

Practical – 1

Steps to install any operating system and Compare two operating system.

Steps to install windows 7

Step 1 : Back – up your files because installation wizard will delete your all data from hard drive you can back-up your file to another drive , an external drive or a cloud based service.



Step 2 : Restart your computer.

Step 3 : Immediately pressed , esc, f2, f10 when it restarts.

Step 4 : Find BIOS boot option menu . The boot options menu of your BIOS may vary in location or name from illustration, but you can find it from search

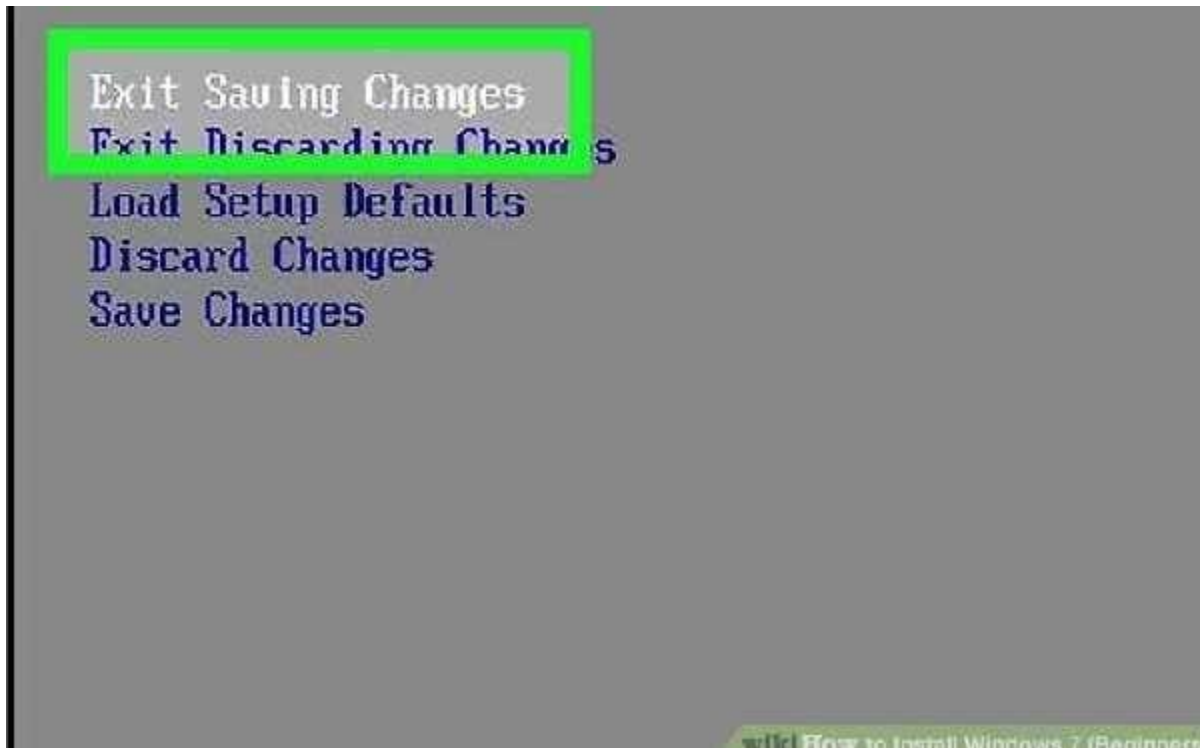


Step 5 : Select the optional disk drive as the first boot device of your computer.



Step 6 : Place the windows 7 installation in disk drive

Step 7 : Save changes in setting press the button indicated on the screen



Step 8 : Shut off your computer.

Step 9 : Start your computer from the disk

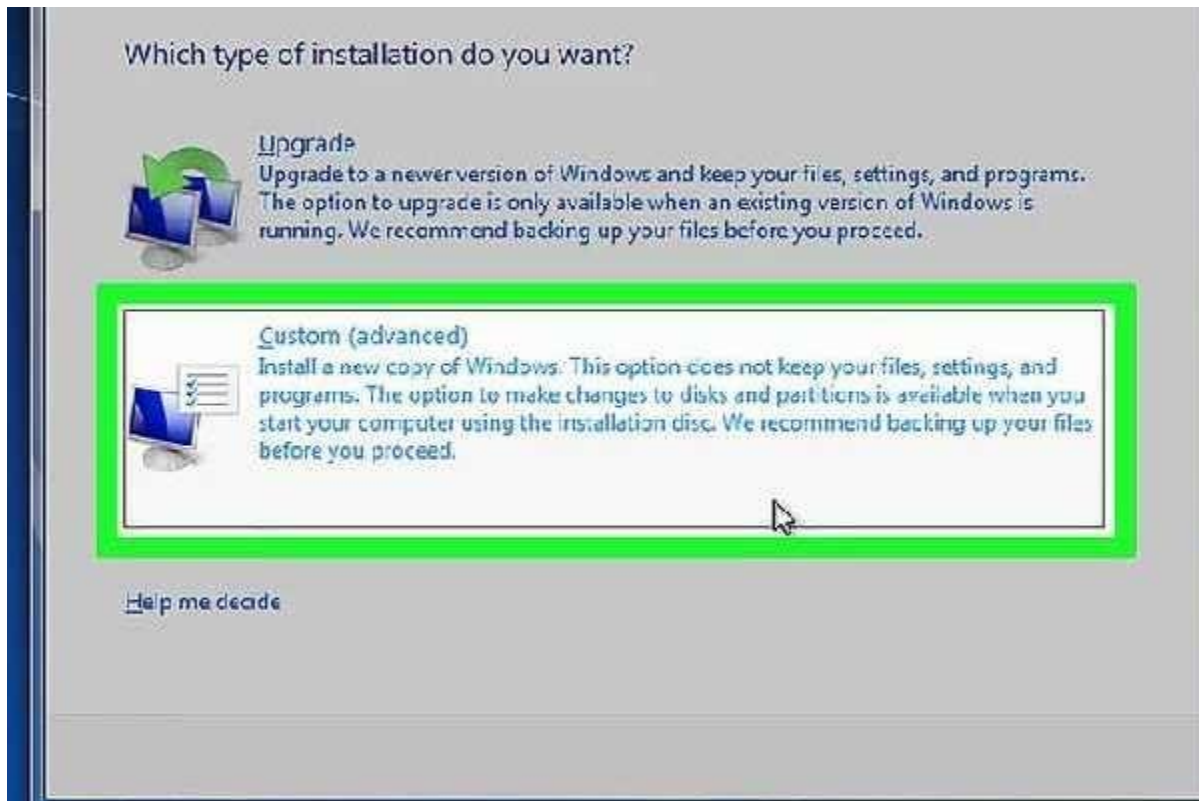


Step 10 : Choose your windows setup option.

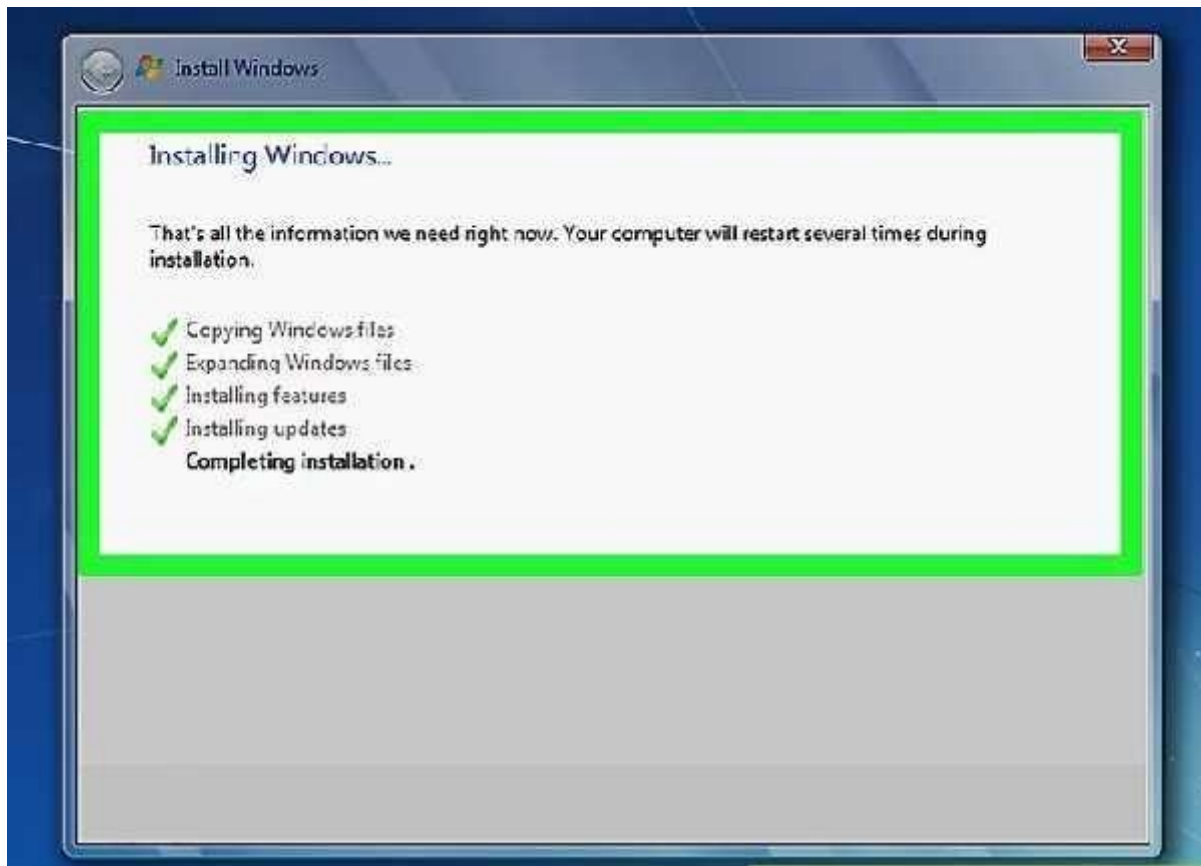
Step 11 : Click on install now button

Step 12 : Accept license term.

Step 13 : Select custom installation.



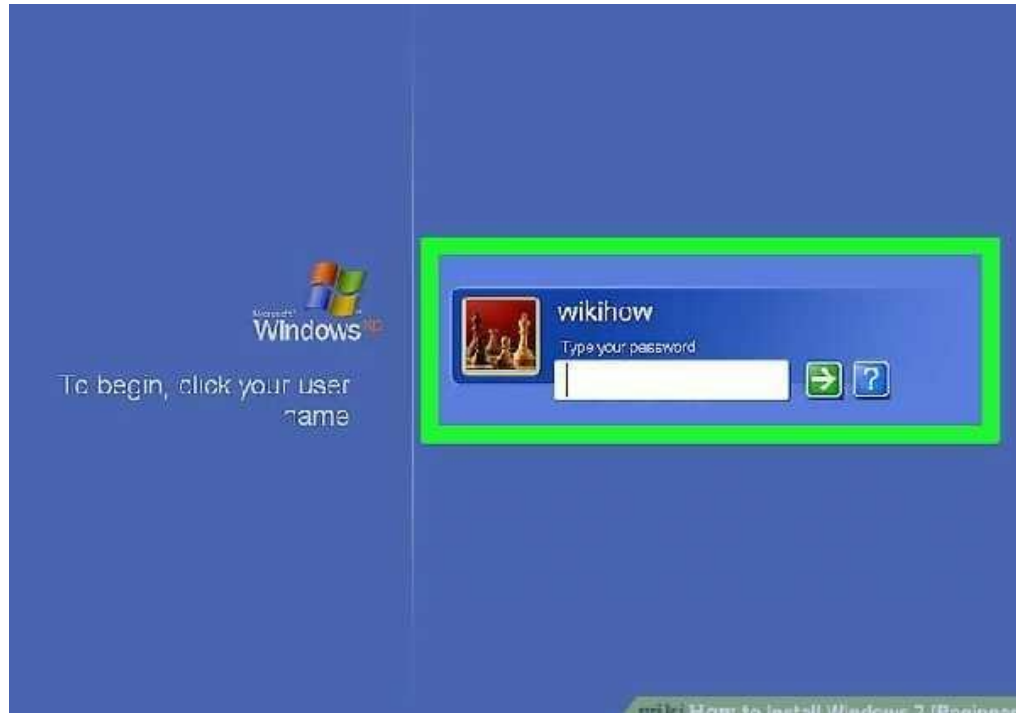
Step 14 : Install windows on your preferred hard drive and partition.



Sub code: 3330701

Sub name: Operating system

Step 15 : Upgrading to windows 7.



Name: Himani Trivedi A.

Enrollment no: 196170307149

- **Difference between Linux and Windows Operating system.**

Linux	Windows
In Linux with GPL- Licensed operating system, users are free to modify the software, can re-use in any number of systems and even they can sell the modified version.	In windows, with Microsoft license, users won't have access to source code (can't modify the software) and based on a number of licenses – we can install only on those number of computers.
Linux is an open source operating system.	Windows are the not the open source operating system.
Linux is free of cost.	While it is costly.
Its file name case-sensitive.	Its file name is case-insensitive.
In Linux, monolithic kernel is used.	Micro kernel is used.
Linux is more efficient in comparison of windows.	Windows are less efficient.
There is forward slash is used for separating the directories.	There is back slash is used for separating the directories.
Linux provides more security than windows.	It provides less security than Linux.
Linux is widely used in hacking purpose based systems.	Windows does not provide much efficiency in hacking.
Linux has access to source code and alters the code as per user need	Whereas Windows does not have access to source code.

Sub code: 3330701

Sub name: Operating system

Linux is complicated to install but has the ability to complete complex tasks easier.	Windows gives user's a simple system to operate but it will take a longer time to install.
Linux supports a wide variety of free software's than windows	Windows have a large collection of video game software.

Practical – 2

Compare various processes scheduling algorithm.

First come first served (fcfs)	Round robin(rr)
First Come First Served (FCFS) is the non-preemptive scheduling algorithm.	Round Robin(RR) is the preemptive scheduling algorithm.
FCFS has the minimal overhead.	While RR has small overhead as it is necessary to record the time elapsed and then switch the process which causes an overhead.
First Come First Served Scheduling Algorithm provides high response time for the processes.	In Round Robin Scheduling Algorithm, for the short processes there is very low response time.
FCFS is inconvenient to use in the time sharing system.	It is mainly designed for the time sharing system and hence convenient to use.
Average waiting time is generally not minimal in First Come First Served Scheduling Algorithm.	In Round Robin Scheduling Algorithm average waiting time is minimal.
The process is simply processed in the order of their arrival in FCFS.	It is similar like FCFS in processing but uses time quantum.

First come first served (fcfs)	Shortest job first (sjf)
First Come First Served (FCFS) executes the processes in the order	Shortest Job First (SJF) executes the processes based upon their

Name: Himani Trivedi A.

Enrollment no: 196170307149

Sub code: 3330701

Sub name: Operating system

in which they arrive i.e. the process that arrives first is executed first.	burst time i.e. in ascending order of their burst times.
FCFS is non preemptive in nature.	SJF is also non-preemptive but its preemptive version is also there called Shortest Remaining Time First (SRTF) algorithm.
FCFS results in quite long waiting time for the processes and thus increases average waiting time.	The average waiting time for given set of processes is minimum.
FCFS leads to the convoy effect.	It does not lead to the convoy effect.
FCFS algorithm is the easiest to implement in any system.	The real difficulty with SJF is knowing the length of the next CPU request or burst.
A process may have to wait for quite long to get executed depending on the burst time of the processes that have arrived first.	A long process may never get executed and the system may keep executing the short processes.
FCFS lead to lower device and CPU utilization thereby decreasing the efficiency of the system.	SJF leads to higher effectiveness of the system due to lower average waiting time.
FCFS results in minimal overhead.	In case of SJF, elapsed time should be recorded, results in more overhead on the processor.
Shortest job first (sjf)	Round-robin (rr)
Shortest Job First (SJF) executes the processes based upon their burst time i.e. in ascending order of their burst times.	Round-Robin (RR) executes the processes based upon the time quantum defined i.e. each process is executed for a fixed amount of time.
SJF is also non-preemptive but its preemptive version is also there called Shortest Remaining	Round-Robin (RR) is preemptive in nature.

Name: Himani Trivedi A.

Enrollment no: 196170307149

Shortest job first (sjf)	Round-robin (rr)
Time First (SRTF) algorithm.	
The average waiting time for given set of processes is minimum.	The average waiting time for given set of processes is quite small and depends on the time quantum.
The real difficulty with SJF is knowing the length of the next CPU request or burst.	It is quite easy to implement RR.
A long process may never get executed and the system may keep executing the short processes.	Each process is executed and every user feels that his work is being done as the CPU gives equal amount of time to each process.
In case of SJF, elapsed time should be recorded, results in more overhead on the processor.	In case of RR, if the time quantum is very small then context switch takes place again and again after very short intervals of time which leads to overhead.

Practical – 3

Test and run basic UNIX command.

1. Help Command

Syntax :- man

- This command is a help command. By writing (man) in the Linux Terminal the help desk is open and more you can explore in that command, there are many instructions we can read and execute in.

\$man ls

It displays help for ls command.

2. Calender Displaying

Syntax :- cal

Sub code: 3330701

Sub name: Operating system

- This command displays the calendar as soon as you write this command in the Linux Terminal it will display the Calendar of that Month.

Example:

\$cal 2019

It displays calendar all the months of the year 2019.

3. Recent Login Activities

Syntax :- who

- This command helps to give the report the recent login activities. As we write the command in the Linux Terminal it will display the List of the recent User login.

Example:

\$who

It displays the recent login activities of users.

4. Current Login Activities

Syntax :- who am i

- This command helps to give the information of the Current User. As you will write the command in the Linux Terminal then it will give the statistics of the current Login of the User like at what time user login username and many more user statistics.

Example:

\$ who am i

It displays the current user login details.

5. Print a Statement

Syntax :- echo

Name: Himani Trivedi A.

Enrollment no: 196170307149

Sub code: 3330701

Sub name: Operating system

- This is a command used in Linux Terminal, which prints the statement what is written in the double-equation.

Example:

\$ echo “ Hello World”

It will print Hello World as output.

6. Basic Calculator

Syntax :- bc

- This is the command of the Linux Terminal that launch the calculator in the terminal, after writing this command you can solve different-different operations by writing them. And also if you want to exit from the calculator in the Linux then you have to type (ctrl+d) command, by writing this command you will exit the Calculator screen and come back to normal bash.

Example:

\$ bc 3 + 5

It will return 8 as output.

7. Full Calender view

Syntax :-ncal

- This is a command same like the cal command of the Linux Terminal but the difference in the command is that the n is extra added and also there is slight difference in the output that in the calender it also display the weeks in the calender.

Example:

\$ ncal

It displays the current month of the year with slight difference than cal.

8. Write Expression in Calculator

Syntax :-expr

Name: Himani Trivedi A.

Enrollment no: 196170307149

Sub code: 3330701

Sub name: Operating system

- This command is used in the to evaluate the experssion thisis written after (bc) command that is of calculator p rograme and all the expression are written with this command, fir st this expr command is written then the expres sion is written.

Example:

```
$ expr 3 + 5
```

It displ ays 8 as output.

9. See the History of all Written Commands

Syntax :- history

- This command is written to see the all the previous commands written in the Linux Terminal, means it c ollects ans list out the history of all the command and dis play that in the List format.

Example:

```
$ history
```

It displ ays all the previous commands used.

10. See current user name.

Syntax :-uname

- This command is w r it ten to see current user name in Linux System.

Example:

```
$ uname
```

It displays the current user name 'Linux'.

11. To Change Password.

Syntax :- passwd

- This command is written to change the password of Linux System User. If password exists insystem.

Example:

```
$ passwd (if password exists)
```

- It displays to enter the current password first, Then

Name: Himani Trivedi A.

Enrollment no: 196170307149

Sub code: 3330701

Sub name: Operating system

Enter new password,

Then Confirm password.

12. To see Current date and time.

Syntax :- date

- This command is written to see the current systemdate, Time and Time-zone in Linux System.

Example:

\$ date

-It displays 29/07/2020 17:35 IST

13. To see current terminal name.

Syntax :- tty

- This command is written to see the current terminal name login in Linux System.

Example:

\$ tty

- It displays /dev/pts/0 (aspath)

14. To set specific values for ECO.

Syntax :- set

- This command is written to set the specific value for 'ECO' in Linux System.

Example:

\$ set 1 2 a3

- If you write,

ECO \$1

Name: Himani Trivedi A.

Enrollment no: 196170307149

Sub code: 3330701

Sub name: Operating system

It displays '1' on screen.

ECO \$2

It displays '2' on screen.

ECO \$3

It displays 'a3' on screen.

ECO \$4

It displays ' ' on screen.

It displays nothing because we haven't set 4th value.

15. Make Directory

Syntax :-mkdir

- This command is written to make directory in Linux system. It creates folder for the user.

Example:

\$ mkdir Sem-3

It create a folder named Sem-3 for user.

You can also create folder within a folder and so on.

Example:

\$ mkdir -p Sem-3/OS/Practicals

It creates folder names Sem-3. OS folder is created within Sem-3 folder.

Practical's folder is created within OS folder.

16. Remove Directory

Syntax:-rmdir

-This command is written to remove directory which is previously created in Linux system. It delete folder for the user.

Name: Himani Trivedi A.

Enrollment no: 196170307149

Sub code: 3330701

Sub name: Operating system

Example:

\$ rmdir OS

It deletes folder OS which is within Sem-3 folder.

17. Change Directory.

Syntax: cd

- This command is written to go to another directory from current directory.

Example:

\$ cd OS

It will go to OS folder from the current directory.

18. To go to previous Directory.

Syntax: cd ..

-This command is written to go back to previous directory or folder.

Example:

\$ cd .. OS

-It will go back to previous folder SEM-3 from OS folder.

19. Present Working Directory

Syntax: pwd

-This command is written to get path till the present directory or folder.

Example:

\$ pwd

-Suppose we are in folder Practical's then the output will be (It will show the path) Sem-3/OS/Practicals.

20. List

Syntax: ls

Name: Himani Trivedi A.

Enrollment no: 196170307149

Sub code: 3330701

Sub name: Operating system

-This command is written to get list of all directories or file in Linux system.

Example:

\$ ls

-Suppose we are on desktop than

It will show all the list of the names of all files and directories of Desktop.

Example-1:

\$ ls -s

- It shows size of all folders and directories with name.

Example-2:

\$ ls -r

-It displays all folders and director ies in descending order (Z-A).

Example-3:

\$ ls -m

-It shows name of all folder and di rectories with comma (,) separated.

Example-4:

\$ ls -a

-It shows name of all folder and di rectories along with hidden folders (. files) and directories created by user.

Example-5:

\$ ls -l

- It shows all folders and directories with their index node number.

Example-6:

Name: Himani Trivedi A.

Enrollment no: 196170307149

Sub code: 3330701

Sub name: Operating system

\$ ls -l

-It shows all folder and directories with time & date when created and by which user it was created. Along with the size of folders and Directories.

Example-7:

\$ ls -*a

-It shows all folder and directories which has last letter 'a' in folder and directory name.

Example-8:

\$ ls -???

-It shows all folders and directories which has exactly three letters in folder and directory name.

Example-9:

\$ ls -*a??

-It shows all folder and directories which has last third letter 'a' and last two letters are not concerned.

Example-10:

\$ ls -l-OS

-It shows information of OS folder [when created, time, date, user and size of folder & directories].

Name: Himani Trivedi A.

Enrollment no: 196170307149

Practical – 4

Test and run Advanced UNIX command.

1. To a create file

Syntax :-

\$ cat>file_name

- This command is written to create file in Linux System. Then Enter data inside the file.

Example:

- **cat> f1**

OS -> Practical -> A3 (Content of file).

It will create a file 'f1' and enter 'OS -> Practical -> A3' as data in the file.

Syntax :-

\$ cat>>file_name

- This command is written to append data inside the existing file in Linux system.

Example:

- **cat>> f1**

Linux Commands (content to append)

It will append 'Linux Commands' inside the file 'f1' as data.

Syntax :-

\$ cat<file_name

- This command is written to display the data inside the existing file in Linux system.

Example:

- **cat< f1**

Sub code: 3330701

Sub name: Operating system

It will display the content of 'f 1' file. OS ->

Practical ->A3

Linux Commands

2. To create multiple files.

Syntax:- touch

- This comm and is written to create multiple files at a time in linux system.

Example:

\$ touch a b c

- It will create the following files namely 'a', 'b', 'c' at a time.

3. To remove files.

Syntax:-rm

- This command is written to remove the existing files from linux system.

Example:

\$ rm -i a

It will confirm again to delete the file

If you want to delete the file than type y.

If you don't want to delete the file than type n.

Example:

\$ rm -r b

It will first open the file then it will remove all data from file and then delete the file.(this happens internally in linux system).

Example:

\$ rm -f c

It will directly delete the file for linux system.

4. To copy data from files

Syntax:- `cp source_file destination_file`

- This command is written to copy the data of one file (source_file) to another file(destination_file) in linux system. If destination_file is not existing then it will create a new file.

Example:

```
$ cp a b
```

It will copy the data of file 'a' into file 'b'.

5. To move data or rename files or folders.

Syntax:- `mv file_1 file_2`

-This command is written to move data of one file(file_1) into another file (file_2).

Example:

```
$ mv f1 f2
```

- it will move the data of file 'f1' into file 'f2'.

Example:

```
$ mv f1 file_1
```

It will rename the file 'f1' to 'file_1'.

6. To compare data of files or folders.

Syntax:- `cmp file_1 file_2`

this command is written to compare the data of file(file_1) with another file(file_2).

Example:

```
$ cmp f1 f2 (if they have same data)
```

It will display nothing because they have same data.

Example:

```
$ cmp f1 f2 (if they don't have same data)
```

It will display line where there are difference of data.

7. To display difference.

Syntax:- diff file_1 file_2

This command is written to see the content of both file with difference in linux system.

Example:

```
$ diff f1 f2
```

8. To display word count.

Syntax:- wc (-l/-c/-w)

- This command is written to display number of lines, words or character in linux system.

Example:

```
$ wc -l f1
```

It will display number of lines in file 'f1'.

It will display 2 on screen

```
$ wc -c f1
```

It will display number of characters in file 'f1'. It will display

36 on screen

```
$ wc -w f1
```

It will display number of words in file 'f1'. It will

display 7 on screen.

9. To display list of processes.

Syntax: ps

-This command is written to display all the list of processes in linux command.

Example:

```
$ps -e
```


This displays list of processes with details of it.

10. To kill a process.

Syntax: kill process_number

- This command is written to kill/stop a process forcefully in linux system.

Example:

\$ kill 343

- This will kill the process 343.

11. To display time[real, user, system]

Syntax: time

-This command is written to display real time, user time, system time in linux system.

Example:

\$ time

- This will display 04:52pm 2:34 3:50

12. To display specific line from end.

Syntax: tail -[number of lines] filename

- This command is written to display specific number of lines from bottom of the file in linux system.

Example: tail -2 os

- This will display last 2 lines of os file.

13. To search specific content.

Syntax: grep "letter/word" filename

- This command is written to display the words/sentence which contains "letter/word" from file in linux system.

Example: grep "T" os

-This will display all words which contains the letter "T".

14. To translate/replace contents.

Syntax: tr [expression] < filename

Sub code: 3330701

Sub name: Operating system

- This command is written to translate/replace the contents of file in linux system.

Example: tr '[a-z]' '[A-Z]' <os

- This will convert all contents of os file to uppercase letters.

15. To cut specific contents.

Syntax: cut [number of letters] filename

- This command is written to display the specific characters from file.

Example: cut -c 1 os

- This displays first character from each line of os file.

16. To paste specific contents.

Syntax: paste [expression] file1 file2

- This command is written to append the contents of file2 to file1.

Example: paste -d% os A3

- This displays content of first line of os. Then it will append '%' Then it will append content of first line of A3. This process will continue up to the end of both files.

Sub code: 3330701

Sub name: Operating system

Practical – 5

Test command related with file editing with VI , VIM, GEDIT , GCC.

VI:-

Command mode:

- The vi editor opens in this mode, and it only understands Commands
- In this mode, you can, move the cursor and cut, copy, paste the text.
- This mode also saves the changes you have made to the file
- Commands are case sensitive. You should use the right letter case.

Editor Insert mode:

- This mode is for inserting text in the file.
- You can switch to the Insert mode from the command mode by pressing 'i' on the keyboard
- Once you are in Insert mode, any key would be taken as an input for the file on which you are currently working.
- To return to the command mode and save the changes you have made you need to press the Esc key

Editing commands:

i - Insert at cursor (goes into insert mode)

a - Write after cursor (goes into insert mode)

A - Write at the end of line (goes into insert mode)

ESC - Terminate insert mode

u - Undo last change

U - Undo all changes to the entire line

o - Open a new line (goes into insert mode)

dd - Delete line

Name: Himani Trivedi A.

Enrollment no : 196170307149

Sub code: 3330701

Sub name: Operating system

VIM:

Vim modes:

- There are some arguments as to how many modes that Vim has, but the modes you're most likely to use are command mode and insert mode.
- These modes will allow you to do just about anything you need, including creating your document, saving your document and doing advanced editing, including taking advantage of search and replace functions.

Basic Vim commands

- The most simple commands allow you to open and close documents as well as saving them.
- As with most other text editors, there are protections in place to help you avoid exiting the editor without having saved what you're working on.

:help [keyword] - Performs a search of help documentation for whatever keyword you enter

:e [file] - Opens a file, where [file] is the name of the file you want opened

:w - Saves the file you are working on

:w [filename] - Allows you to save your file with the name you've defined

:wq - Save your file and close Vim

:q! - Quit without first saving the file you were working on

GEDIT

Opening gedit:

- gedit incorporates a graphical user interface (GUI) and is opened by going to

Applications → Accessories → Text Editor or by pressing Alt+F2 and typing gedit

Command Line Tips:

- To open a specific file:

Name: Himani Trivedi A.

Enrollment no : 196170307149

Sub code: 3330701

Sub name: Operating system

gedit filename

- To open multiple files:

gedit file1 file2

To edit system files such as sources.list and fstab, open it with administrative privileges. It is NOT recommended to manually run graphical applications with administrative privileges, but in case you insist to do it, be sure to use gksudo rather than sudo.

gksudo gedit

To open at a specific line number, useful when an error message includes the line number, include "+<line number>". (gksudo is used in this example since the file is a system file owned by root):

gksudo gedit +21 /etc/apt/sources.list

GCC

- GCC stands for GNU Compiler Collections which is used to compile mainly C and C++ language.
- It can also be used to compile Objective C and Objective C++.
- The most important option required while compiling a source code file is the name of the source program, rest every argument is optional like a warning, debugging, linking libraries, object file etc.
- The different options of gcc command allow the user to stop the compilation process at different stages.

Syntax:

gcc [-c|-S|-E] [-std=standard]

Name: Himani Trivedi A.

Enrollment no : 196170307149

Sub code: 3330701

Sub name: Operating system

Practical – 6

Create shell script to print hello world.

```
echo "Hello World";
```

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ gedit test2
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ sh test2
Hello World!!
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

Practical – 7

Create shell script read and display content of File.

```
echo "Enter file name = "
read f1
cat > $f1
echo "Reading Completed"
cat $f1
```

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ sh test3
Enter file name:
test2
Hello
hii
i am Himani Trivedi
Reading Completed
Hello
hii
i am Himani Trivedi
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat test2
Hello
hii
i am Himani Trivedi
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

Name: Himani Trivedi A.

Enrollment no : 196170307149

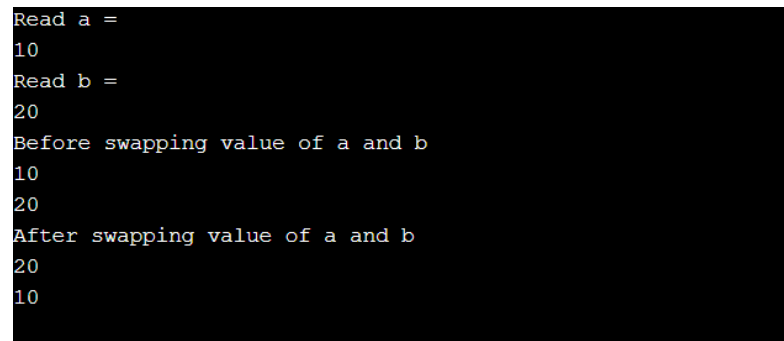
Sub code: 3330701

Sub name: Operating system

Practical – 8

Create a shell script read from command line

```
echo "Read a = "  
read a  
echo "Read b = "  
read b  
echo "Before swapping value of a and b"  
echo $a  
echo $b  
c=$a  
a=$b  
b=$c  
echo "After swapping value of a and b"  
echo $a  
echo $b
```



```
Read a =  
10  
Read b =  
20  
Before swapping value of a and b  
10  
20  
After swapping value of a and b  
20  
10
```

Name: Himani Trivedi A.

Enrollment no : 196170307149

Sub code: 3330701

Sub name: Operating system

Practical – 9

create shell script to append content of one file Another file

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat test3
echo "Enter 1st File name:"
read f1
echo "Enter 2nd File name:"
read f2
echo "Content of 1st file:"
cat $f1
cat $f2>>$f1
echo "After Appending:"
cat $f1
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

```
echo "enter file name"
read f1
echo "enter file name"
read f2
echo "Enter content of file 1 "
cat > $f1
echo "Enter content of file 2 "
cat > $f2
echo "Content of file one before appending"
cat $f1
cat $f2 >> $f1
echo "Content of file one after appending"
cat $f1
```

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ sh test3
Enter 1st File name:
test1
Enter 2nd File name:
test2
Content of 1st file:
Hii...
I am Himani.

After Appending:
Hii...
I am Himani.

This is the script for appending one file into another file.
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ █
```

Name: Himani Trivedi A.

Enrollment no : 196170307149

Sub code: 3330701

Sub name: Operating system

Practical – 10

Create shell script accept string in lower case And convert into upper case.

```
echo "Enter ur name:"
read str
echo $str | tr "[a-z]" "[A-Z]"
```

```
[onworks@localhost ~]$ gedit test1
[onworks@localhost ~]$ sh test1
Enter ur name:
himani trivedi
HIMANI TRIVEDI
[onworks@localhost ~]$
```

Practical – 11

Write shell script to find number of character ,Words and line of given file

```
echo "Enter file name "
read f1
c=`cat $f1 | wc -c`
w=`cat $f1 | wc -w`
l=`cat $f1 | wc -l`
echo "Character = $c"
echo "Words = $w"
echo "Line = $l"
```

```
[onworks@localhost ~]$ cat test1
Hii..
It's Himani
This is RedHat verstion of Linux.

Thank you!!
[onworks@localhost ~]$ sh test2
Enter file name:
test1
test1 file have,
Characters=65
Words=11
Lines=5
[onworks@localhost ~]$
```

Name: Himani Trivedi A.

Enrollment no : 196170307149

Sub code: 3330701

Sub name: Operating system

Practical – 12

Write shell script to read a number and reverse It

```
echo "Enter a number"
```

```
read n
```

```
sd=0
```

```
rev=0
```

```
while [ $n -gt 0 ]
```

```
do
```

```
    sd=$(( $n % 10 ))
```

```
    rev=$(( ( $rev * 10 ) + $sd ))
```

```
    n=$(( $n / 10 ))
```

```
done
```

```
echo "Reverse number of entered digit is $rev"
```



```
onworks@onworks-Standard-PC-1440FX-PIIX-1996:~$ sh test1
Enter a number
123
Reverse number of entered digit is 321
onworks@onworks-Standard-PC-1440FX-PIIX-1996:~$
```

Name: Himani Trivedi A.

Enrollment no : 196170307149

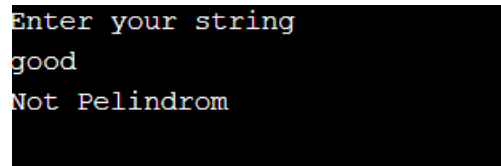
Sub code: 3330701

Sub name: Operating system

Practical – 13

Write shell script to check string is palindrome Or not

```
echo "Enter your string "  
read s  
l=`expr $s | wc -c`  
l=`expr $l - 1`  
i=1  
j=`expr $l / 2`  
while [ $i -le $j ]  
do  
k=`echo $s | cut -c $i`  
r=`echo $s | cut -c $l`  
if [ $k != $r ]  
then  
echo "Not Palindrom"  
exit  
fi  
i=`expr $i + 1`  
l=`expr $l - 1`  
done  
echo "Palindrom"
```



```
Enter your string  
good  
Not Palindrom
```

Name: Himani Trivedi A.

Enrollment no : 196170307149

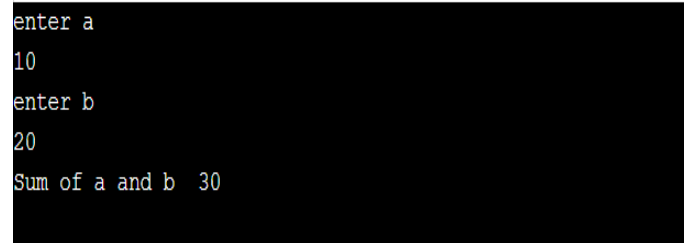
Sub code: 3330701

Sub name: Operating system

Practical – 14

Write shell script to add two no

```
echo "enter a "  
read a  
echo "enter b "  
read b  
c=`expr $a + $b`  
echo "Sum of a and b " $c
```



```
enter a  
10  
enter b  
20  
Sum of a and b 30
```

Practical – 15

Write shell script to read a number and reverse It.

```
echo "Enter a number"  
read n  
sd=0  
rev=0  
while [ $n -gt 0 ]  
do  
    sd=$(( $n % 10 ))  
    rev=$(( $rev * 10 + $sd ))  
    n=$(( $n / 10 ))  
done
```

Name: Himani Trivedi A.

Enrollment no : 196170307149

Sub code: 3330701

Sub name: Operating system

echo "Reverse number of entered digit is \$rev"

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop$ cat test1
echo "Enter no:"
read n
m=$n
s=0
r=0
while [ $n -gt 0 ]
do
    s=$(( $n % 10 ))
    r=$(( ($r * 10) + $s ))
    n=$(( $n / 10 ))
done
echo "Reverse of $m is $r"
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop$ sh test1
Enter no:
129056
Reverse of 129056 is 650921
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop$
```

Name: Himani Trivedi A.

Enrollment no : 196170307149