C:\Users\Admin > cd..
C:\Users > cd..

i) output- C:\> date

ii) output - C:\> time

iii) output- C:\> path

iv) output- C:\>

v) output- C:\> mkdir Aashish — (before create then after used the cd command)
C:\> cd Aashish

practical-2

2. windows (DOS) commands-

g.) Date, time, path, md, rd, cd

(i) Date- Displays or sets the system date.

Syntax-

DATE [mm-DD-YYYY]

(ii) time- Displays or sets the system time.

Syntax-

TIME [HH:MM:SS]

(iii) path- Displays or sets a search path for executable file.

(iv) rd- rd is used to delete empty directories and optionally, to delete directories and all of their contents. or also known as rmdir remove directories.

Syntax-

rd [d:]path. or. $rmdir syit.

(v) cd- To navigate between different folder.

Syntax-

$cd Ashish

Teacher's Signature with Date :

i) output - C:\> copy con q1.txt
This is my first file
^Z — (Ctrl + Z)
1 file(s) copied
C:\> type q1.txt — (show copied file)

ii) output - C:\> xcopy Aathish Shyam.

iii) output - C:\>del q1.txt

iv) output - C:\> move q1.txt q6.txt
1 file(s) moved

v) output - C:\> clear

vi) output - C:\Windows\system32> chkdsk
— (Run as Administrator)

h) copy, xcopy, del, move, cls, chkdsk.

(i) copy - copy files from one directory to another.

Syntax -

$copy q1.txt b1.txt.
con

(ii) xcopy - copies files or groups of files to and from directories.

Syntax -

xcopy [d:] [path] filename. /xcopy Aathish Shyam.

(iii) del - The del command is used to delete files from a disk.

Syntax -

del filename.

(iv) move - Rename and Replace the files.

Syntax -

$mv q1.txt b1.txt.

(v) cls - clear terminal.

Syntax - $clear.

(vi) chkdsk - The chkdsk command is used to check a disk for errors.

Syntax - chkdsk

Teacher's Signature with Date :

i) output- C:\> doskey

ii) output- C:\> echo "Good Afternoon"
Good Afternoon

i) doskey, echo.

ii) doskey – The Doskey command loads the Doskey program into memory. which allows you to: Recall Dos commands, Edit command lines and create and run macros.

Syntax-
C:\> doskey.

iii) echo – Display active processes on the terminal.

Syntax-
echo "Good Morning".

Teacher's Signature with Date :

i) output- C:\>fc a2.txt a5.txt

ii) Output- C:\> find "This" a1.txt

iii) output- C:\> rename a6.txt myfile2.txt

iv) Output- C:\> ~~mkdir Aathish~~
~~type Aath~~
C:\> type a6.txt

v) Output- C:\>ver

i) fc, find, rename, type, ver.

i) fc- The Fc command in DOS is a command-line program that compares two or more files and displays the difference between them.
syntax-
fc a2.txt a5.txt

(ii) find- FIND is a filter command (reads from input, transforms it, and outputs it to the screen, to a file, or to a printer.
Syntax-
find "This" a1.txt

(iii) rename- Rename and Replace the files.

Syntax-
rename a6.txt myfile2.txt

(iv) type - The file is displayed with limited on screen-formatting.

Syntax-
type a6.txt.

(v) ver - The ver comman is used to displays the version number of the currently active DOS.

Syntax-
$ver.

Teacher's Signature with Date : 10

practical-3

3. Linux Commands-

C.) pwd, cd, ls, mkdir, rmdir.

(i) pwd – The pwd command is used to displays the current working directory's full path.

syntax–

pwd.

(ii) cd – To navigate between different folders.

syntax –

cd Ashish.

(iii) ls – Displays information about files in the current directory.

syntax–

$ls.

(iv) mkdir – Creates a directory.

syntax–

$mkdir syit

(v) rmdir – Removes empty directory from the directory list

syntax–

$rmdir syit.

Teacher's Signature with Date : _____

d) touch, rm, cp, mv, rename, head, tail, cat, more, chmod.

(i) touch – Create empty files.

Syntax –

$touch.

(ii) rm – Delete files.

Syntax –

$rm C1.txt

(iii) cp – Copy files from one directory to another.

Syntax –

$cp a1.txt b1.txt

(iv) mv – Rename and Replace the files.

Syntax –

$mv a1.txt b1.txt

(v) rename – Rename and Replace the files.

syntax –

rename a1.txt a2.txt

Teacher's Signature with Date : _____

(vi) head – Extract first lines of file.

Syntax –

$ head +2 911.txt.

(vii) tail – Extract last lines of file.

Syntax –

$ tail -1 911.txt

(viii) cat – The cat command is used to read, display and concatenate text files.

Syntax –

$ Cat 96.txt

(ix) more – The more command is used to command line utility that allows users to view text files one page at a time.

Syntax –

$ more.

(x) chmod – the chmod (change mode) command in linux is used to change file and directory permissions.

Syntax –

$ chmod +x k1.txt.

Teacher's Signature with Date : _____

e) ps, top, bg, fg.

i) ps- Display the process in terminal.

syntax-

$ps.

(ii) top- Display current active process.

syntax-

$top.

(iii) bg- Background process.

Syntax-

$bg %1..

(iv) fg- foreground process.

Syntax-

$fg %1.

(v) Kill- The Kill command sends a signal to a process.

Syntax-

Kill 338)

kill all bash

Teacher's Signature with Date :

f7 grep, locate, find.

i) grep- search for a specific in an output.

syntax-

$grep first C1.txt

ii) locate- Find a file in the database.

Syntax-

$locate 91.txt.

iii) find- Find is a filter command (ready from input transforms it, and outputs it to the screen to a filter or to a printer.

syntax-

find "this "91.txt.

Teacher's Signature with Date: 1.22

g.) date, cal, uptime, uname, man, du

(i) date - Displays or sets the system date.

syntax -

$ DATE [mm-DD-YYYY]

(ii) cal - Cal command is a calendar command in linux which is used to see the calendar of a specific month or a whole year.

syntax -

$ cal

(iii) uptime - The uptime command in linux shows how long a system has been running, the current time, and the number of active user

syntax -

$uptime.

(iv) Uname - the uname command writes to standard output the name of the operating system that you are using.

syntax -

$unam

$uname -r

(v) man - man command in linux is used to display the user manual of any command that we can run on the terminal.

Teacher's Signature with Date :

syntax-

$man ls

(vi) du- The du command in linux, or "disk usage", is a console command that reports how much disk space is used by files and dictionaries.

syntax-

$du.

h.) Compression : tar, gzip

(i) tar- The tar command in linux is used to create, extract, maintain, and modify archives of the files and directories.

syntax-

$tar cvf file1.tar *.txt

$tar xvf file1.tar *.txt.

(ii) gzip- The gzip command in linux is a command-line tool that compresses files into the GZIP file format.

syntax-

$gunzip f1.txt

Teacher's Signature with Date : (be)

Practical-4

4. working with Linux Desktop and utilities.

b.) The vi editor.

- The default editor that comes with the Linux/Unix operating system is called vi (visual editor). Using Vi editor, we can edit an existing file or create a new file from scratch. we can also use this editor to just read a text file. The advanced version of the vi editor is the vim editor.

modes of Operation in the vi editor.

1. TO create file-

$vi filename

ex. vi ks.txt

press insert button

type the data

good morning everyone

2. To save the file-

press Esc button once then shift + colon (your cursor position at end of the editor) wq.

:wq.

	Commands	Description
1.)	'k'	move the cursor up one line
2.)	'j'	move the cursor down one line

Teacher's Signature with Date :

	Commands	Description.
3)	'h'	Moves the cursor to the left one-character position.
4)	'l'	Moves the cursor to the right one-character position.
5)	'0'	positions cursor at beginning of line.
6)	'w'	positions cursor to the next word.
7)	'B'	positions cursor to previous word.
8)	$	positions cursor at end of line.
9)	'('	positions cursor to beginning of current sentence.
10)	')'	positions cursor to beginning of next sentence.
11)	'H'	Move to top of screen.
12)	'nH'	moves to nth line from the top of the screen.
13)	'M'	move to middle of screen.
14)	'L'	move to bottom of screen.
15)	'nl'	moves to nth line from the bottom of the screen.
16)	colon followed by a number position.	The cursor on the line number is represented by the number after the colon. for e.g. ":10" positions the cursor on line 10.

To move around within a file without affecting text must be in command mode (press Esc twice)

Teacher's Signature with Date :

(i) Graphic User Interface.

- The common and useful component of electronics such as computers, tablets, and smartphones is a visual interface called a Graphical User Interface (GUI).
- GUI is the interface that uses graphical elements to let people interact as per requirement with electronic devices including computers, laptops, tablets, and smart phones. In terms of human-computer interaction systems or technology. It's a very important component of software application programming since it substitutes actions for the text-based commands in the system, whether It's a text file, object, image or video as per requirement.

Components of GUI.-

i) pointers - The pointer appears on the User screen as a marking symbol. The pointer moves on to choose instructions and objects as per requirement.

ii) Icons - Icons allude to tiny visual representations of windows, documents, actions, and other things.

iii) pointing tool - At the initial stages, the pointing tool enables the user to select and move the required pointer items on the screen, including mouse & trackbo

iv) Desktop - The desktop is the screen that is the contained within the icons and user beneficial.

Teacher's Signature with Date :

d) working with Terminal

- Basic Navigation -
  - ls - list files in the current directory.
  - cd - change directory.
  - pwd - print the current working directory.
- windows -
  - dir - list files in the current directory.
  - cd - change directory.
  - echo %cd% - print the current working directory.
- File operations -
  - cp - copy a file
  - mv - move or rename a file.
  - rm - Remove a file.
- Creating and Editing files -
  - touch - create an empty file.
  - nano - Edit a file with nano text editor.
  - cat - Display file contents.
- windows -
  - type - Display running processes. file contents.
  - notepad - Edit a file. with Notepad.
- System Information -
  - top - Display running process.
  - df -h. - show disk space usage.
  - free -h. - Display memory usage.

Teacher's Signature with Date :

(i) Adjusting display resolution.

=> Steps to change Screen Resolution in Linux.

XRandR is a powerful command-line interface to interact with displays Anything related to setting the size, orientation or reflection of traditional linux displays is handled by xrandr. we will be using this utility to change screen resolution.

Step-1- List all the available displays.

Let's get the xrandr utility to display the name of the active moniter.

$xrandr --listactivemonitors.

Step-2 - List all the available resolutions for each displays.

Next, let's list out all the resolutions postibley on your moniters. This will help us set up the proper aspect ratio.

$xrandr

The output tells us the current resolution (1600 x 900) and all the available resolutions for eDP-1. If you have multiple screens hooked up. it will list the available resolutions for all those screens.

Teacher's Signature with Date :

f) Using the browsers.

- Using a web browser in Linux is quite similar to using one in other operating systems.

1. Using package managers-
   sudo apt update
   sudo apt install firefox.
   sudo dnf install firefox
   - Arch Linux -
     sudo pacman -S firefox.

2. Downloading Directly - you can download browsers like chrome or Brave directly from their official website and follow their installation instructions

- Launching a Browser -
- Via Terminal - you can start a browser by typing its name. for example.
  firefox &
- Via GUI - use the applications menu to find and launch your installed browser.

Teacher's Signature with Date :

9.) Configuring simple networking.

– Configuring simple networking in Linux involves setting up network interfaces, usually through command-line tools.

1. Identify your network interfaces –
First check which network interfaces are available –
ip a. — this command list all network interfaces.

2. Configuring a network interfaces –
Using ip command
to set a static IP address, you can use the ip command –
Sudo ip addr add 192.168.1.100/24 dev eth0
Sudo ip link set eth0 up.
Replace 192.168.1.100/24 with your desired IP and eth0 with your network interface name.

To configure a DHCP connection.
if you want to configure the interface to use DHCP –
Sudo dhclient eth0.

3. Configure Default Gateway –
To set the default gateway.
Sudo ip route add default via 192.168.1.1
Replace 192.168.1.1 with your gateway IP.

Teacher's Signature with Date :

b) Creating users and shares.

- Creating users and shares in Linux typically involves adding users and configuring file sharing often using samba for sharing with Windows clients or NFS for Linux clients.

Creating Users -

1. Add a New User - Use the adduser command to create a new user.

   Sudo adduser username.

2. Set User permissions - To modify user permissions or add the user to specific groups, use the usermod command.

   Sudo usermod -aG groupname username.

Creating Shares -

1. Create a share directory - Create a directory to share.

   Sudo mkdir /srv/samba/share
   sudo mkdir nobody:nogroup /srv/samba/share
   sudo chmod 0777 /srv/samba/share

2. Export the shares - Sudo exportsfs -a

Teacher's Signature with Date :

Output-

vi try1.c – (to write program)
gcc try1.c – (to compile this program)
./a.out try1.c – (to run this program)

Enter a number : 13
13 is odd
./a.out try1.c – (to run this program Again)
Enter a number : 10
10 is even

practic-5.

5.) Running c/ctt python programs in linux.

i) Running c programs in linux.

ii) write a c program even and odd in linux.

program -

```
#include<stdio.h>
main()
{
int n;
printf("Enter a number:");
scanf("%d", &n);
if(n%2==0)
{
printf("Number is even");
}
else
{
printf("Number is odd");
}
}
```

iii) write a c program simple interest in linux.

program -

```
#include<stdio.h>
main()
{
```

Teacher's Signature with Date :

Output- vi try2.c -(to write the program)
gcc try2.c -(to compile this program)
./a.out try2.c -(to run this program)

Enter the principal amount, number of year and rate respectively: 45
interest = 0.00

Output -

Vi try3.c -(to write the program)
gcc try3.c -(to compile this program)
./a.out try3.c -(to run this program)

enter a number: 5
cube of 5 is 125

Output-

Vi try4.c -(to write the program)
gcc try4.c -(to compile this program)
./a.out try4.c -(to run this program)

enter the number of factorial: 5
factorial fo of 5 is 120

```
float p,n,r, interest;
printf("Enter a principal amount, number of year
        and rate respectively:");
scanf("%f%f%f", &p, &n, &r);
interest = (p*n*r)/100;
printf("interest = %f", interest);
}
```

iii) write a C program cube of a number in linux.

program-

```
#include<stdio.h>
main()
{
int n;
printf("enter a number:");
scanf("%d", &n);
printf("Cube of %d is: %d", n, (n*n*n));
}
```

iv) write a C program factorial of a number in linux.

program-

```
#include<stdio.h>
main()
{
int i, fact = 1, terms;
printf("enter the number of factorial:");
scanf("%d", &terms);
```

Teacher's Signature with Date : _____

Output-

vi trys.c – (to write the program)
gcc trys.c – (to compile this program)
./a.out trys.c – (to run this program)

Enter a number: 10
positive

gcc trys.c – (Again to compile this program)
./a.out trys.c – (Again to run this program)

Enter a number: -10
Negative

```
for (i=1; i<=termy; i++)
{
fact = fact * i;
}
printf("factorial of %d is: %d", termy, fact);
}
```

v) write a C program to print the positive and negative number in linux.

program-

```
#include<stdio.h>
main()
{
int n;
printf("Enter a number:");
scanf("%d", &n);
if(n<0)
printf("Negative number");
else
printf("positive number");
}
```

Teacher's Signature with Date : 10

practical-6

6. Installing utility software on Linux and windows.

- A package management system is comprised of sets of tools and file formats that are used together to install, update, and uninstall Linux apps. The two most common package management systems are from Red Hat and Debian. Red Hat, CentOS, and Fedora all use the rpm system. (.rpm files) while Debian, Ubuntu, Mint and Ubuntu use dpkg (.deb files).

If you are a Linux user or system administrator chances are you've come across Yum (yellowdog Updater modified) at some point in your journey. yum is a powerful package management tool that simplifies the process of installing, updating and managing software on Red Hat based Linux distributions like CentOS and Fedora. In this article, we will delve into the most common yum commands, providing detailed explanations and real-world examples to help you harness its full potential.

Syntax-

#yum [options] [command] [package(s)]

Teacher's Signature with Date : _____

1) For install - This command is used to install packages on your system using 'yum'. Replace package-name with the name of the package you want to install. yum will automatically handle dependencies and download the necessary files from repositories.

#sudo yum install package-name.

2) For updation - Keeping your system up to date is important for security and performance. Running this command will check for updates for all installed packages and install any available updates.

#sudo yum update.

3) Remove or delete Software - Use this command to remove a package from your system. Replace 'package-name' with the name of the package you want to uninstall. yum will also remove any dependencies that are no longer needed.

#sudo yum remove package-name.

4) Search - This command allows you to search for packages by providing a keyword. yum will display a list packages that match the keyword. making it easier to find the package you need.

#Yum search keyword.

Teacher's Signature with Date :

Vi K1.sh — (to write program)
chmod +x K1.sh — (to compile this program)
./K1.sh — (to run this program)

Output –

```
Enter a: 10
Enter b: 2
Addition of a and b are: 12
subtraction of a and b are: 8
multiplication of a and b are: 20
Division of a and b are: 5
modulus of a and b are: 0
Increment operator when applied on 11 results
    into a: 11
Decrement operator when applied on 1 results
    into b: 1
```

Practical-7.

7. Introduction to linux shell scripting.

1.) Basic operators.

```
#!/bin/bash

read -r -p "Enter a:" a
read -r -p "Enter b:" b
add=$((a+b))
echo "Addition of a and b are:" ${add}
sub=$((a-b))
echo "Subtraction of a and b are:" ${sub}
mul=$((a*b))
echo "multiplication of a and b are:" ${mul}
div=$((a/b))
echo "Division of a and b are:" ${div}
mod=$((a%b))
echo "modulus of a and b are:" ${mod}
((++a))
echo "Increment operator when applied on $a
      results into a:" "${a}"
((--b))
echo "Decrement operator when applied on $b
      results into b:" "${b}"
```

Teacher's Signature with Date :

Output-

```
Vi s1.sh      — (to write program)
chmod +x s1.sh — (to compile program)
./s1.sh       — (to run this program)
```

```
if condition
a is greater
```

Output-

```
Vi s2.sh  — (to write program)
chmod +x s2.sh — (to compile program)
./s1.sh  — (to run this program)
```

a is less than b

9) Decision Making

i) If condition-

program-

```
#!/bin/bash
echo "if condition"
a=10
b=20
if [ $a -lt $b ]
then
echo "a is greater"
fi
```

ii) If — else condition-

program-

```
#!/bin/bash
echo "if else condition"
a=5
b=20
if [ $a -gt $b ]
then
echo "a is greater"
else
echo "a is less than b"
fi
```

Teacher's Signature with Date :

Output-

vi s4.sh – (to write program)
chmod +x s4.sh – (to compile program)
./s4.sh – (to run this program)

Enter a number: 5
positive

chmod +x s4.sh – (Again to compile program)
./s4.sh – (Also again run this program)
Enter a number: -5
negative

chmod +x s4.sh – (to compile program)
./s4.sh (to run this program)
Enter a number: 0
neither positive nor negative

iii) If – elif – else condition-

program-

```
#!/bin/bash
echo "Enter a number"
read num
if [ $num -lt 0 ]
then
echo "negative"
elif [ $num -gt 0 ]
then
echo "positive"
else
echo "neither positive nor negative"
fi
```

Teacher's Signature with Date :

Output-

Vi t1.sh – (to write program)
chmod +x t1.sh – (to compile program)
./t1.sh – (to run this program)

1
2
3
4
5
6

Output-

Vi t2.sh – (to write program)
chmod +x t2.sh – (to compile program)
./t2.sh – (to run this program)

1
2
3
4
5
6
7

i) Regular

ii) Looping –

i) program –

```
#!/bin/bash
n=7
for (( i=1; i<n; i++))
{
echo $i;
}
```

ii) program –

```
#!/bin/bash
for n in {1..7}
do
echo $n;
done
```

Teacher's Signature with Date :

Output-

vi s1.sh — (to write program)
chmod +x s1.sh — (to compile program)
./s1.sh — (to run this program)

Enter the day number
4
wednesday.

Output-

vi s2.sh — (to write program)
chmod +x s2.sh — (to compile program)
./s2.sh — (to run this program)

echo Enter the file Name
~~read~~ K1.sh

The file exists.

i) Regular Expression.
ii) special variable and command line arguments.

1) program-

```
echo Enter the day number
read num
case $num in
1) echo "Sunday";;
2) echo "Monday";;
3) echo "Tuesday";;
4) echo "Wednesday";;
5) echo "Thursday";;
6) echo "Friday";;
7) echo "Saturday";;
*) echo Enter the number bet 1 to 7;;
esac
```

2) program-

```
echo Enter the file name
read fname
if [ -f $fname ]
then
echo The file exists
else
echo File with the name $fname
     does not exists
fi
```

Teacher's Signature with Date :