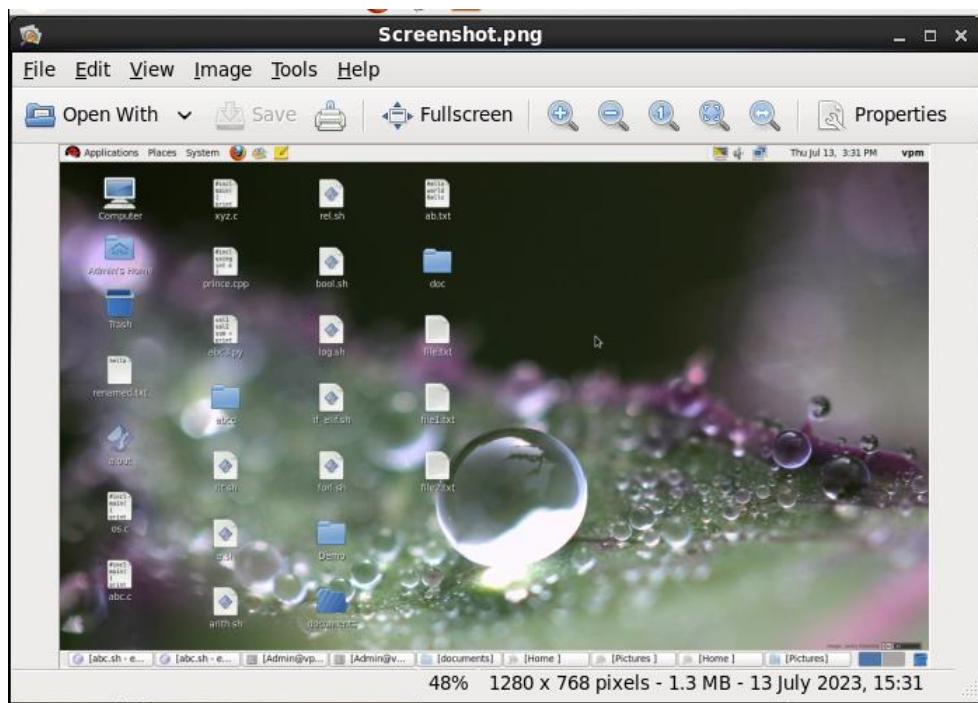
Graphics in Linux can be managed by GThumb. Which comes by default in RedHat Linux 6.0 by default.

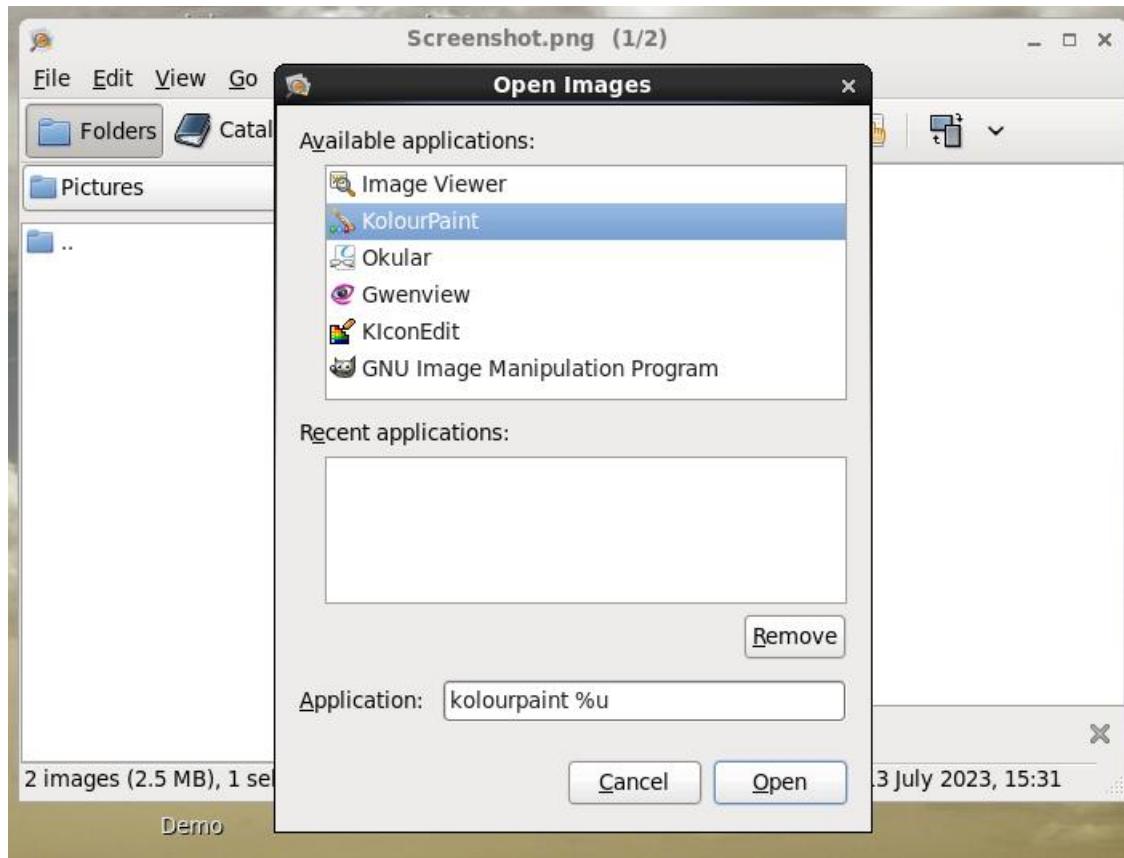
To start Gthumb

1. Click on Applications Menu > Graphics> gthumb Image viewer.
2. The **File Menu** has options for opening a new window. It has the option **Open** and **Open with** for opening the files already existing in the system. Also has options like **Save** and **Save As**. For printing an image **Print** option is available. The **Properties** option is used to view the details of the current open image.



SYIT SEM-III OPERATING SYSTEM BY: MEGHA SHARMA

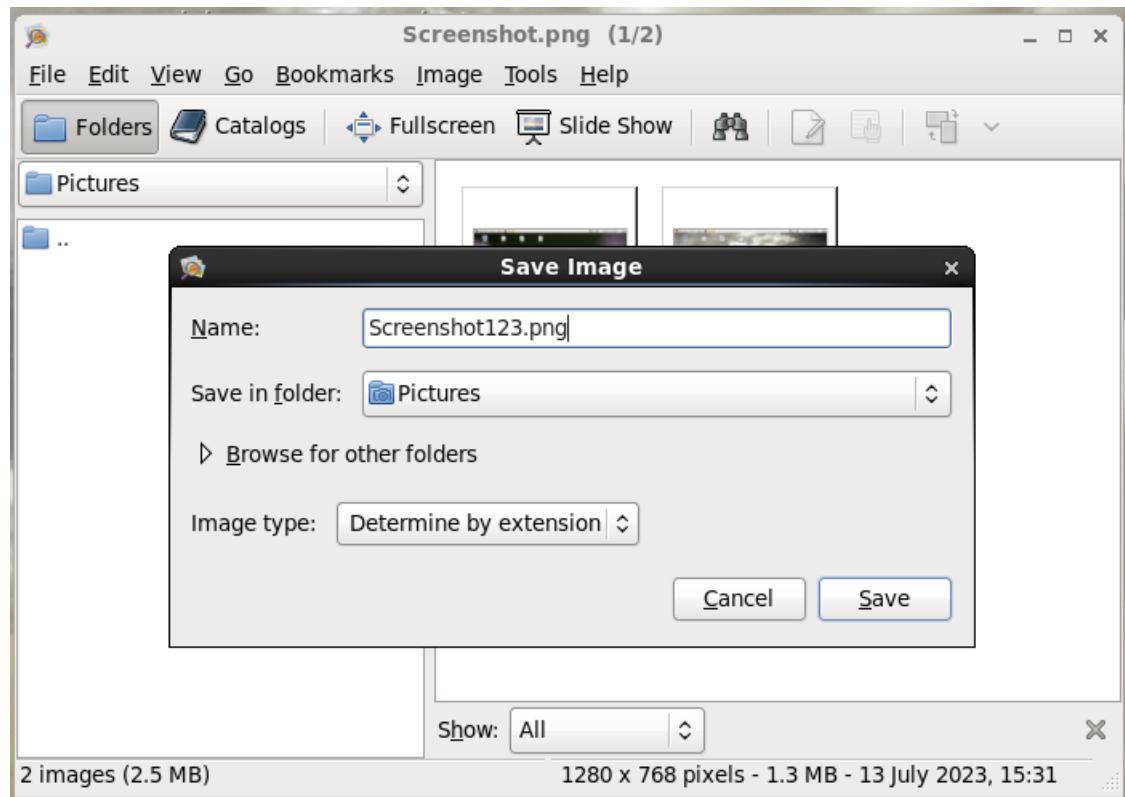




Open With (KolourPaint)

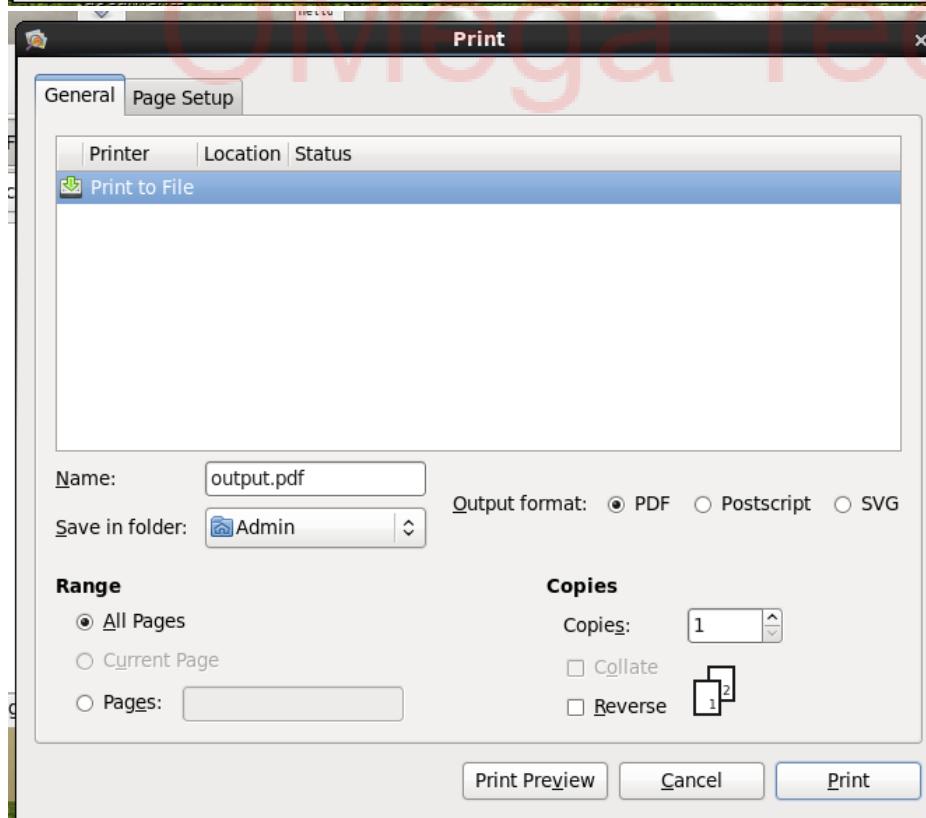
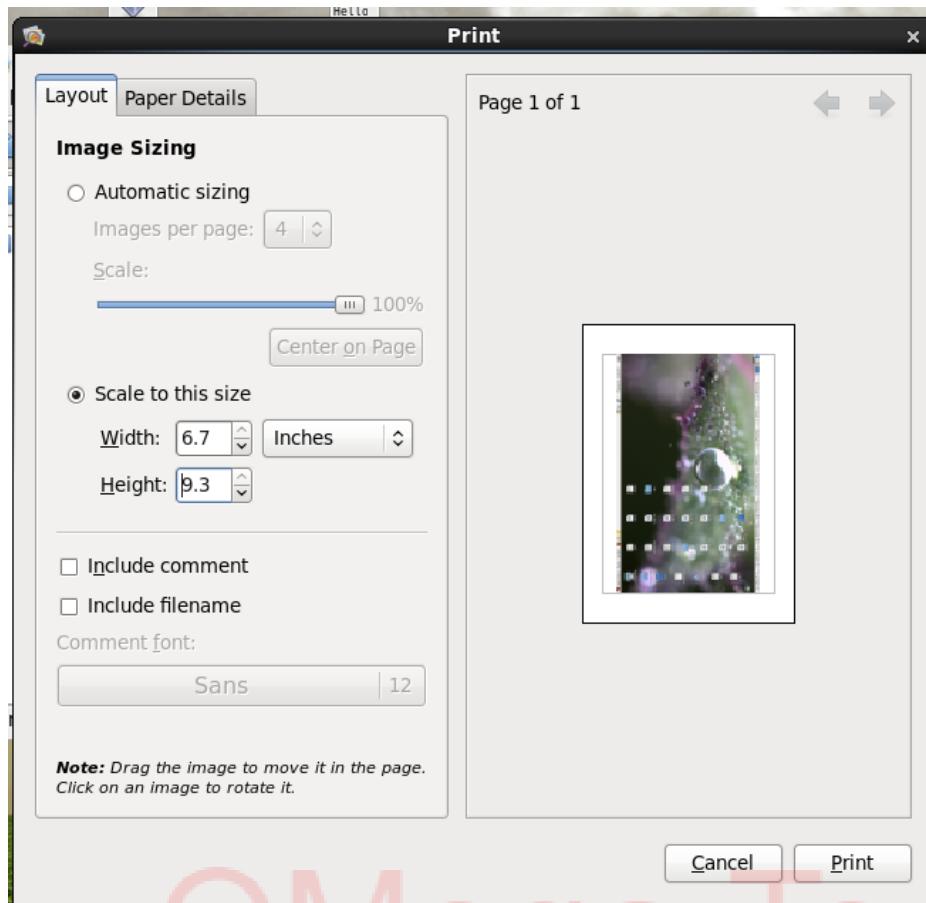


Save& save as

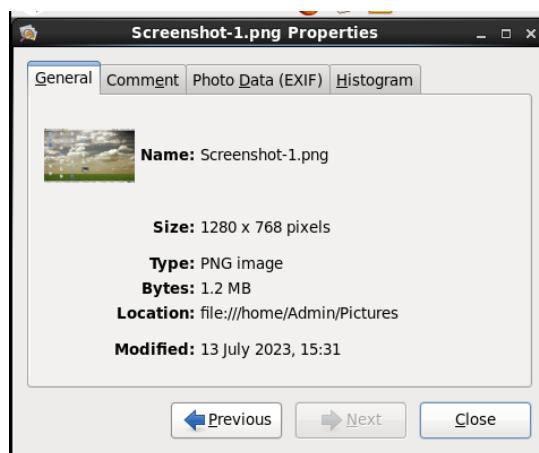
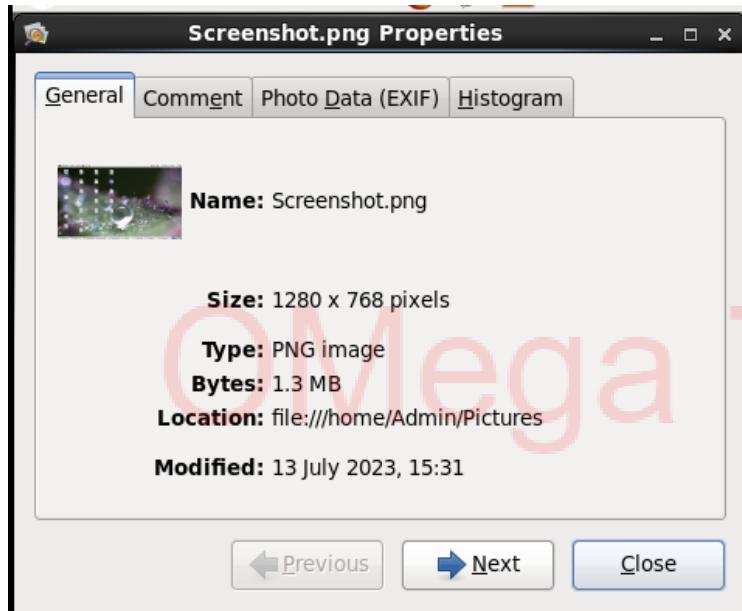
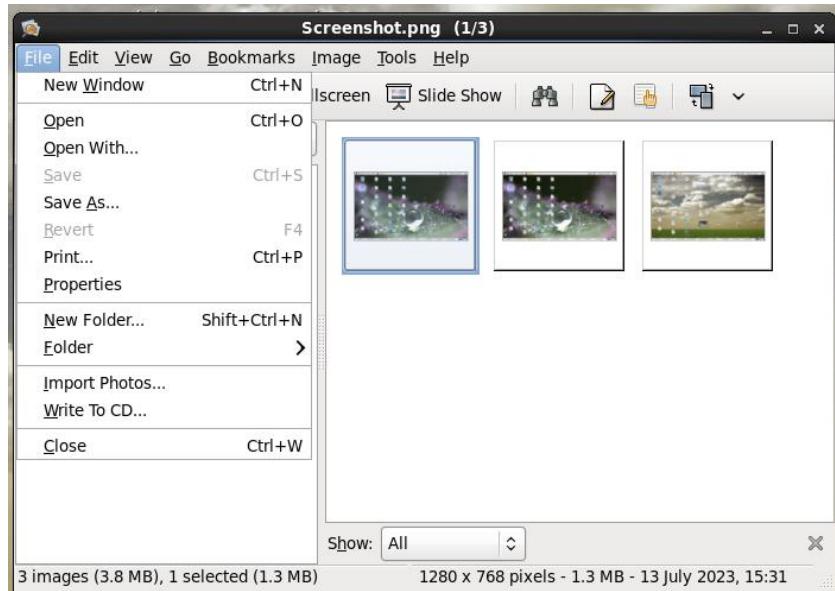


OMega TechEd

Print



The Properties



3. **The Edit menu:** Like any normal edit menu provides facility for **UNDO, REDO, RENAME, COPY, MOVE, DUPLICATE, MOVE TO TRASH** facilities.

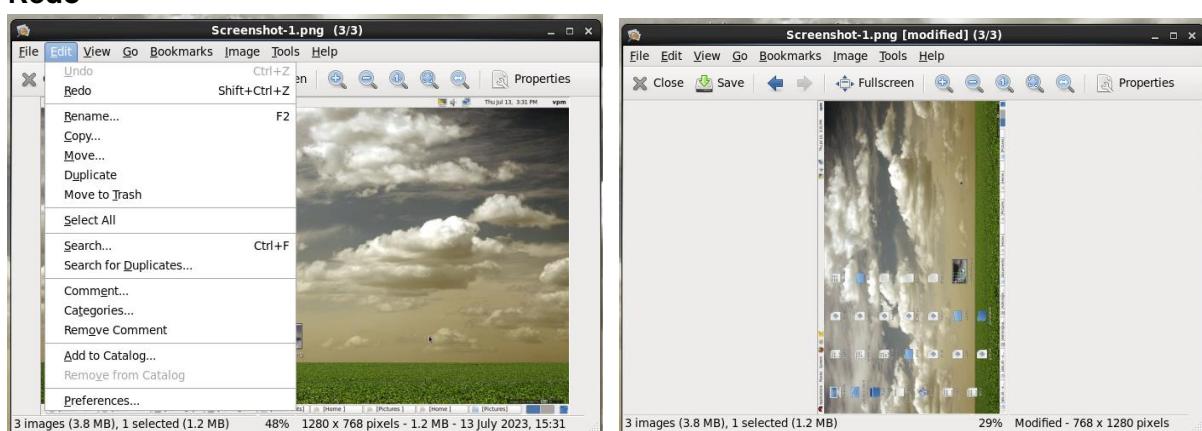
The Edit Menu: undo, redo, rename, copy, move, duplicate, move to trash.



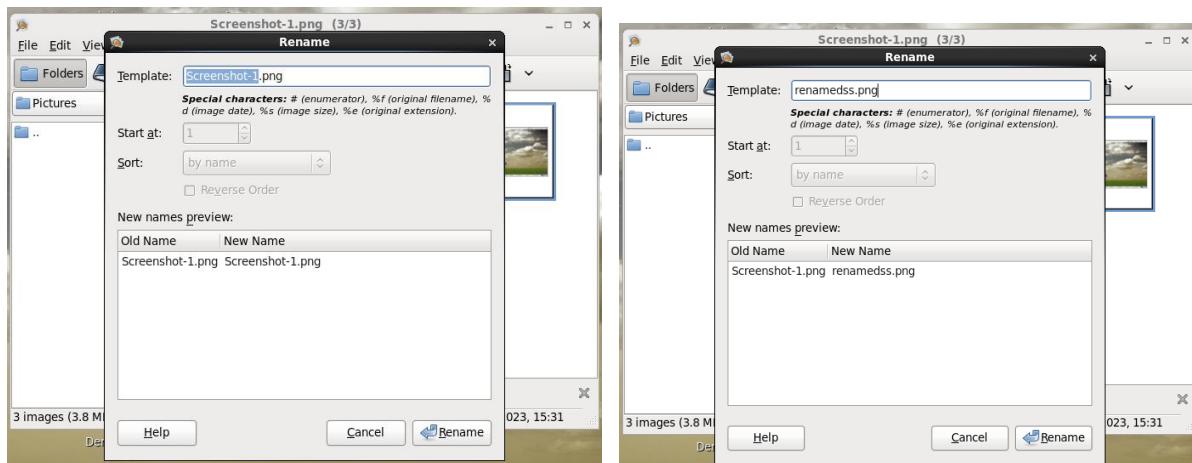
Undo



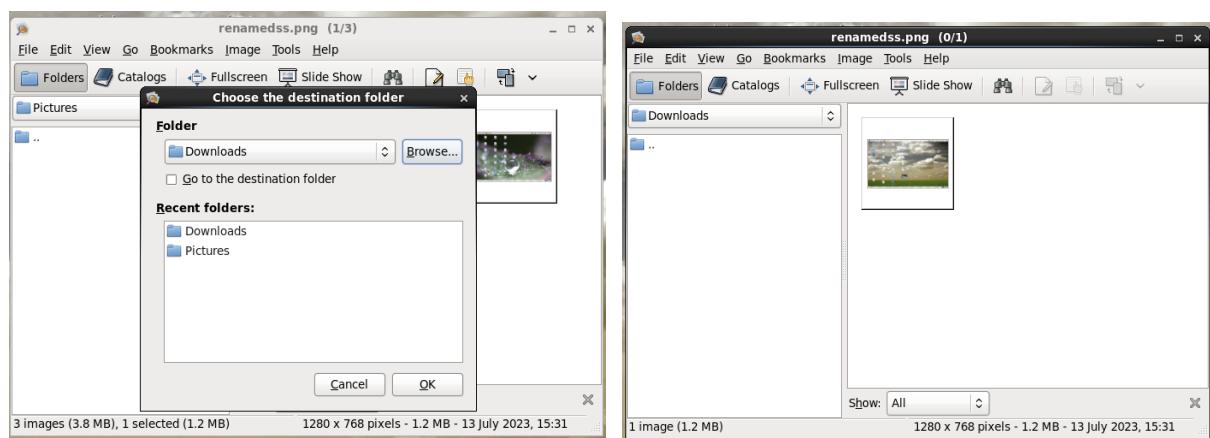
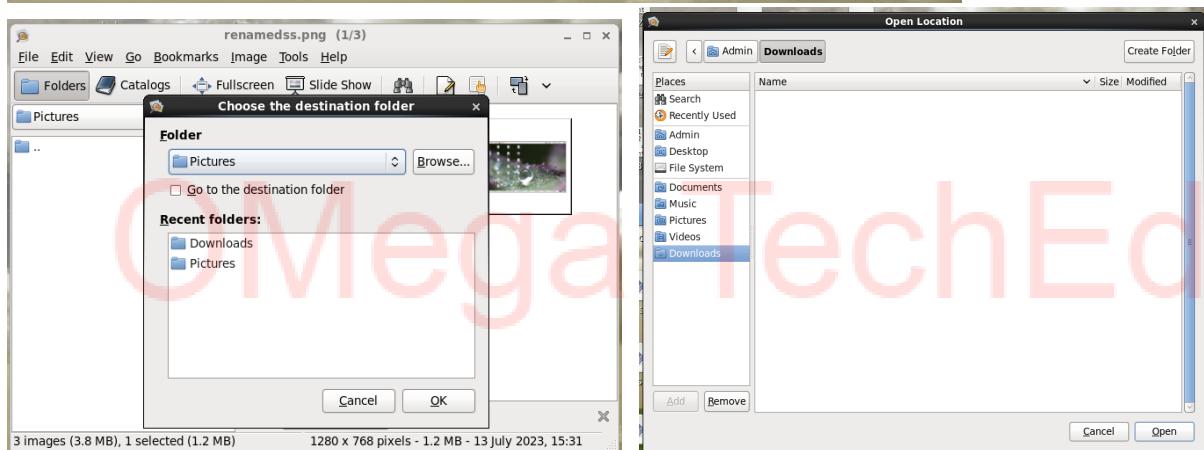
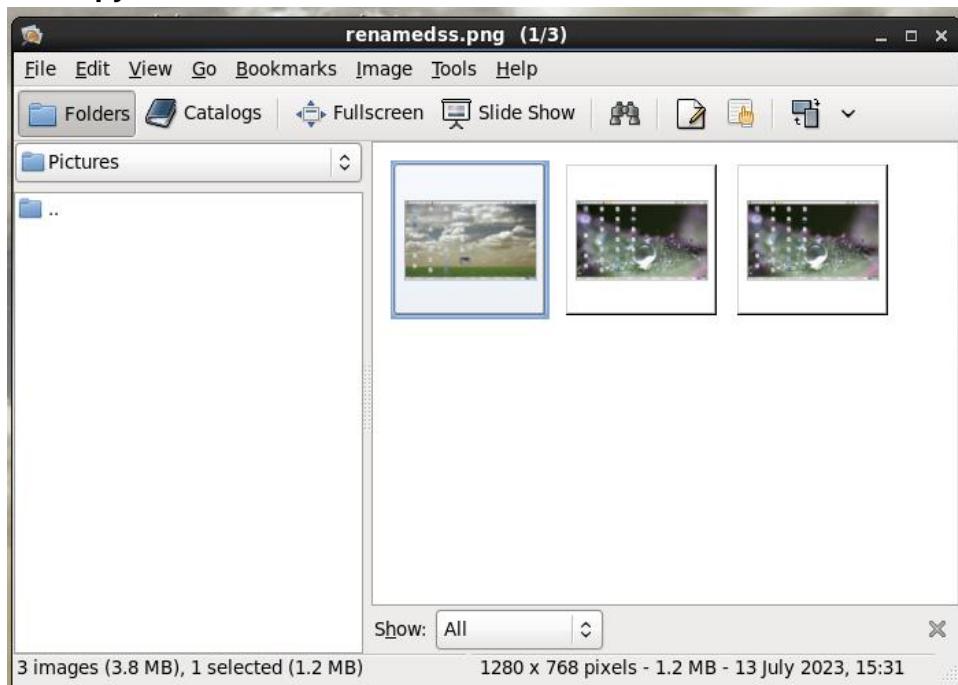
Redo



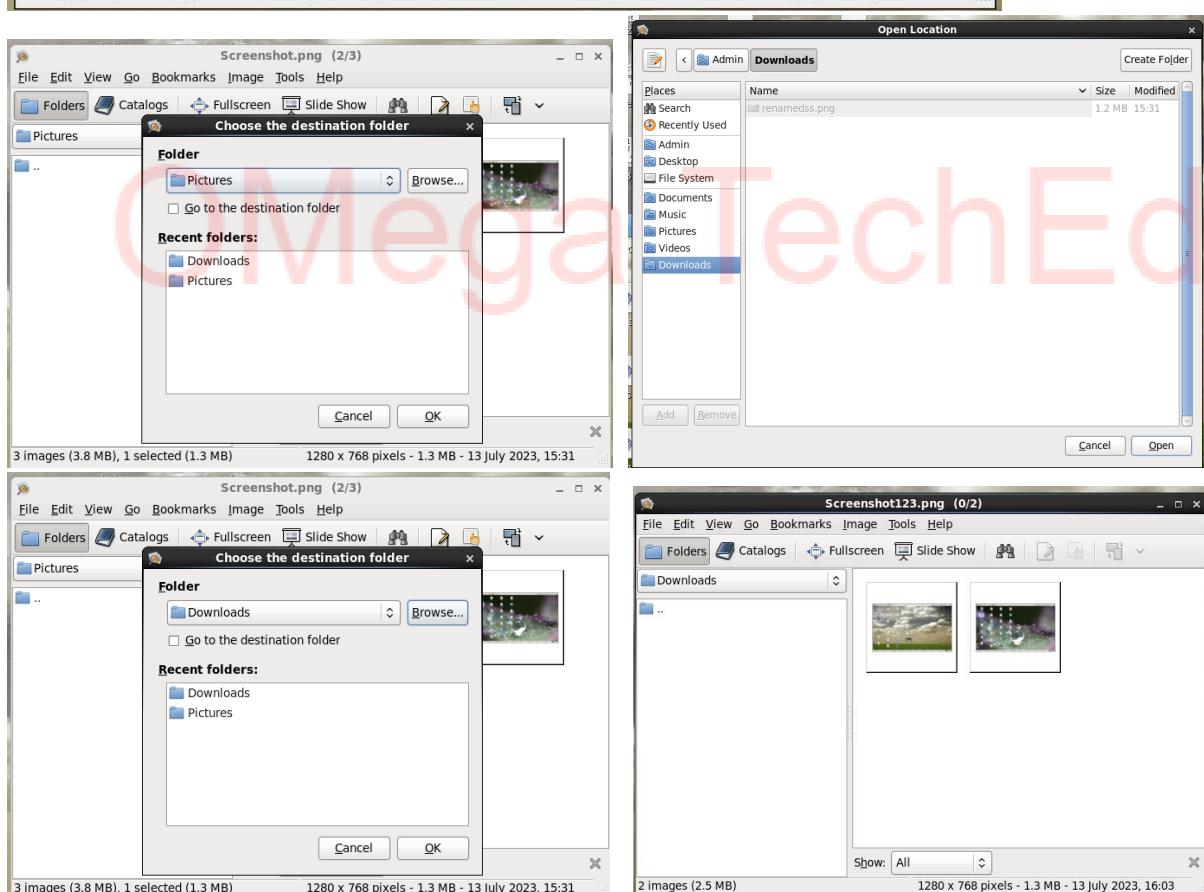
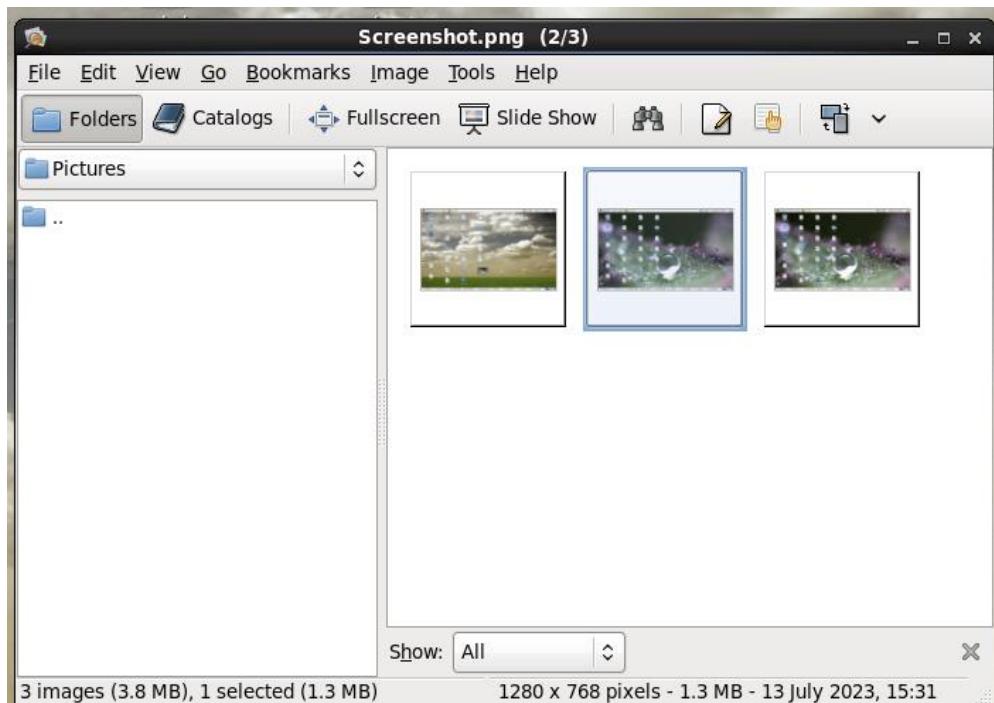
Rename



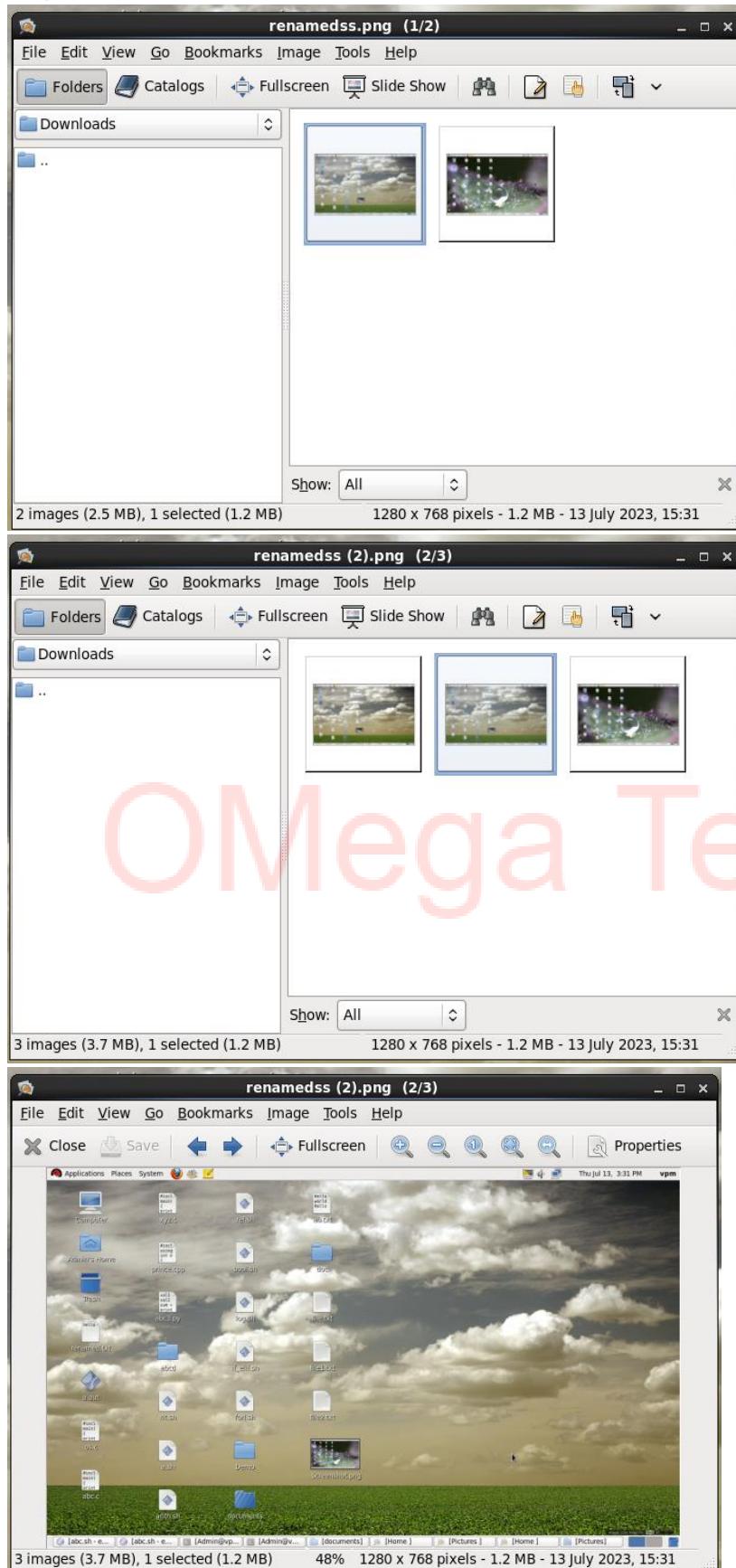
Copy



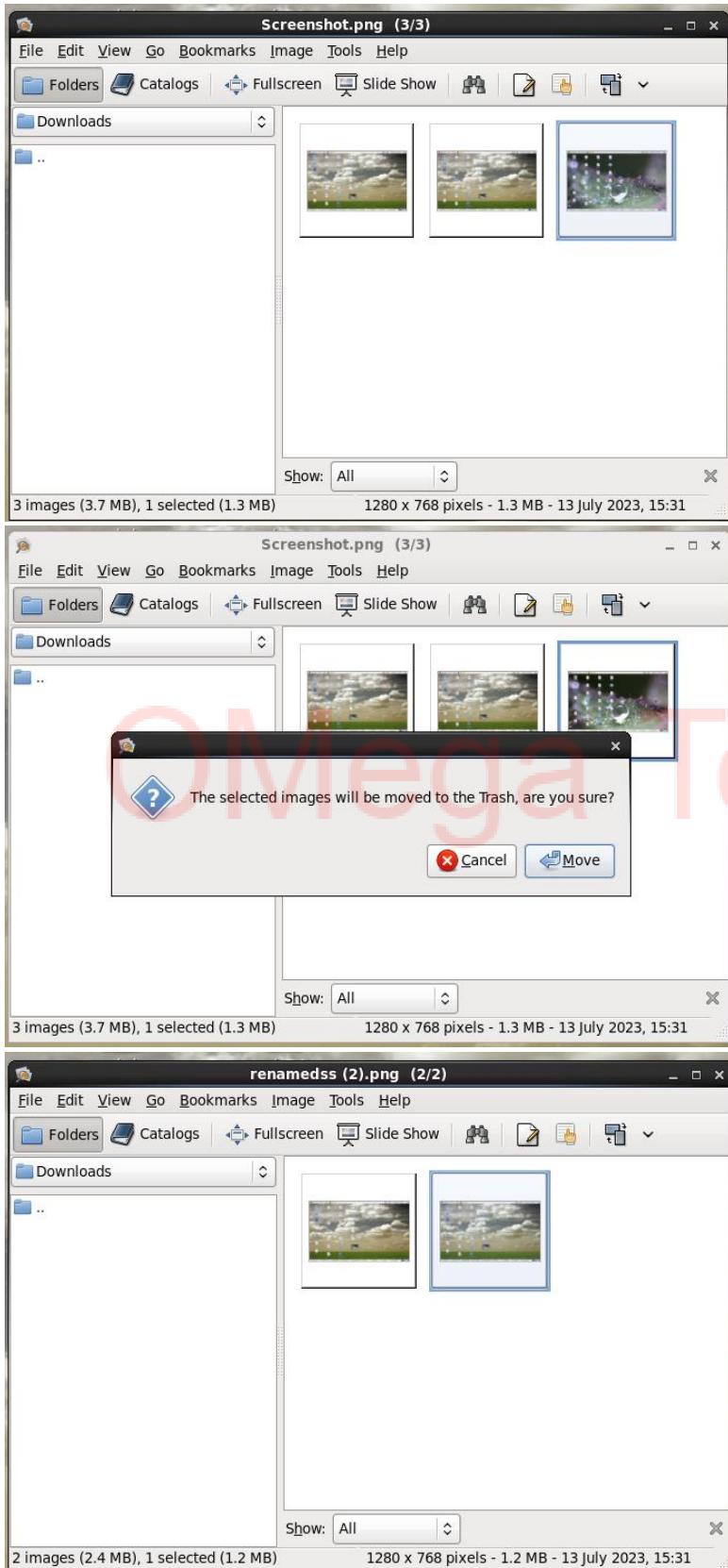
Move



Duplicate

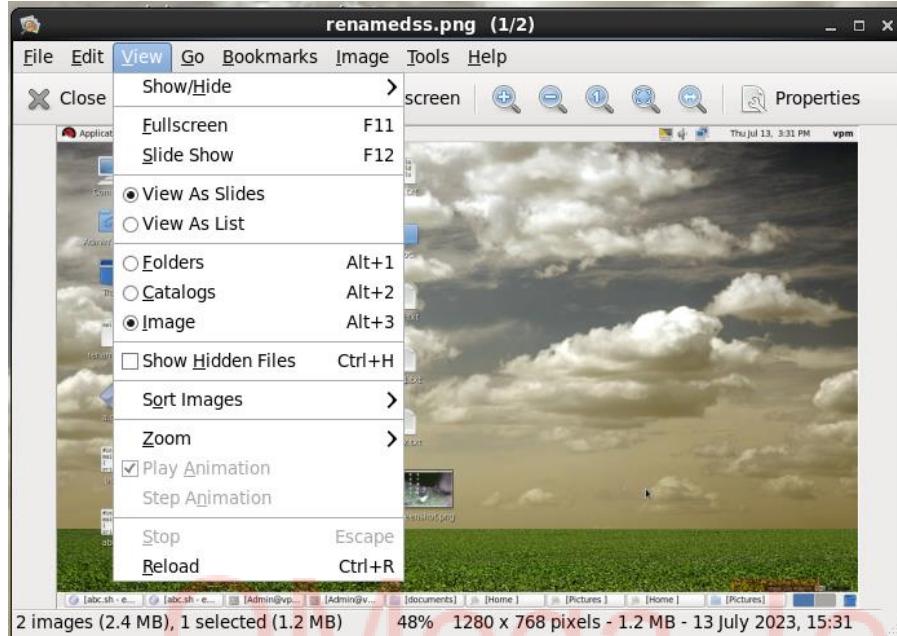


Move to trash

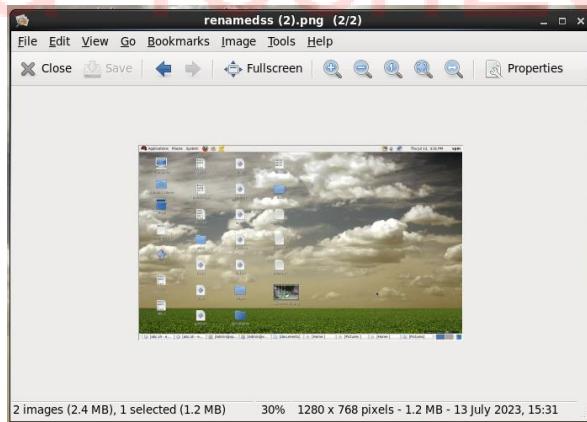
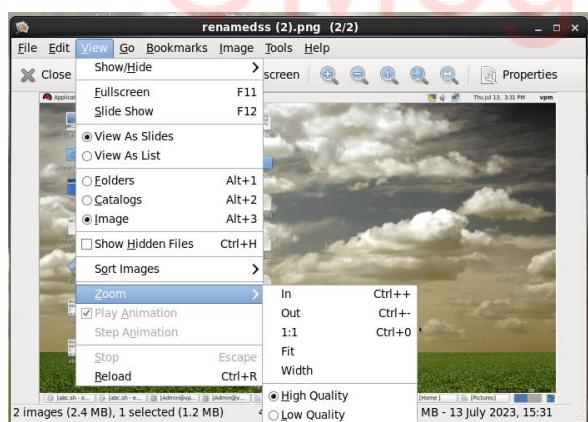


4. The View menu helps the user to display the image as per the requirement of the users, like Zoom, Full screen etc.

The View Menu: Zoom, Full screen



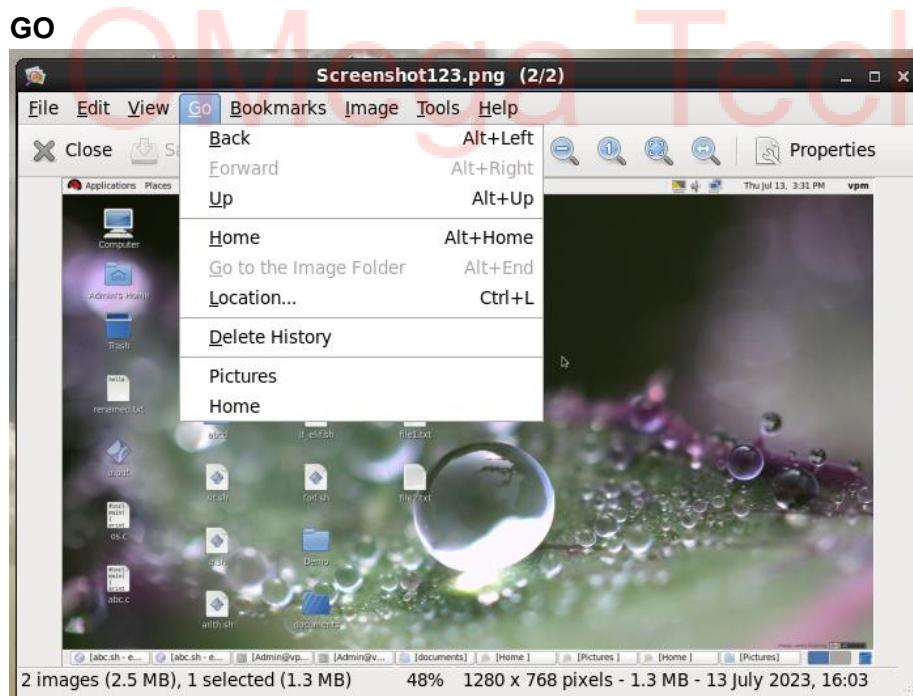
Zoom



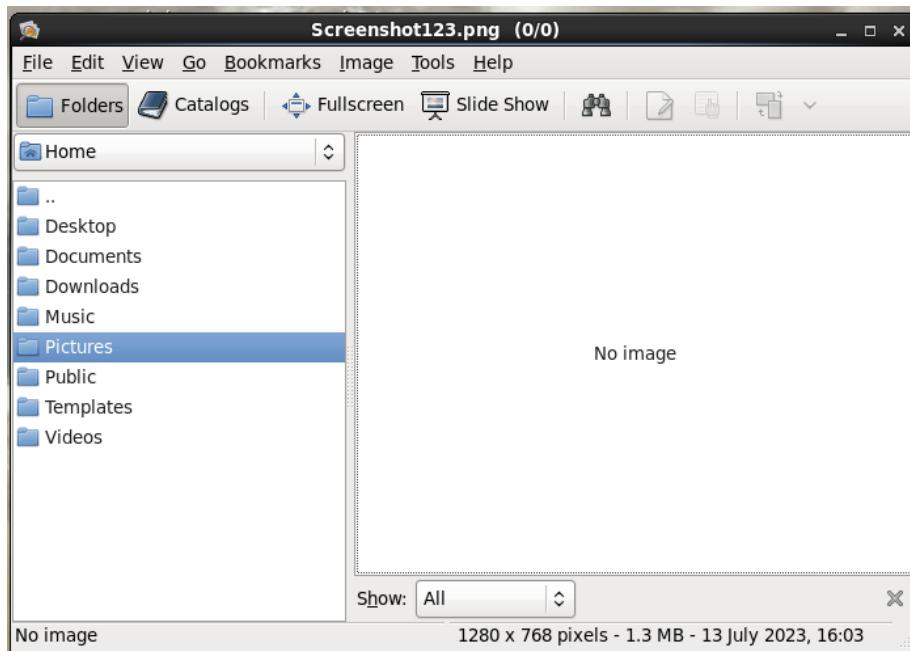
Full screen



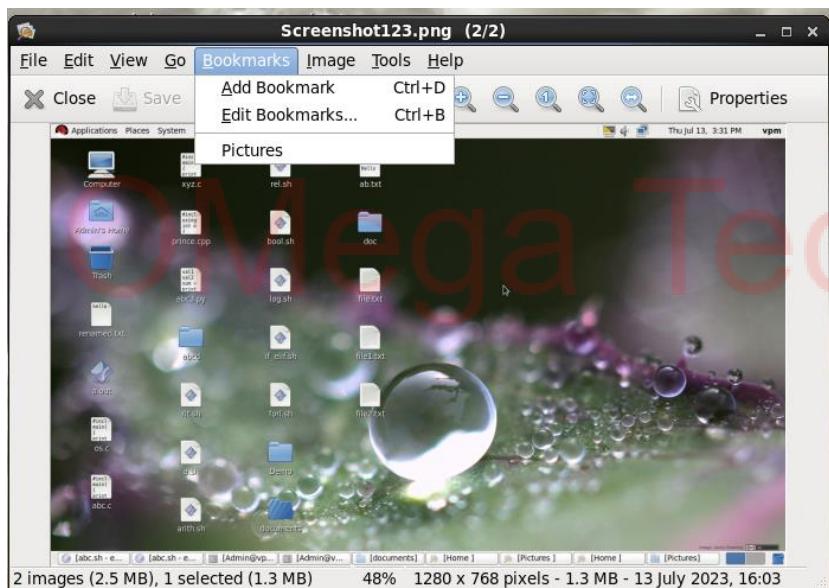
5. **The GO and BOOKMARK menus** are used for navigation purposes as in image as well as through system.



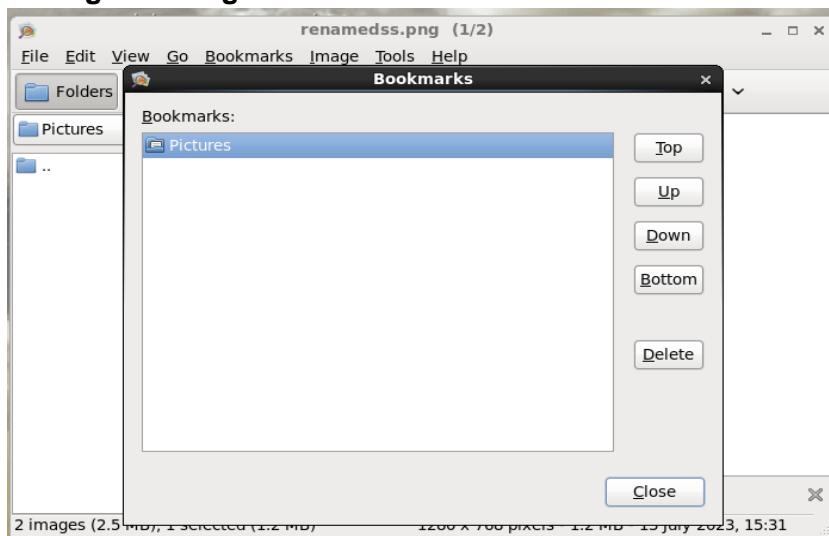
Using the back option, we can go back to our current folder.



BOOKMARK

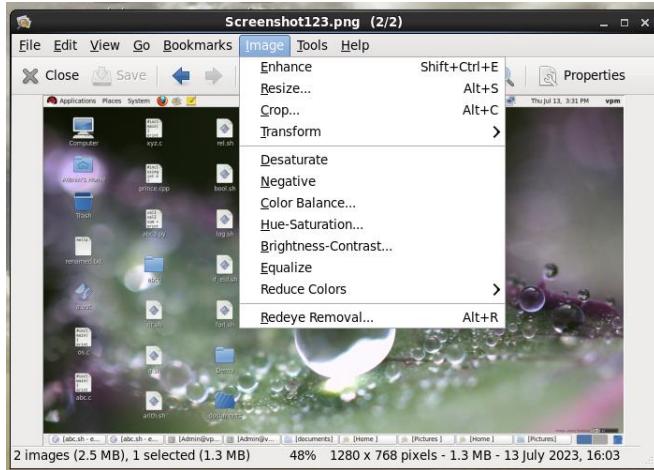


Adding & Editing Bookmarks.



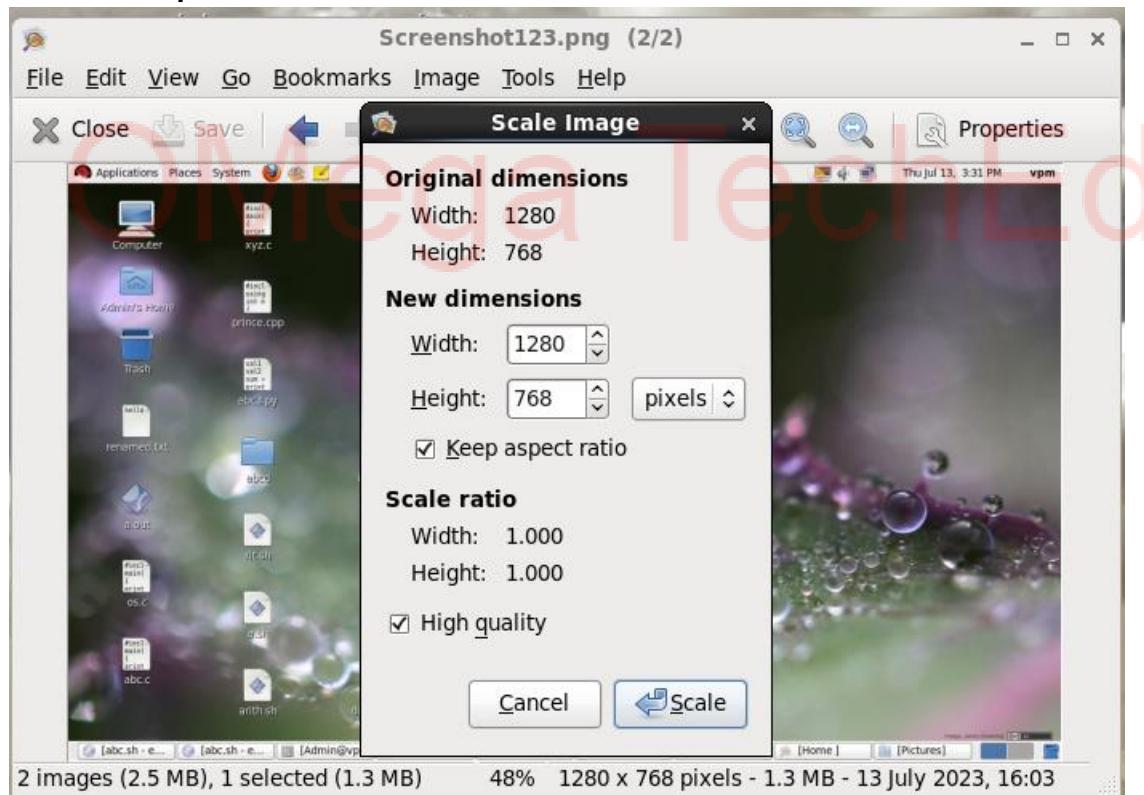
6. **The Image Menu:** This menu provides several options for image manipulation.

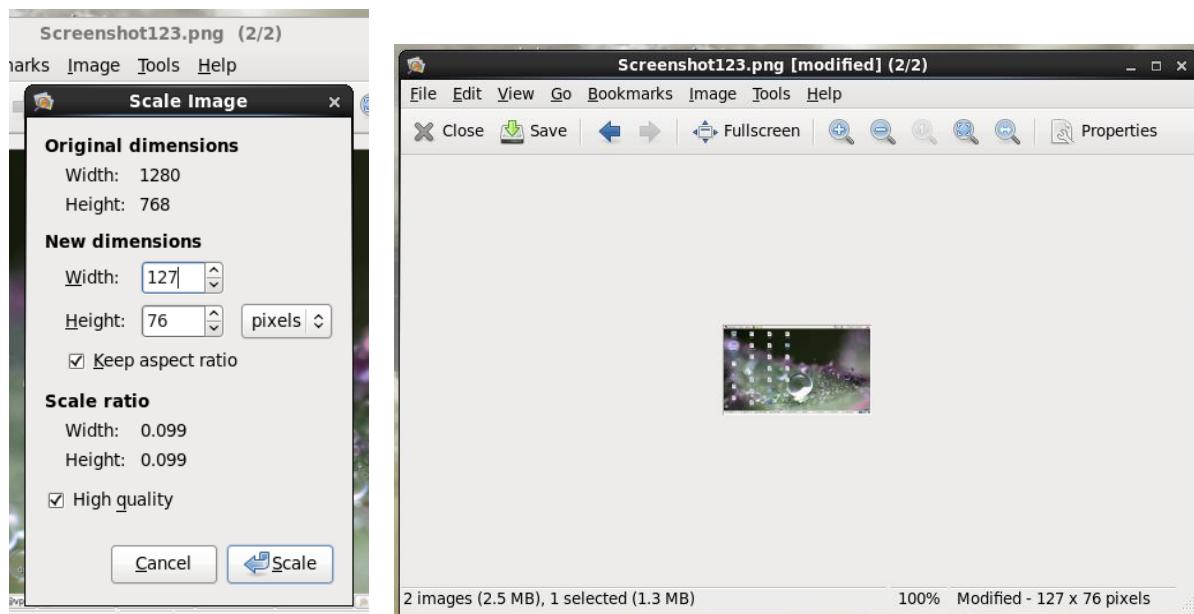
The Image Menu:



7. **Resize option** gives as facility to define new dimension to image.

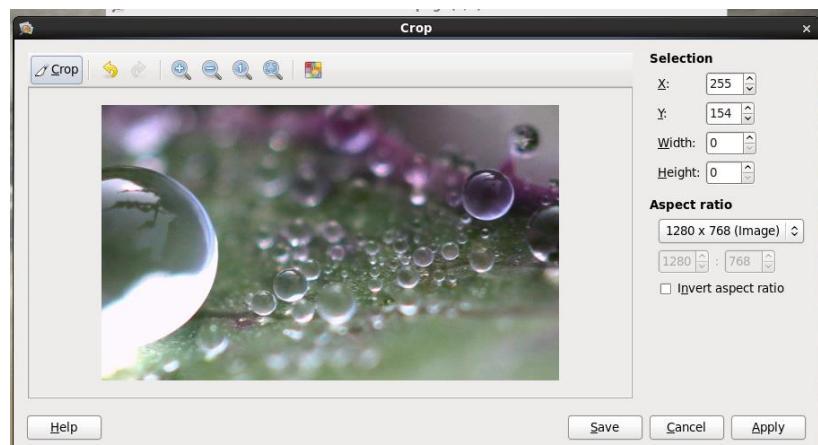
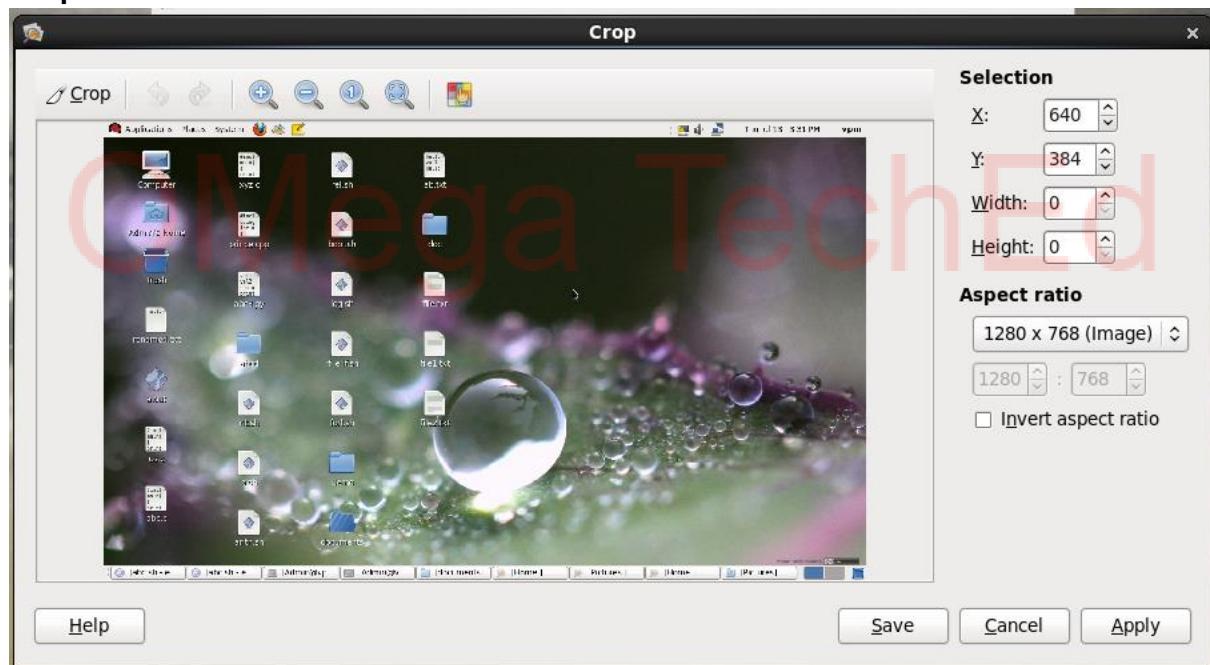
Resize option

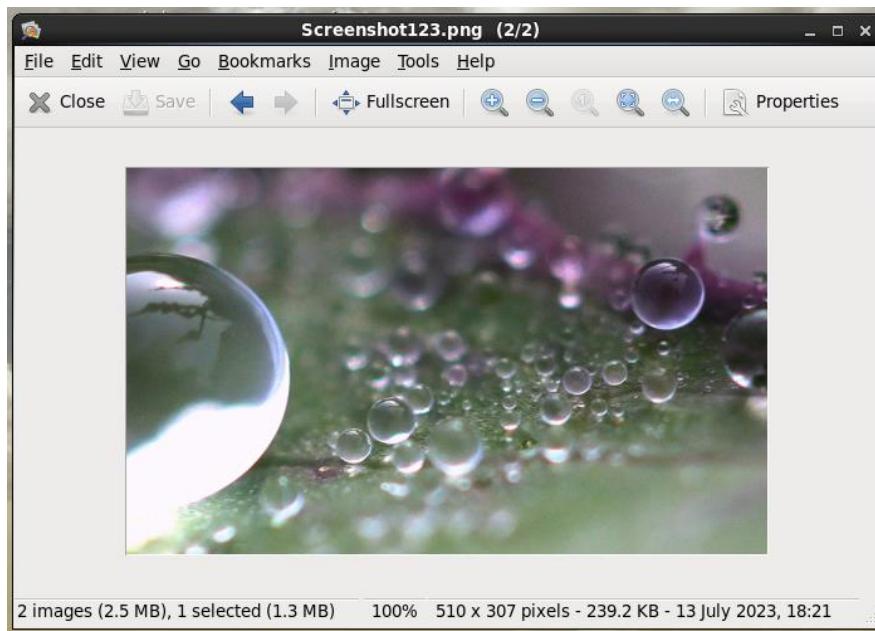




8. **Crop** allows user to crop the image and remove unwanted portions of image.

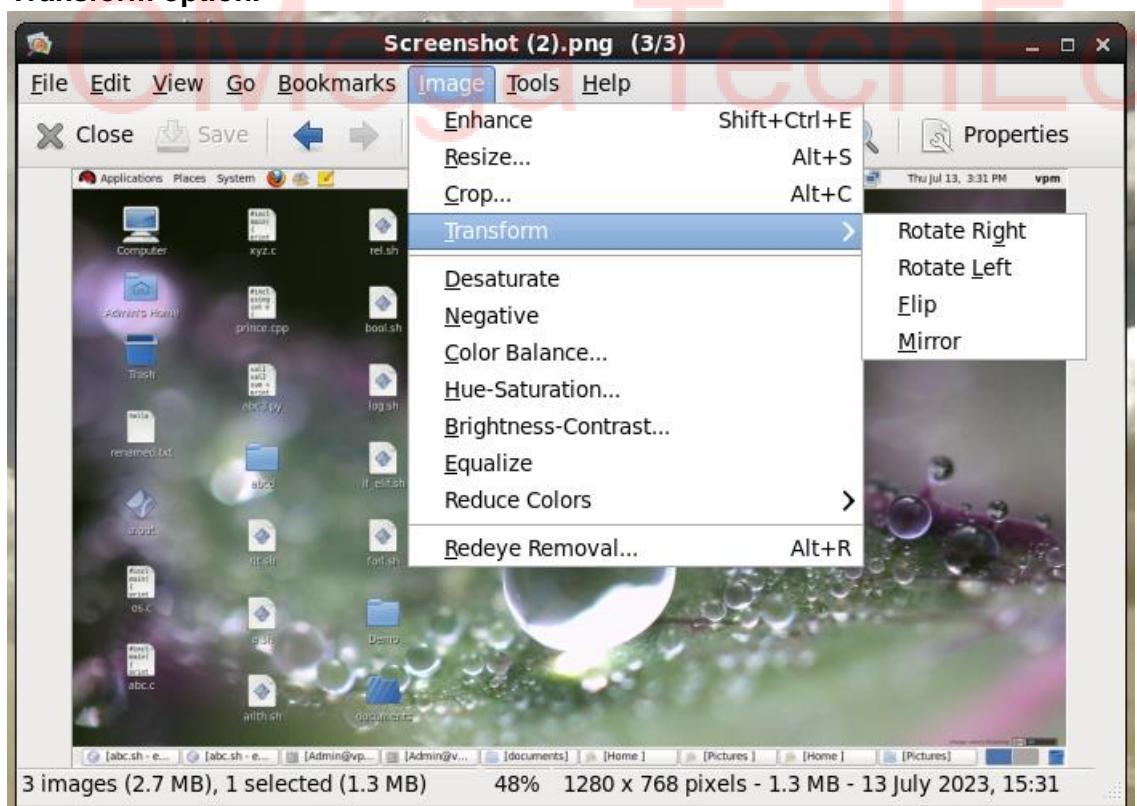
Crop



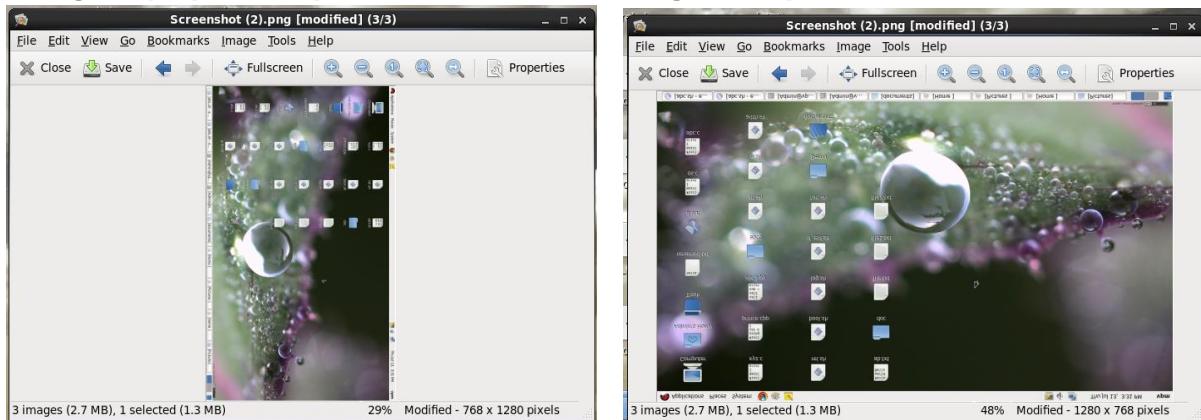


8. **Transform option** is used to rotate image Right, Left or Flip and can create mirror of image.

Transform option.

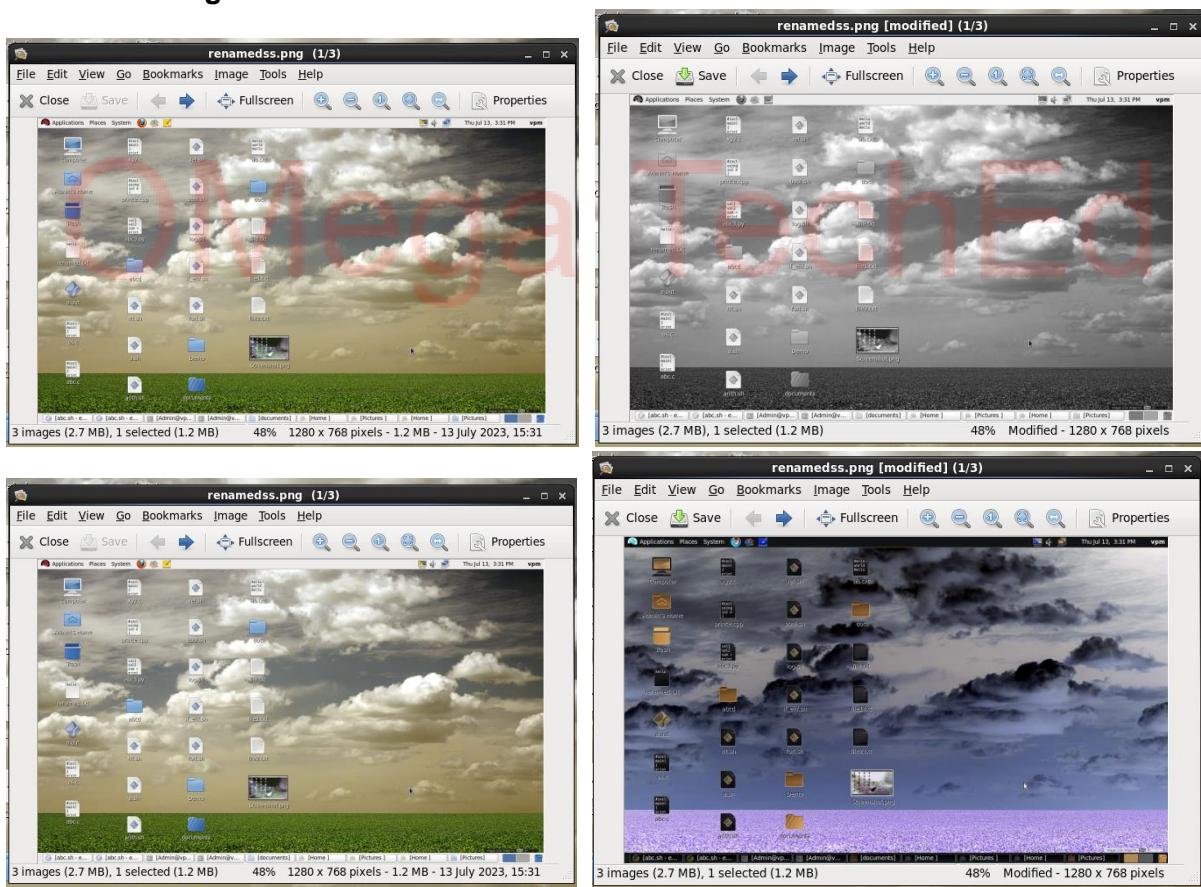


Using Crop option, implementation of Rotate Right & Flip.



10 Desaturate and Negative, automatically desaturate (Black and White) and negates the selected image.

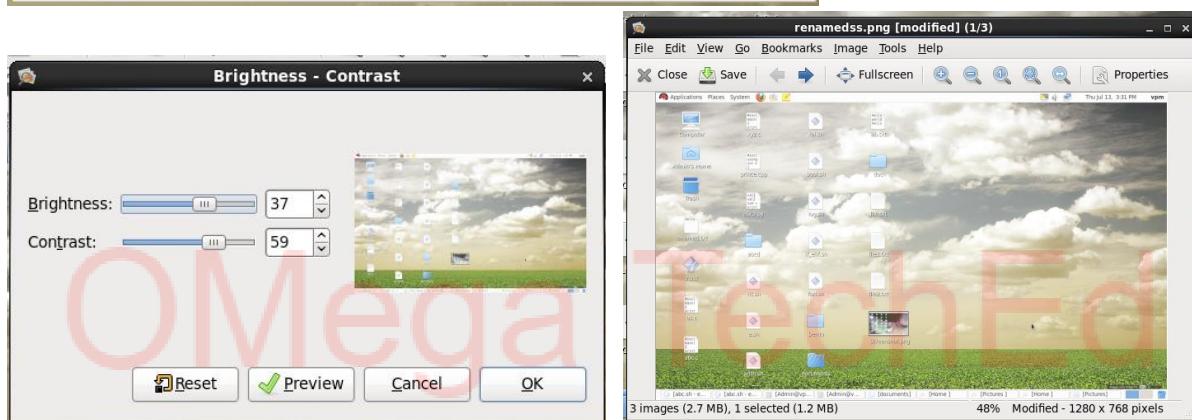
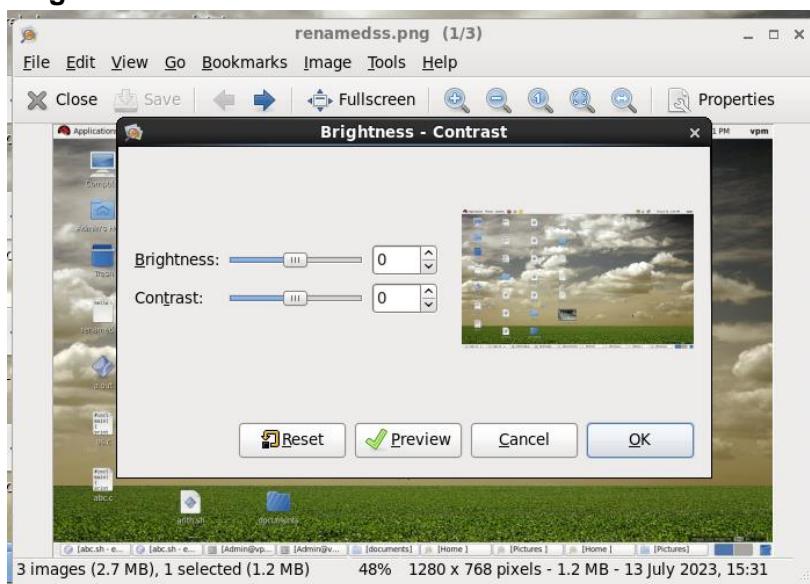
Desaturate Negative



We can adjust **Brightness and Contrast** using the option. Also enhance the image by using **Equalize** option. We can **Reduce** the color as Web Palate or Black and White.

Brightness & Contract, Equalize, Reduce colors.

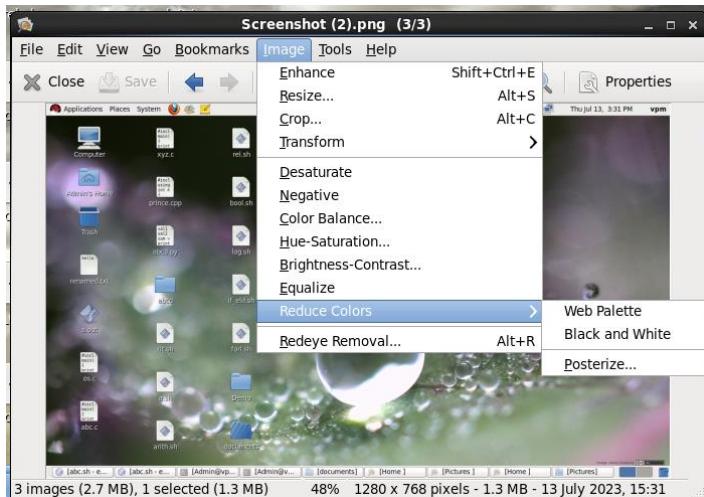
Brightness & Contract



Equalize



Reduce colors



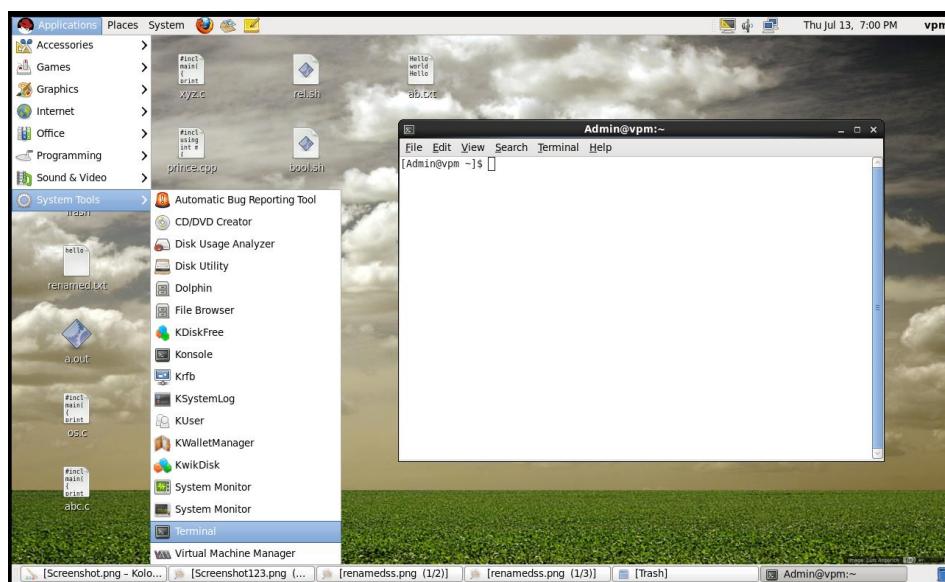
Using Reduce Colors options, Implementation of Black and White.



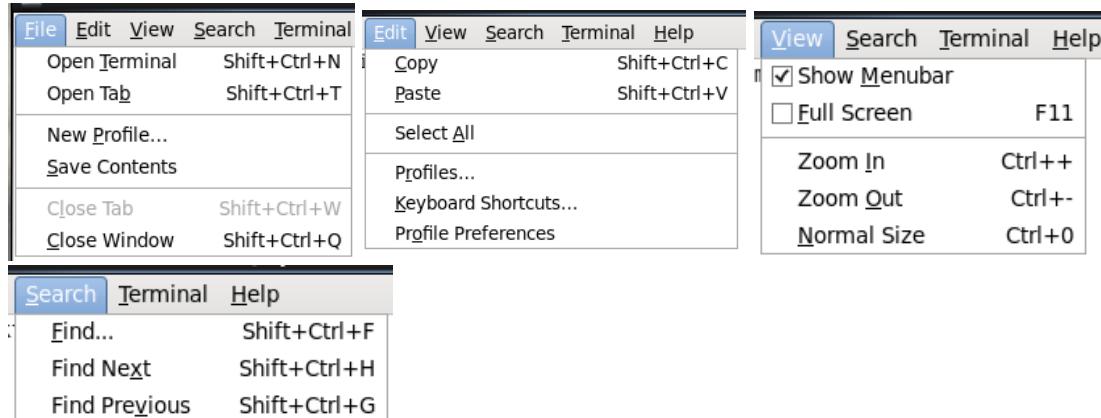
Qnnecta TechEd

4c) Working with terminal.

The terminal editor comes with following Menus.

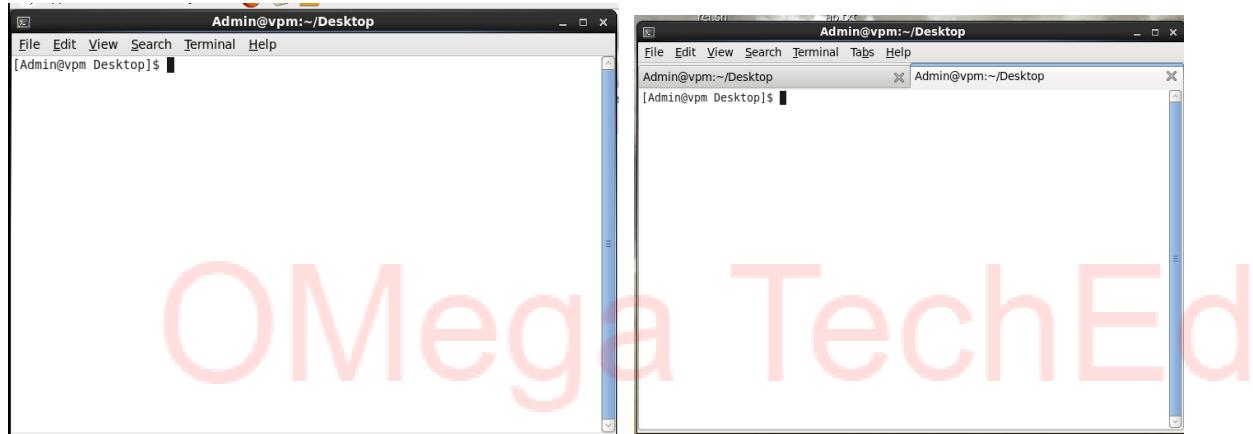


File, Edit, View, Search



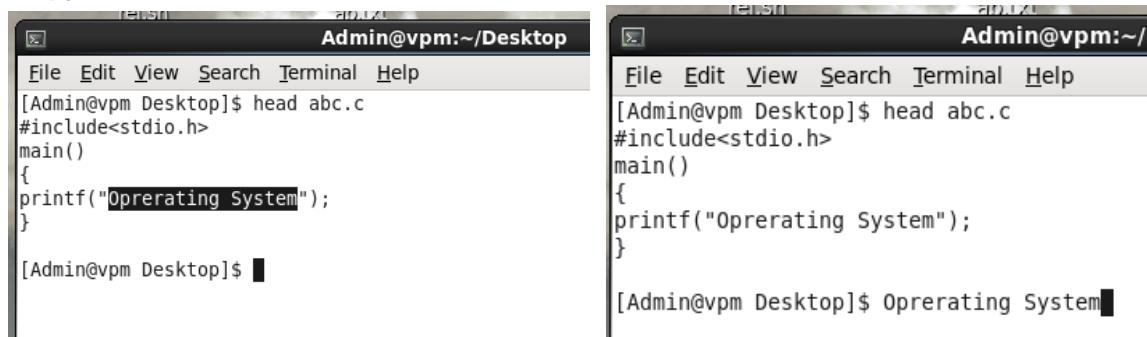
File: The **File Menu** has options for opening a new terminal.

1.File: Open new terminal & Open Tab

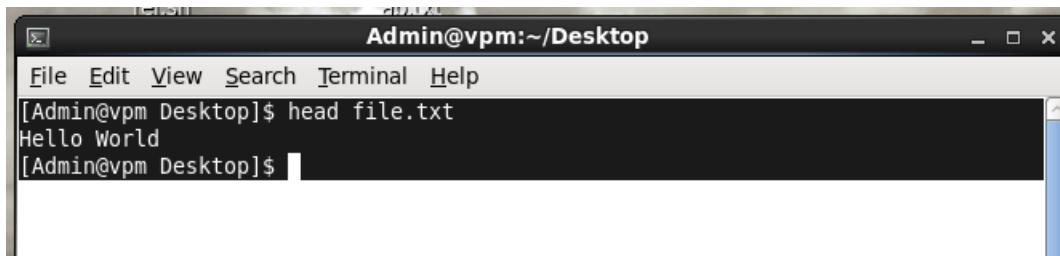


Edit: Like any normal edit menu provides facility for **COPY, PASTE, SELECTION** facilities.

Copy & Paste

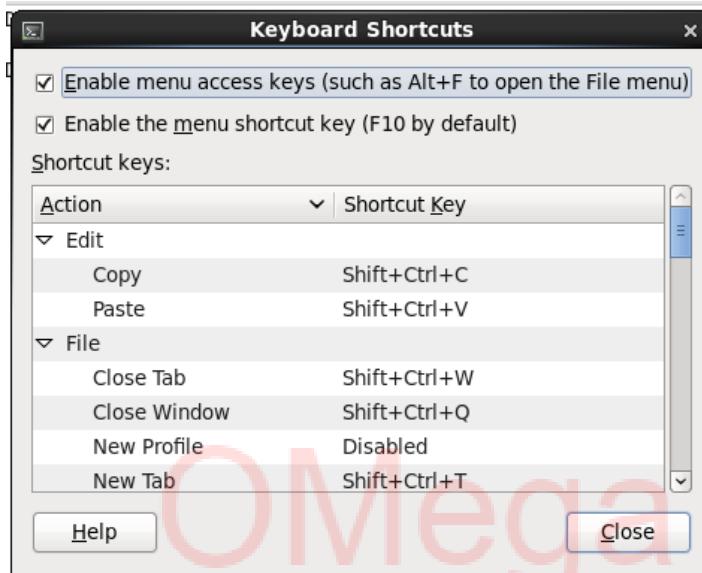


Selections

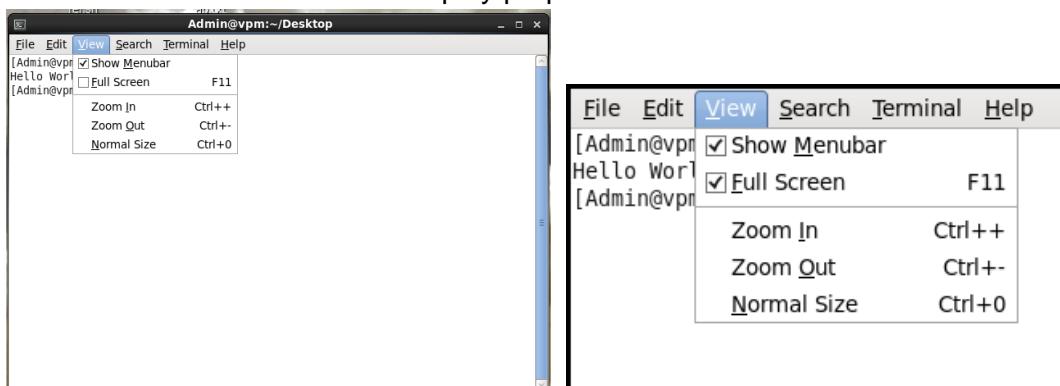


```
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ head file.txt
Hello World
[Admin@vpm Desktop]$
```

Keyboards Shortcuts

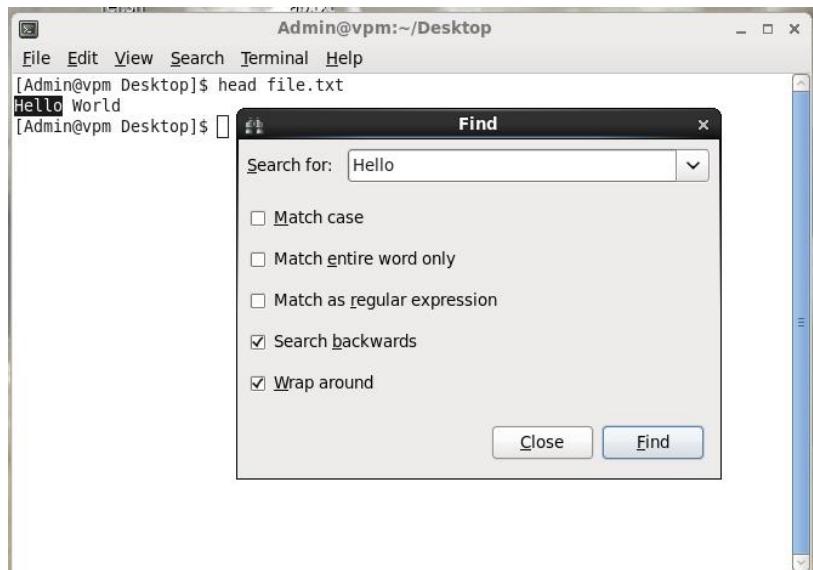


View: In this Menu the editors display properties are handled.



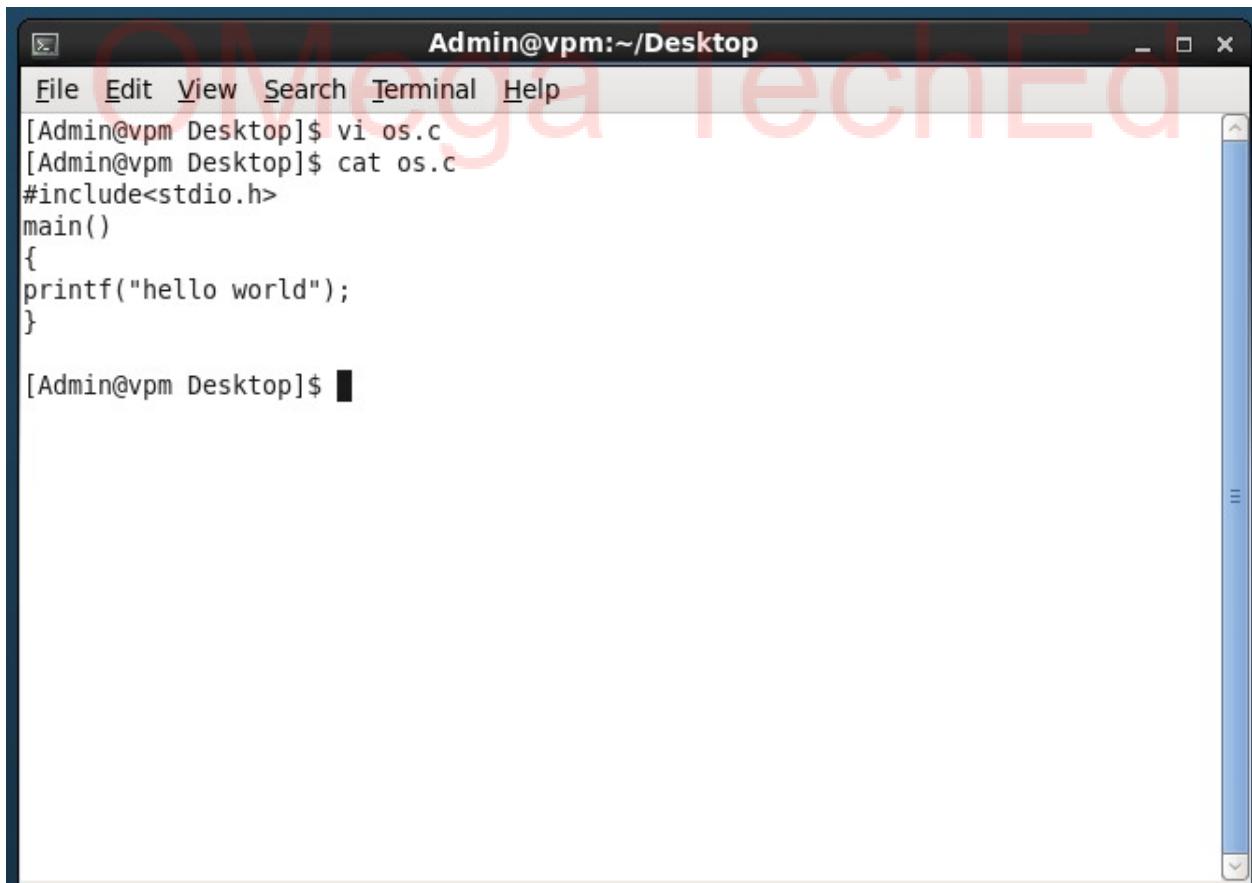
Search: The search menu provides an option for finding the text which is typed in terminal.

Search



Terminal: This menu allows user to set the size of terminal windows as well as to set the title and Character encoding.

Help: It is used to get the editing instructions for new users about how to handle Terminal.



```
Admin@vpm:~/Desktop
```

```
File Edit View Search Terminal Help
```

```
[Admin@vpm Desktop]$ vi os.c
```

```
[Admin@vpm Desktop]$ cat os.c
```

```
#include<stdio.h>
```

```
main()
```

```
{
```

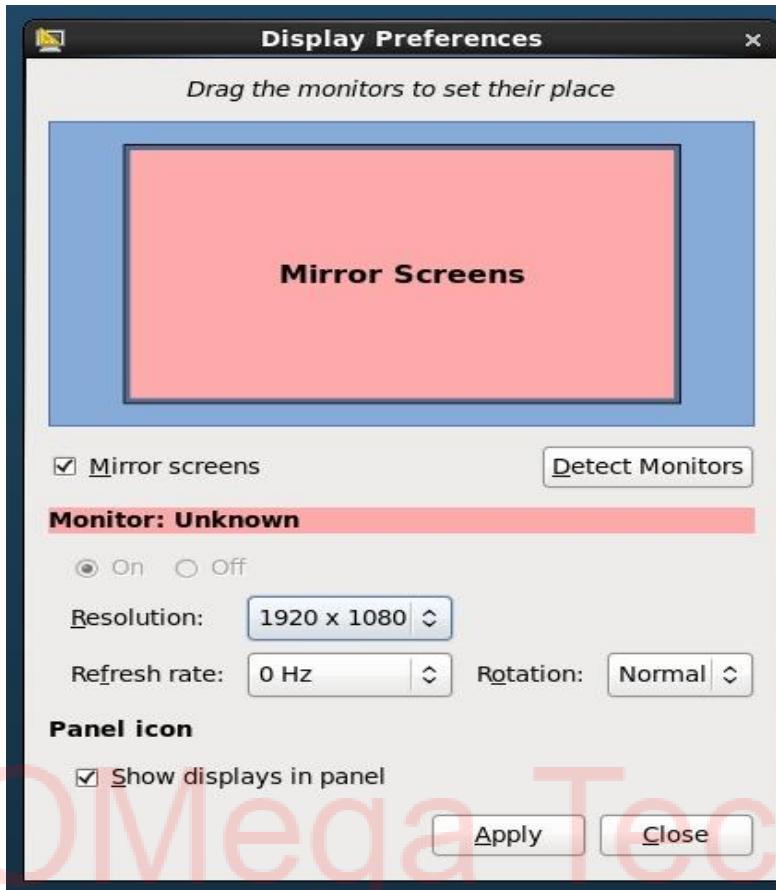
```
printf("hello world");
```

```
}
```

```
[Admin@vpm Desktop]$
```

4d) Adjusting display resolution.

Click on the System menu> Select Preferences> click display to open the display preferences.



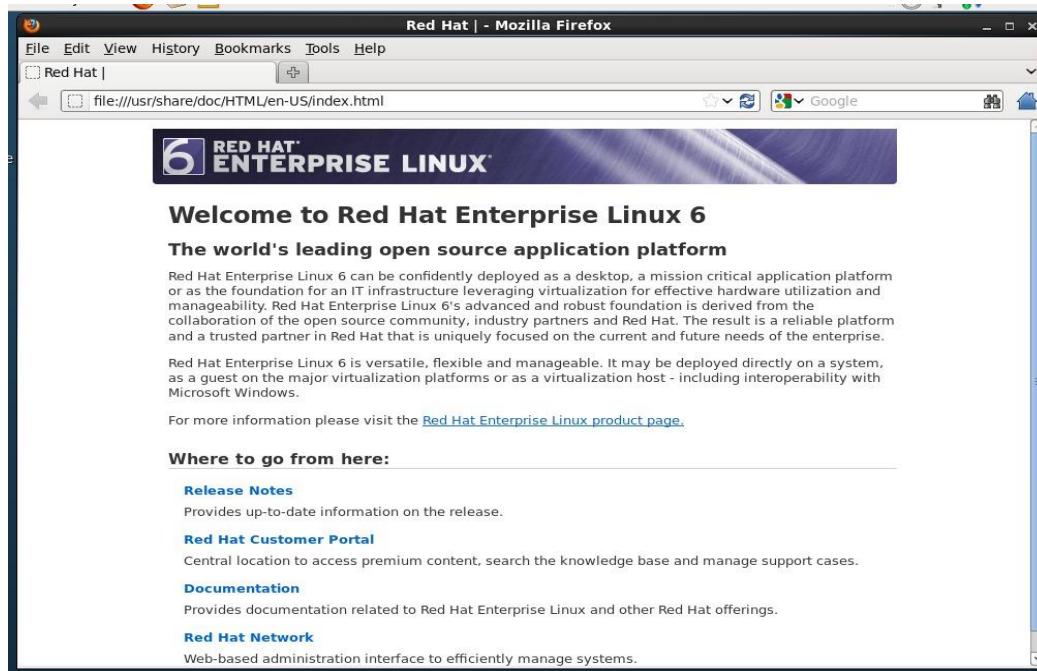
In **Display preferences** select the display whose resolution is to be changed if the several monitors are attached. Select the resolution from the Resolution drop-down list. Also, here we can set the refresh rate of monitor with other available options.

After finishing the changes Click on Apply button to save the changes.

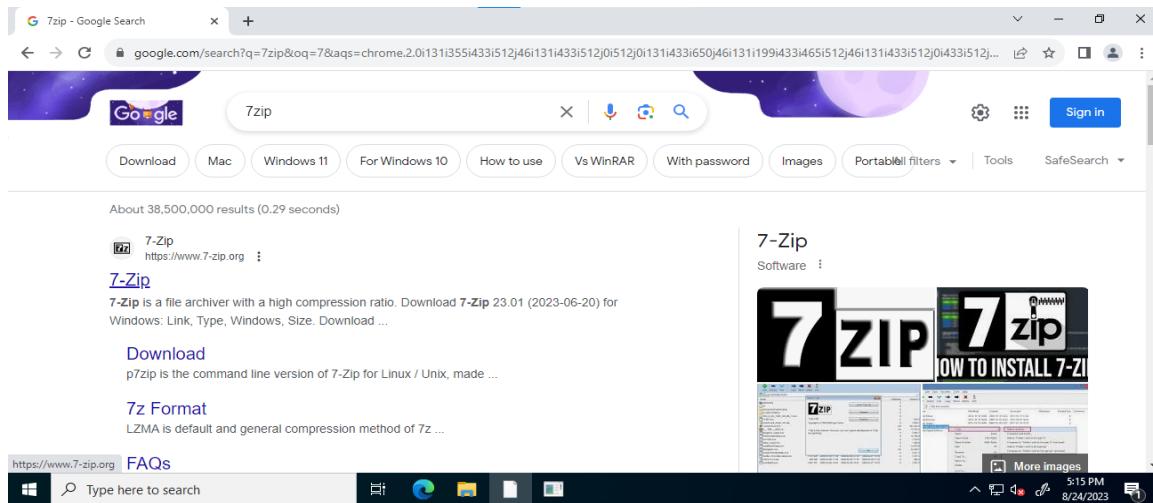
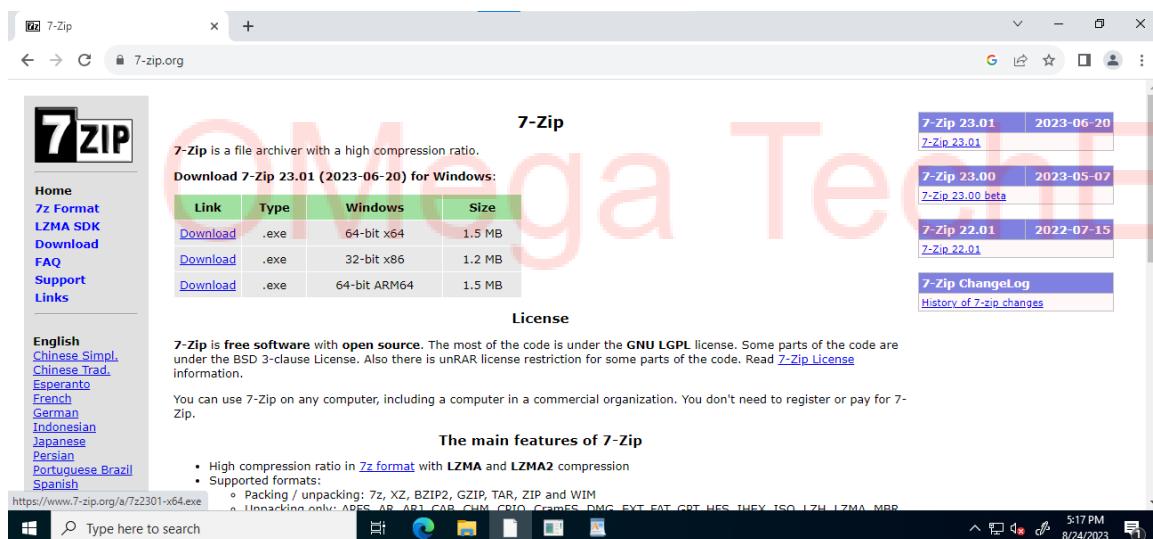
4e) Using the browsers.

A web browser is a type of software that allows us to find and view websites on the internet.

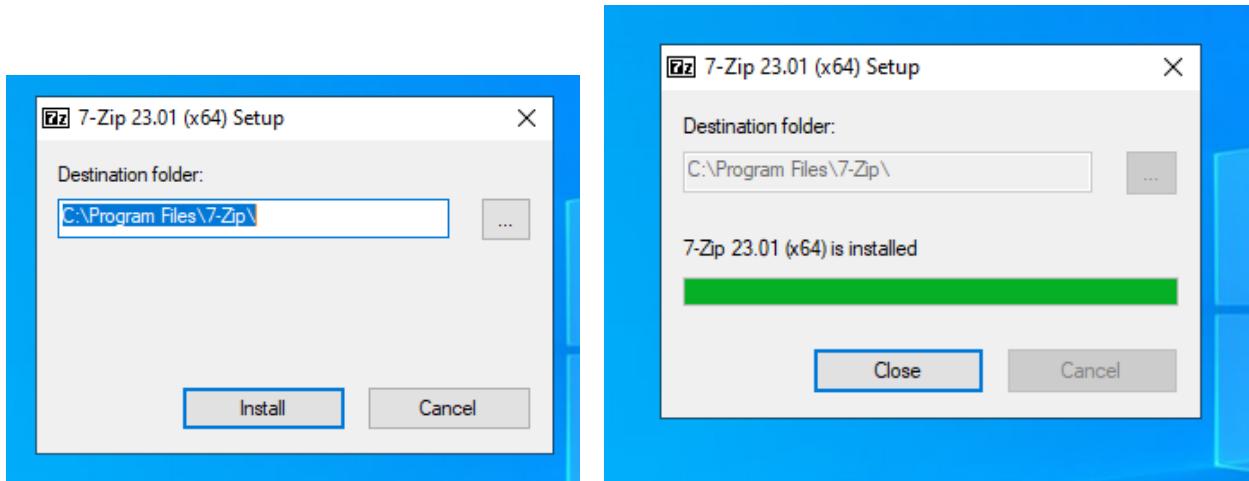
Example: Google Chrome, Internet Explorer, Safari, and Mozilla Firefox.



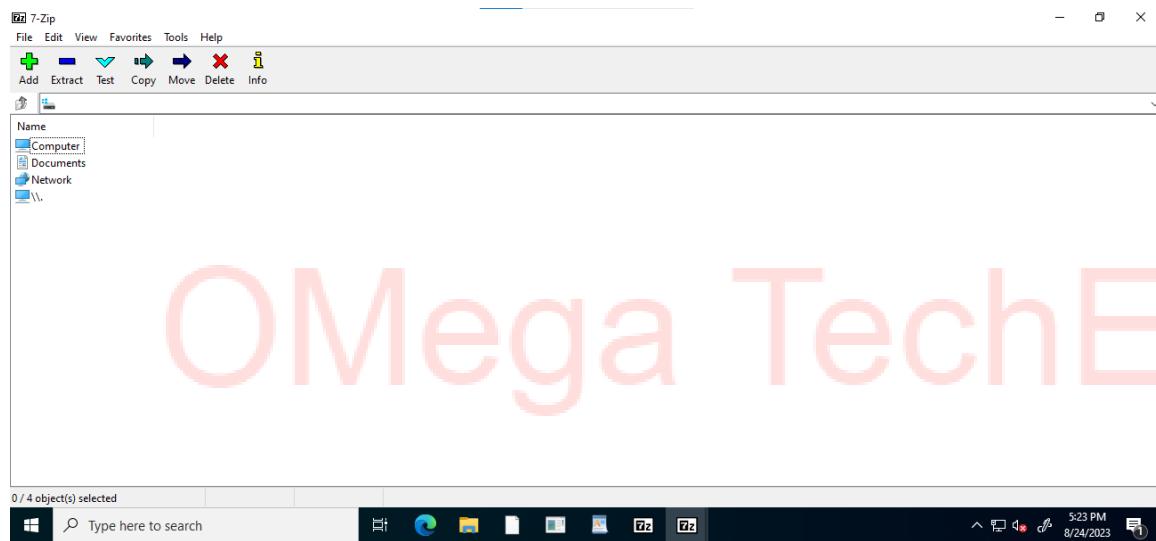
OMega TechEd

Practical 5**5a) Installing utility software 7zip on windows.****Step 1:** - Search for 7zip software on the Google.**Step:-** Select Download option on the website.

Step 3:- Click the Install button and installed it then close it.



Step 4: - Now our utility software 7zip is installed successfully!!



Practical 6

6a) Running C/C++/Python programs in Linux.

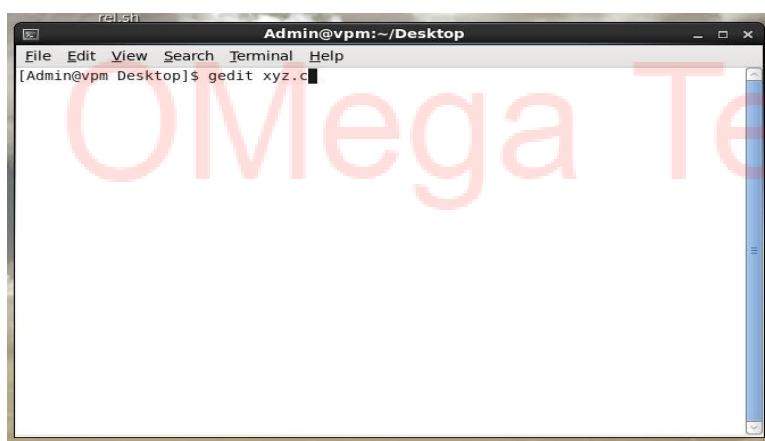
C Program in Linux: -

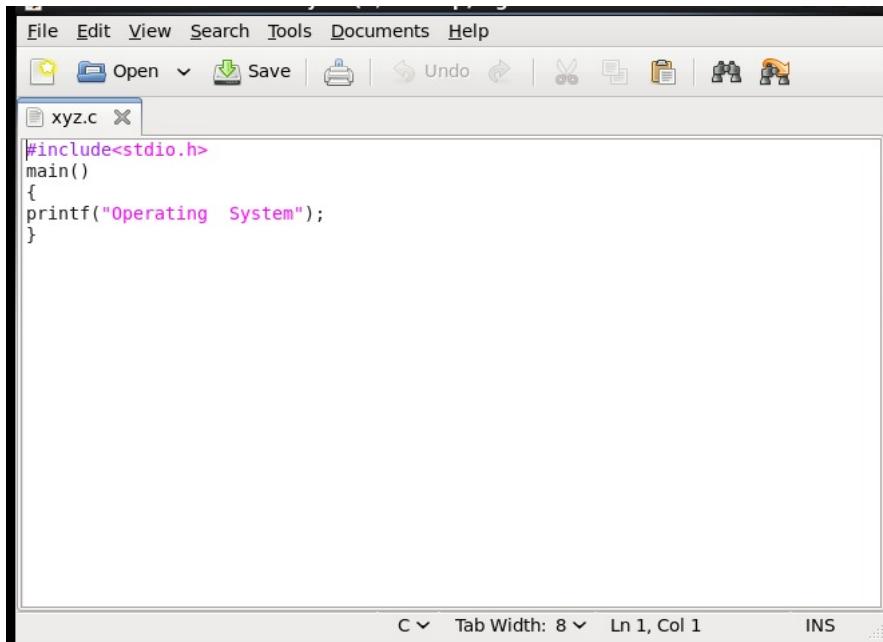
Method 1:

1. Use the vim editor. Open file using vim file. c (file name can be anything but it should end with dot c extension) command.
2. Press i to go to insert mode. Type in your program.
3. Press Esc button and then type: wq. It will save the file.
4. Compile using gcc file.c.
5. To Run the program: type ./a.out

Method 2:

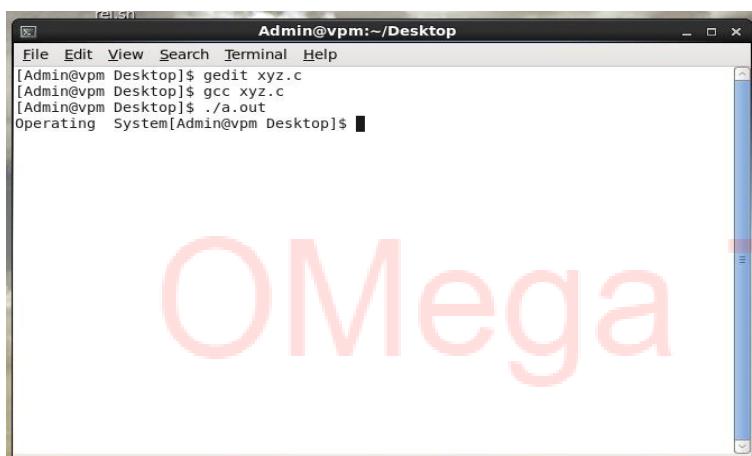
1. Open the terminal.
2. Create file using gedit for example gedit xyz.c
3. After this new window opens, type your C program Save with CTRL+S and close the window.
4. Compile using gcc file.c.
5. To Run the program: type ./a.out





```
File Edit View Search Tools Documents Help
Open Save Undo Redo Cut Copy Paste Find Replace
xyz.c
#include<stdio.h>
main()
{
printf("Operating System");
}
```

C Tab Width: 8 Ln 1, Col 1 INS



```
File Edit View Search Terminal Help
Admin@vpm:~/Desktop
[Admin@vpm Desktop]$ gedit xyz.c
[Admin@vpm Desktop]$ gcc xyz.c
[Admin@vpm Desktop]$ ./a.out
Operating System[Admin@vpm Desktop]$
```

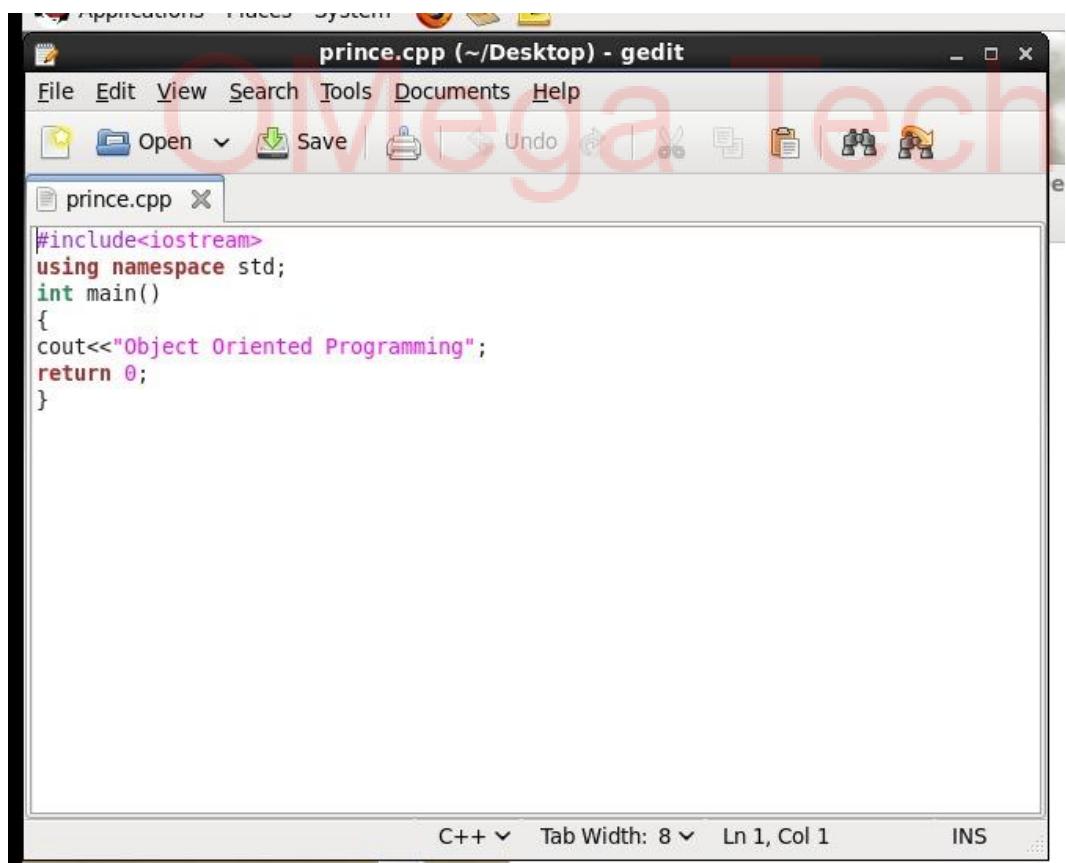
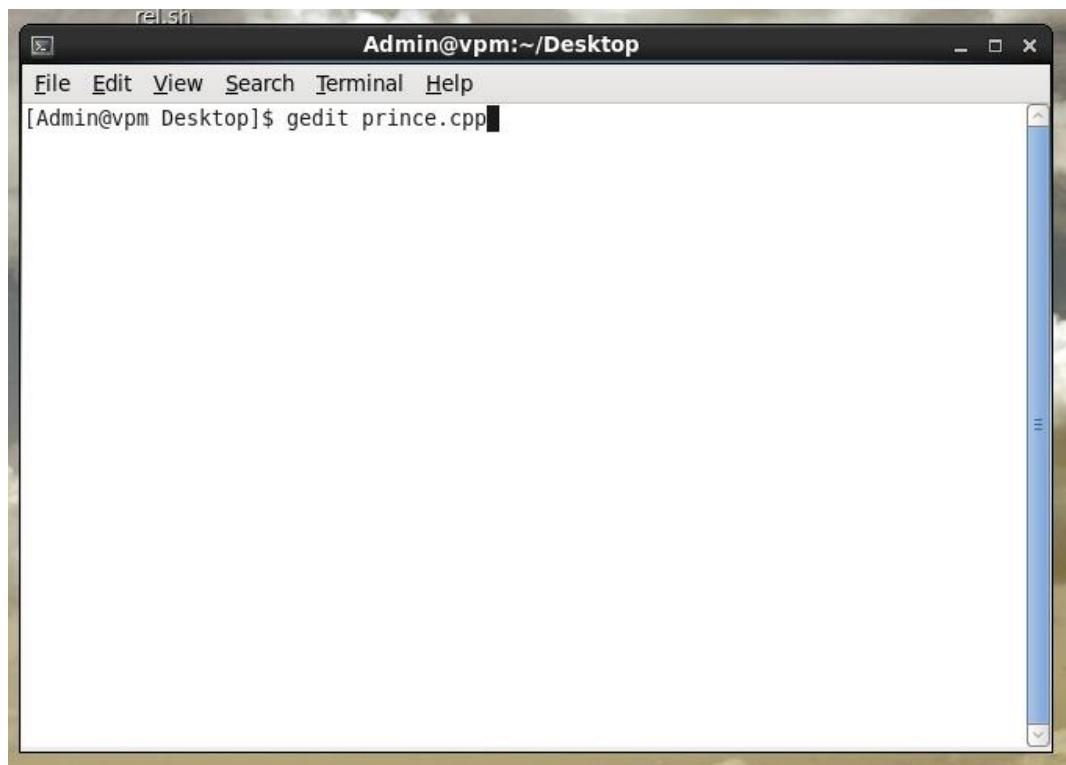
b) C++ Program in Linux: -

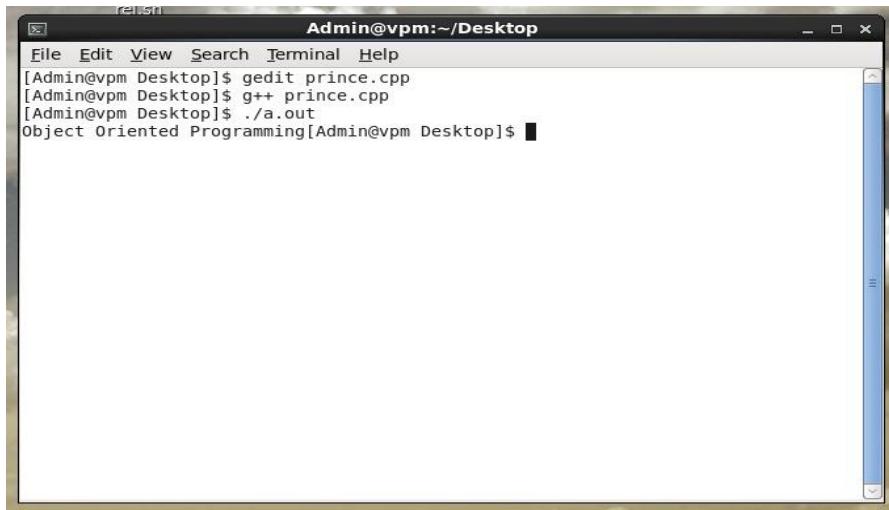
Method 1:

1. Use the vim editor. Open file using vim filename. cpp
2. Press i to go to insert mode. Type in your program.
3. Press Esc button and then type: wq. It will save the file. ...
4. Compile using g++ filename.c.
5. To Run the program: type ./a.out

Method 2:

1. Open the terminal.
2. Create file using gedit for example gedit prince.cpp
3. After this new window opens, type your C program Save with CTRL+S and close the window.
4. Compile using g++ filename.cpp.
5. To Run the program: type ./a.out





Admin@vpm:~/Desktop

```
[Admin@vpm Desktop]$ gedit prince.cpp
[Admin@vpm Desktop]$ g++ prince.cpp
[Admin@vpm Desktop]$ ./a.out
Object Oriented Programming[Admin@vpm Desktop]$
```

Note: gcc is used to compile C program while g++ is used to compile C++ program.

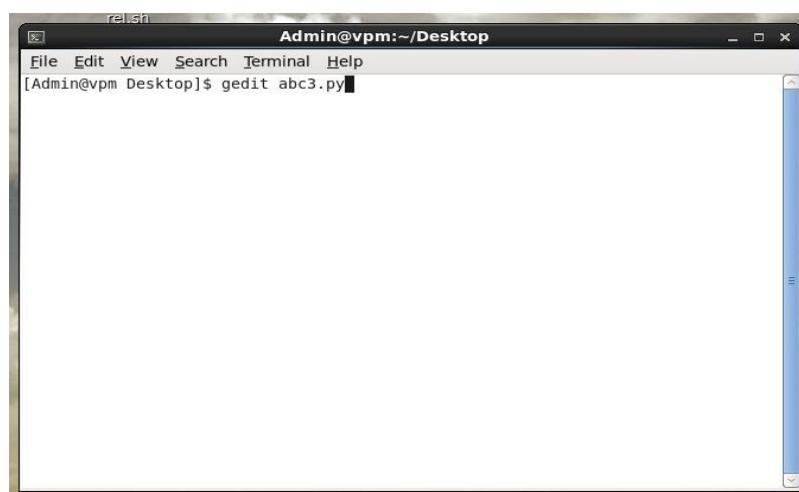
c) Python Program in Linux: -

Method 1:

- a. Use the vim editor. Open file using, vim filename.py
- b. Press i to go to insert mode. Type in your program.
- c. Press Esc button and then type: wq. It will save the file. ...
- d. To Run the program: type python filename.py

Method 2:

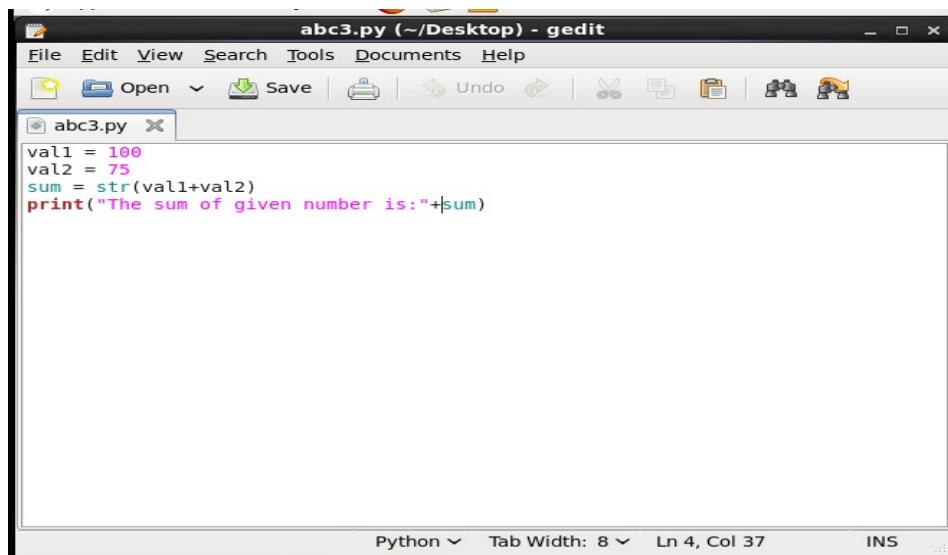
1. Open the terminal.
2. Create file using gedit for example gedit abc.py
3. After this new window opens, type your python program Save with CTRL+S and close the window.
4. To Run the program: type python filename.py



Admin@vpm:~/Desktop

```
[Admin@vpm Desktop]$ gedit abc3.py
```

SYIT SEM-III OPERATING SYSTEM BY: MEGHA SHARMA



abc3.py (~/Desktop) - gedit

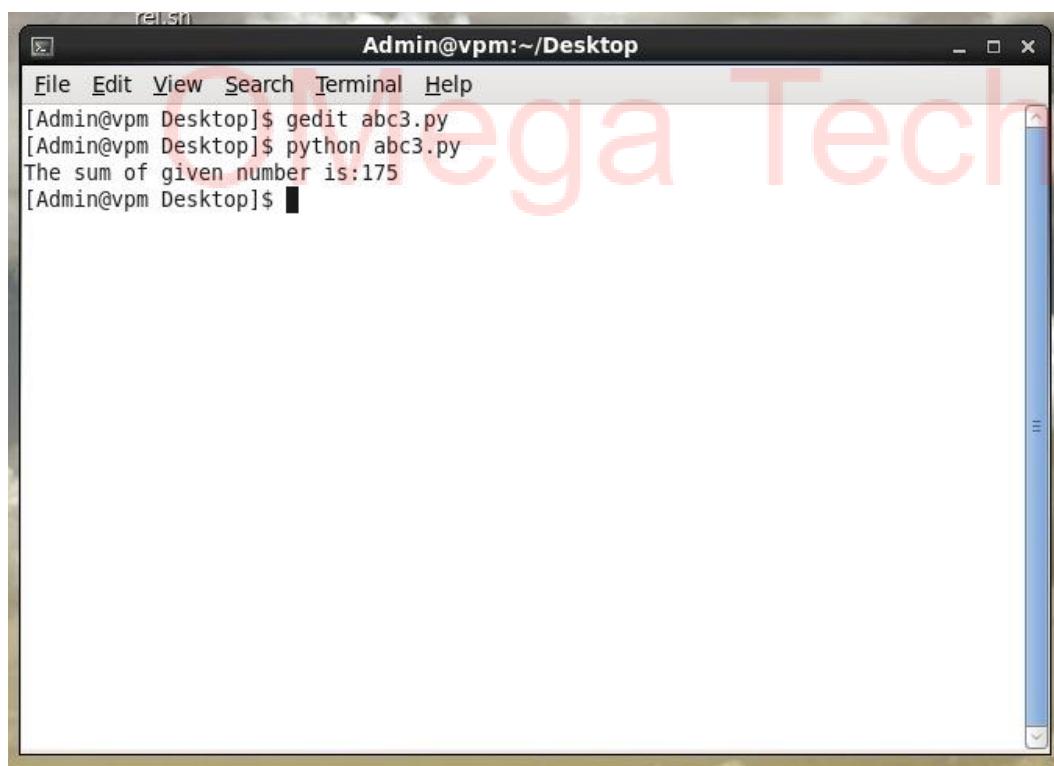
File Edit View Search Tools Documents Help

abc3.py

```
val1 = 100
val2 = 75
sum = str(val1+val2)
print("The sum of given number is:"+sum)
```

Python Tab Width: 8 Ln 4, Col 37 INS

This screenshot shows a Gedit text editor window with the file 'abc3.py' open. The code calculates the sum of two variables, val1 and val2, and prints the result. The code is highlighted in different colors: val1 and val2 are blue, sum is red, and the print statement is green. The status bar at the bottom shows 'Python' and 'Tab Width: 8'.



re13n Admin@vpm:~/Desktop

File Edit View Search Terminal Help

[Admin@vpm Desktop]\$ gedit abc3.py
[Admin@vpm Desktop]\$ python abc3.py
The sum of given number is:175
[Admin@vpm Desktop]\$

This screenshot shows a terminal window with the title 're13n' and the user 'Admin@vpm:~/Desktop'. The terminal shows the command 'python abc3.py' being run, followed by the output 'The sum of given number is:175'.

Practical 7

7. Introduction to Linux Shell Scripting

Welcome to Vim editor

1. **Vim editor** is available in almost all linux distributions.
2. It works the same across different platforms and distributions.
3. It is user friendly.
4. Commands are case sensitive.

Using Vim Editor:

1. Open the Vim Editor:

Launch your Linux terminal and use the vim command followed by the name of the file you want to edit.

2. Entering Insert Mode:

Press the "i" key on the keyboard to enter insert mode.

In **insert mode**, you can type and edit text.

3. Editing Text:

In insert mode, you can freely type and **edit your content** as needed.

4. Exiting Insert Mode:

To exit insert mode and return to normal mode, press the **ESC** key.

5. Saving and Exiting:

To **save changes** and **exit Vim**, type **:wq** and **press Enter**.

6. Running a Shell Script:

In the terminal, use the **sh** command to execute your shell script created using Vim editor:

Syntax:- sh <file name>

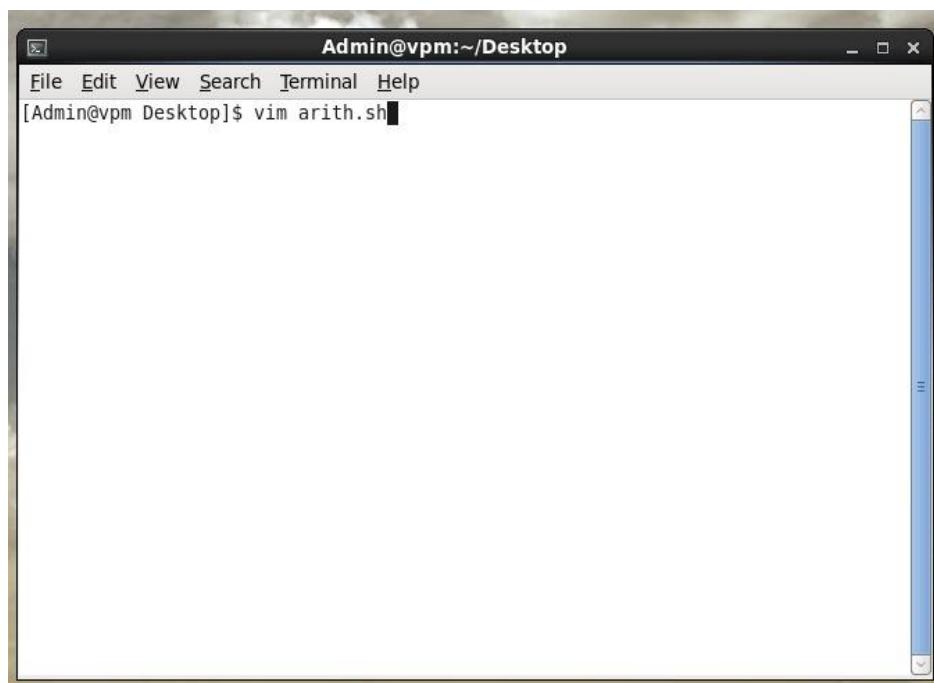
7a) Basic operators: - a) Arithmetic b) Relational c) Logical d) Boolean

Arithmetic Operators: - The following arithmetic operators are supported by Bourne Shell.

Assume variable **a** holds 10 and variable **b** holds 20 then –

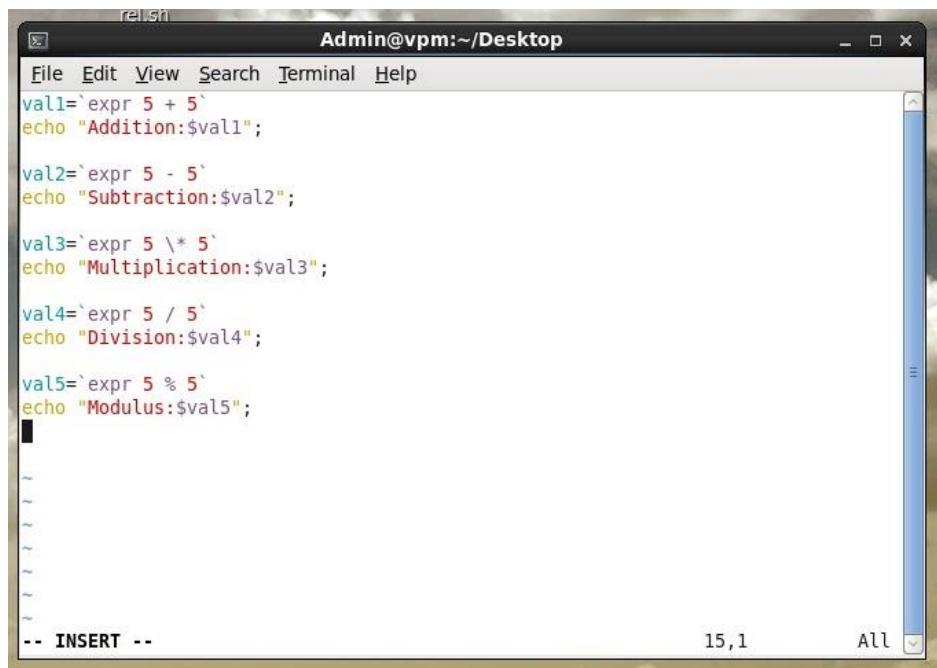
a) Operator	Description	Example
+(Addition)	Adds values on either side of the operator	`expr \$a + \$b` will give 30
-(Subtraction)	Subtracts right hand operand from left hand operand	`expr \$a - \$b` will give -10
*(Multiplication)	Multiplies values on either side of the operator	`expr \$a * \$b` will give 200
/(Division)	Divides left hand operand by right hand operand	`expr \$b / \$a` will give 2
% (Modulus)	Divides left hand operand by right hand operand and returns remainder	`expr \$b % \$a` will give 0
= (Assignment)	Assigns right operand in left operand	<code>a = \$b</code> would assign value of b into a
== (Equality)	Compares two numbers, if both are same then returns true.	<code>[\$a == \$b]</code> would return false.
!= (Not Equality)	Compares two numbers, if both are different then returns true.	<code>[\$a != \$b]</code> would return true.

It is very important to understand that all the conditional expressions should be inside square braces with spaces around them, for example `[$a == $b]` is correct whereas, `[$a==$b]` is incorrect.



The screenshot shows a terminal window with the following details:

- Title Bar:** Admin@vpm:~/Desktop
- Menu Bar:** File Edit View Search Terminal Help
- Command Line:** [Admin@vpm Desktop]\$ vim arith.sh



relish Admin@vpm:~/Desktop

```
File Edit View Search Terminal Help
val1=`expr 5 + 5`
echo "Addition:$val1";

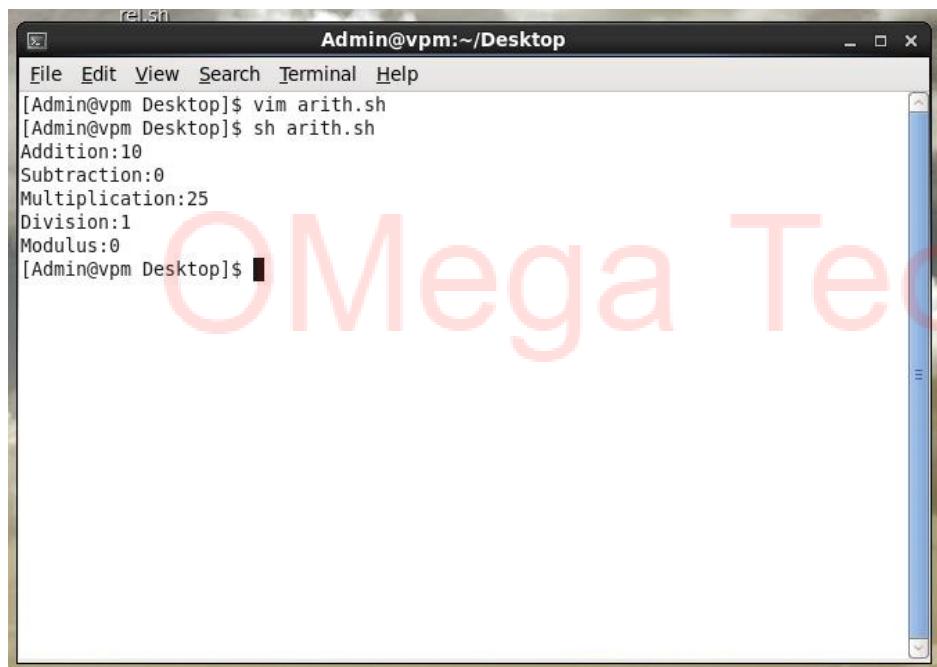
val2=`expr 5 - 5`
echo "Subtraction:$val2";

val3=`expr 5 \* 5`
echo "Multiplication:$val3";

val4=`expr 5 / 5`
echo "Division:$val4";

val5=`expr 5 % 5`
echo "Modulus:$val5";
~
```

-- INSERT -- 15,1 All



relish Admin@vpm:~/Desktop

```
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ vim arith.sh
[Admin@vpm Desktop]$ sh arith.sh
Addition:10
Subtraction:0
Multiplication:25
Division:1
Modulus:0
[Admin@vpm Desktop]$
```

b) Relational Operators: -

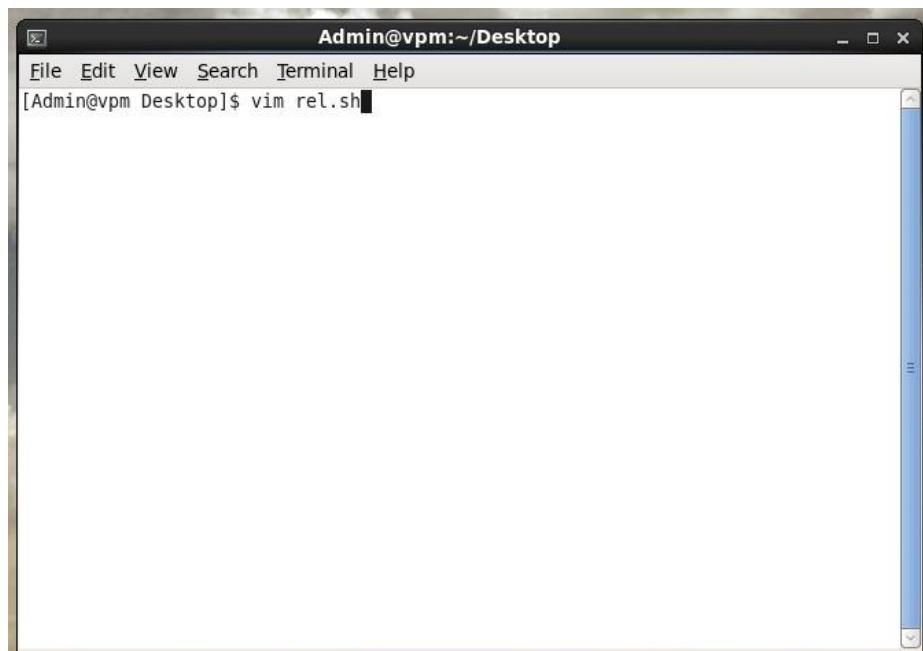
Bourne Shell supports the following relational operators that are specific to numeric values. These operators do not work for string values unless their value is numeric.

For example, the following operators will work to check a relation between 10 and 20 as well as in between "10" and "20" but not in between "ten" and "twenty".

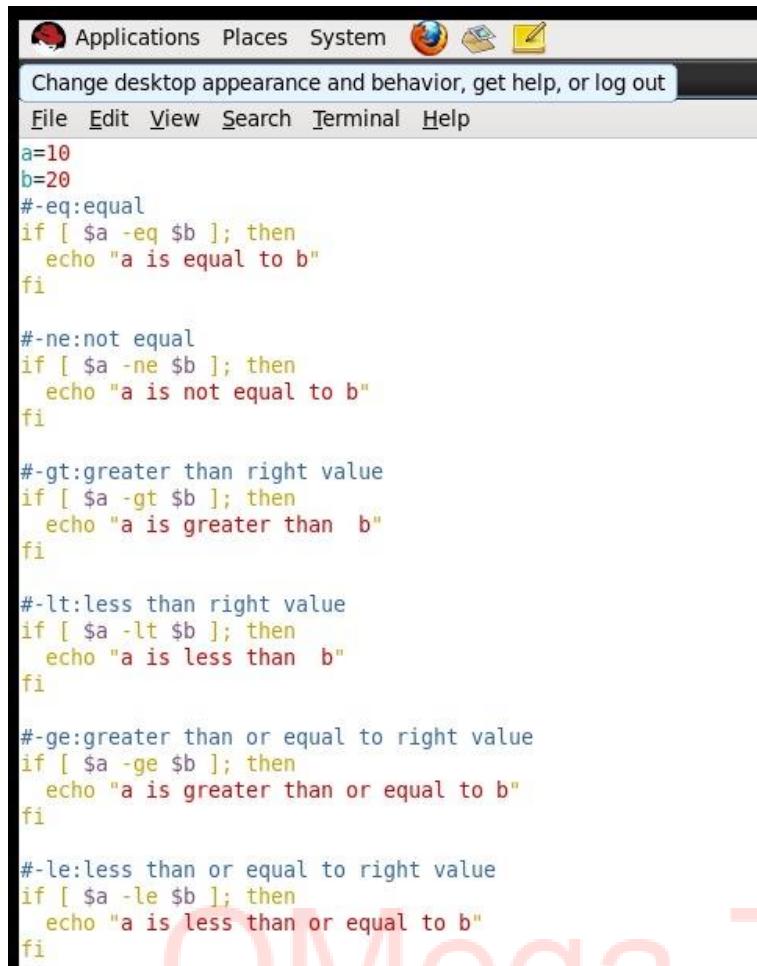
Assume variable **a** holds 10 and variable **b** holds 20 then –

Operator	Description	Example
-eq	Checks if the value of two operands are equal or not; if yes, then the condition becomes true.	[\$a -eq \$b] is not true.
-ne	Checks if the value of two operands are equal or not; if values are not equal, then the condition becomes true.	[\$a -ne \$b] is true.
-gt	Checks if the value of left operand is greater than the value of right operand; if yes, then the condition becomes true.	[\$a -gt \$b] is not true.
-lt	Checks if the value of left operand is less than the value of right operand; if yes, then the condition becomes true.	[\$a -lt \$b] is true.
-ge	Checks if the value of left operand is greater than or equal to the value of right operand; if yes, then the condition becomes true.	[\$a -ge \$b] is not true.
-le	Checks if the value of left operand is less than or equal to the value of right operand; if yes, then the condition becomes true.	[\$a -le \$b] is true.

It is very important to understand that all the conditional expressions should be placed inside square braces with spaces around them. For example, [\$a <= \$b] is correct whereas, [\$a <= \$b] is incorrect.



A screenshot of a terminal window titled "Admin@vpm:~/Desktop". The window has a standard Linux-style interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar. The main area of the terminal shows the command `[Admin@vpm Desktop]$ vim rel.sh` being typed. The terminal is running in a dark-themed window, and the text is white on a dark background.



Applications Places System

Change desktop appearance and behavior, get help, or log out

File Edit View Search Terminal Help

```
a=10
b=20
#-eq:equal
if [ $a -eq $b ]; then
  echo "a is equal to b"
fi

#-ne:not equal
if [ $a -ne $b ]; then
  echo "a is not equal to b"
fi

#-gt:greater than right value
if [ $a -gt $b ]; then
  echo "a is greater than b"
fi

#-lt:less than right value
if [ $a -lt $b ]; then
  echo "a is less than b"
fi

#-ge:greater than or equal to right value
if [ $a -ge $b ]; then
  echo "a is greater than or equal to b"
fi

#-le:less than or equal to right value
if [ $a -le $b ]; then
  echo "a is less than or equal to b"
fi
```



Admin@vpm:~/Desktop

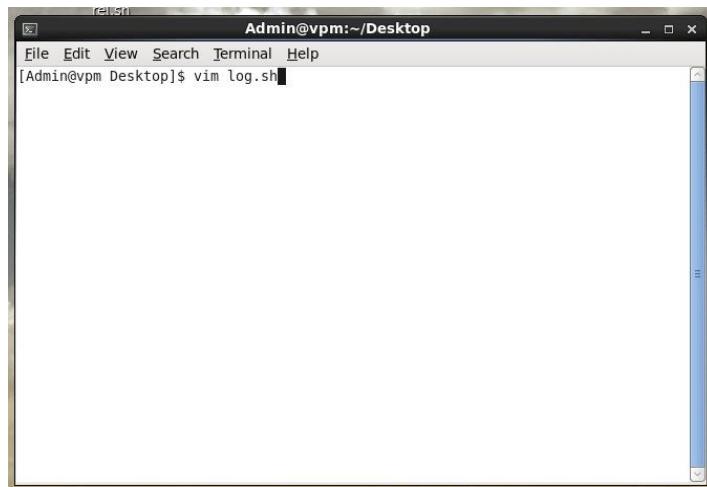
File Edit View Search Terminal Help

```
[Admin@vpm Desktop]$ vim rel.sh
[Admin@vpm Desktop]$ sh rel.sh
a is not equal to b
a is less than b
a is less than or equal to b
[Admin@vpm Desktop]$
```

c)Logical Operators: -

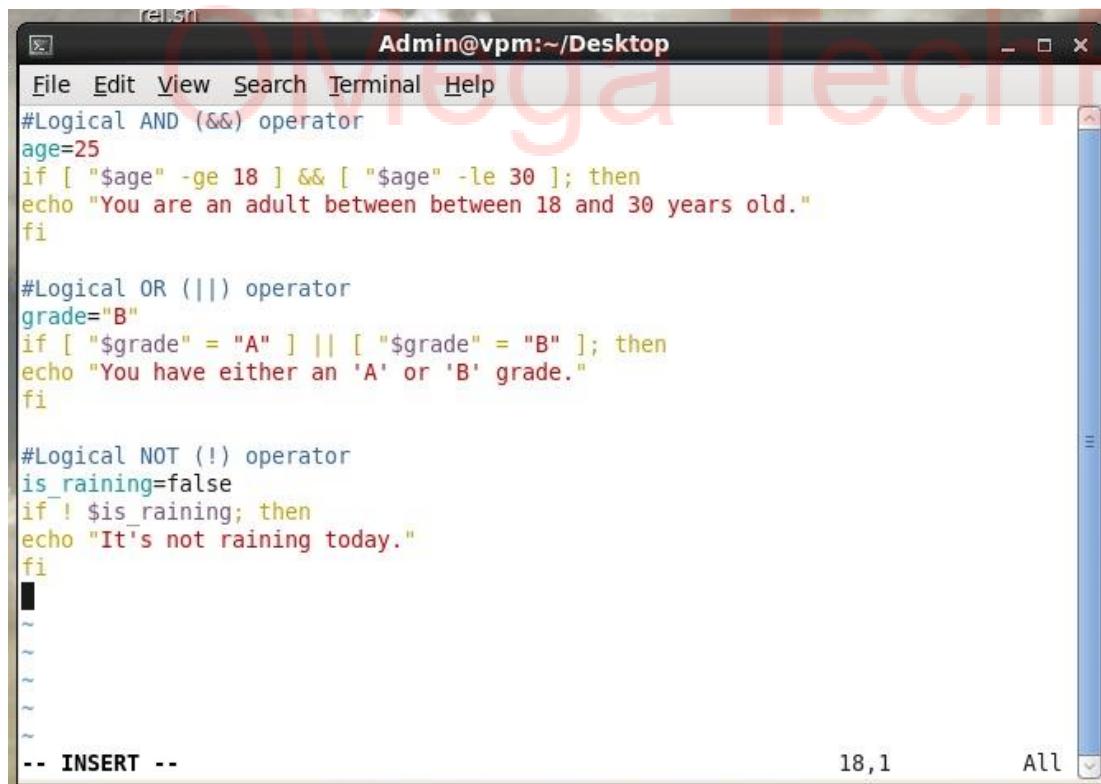
These are used to perform logical operations. They are of 3 types:

- **Logical AND (&&):** This is a binary operator, which returns true if both the operands are true otherwise returns false.
- **Logical OR (||):** This is a binary operator, which returns true if either of the operand is true or both the operands are true and return false if none of them is false.
- **Not Equal to (!):** This is a unary operator which returns true if the operand is false and returns false if the operand is true.



Admin@vpm:~/Desktop

```
[Admin@vpm Desktop]$ vim log.sh
```



relish

Admin@vpm:~/Desktop

```
File Edit View Search Terminal Help
```

```
#Logical AND (&&) operator
age=25
if [ "$age" -ge 18 ] && [ "$age" -le 30 ]; then
echo "You are an adult between between 18 and 30 years old."
fi

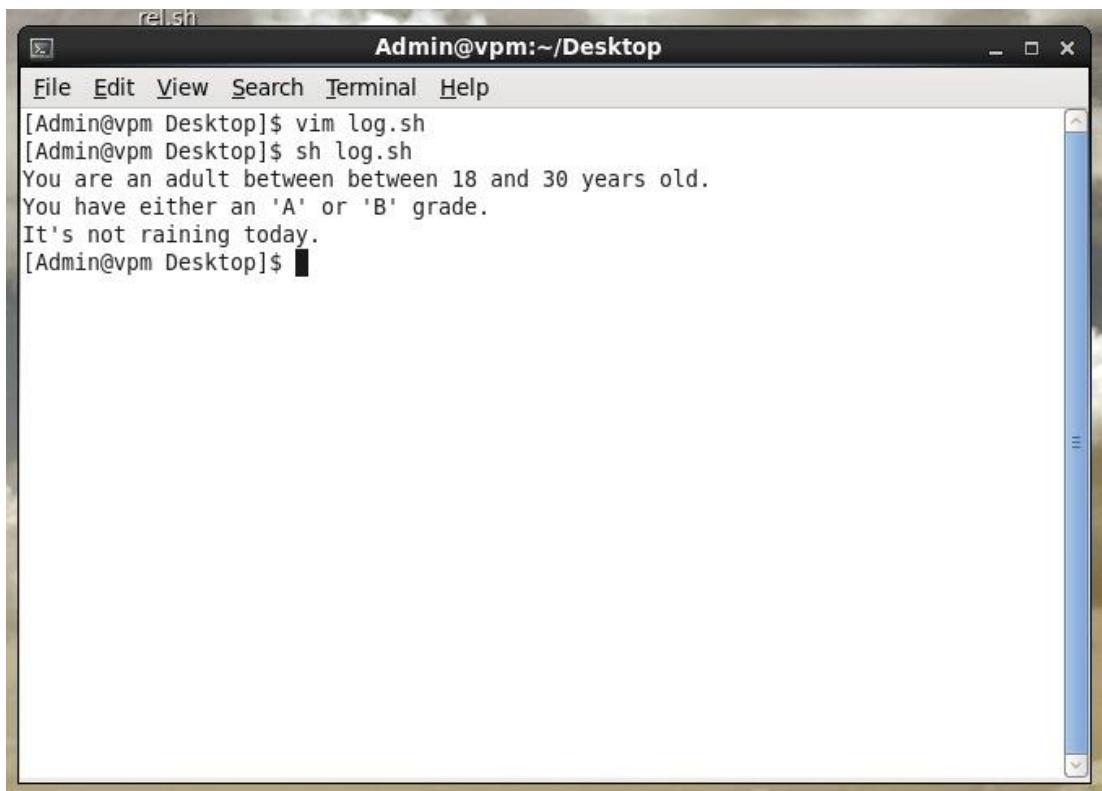
#Logical OR (||) operator
grade="B"
if [ "$grade" = "A" ] || [ "$grade" = "B" ]; then
echo "You have either an 'A' or 'B' grade."
fi

#Logical NOT (!) operator
is_raining=false
if ! $is_raining; then
echo "It's not raining today."
fi
```

-- INSERT --

18,1

All



```

relish
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ vim log.sh
[Admin@vpm Desktop]$ sh log.sh
You are an adult between between 18 and 30 years old.
You have either an 'A' or 'B' grade.
It's not raining today.
[Admin@vpm Desktop]$ 
```

d) Boolean Operators: -

The following Boolean operators are supported by the Bourne Shell.

Assume variable **a** holds 10 and variable **b** holds 20 then –

Operator	Description	Example
!	This is logical negation. This inverts a true condition into false and vice versa.	[! false] is true.
-o	This is logical OR . If one of the operands is true, then the condition becomes true.	[\$a -lt 20 -o \$b -gt 100] is true.
-a	This is logical AND . If both the operands are true, then the condition becomes true otherwise false.	[\$a -lt 20 -a \$b -gt 100] is false.



```

relish
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ vim bool.sh
[Admin@vpm Desktop]$ 
```

rel.sh

Admin@vpm:~/Desktop

```
a=10
b=20
age=15

#Logical Negation (!)
if [ ! $age -ge 18 ]; then
    echo "You are not yet an adult."
fi

#Logical OR (-o)
if [ $a -lt 20 -o $b -gt 100 ]; then
    echo "Logical OR condition is true."
fi

#Logical AND (-a)
if [ $a -lt 20 -a $b -gt 100 ]; then
    echo "Logical AND condition is true."
else
    echo "Logical AND condition is false."
fi
```

-- INSERT --

21,1

All

rel.sh

Admin@vpm:~/Desktop

```
[Admin@vpm Desktop]$ vim bool.sh
[Admin@vpm Desktop]$ sh bool.sh
You are not yet an adult.
Logical OR condition is true.
Logical AND condition is false.
[Admin@vpm Desktop]$
```

7b) Decision Making:

Unix Shell supports conditional statements which are used to perform different actions based on different conditions.

The if...else statements

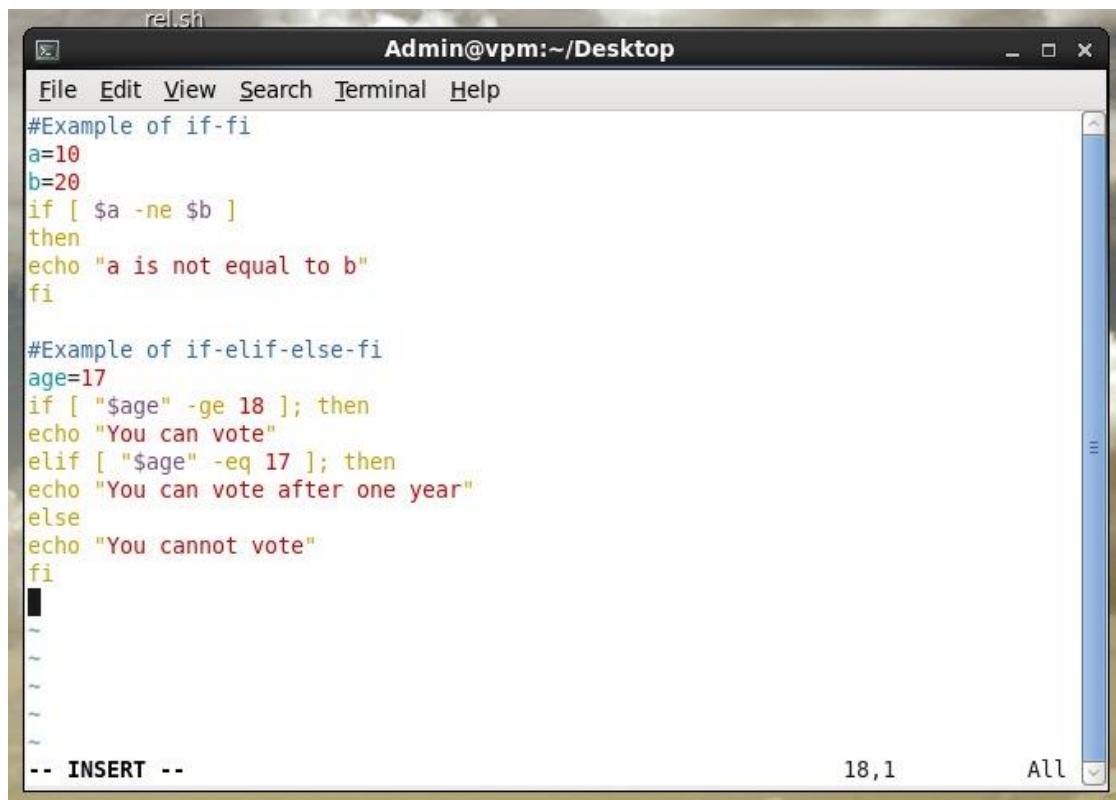
If else statements are useful decision-making statements which can be used to select an option from a given set of options.

Unix Shell supports following forms of **if...else** statement –

- if...fi statement
- if...else...fi statement
- if...elif...else...fi statement



```
rel.sh
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ vim if_elif.sh
```

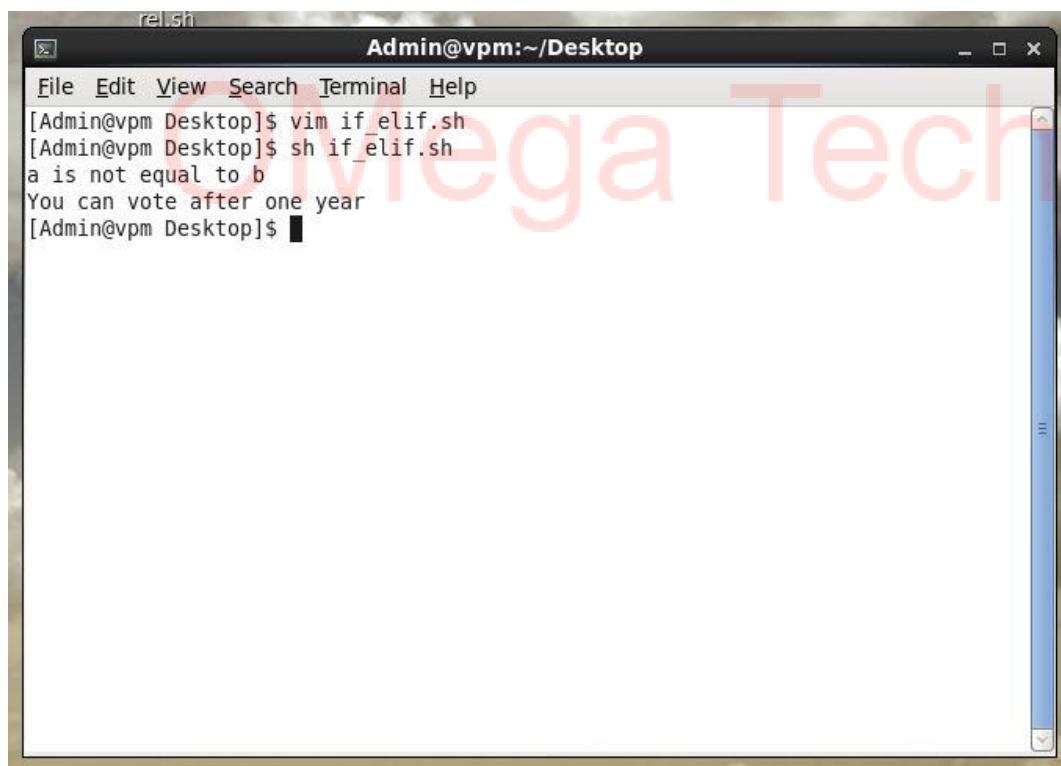


rel.sh Admin@vpm:~/Desktop

```
File Edit View Search Terminal Help
#Example of if-fi
a=10
b=20
if [ $a -ne $b ]
then
echo "a is not equal to b"
fi

#Example of if-elif-else-fi
age=17
if [ "$age" -ge 18 ]; then
echo "You can vote"
elif [ "$age" -eq 17 ]; then
echo "You can vote after one year"
else
echo "You cannot vote"
fi
-
-
-
-
-
-- INSERT --
```

18,1 All



rel.sh Admin@vpm:~/Desktop

```
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ vim if_elif.sh
[Admin@vpm Desktop]$ sh if_elif.sh
a is not equal to b
You can vote after one year
[Admin@vpm Desktop]$
```

7c) loop: -

A loop is a powerful programming tool that enables you to execute a set of commands repeatedly. In this chapter, we will examine the following types of loops available to shell programmers –

- The while loop
- The for loop
- The until loop
- The select loop

We will use different loops based on the situation. For example, the **while** loop executes the given commands until the given condition remains true; the **until** loop executes until a given condition becomes true.



relish
Admin@vpm:~/Desktop
File Edit View Search Terminal Help
[Admin@vpm Desktop]\$ vim forl.sh



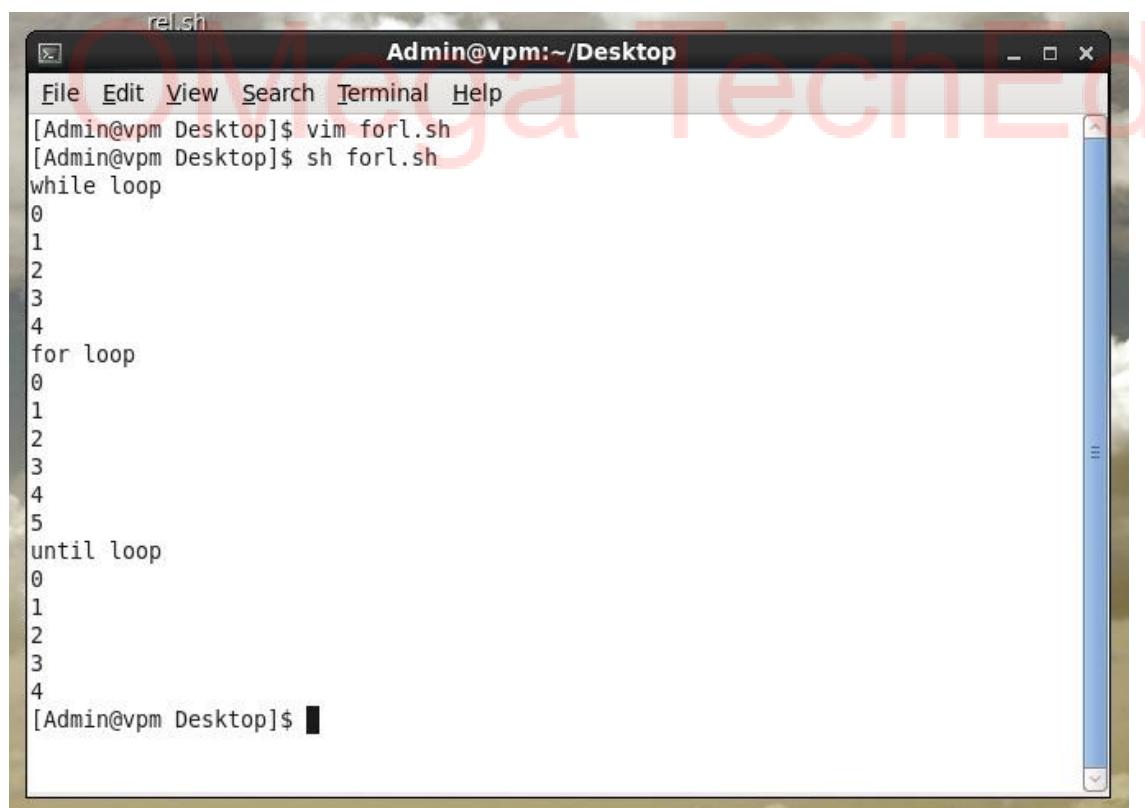
relish

Admin@vpm:~/Desktop

```
File Edit View Search Terminal Help
#While loop
echo "while loop"
a=0
while [ $a -lt 5 ]
do
    echo $a
    a=`expr $a + 1`
done
#for loop
echo "for loop"
for var in 0 1 2 3 4 5
do
    echo $var
done
#Until loop
echo "until loop"
b=0
until [ ! $b -lt 5 ]
do
    echo $b
    b=`expr $b + 1`
done
-- INSERT --
```

23,1 All

This screenshot shows a terminal window titled 'relish' with the command 'Admin@vpm:~/Desktop'. The window displays a shell script with syntax highlighting. The script contains three loops: a while loop that prints numbers from 0 to 4, a for loop that prints numbers from 0 to 5, and an until loop that prints numbers from 0 to 4. The status bar at the bottom shows '23,1' and 'All'.



relish

Admin@vpm:~/Desktop

```
File Edit View Search Terminal Help
[Admin@vpm Desktop]$ vim forl.sh
[Admin@vpm Desktop]$ sh forl.sh
while loop
0
1
2
3
4
for loop
0
1
2
3
4
5
until loop
0
1
2
3
4
[Admin@vpm Desktop]$
```

This screenshot shows a terminal window titled 'relish' with the command 'Admin@vpm:~/Desktop'. The window displays the output of a shell script named 'forl.sh'. The script contains three loops: a while loop that prints numbers from 0 to 4, a for loop that prints numbers from 0 to 4, and an until loop that prints numbers from 0 to 4. The status bar at the bottom shows the command '[Admin@vpm Desktop]\$'.

Practical 8

Case Study of Server O.S.: -

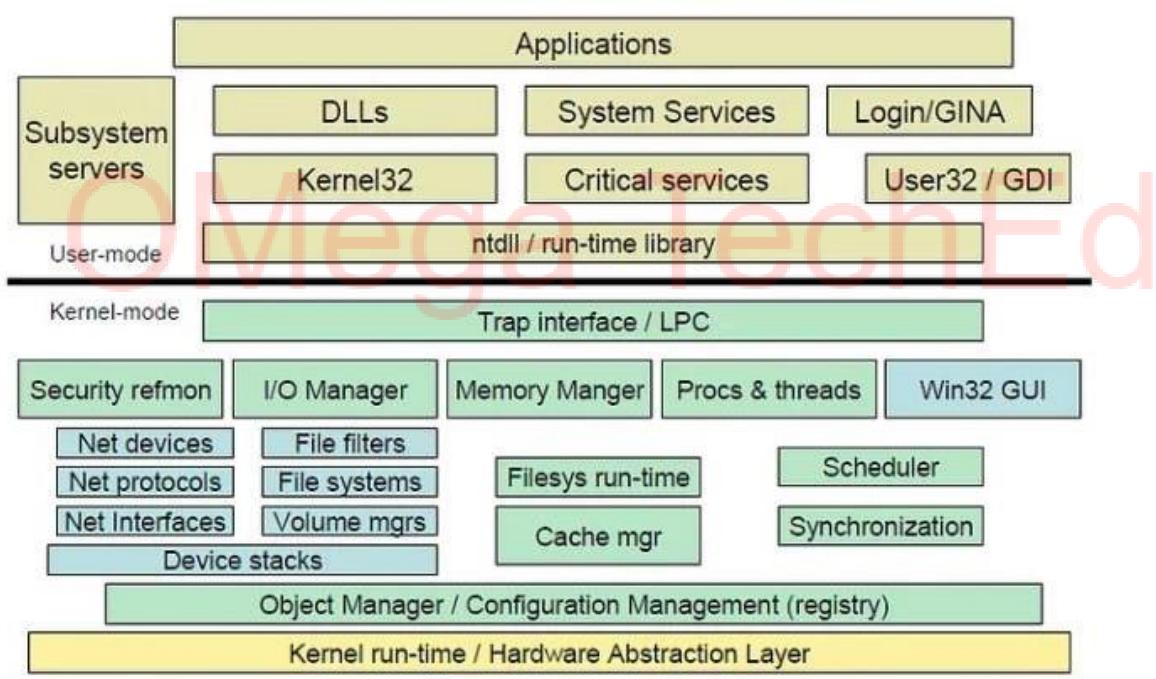
Windows Server 2022 Operating System:

Windows Server 2022 is a server operating system developed by Microsoft.

Architecture:

Windows Server 2022 follows a similar architecture to other Windows Server versions. It is based on the Windows NT kernel, which provides core functionality and services. It supports both 32-bit and 64-bit architectures.

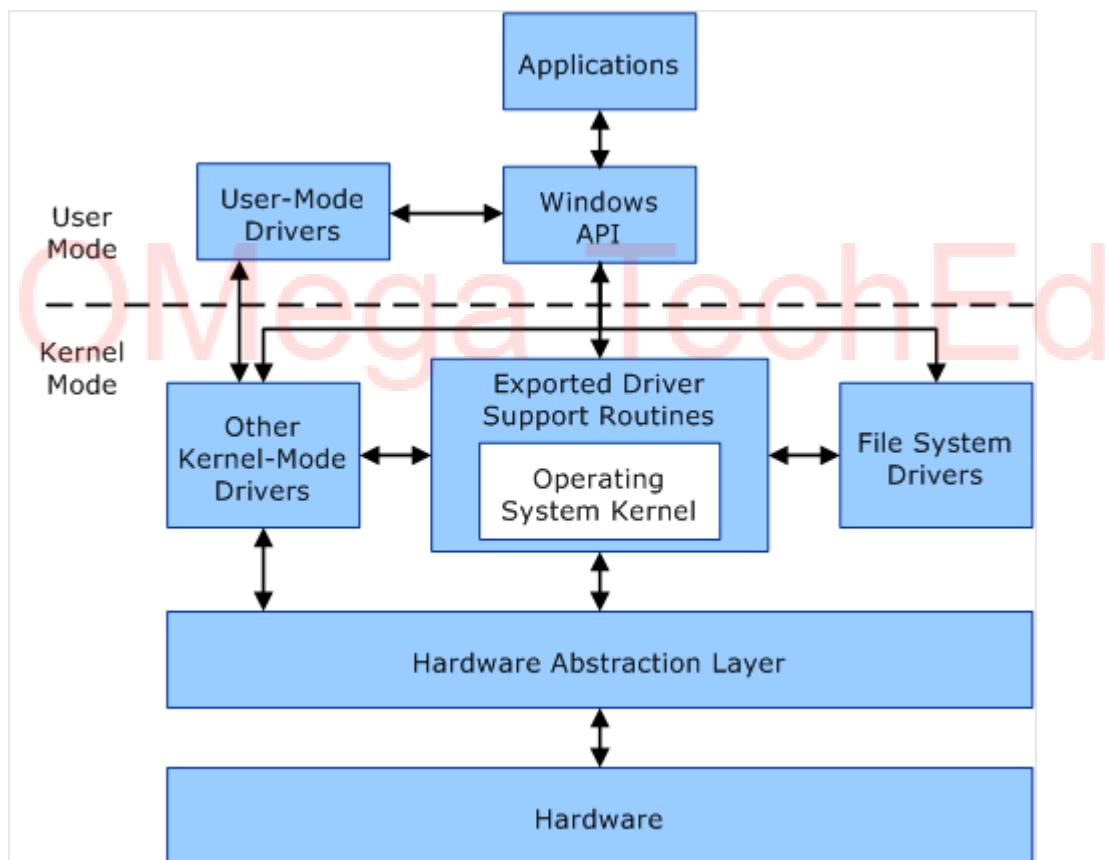
Windows Architecture



Components:

There are many components of the Windows OS Server:

1. **Kernel:** The Windows NT kernel forms the core of the operating system ,managing processing, memory, and system resources.
2. **User Mode:** This layer provides the user interface and runs user applications,including graphical interfaces and command-line tools.
3. **Hardware Abstraction Layer HAL:** HAL abstracts hardware-specific details,allowing the operating system to work with different hardware configurations.
4. **File System:** Window Server 2022 supports multiple file systems, including NTFS(New Technology File System), ReFS (Resilient File System), and FAT32 (File Allocated Table).



Services:

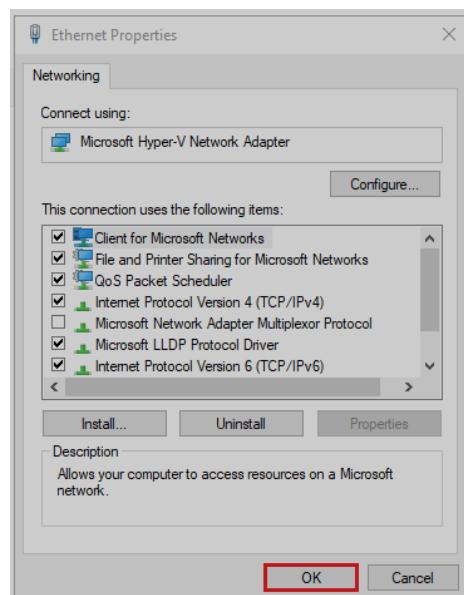
Windows Server 2022 offers various services to support server functionality. Some important services:

1. **Active Directory Domain Services (AD DS):** Provides centralized user and computer management, authentication, and security policies.
2. **Domain Name System (DNS):** Translates domain names into IP addresses, enabling network communication.
3. **Dynamic Host Configuration Protocol (DHCP):** Assigns IP addresses and network configuration settings to devices on a network.
4. **File and Storage Services:** Offers file sharing, storage management, and data deduplication features.
5. **Hyper-V:** Provides virtualization capabilities, allowing multiple virtual machines to run on a single physical server.
6. **Internet Information Services (IIS):** Supports hosting websites and web applications.
7. **Remote Desktop Services (RDS):** Enables remote access to Windows-based applications and desktops.

Configuration:

Windows Server 2022 can be configured using various tools, including:

1. **Server Manager:** Provides a graphical interface to manage server roles, features, and configuration settings.
2. **PowerShell:** A powerful command-line interface for system administration and automation.
3. **Group Policy:** Allows centralized management of security policies, user settings, and system configurations across multiple servers.
4. **Windows Admin Center:** A web-based management tool that simplifies server administration tasks.



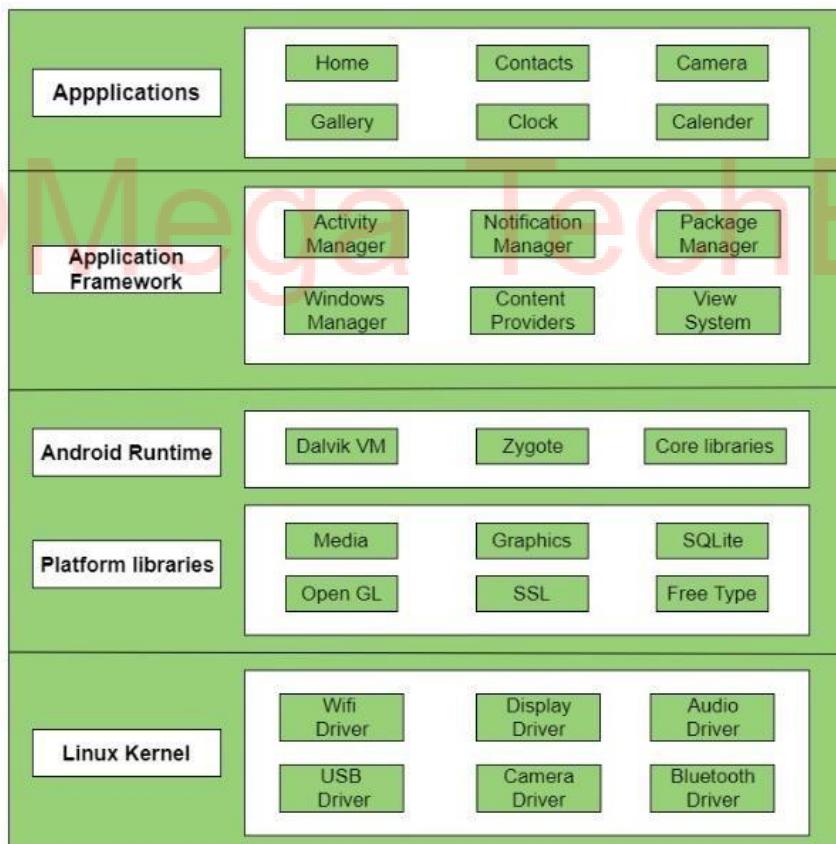
Practical 9

Case Study of Android O.S.: -

Architecture:

Android follows a layered architecture that consists of the following components:

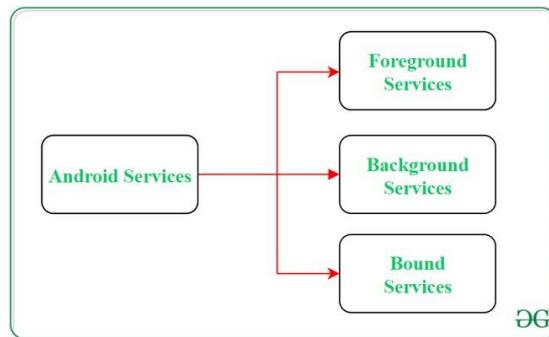
1. **Linux Kernel:** Android is built on top of the Linux kernel, which provides core operating system services such as process management, memory management, device drivers, and security.
2. **Libraries:** Android includes a set of libraries written in C/C++ that provide essential functionality to applications, such as graphics rendering, database access, and Networking.
3. **Android Runtime (ART):** It is the runtime environment in which Android applications run. ART utilizes ahead-of-time (AOT) and just-in-time (JIT) compilation techniques to optimize application performance.
4. **Application Framework:** The framework provides a set of APIs and tools for developers to build applications. It includes components such as activities, services, content providers, and broadcast receivers.
5. **Applications:** These are the end-user applications that run on the Android platform, such as web browsers, email clients, and social media apps.



Services:

Android OS provides various services that enhance the functionality of the platform, including:

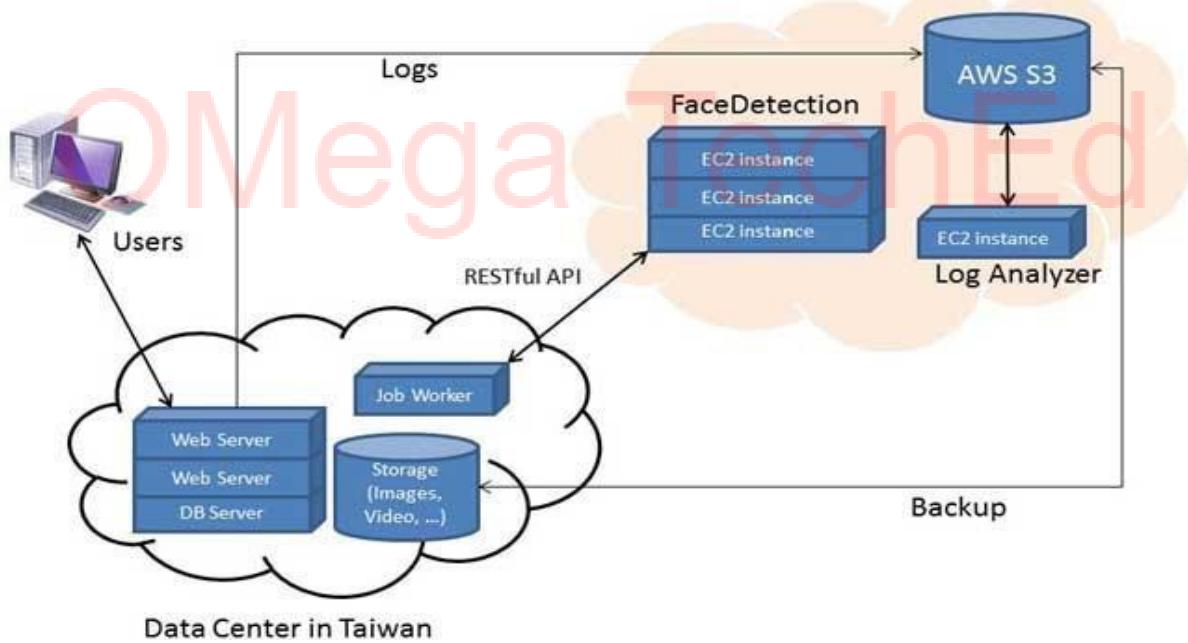
- 1. Activity Manager:** Manages the lifecycle of activities and their interactions.
- 2. Package Manager:** Handles the installation, removal, and management of applications.
- 3. Notification Manager:** Manages the display of notifications to the user.
- 4. Location Manager:** Provides access to location-based services such as GPS and network location.
- 5. Telephony Manager:** Offers access to telephony-related services such as making calls and sending messages.



OMega TechEd

Practical no-10**Case Study of AWS**Introduction:

Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 200 AWS services are available. New services can be provisioned quickly, without the upfront fixed expense. This allows enterprises, start-ups, small and medium-sized businesses, and customers in the public sector to access the building blocks they need to respond quickly to changing business requirements.

AWS Infrastructure

Services:

AWS provides a vast array of services across categories like computing storage databases networking analytics machine learning security and more.

Some notable Services are included:

- Computer services
- Storage services
- Database services
- Networking services
- Analytics and big data
- AI and Machine learning
- Security and Compliance.

AWS Configuration:

AWS Management Console: A web-based interface for managing AWS resources, configuring services, and monitoring usage.

AWS Command line interface: A command line tool for managing AWS resources, automating tasks and scripting.

AWS CloudFormation: A service for infrastructure as code allowing the provisioning and management of AWS resources using templates.

References:

Operating Systems by Tushar Sambare & Sonali Sambare SHETH Publication.
<https://www.tutorialspoint.com>

For the video demonstration of the practical click on the link:

Operating System Practicals