

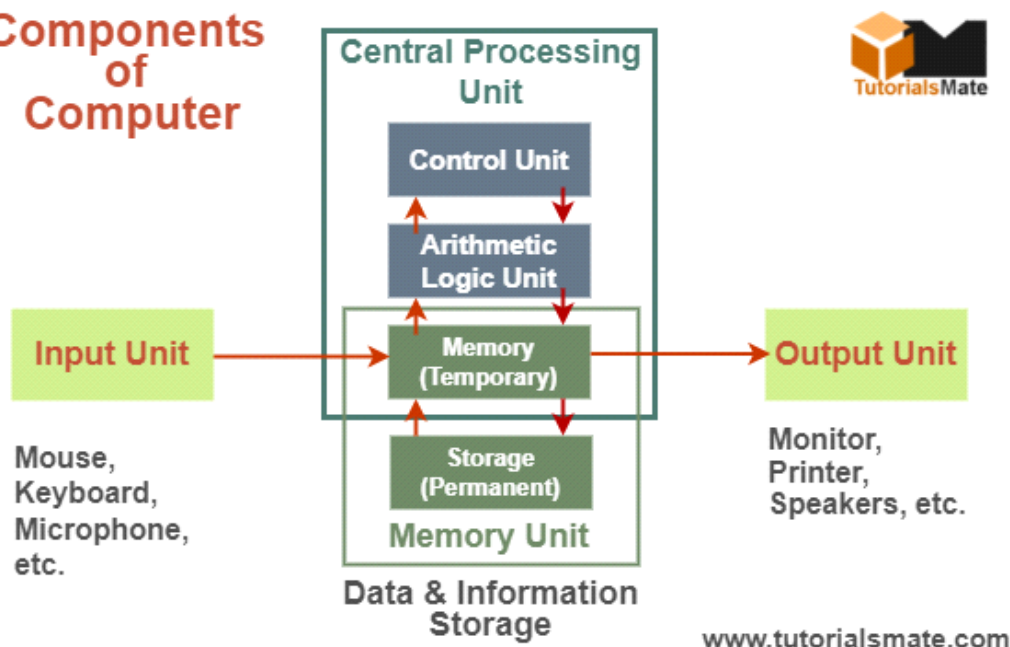
AIM:Explain about Components of a computer.

A computer system is mainly made up of three primary components, namely:

- Input Unit
- Central Processing Unit
- Output Unit

All these components participate in almost every task or activity performed on the computer system. While there are many other essential components, the above ones are the building blocks of any computer system, which ensure the smooth functioning of the computer.

Components of Computer



Input Unit:

The input unit contains the devices required to input data into a computer system. Since computers do not react or operate on their own, the input unit is one of the main components of a computer. This unit establishes a link between the user and the computer so that the user can direct commands and data into the computer.

The input unit transmits the data to the computer's primary memory (main memory), which is then processed by the computer's processor according to given instructions.

Eg: Mouse, Keyboard, Microphone.

Central Processing Unit:

The Central Processing Unit (CPU) is an essential electronic hardware component that controls and processes all functions in a computer system, including arithmetic and logical operations. The CPU is called the brain of the computer because it controls the operation of all the parts of the computer. Despite this, the computer's processor (CPU) also handles the operation of many

other units within the computer system, such as the arithmetic logic unit, control unit and memory unit. The Control Unit (CU) and the Arithmetic Logic Unit (ALU) are commonly called components of the CPU because they both together make up the CPU.

Control Unit:Control Unit (CU) is the main unit of the computer system as it controls all the operations and activities of the computer.

Arithmetic Logic Unit:It handles basic to complex operations of addition, subtraction, multiplication, and division .It also performing logical operations like AND, OR, equal to, less than, and greater than, etc.

Memory Unit:it is required by the CPU to store data and instructions.

Output Unit:

The output unit contains the devices needed to receive and view information from a computer system. The devices used to receive information or results from the computer are called output device. The output unit also establishes a link between the user and the computer. The output unit retrieves processed data from the computer's primary memory (main memory), which is converted into a human-understandable form before being displayed by the corresponding output device.

Eg:Monitor,Printer,Speakers.

AIM: Physical identification of major components of a computer system such as mother board, RAM modules, daughter cards, bus slots, internal storage devices, interfacing ports.

Motherboard:

The motherboard is defined as a circuit board for the computer system, also called logic board or mainboard. In the computer system, the biggest component is the motherboard that controls all the components of the computer system and establishes a link between all components. From the motherboard, different components like ROM, CPU, RAM, PCI slots, USB ports, and other components are connected. The computer system starts using the motherboard and these components act as the backbone for starting the system.

Components of motherboard:

1. Keyboard and mouse
2. Universal Serial Bus (USB)
3. Parallel port
4. CPU chip
5. RAM slots
6. Floppy Controller
7. IDE controller
8. PCI slot
9. ISA slot
10. CMOS Battery
11. AGP slot
12. CPU slot
13. Power supply slot

Types of Motherboard:

1. AT Motherboard
2. ATX Motherboard
3. LPX Motherboard
4. BTX Motherboard
5. Pico BTX motherboard

6. Mini ITX motherboard

RAM Modules:

There are two basic distinctions of memory. One is volatile memory where the data is lost as soon as power is removed, and one is non-volatile that can store the data without power.

Random access memory (RAM) is used as read-write memory, which the processor can use as a scratch pad and modify rapidly.

Types of RAM modules:

- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Synchronous Dynamic RAM (SDRAM)
- Single Data Rate Synchronous Dynamic RAM (SDR SDRAM)
- Double Data Rate Synchronous Dynamic RAM (DDR SDRAM, DDR2, DDR3, DDR4)

Daughter cards:

The daughter card is a computer hardware. It is also known as the piggyback board, riser card, daughter board, daughter card. A daughter board is a printed circuit board which is connected to the motherboard or expansion card. As compared to the motherboard, it is smaller in size.

Types of daughter cards:

- MIDI
- Modem
- MPEG decoder
- Network Card
- Sound card
- Tuner card
- Video capture card
- Video card

Bus slots/Expansion slots:

An expansion slot is a socket on the motherboard that is used to insert an expansion card (or circuit board), which provides additional features to a computer such as video, sound, advanced graphics, Ethernet or memory. The expansion card is plugged directly into the expansion port so that the motherboard has direct access to the hardware. However, since all computers have a limited number of expansion slots.

Types of Bus slots:

- **PCI Express** – Video card
- **AGP** – Video card

- **ISA** – Network card, Sound card, Video card
- **AMR** – Modem, Sound card
- **CNR** – Modem, Network card, Sound card
- **EISA** – SCSI, Network card, Video card
- **PCI** – Network card, SCSI, Sound card, Video card

Internal storage devices:

Most computers have some form of internal storage. The most common type of internal storage is the **hard disk**. At the most basic level, internal storage is needed to hold the operating system so that the computer is able to access the input and output devices. Internal storage allows the data and applications to be loaded very rapidly into memory, ready for use. The data can be accessed much faster than data which is stored on an external storage device. There are three main categories of Internal storage devices: **optical, magnetic and semiconductor**.

Interfacing ports:

A port is a physical docking point using which an external device can be connected to the computer. It can also be programmatic docking point through which information flows from a program to the computer or over the Internet. It is an interface between the motherboard and an external device of the computer. There are different types of ports available:

- Serial port
- Parallel port
- USB port
- PS/2 port
- VGA port
- Modem port
- FireWire Port
- Sockets
- Infrared Port
- Game Port
- Digital Video Interface(DVI) Port
- Ethernet Port

AIM:SMPS

Switched-Mode Power Supply (SMPS) is an electronic circuit which converts the power using switching devices that are turned on and off at high frequencies, and storage components such as inductors or capacitors to supply power when the switching device is in its non-conduction state. It can be abbreviated as SMPS.

SMPS Parts and connectors:

Power-IN: A power cable is inserted to MAIN, the other end of which is connected to mains supply. The input supply gets converted to DC supply.

Power-OUT: The power-OUT connector is connected directly to the Power-IN connector from inside the supply unit. It supplies the same AC supply that is fed to power-IN socket. The power-OUT connector is used to give supply to monitors or any display unit.

FAN:At the back side of Computer-SMPS, find a FAN at the right side. The FAN blows the air out and is only used to dissipate the internal heat from the SMPS since the switching is done at high frequencies which create a lot of heat inside.

ATX Connector:It is a 24-pin female connector which is used to supply DC supply to the motherboards. Various color-coded wires connect to this connector and each colored wire supplies distinct DC voltage.

ATX-12V connector: Latest SMPS power supplies are accompanied by an extra 4-pin connector which supplies 12 volts to energize the central processing unit and other components of a mother board.

AT Connectors:Earlier motherboards used to support AT connectors(6-pin each) also called P8 and P9 connectors to supply power to these motherboards(upto 486 boards).

4-PIN connectors: There are multiple 4-pin connectors that draw out from the SPMS unit. These connectors are used to supply DC power to various peripherals of computer like a floppy disk drive, hard disk drive or DVD-writers.

SATA-output connector. To feed the power to latest SATA hard drives, these connectors are used.

PROGRAM NO:4

DATE:30-03-2022

AIM: Specifications of desktop computers.

- ☐ Motherboard: Gigabyte B560M-DS3H Motherboard
- ☐ Case: In Win CE685 PC case with 300 watt 80PLUS certified power supply
- ☐ CPU: Intel Core i5-11400 Processor
- ☐ RAM: 8GB DDR4
- ☐ NIC: On-board 10/100/1000 Mbps based Ethernet NIC
- ☐ HDD: 250 GB Western Digital M.2 NVMe SSD Part #WDS250G2B0C
- ☐ Video: Integrated Intel HD Graphics with DVI Digital Output Interface
- ☐ Optical Drive: (optional)
- ☐ Audio: Onboard HD audio
- ☐ Externally powered Speakers: (optional)
- ☐ 6 External Powered USB Ports and 2 on front of case
- ☐ Operating System: Microsoft Windows 7 Home Basic(or other least expensive Microsoft Windows operating system)
- ☐ All appropriate cables necessary
- ☐ HIDs: USB Optical mouse, mouse pad, USB keyboard, all necessary cables
- ☐ Monitor: 20" Wide Screen LCD Monitor (1920×1080 native resolution) with DVI Digital Input Interface

AIM: Install latest version of Ubuntu on a virtual box

Download VirtualBox:

- Go to VirtualBox website to download the binary for your current operating system.
- Since our host machine is running on Windows, I'll choose 'x86/amd64' from Windows hosts.
- When download is finished, run the executable file.
- Continue with the installation of VirtualBox with the defaults.
- This will open VirtualBox at the end of the installation

Create Virtual Machine:

- Open VirtualBox.
- Click 'New' button to open a dialog.
- Type a name for the new virtual machine. VirtualBox automatically changes 'Type' to Linux and 'Version' to 'Ubuntu (64 bit)'. Click 'Next' button.
- Choose the memory size. The memory size depends on your host machine memory size. Click 'Next' button.
- Accept the default 'Create a virtual hard drive now' and click 'Create' button.
- Continue to accept the default 'VDI' drive file type and click 'Next' button.
- Change the storage type from the default 'Dynamically allocated' to 'Fixed size' to increase performance and click 'Next' button.
- Choose the file location and size.
- Click 'Create' button and VirtualBox will generate Ubuntu virtual machine.

The virtual machine is created

Install Ubuntu in Virtual Machine:

- Select your new virtual machine and click 'Settings' button.
- Click on 'Storage' category and then 'Empty' under Controller:IDE.
- Click "CD/DVD" icon on right hand side and select the ubuntu ISO file which is already downloaded.
- Click 'OK' button to continue.
- Click on the new Ubuntu virtual machine and hit 'Start' button
- Click 'Install Ubuntu' button.
- Click 'Continue' button.
- Make sure 'Erase disk and install Ubuntu' option is selected and click 'Install Now' button.
- Ubuntu will ask you a few questions like location, keyboard layout, name etc. If the default is good, click 'Continue' button.
- The installation will continue until it is finished.
- After installation is complete, click 'Restart Now' button.
- The Ubuntu Desktop OS is ready.

AIM: Study of a terminal-based text editor such as Vim

Vim is an advanced and highly configurable text editor built to enable efficient text editing.

To install vim on ubuntu run the command:

```
sudo apt-get install vim
```

You can open vim by running vim command on the terminal.

```
vim
```

To open a file in vim editor just write the file name after the vim command in the terminal as follows:

```
vim filename.txt
```

Then the file will be opened.

Write into file

write some content in to write data it needs to go in insert mode. To go into write mode type i.

After going into insert mode you will see INSERT in the status bar. After that, we can write any data in it.

Save and Exit:

We have written the data into a file now the task is to save and close the file to do that first exit from insert mode by pressing the Esc key. To write a command first type semicolon (:) and then type the command wq! And then hit ENTER.

```
:wq!
```

Exit without saving the file:

To exit from the file without saving the file just use the command q! As follows

```
:q!
```

AIM: Basic Linux commands**1. pwd Command**

The pwd command is used to display the location of the current working directory.

```
student@student-Pegatron:~$ pwd
```

```
/home/student
```

2. mkdir Command

The mkdir command is used to create a new directory under any directory.

```
student@student-Pegatron:~/example1$ mkdir mca2
```

```
student@student-Pegatron:~/example1$ ls
```

```
mca mca2
```

3. rmdir Command

The rmdir command is used to delete a directory.

```
student@student-Pegatron:~/example1$ rmdir mca2
```

```
student@student-Pegatron:~/example1$ ls
```

```
mca
```

4. ls Command

The ls command is used to display a list of content of a directory.

```
student@student-Pegatron:~/Desktop/thasni/netwrk/network$ ls
```

```
arithmetic.sh file1.sh ifloop.sh sum1.sh while1.sh
```

```
arithmetricfn.sh file2.sh mark.sh sumfn.sh while2.sh
```

```
break1.sh for1.sh multiplication.sh sumofdigit.sh
```

5. cd Command

The cd command is used to change the current directory.

```
student@student-Pegatron:~/Desktop/thasni/netwrk$ cd
```

```
student@student-Pegatron:~$
```

6. mv Command

The mv command is used to move a file or a directory from one location to another location.

```
student@student-Pegatron:~/example1/mca$ ls
```

```
break1.sh for1.sh
```

```
student@student-Pegatron:~/example1/mca$ mv break1.sh continue.sh
```

```
student@student-Pegatron:~/example1/mca$ ls
```

```
continue.sh for1.sh
```

7. cat Command

The cat command is used to create a file, display content of the file, copy the content of one file to another file, and more.

```
student@student-Pegatron:~/Desktop/thasni/netwrk/network$ cat for1.sh
```

```
#!/bin/bash
```

```
for i in 1 2 3 4 5
```

```
do
```

```
echo "Welcome $i times"
```

```
done
```

8. rm Command

The rm command is used to remove a file.

```
student@student-Pegatron:~/Desktop/s2mca$ ls
```

```
break1.sh evenodd.sh greatest.sh ifloop.sh mark.sh sum.sh while2.sh
```

```
student@student-Pegatron:~/Desktop/s2mca$ rm evenodd.sh
```

```
student@student-Pegatron:~/Desktop/s2mca$ ls
```

```
break1.sh greatest.sh ifloop.sh mark.sh sum.sh while2.sh
```

9. cp Command

The cp command is used to copy a file or directory.

```
student@student-Pegatron:~/Desktop/s2mca$ cp copy example4
```

```
student@student-Pegatron:~/Desktop/s2mca$ cat example4
```

```
hai friends
```

10. head Command

The head command is used to display the content of a file. It displays the first 10 lines of a file.

```
student@student-Pegatron:~/Desktop/s2mca$ head -2 employeee.sh
```

```
#!/bin/bash
```

```
echo "enter the employee number"
```

11. tail Command

The tail command displays the last ten lines of the file content. It is useful for reading the error message.

```
student@student-Pegatron:~/Desktop/s2mca$ tail -2 employeee.sh
```

```
echo "Gross salary:$gross"|tee cat >> emp.dat
```

```
echo "Net salary:$netsal"|tee cat >>emp.dat
```

12. more command

The more command is used to display the file content in the same way that the cat command does. The only difference between both commands is that, in case of larger files, the more command displays screenful output at a time.

13. less Command

The less command is similar to the more command. It also includes some extra features such as adjustment in width and height of the terminal.

14. useradd Command

The useradd command is used to add or remove a user on a Linux server.

```
student@student-Pegatron:~$ sudo useradd user1
```

```
[sudo] password for student:
```

```
student@student-Pegatron:~$ sudo passwd user1
```

```
Enter new UNIX password:
```

Retype new UNIX password:

passwd: password updated successfully

student@student-Pegatron:~\$ sudo mkdir /home/user1

15. passwd Command

The passwd command is used to create and change the password for a user.

student@student-Pegatron:~\$ sudo passwd user1

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

16. cut Command

The cut command is used to select a specific column of a file.

student@student-Pegatron:~/Desktop/s2mca\$ cat copy

hai friends

student@student-Pegatron:~/Desktop/s2mca\$ cut -b -2 copy

ha

17. grep Command

The grep is the most powerful and used filter in a Linux system. The 'grep' stands for "**global regular expression print.**" It is useful for searching the content from a file.

student@student-Pegatron:/home\$ cat /etc/passwd | grep user1

user1:x:1001:1001::/home/user1:

18. wc Command

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

student@student-Pegatron:~/Desktop/s2mca\$ wc sum.sh

7 15 113 sum.sh

19. find Command

The find command is used to find a particular file within a directory.

The following symbols are used after the find command:

(.) : For current directory name

(/) : For root

```
student@student-Pegatron:~/Desktop/s2mca$ find .
```

```
./mark.sh
```

```
./greatest.sh
```

```
./while2.sh
```

```
./copy
```

```
./file5
```

```
./sum.sh
```

```
./break1.sh
```

```
./employeee.sh
```

```
./ifloop.sh
```

20. df Command

The df command is used to display the disk space used in the file system. It displays the output as in the number of used blocks, available blocks, and the mounted directory.

```
student@student-Pegatron:~/Desktop/s2mca$ df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
------------	-----------	------	-----------	------	------------

udev	2963740	0	2963740	0%	/dev
------	---------	---	---------	----	------

tmpfs	597048	8892	588156	2%	/run
-------	--------	------	--------	----	------

/dev/sda5	144052920	8633704	128078672	7%	/
-----------	-----------	---------	-----------	----	---

tmpfs	2985228	276	2984952	1%	/dev/shm
-------	---------	-----	---------	----	----------

tmpfs	5120	4	5116	1%	/run/lock
-------	------	---	------	----	-----------

tmpfs	2985228	0	2985228	0%	/sys/fs/cgroup
-------	---------	---	---------	----	----------------

tmpfs	597048	88	596960	1%	/run/user/1000
-------	--------	----	--------	----	----------------

```
/dev/sda6 92155928 6509596 80941976 8% /media/student/cde88486-b303-4d69-85cc-8b381ac8b53c
```

21. ssh Command

Linux ssh command is used to create a remote connection through the ssh protocol.

```
student@student-Pegatron:~/Desktop/s2mca$ ssh
```

```
usage: ssh [-1246AaCfGgKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
```

```
[-D [bind_address:]port] [-E log_file] [-e escape_char]
```

```
[-F configfile] [-I pkcs11] [-i identity_file] [-L address]
```

```
[-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
```

```
[-Q query_option] [-R address] [-S ctl_path] [-W host:port]
```

```
[-w local_tun[:remote_tun]] [user@]hostname [command]
```

22. man command

man command shows the manual pages of the cd command. Typing in the command name and the argument helps it show which ways the command can be used.

CUT(1) User Commands CUT(1)

NAME

cut - remove sections from each line of files

SYNOPSIS

cut OPTION... [FILE]...

DESCRIPTION

Print selected parts of lines from each FILE to standard output.

With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options N too.

-b, --bytes=LIST

select only these bytes

-c, --characters=LIST

select only these characters

-d, --delimiter=DELIM

23. echo

The echo command helps us move some data, usually text into a file.

```
student@student-Pegatron:~/Desktop/s2mca$ echo "hai"
```

```
hai
```

24. tar

It can be used to compress and uncompress different types of tar archives like .tar, .tar.gz, .tar.bz2, etc. It works on the basis of the arguments given to it.

25. chmod command

chmod to make a file executable and to change the permissions granted to it in Linux.

```
student@student-Pegatron:~/Desktop/s2mca$ chmod +x sum.sh
```

26.chown command

chown command is used to change the file Owner or group. Whenever you want to change ownership you can use chown command.

```
student@student-Pegatron:/home$ sudo chown user1:user1 /home/user1
```

```
student@student-Pegatron:/home$ cat /etc/passwd | grep user1
```

```
user1:x:1001:1001::/home/user1:
```

27.read command

read command in Linux system is used to read from a file descriptor.

```
student@student-Pegatron:~/Desktop/s2mca$ echo "what is your name";read name;echo  
"hello $name";
```

```
what is your name
```

```
thasni
```

```
hello thasni
```


28. paste command

Paste command is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output.

```
student@student-Pegatron:~/Desktop/s2mca$ paste -d "|" copy sum.sh
```

```
hai friends|#!/bin/bash
```

```
|echo "First number"
```

```
|read num1
```

```
|echo "second number"
```

```
|read num2
```

```
|sum=$(( $num1+$num2 ))
```

```
|echo "Sum is:$sum"
```

29.expr command

The expr command in Unix evaluates a given expression and displays its corresponding output.

```
student@student-Pegatron:~/Desktop/s2mca$ x=network
```

```
student@student-Pegatron:~/Desktop/s2mca$ sub=`expr substr $x 2 3`
```

```
student@student-Pegatron:~/Desktop/s2mca$ echo $sub
```

```
etw
```

30.Redirection

Redirection is a feature in Linux such that when executing a command, you can change the standard input/output devices.

31.Piping

A pipe is a form of redirection (transfer of standard output to some other destination).

The Unix/Linux systems allow stdout of a command to be connected to stdin of another command by using the pipe character '|'.

32.usermod Command

usermod command or modify user is a command in Linux that is used to change the properties of a user in Linux through the command line.

```
student@student-Pegatron:~/Desktop/s2mca$ sudo usermod -c "this is a msg" user1
```

```
[sudo] password for student:
```

```
student@student-Pegatron:~/Desktop/s2mca$ cat /etc/passwd | grep user1
```

```
user1:x:1001:1001:this is a msg:/home/user1:
```

33.userdel Command

For deleting the user

```
sudo userdel -r user1
```

34.topCommand

top command is used to show the Linux processes.Usually, this command shows the summary information of the system and the list of processes or threads

```
student@student-Pegatron:~/Desktop/s2mca$ cat sum.sh | head -4 | tail -2
```

```
read num1
```

```
echo "second number"
```

35.ps Command

ps command is used to list the currently running processes and their PIDs along with some other information depends on different options.

```
student@student-Pegatron:~/Desktop/s2mca$ ps
```

```
PID TTY TIME CMD
```

```
15810 pts/1 00:00:00 bash
```

```
20811 pts/1 00:00:00 ps
```

36.scp Command

scp (secure copy) command in Linux system is used to copy file(s) between servers in a secure way.

Syntax:scp [-346BCpqrTv] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit] [-o ssh_option] [-P port] [-S program] [[user@]host1:]file1 ... [[user@]host2:]file2

37.ssh-keygen

Use the ssh-keygen command to generate a public/private authentication key pair. Authentication keys allow a user to connect to a remote system without supplying a password. Keys must be generated for each user separately. If you generate key pairs as the root user, only the root can use the keys.

38.Ssh-copy-id

One of the useful tools of SSH is ssh-copy-id that is used to install the SSH key on the remote server to login into the server without providing the password for the login. So, the single sign-on and automated password-less login using SSH can be implemented easily using this tool.

AIM: File system hierarchy in a common Linux distribution

- 1./ (Root):** Primary hierarchy root and root directory of the entire file system hierarchy.
- 2. /bin :** Essential command binaries that need to be available in single-user mode; for all users, e.g., cat, ls, cp.
- 3. /boot :** Boot loader files, e.g., kernels, initrd.
- 4. /dev :** Essential device files, e.g., /dev/null.
- 5. /etc :** Host-specific system-wide configuration files.
- 6. /home :** Users' home directories, containing saved files, personal settings, etc
- 7. /lib :** Libraries essential for the binaries in /bin/ and /sbin/.
- 8. /media :** Mount points for removable media such as CD-ROMs (appeared in FHS-2.3).
- 9. /mnt :** Temporarily mounted filesystems.
- 10. /opt :** Optional application software packages.
- 11. /sbin :** Essential system binaries, e.g., fsck, init, route.
- 12. /srv :** Site-specific data served by this system, such as data and scripts for web servers, data offered by FTP servers, and repositories for version control systems.
- 13. /tmp :** Temporary files. Often not preserved between system reboots, and may be severely size restricted.
- 14. /usr :** Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.
- 15. /proc :** Virtual filesystem providing process and kernel information as files. In Linux, corresponds to a procfs mount. Generally automatically generated and populated by the system, on the fly.

PROGRAM NO:9

DATE:28-04-2022

AIM: Check a number whether even or odd in shell script.

SOURCE CODE:

```
#!/bin/bash

Clear

echo "EVEN OR ODD IN SHELL SCRIPT"

echo -n "Enter a number"
read n
echo -n "Result "
if [ `expr $n % 2` == 0 ]
then
    echo "$n is even"
else
    echo "$n is odd"
fi
```

OUTPUT:

---- EVEN OR ODD IN SHELL SCRIPT ----

Enter a number:71

RESULT: 71 is Odd

PROGRAM NO:10

DATE:05-05-2022

AIM: Shell script to find the greatest of three numbers

SOURCE CODE:

```
echo "Enter Num1"
read num1
echo "Enter Num2"
read num2
echo "Enter Num3"
read num3
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
    echo "Greater Number is" $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
    echo "Greater Number is" $num2
else
    echo "Greater Number is" $num3
fi
```

OUTPUT:

Output:

Enter Num1

41

Enter Num2

50

Enter Num3

13

Greater Number is 50

PROGRAM NO:11

DATE:11-05-2022

AIM: Arithmetic operations using shell programming (using switch - case)

SOURCE CODE:

```
echo "Enter Two Numbers"

read a

read b

echo "What do you want to do? (1 to 5)"

echo "1) Sum"

echo "2) Difference"

echo "3) Product"

echo "4) Quotient"

echo "5) Remainder"

echo "Enter your Choice"

read n

case "$n" in

1) echo "The Sum of $a and $b is `expr $a + $b`";;

2) echo "The Difference between $a and $b is `expr $a - $b`";;

3) echo "The Product of the $a and $b is `expr $a \* $b`";;

4) echo "The Quotient of $a by $b is `expr $a / $b`";;

5) echo "The Remainder of $a by $b is `expr $a % $b`";;

Esac
```

OUTPUT:

Enter Two Numbers

14

18

What do you want to do? (1 to 5)

1) Sum

2) Difference

3) Product

4) Quotient

5) Remainder

Enter your Choice

1

The Sum of 14 and 18 is 32

student@student-Pegatron:~/Documents\$./pgm5.sh

Enter Two Numbers

45

5

What do you want to do? (1 to 5)

1) Sum

2) Difference

3) Product

4) Quotient

5) Remainder

Enter your Choice

5

The Remainder of 45 by 5 is 0

PROGRAM NO:12

DATE:20-05-2022

AIM: Write a shell script program to Check whether a number is prime or not.

SOURCE CODE:

```
echo -e "Enter Number : \c"

read n

for((i=2; i<=$n/2; i++))

do

    ans=$(( n%i ))

    if [ $ans -eq 0 ]

    then

        echo "$n is not a prime number."

        exit 0

    fi

done

echo "$n is a prime number."
```

OUTPUT:

Enter Number : 11

11 is a prime number.

PROGRAM NO:13

DATE:30-05-2022

AIM: Find the Sum of digits of a number

SOURCE CODE:

```
echo "Enter a number"

read num

sum=0

while [ $num -gt 0 ]
do
    mod=$((num % 10))  #It will split each digits
    sum=$((sum + mod)) #Add each digit to sum
    num=$((num / 10))  #divide num by 10.
done

echo "Sum of Digit is" $sum
```

OUTPUT:

Enter a number

142

Sum of Digit is 7

PROGRAM NO:14

DATE:03-06-2022

AIM: Shell Script program to find a factorial of a number

SOURCE CODE:

```
echo "Enter a number"

read num

fact=1

while [ $num -gt 1 ]

do

    fact=$((fact * num)) #fact = fact * num

    num=$((num - 1))    #num = num - 1

done

echo "Factorial of a number is $fact"
```

OUTPUT:

Enter a number

4

Factorial of a number is 24

PROGRAM NO:15

DATE:09-06-2022

AIM: Print even series in shell script up to n terms

SOURCE CODE:

```
echo "-----EVEN SERIES-----"
```

```
echo -n "Enter a number: "
```

```
checker=0
```

```
read num
```

```
while test $checker -le $num
```

```
do
```

```
ii=`expr $checker % 2`
```

```
    if test $ii -eq 0
```

```
    then
```

```
        echo "$checker"
```

```
    fi
```

```
checker=`expr $checker + 1`
```

```
done
```

OUTPUT:

```
-----EVEN SERIES-----
```

```
Enter a number: 15
```

```
0
```

```
2
```

```
4
```

```
6
```

```
8
```

```
10
```

```
12
```

```
14
```

PROGRAM NO:16

DATE:22-06-2022

AIM: Write a shell program to perform get mark details of a student and display total and grade.

SOURCE CODE:

```
clear
echo -----
echo '\t Student Mark List'
echo -----
echo Enter the Student Name
read name
echo Enter the Register number
read rno
echo Enter the Mark1
read m1
echo Enter the Mark2
read m2
echo Enter the Mark3
read m3
echo Enter the Mark4
read m4
echo Enter the Mark5
read m5
tot=$(expr $m1 + $m2 + $m3 + $m4 + $m5)
avg=$(expr $tot / 5)
echo -----
echo '\tStudent Mark List'
echo -----
```

```

echo "Student Name   : $name"
echo "Register Number : $rno"
echo "Mark1          : $m1"
echo "Mark2          : $m2"
echo "Mark3          : $m3"
echo "Mark4          : $m4"
echo "Mark5          : $m5"
echo "Total          : $tot"
echo "Average        : $avg"
if [ $m1 -ge 35 ] && [ $m2 -ge 35 ] && [ $m3 -ge 35 ] && [ $m4 -ge 35 ] && [ $m5 -ge 35 ]
then
    echo "Result      : Pass"
    if [ $avg -ge 90 ]
    then
        echo "Grade      : S"
    elif [ $avg -ge 80 ]
    then
        echo "Grade      : A"
    elif [ $avg -ge 70 ]
    then
        echo "Grade :B"
    elif [ $avg -ge 60 ]
    then
        echo "Grade :C"
    elif [ $avg -ge 50 ]
    then
        echo "Grade :D"

```

```
elif [ $avg -ge 35 ]
then
    echo "Grade :E"
fi
else
    echo "Result :Fail"
fi
echo -----
```

OUTPUT:

Student Mark List

Enter the Student Name

Riya

Enter the Register number

1527519

Enter the Mark1

42

Enter the Mark2

47

Enter the Mark3

49

Enter the Mark4

54

Enter the Mark5

50

Student Mark List

Student Name : Riya

Register Number : 1527519

Mark1 : 42

Mark2 : 47

Mark3 : 49

Mark4 : 54

Mark5 : 50

Total : 242

Average : 48

Result : Pass

Grade : E

PROGRAM NO:17

DATE:24-06-2022

AIM: Arithmetic operations using shell programming (using function)

SOURCE CODE:

```
#!/bin/sh

# Define your function here
add () {

#echo " Sum of $num1 and $num2 is:" $((($num1 + $num2))

echo "The Sum of $num1 and $num2 is `expr $num1 + $num2`"

}

sub () {
echo "The Difference between $num1 and $num2 is `expr $num1 - $num2`"

}

product(){

echo "The Product of the $num1 and $num2 is `expr $num1 \* $num2`"
}

Quotient(){
echo "The Quotient of $num1 by $num2 is `expr $num1 / $num2`"
}

Remainder(){
echo "The Remainder of $num1 by $num2 is `expr $num1 % $num2`"
}
input ()
{
echo "Enter the First Number"
read num1
echo "Enter the Second Number"
read num2
}
# Invoke your function

input
echo "What do you want to do? (1 to 5)"
echo "1) addition"
echo "2) Difference"
echo "3) Product"
echo "4) Quotient"
echo "5) Remainder"
```

```
echo "Enter your Choice"
read n
case "$n" in
1) add ;;
2) sub;;
3) product;;
4) Quotient;;
5) Remainder;;
Esac
```

OUTPUT:

```
Enter the First Number
7
Enter the Second Number
14
What do you want to do? (1 to 5)
1) addition
2) Difference
3) Product
4) Quotient
5) Remainder
Enter your Choice
1
The Sum of 7 and 14 is 21
student@student-Pegatron:~/Documents$ ./pgm12.sh
Enter the First Number
12
Enter the Second Number
4
What do you want to do? (1 to 5)
1) addition
2) Difference
3) Product
4) Quotient
5) Remainder
Enter your Choice
3
The Product of the 12 and 4 is 48
```

PROGRAM NO:18

DATE:27-06-2022

AIM: Write a shell script to accept the name of the file from standard input and perform the following tests on it

- a) File executable**
- b) File readable**
- c) File writable**
- d) Both readable & writable**

SOURCE CODE:

```
echo "enter a file name"
read file

if [ -e $file ]

then

echo "$file exists"

if [ -f $file ]

then

echo "$file is an ordinary file"

if [ -r $file ]

then
echo "$file has read access"
else
echo "$file does not have read access"
fi
if [ -w $file ]
then
echo "$file has write permission"
else
echo "$file does not have write permission"
fi
if [ -x $file ]
then
echo "$file has execute permission"
else
echo "$file does not have execute permission"
fi
if [ -r $file ] && [ -w $file ]
then
echo "$file has both read and write operations"
fi
```

```
elif [ -d $file ]
then
echo "$file is a directory"
fi
else
echo "$file does not exist"
fi
```

OUTPUT:

```
enter a file name
Sample
Sample exists
Sample is an ordinary file
Sample has read access
Sample has write permission
Sample does not have execute permission
Sample has both read and write operations
```

AIM: Write a menu driven shell script to copy, edit, rename and delete a file.

SOURCE CODE:

```
ch=0
while [ $ch -ne 9 ]
do
clear
echo "1.Display current dir"
echo "2.Listing the dir"
echo "3.Make a dir"
echo "4.Copy a file"
echo "5.Rename file"
echo "6.Delete file"
echo "7.Edit file"
echo "8.open or display file"
echo "9.Exit"
echo "Enter your choice"
read ch
case $ch in
1)echo "Current Dir is : "
pwd;;
2)echo "Directories are"
ls;;
3)echo "Enter dir name to create"
read d
mkdir $d
echo $d" Dir is created";;
4)echo "Enter filename from copy"
read f1
echo "Enter filename2 to be copied"
read f2
cp $f1 $f2
echo $f2" is copied from "$f1;;
5)echo "Enter file name to rename"
read f1
echo "Enter new name of file"
read f2
mv $f1 $f2
echo $f1" is renamed as "$f2;;
6)echo "Enter any filename to be delete"
read f1
rm $f1
echo $f1" is deleted";;
7)echo "Enter any file to be editing "
read f1
vi $f1;;
8) echo "Enter the file name you want to open"
```

```

read f1
cat $f1;;
9)echo "Have a nice time"
exit;;
*)echo "Invalid choice entered";;
esac
echo "are you continue (1 for yes / 0 for No)"
read temp
if [ $temp -eq 0 ]
then
ch=9
fi
done

```

OUTPUT:

```

1.Display current dir
2.Listing the dir
3.Make a dir
4.Copy a file
5.Rename file
6.Delete file
7.Edit file
8.open or display file
9.Exit
Enter your choice
1
Current Dir is :
/home/student/Documents
are you continue (1 for yes / 0 for No)
1
1.Display current dir
2.Listing the dir
3.Make a dir
4.Copy a file
5.Rename file
6.Delete file
7.Edit file
8.open or display file
9.Exit
Enter your choice
2
Directories are
pgm10.sh pgm12.sh pgm14.sh pgm2.sh pgm4.sh pgm6.sh pgm8.sh S2_MCA SMPS.odt
pgm11.sh pgm13.sh pgm1.sh pgm3.sh pgm5.sh pgm7.sh pgm9.sh Sample
are you continue (1 for yes / 0 for No)
1

1.Display current dir
2.Listing the dir

```

3.Make a dir
4.Copy a file
5.Rename file
6.Delete file
7.Edit file
8.open or display file
9.Exit
Enter your choice
5
Enter file name to rename
Sample
Enter new name of file
Sample_1
Sample is renamed as Sample_1
are you continue (1 for yes / 0 for No)
0

PROGRAM NO:20

DATE:04-07-2022

AIM: Write a shell script to accept empno,empname,basic. Find DA,HRA,TA,PF using following rules. Display empno, empname, basic, DA,HRA,PF,TA,GROSS SAL and NETSAL.

Also store all details in a file called emp.dat Rules:

HRA is 18% of basic if basic > 5000 otherwise 550

DA is 35% of basic

PF is 13% of basic

IT is 14% of basic

TA is 10% of basic

SOURCE CODE:

```
echo "Enter the EmployeeID (empno)"
```

```
read empno
```

```
echo "Enter the Name of Employee"
```

```
read empname
```

```
echo "enter the basic salary:"
```

```
read bsal
```

```
bsalp=`expr $bsal / 100`
```

```
if [ $bsal -gt 5000 ]
```

```
then
```

```
hra=`expr $bsalp \* 18`
```

```
else
```

```
hra=550
```

```
fi
```

```
da=`expr $bsalp \* 35`
```

```
pf=`expr $bsalp \* 13`
```

```
it=`expr $bsalp \* 14`
```



```

ta=`expr $bsalp \* 10`
gross=`expr $bsal + $hra + $da + $ta`
netsal=`expr $gross - $pf - $it`
echo "Empno : $empno"|tee cat >> emp.dat
echo "Empname : $empname"|tee cat >> emp.dat
echo "Basic : $bsal"|tee cat >> emp.dat
echo "HRA(House Rent Allowance): $hra"|tee cat >> emp.dat
echo "PF (Provident fund):$pf"|tee cat >> emp.dat
echo "TA (Travelling Allowance): $ta"|tee cat >> emp.dat
echo "IT (Income Tax) : $it"|tee cat >> emp.dat
echo "Gross salary : $gross"|tee cat >> emp.dat
echo "netsalary : $netsal"|tee cat >>emp.dat

```

OUTPUT:

Enter the EmployeeID (empno)

1527518

Enter the Name of Employee

Ram

enter the basic salary:

30000

Empno : 1527518

Empname : Ram

Basic : 30000

HRA(House Rent Allowance): 5400

PF (Provident fund):3900

TA (Travelling Allowance): 3000

IT (Income Tax) : 4200

Gross salary : 48900netsalary : 40800

PROGRAM NO:21

DATE:04-07-2022

AIM: Write a shell script to generate the mathematical tables.

SOURCE CODE:

```
echo "Enter a number:"

read n

echo "enter a table range"

read r

echo "Multiplication table of $n is:"

for (( i=1; i<=$r; i++ ))
do
result=$(( $n * $i ))
echo "$n "*" $i = $result
done
```

OUTPUT:

Enter a number:

5

enter a table range

10

Multiplication table of 5 is:

5 * 1 = 5

5 * 2 = 10

5 * 3 = 15

5 * 4 = 20

5 * 5 = 25

5 * 6 = 30

5 * 7 = 35

5 * 8 = 40

$$5 * 9 = 45$$

$$5 * 10 = 50$$

PROGRAM NO:22

DATE:08-07-2022

AIM: Installation and configuration of LAMP stack. Deploy an open source application such as phpmyadmin and Wordpress.

LAMP is an open-source Web development platform that uses Linux as the operating system, Apache as the Web server, MySQL as the relational database management system and PHP/Perl/Python as the object-oriented scripting language.

Update your system

sudo apt update

Install Apache using apt:

sudo apt install apache2

Confirm that Apache is now running with the following command:

sudo systemctl status apache2

if it is not working

sudo systemctl start apache2

Once installed, test by accessing your server's IP in your browser:

<http://youripaddress>

Install mariadb:

(MariaDB is a fork of MySQL from some of the original MySQL team and is a drop-in replacement.)

sudo apt install mariadb-server mariadb-client

Check mariadb Installation

sudo systemctl status mysql

(if it is not working sudo systemctl start mysql)

Secure your newly installed MariaDB service:

sudo mysql_secure_installation

(This will set password for mariadb, and strengthen the security by asking some questions like disallow root login remotely? Remove test database? Etc)

Install PHP and commonly used modules

```
sudo apt install php libapache2-mod-php php-opcache php-cli php-gd php-curl  
php-mysql
```

Restart apache2

```
sudo systemctl restart apache2
```

Now you can check php installation

```
sudo echo "<?php phpinfo(); ?>" | sudo tee -a /var/www/html/phpinfo.php
```

Open a browser

<http://127.0.0.1/phpinfo.php>

Install phpmyadmin

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

(It ask for webserver select apache2, select db-configuration and set password)

Restart apache2

```
sudo systemctl restart apache2
```

Check phpmyadmin

Open a browser

<http://localhost/phpmyadmin>

If you get a 404 error upon visiting <http://localhost/phpmyadmin>: You will need to configure apache2.conf to work with Phpmyadmin.

```
$ gksudo gedit /etc/apache2/apache2.conf
```

Include the following line at the bottom of the file, save and quit.

```
$ Include /etc/phpmyadmin/apache.conf
```

username : root

password : yourpassword

If any problem for login run the following command

```
sudo mysql
```

```
ALTER USER root@localhost IDENTIFIED BY "yourpassword";
```

Install word press on ubuntu Install word press on ubuntu

WordPress is a free, open-source website creation platform. On a more technical level, WordPress is a content management system (CMS) written in PHP that uses a MySQL database.

Creating html directory

```
cd /var/www/html
```

Download wordpress zip file from wordpress.org (latest.zip)

```
sudo wget https://wordpress.org/latest.zip
```

unzip the zip file

```
sudo unzip latest.zip
```

remove the zip file

```
sudo rm -r latest.zip
```

move the wordpress file to WP directory

```
sudo mv wordpress/ wp
```

```
mysql -u root -p (For entering mysql in terminal)
```

Give wp directory ownership to a user

```
sudo chown -R shameer /var/www/html/wp
```

```
sudo chown -R www-data:www-data /var/www/html/wp
```

open the web browser

<http://localhost/wp/wp-admin/> (word press login user name: admin, password: admin)