Experiment No:1

Date:17-03-2022

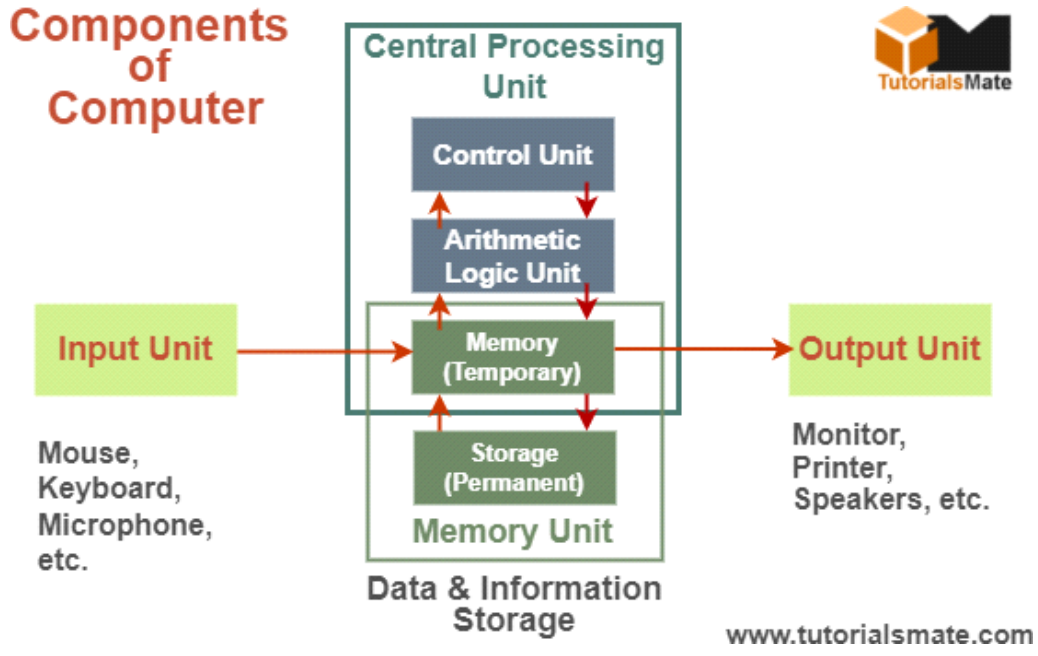
Explain about Components of a computer.

Basic 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.



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 arithemetic 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.

Arithemetic 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.

Experiment No:2

Date:04-04-2022

Physical identification of major components of a computer system such as Mother board,RAM modules,Daughter cards/Expansion cards,Bus slots,Internal storage device,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
- Vedio 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

Experiment No:3

Date:13-04-2022

Short note about SMPS(Switched-Mode Power Supply)

Switched-Mode Power Supply

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 indicators 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.

Experiment No:4

Date:27-04-2022

Explain linux file hierarchy structure

Linux file hierarchy structure

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.

Experiment No:5

Date:17-05-2022

Ubuntu installation using 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.

Experiment No:6

Date:18-05-2022

Basic Linux commands

1. Date command

date command prints the current date time

student@student-Pegatron:~$ date

Wed May 18 17:42:25 IST 2022

2. cal command

cal command is used to display a calendar in your shell, by default it will display the current month.

student@student-Pegatron:~$ cal

May 2022

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

3. whoami command

whoami command will tell you which user account you are using in this system.

student@student-Pegatron:~$ whoami

student

4. id command

id prints real user id, and various other details related to the account.

student@student-Pegatron:~$ id

5. pwd command

pwd command, short for print working directory, will help you to find out the absolute path of the current directory.

student@student-Pegatron:~$ pwd

/home/student

6.cd command

short for change directory. This command will help you to change your current directory.

student@student-Pegatron:~$ cd Desktop

student@student-Pegatron:~/Desktop$ cd ..

student@student-Pegatron:~$

7. ls command

ls command to list the files and directories inside any given directory.

student@student-Pegatron:~$ ls

cat emp.dat first.sh sallary.sh Templates

8. mkdir command

create new directories using mkdir command.

student@student-Pegatron:~$ mkdir myFolder

9. rm command

rm command is used to remove a file, or directory. The -r option is being used to remove in a recursive way. With -f

student@student-Pegatron:~$ rm first.sh

10.Copying a file using cp command

cp command to copy a file in the Linux shell.

$ cp hello.txt hello2.txt

11. Renaming or moving a file

The mv command is used to rename or move a file or directory.

student@student-OptiPlex-3050:~$ mv fact.c factNew.c

12. tree command

tree command prints the directory structure in a nice visual tree design way.

13. Using > to redirect output to a file

to redirect the output of one command to a file, if the file exists this will remove the old content and only keep the input.

14. man pages

man shows the system's manual pages. This is the command we use to view the help document (manual page) for any command.

The general syntax is man section command

student@student-OptiPlex-3050:~$ man ls

15. Open File Using cat Command

cat<file name>

student@student-OptiPlex-3050:~$ cat prime.cpp

16. tar command

Syntax:

tar [options] [archive-file] [file or directory to be archived]

Creating an uncompressed tar Archive using option -cvf

student@student-OptiPlex-3050:~$ tar cvf myFile.tar *.c

17. Grep Command

The grep command is a filter that is used to search for lines matching a specified pattern and print the matching lines to standard output.

18. chmod - To change access permissions, change mode.

19. Finding the IP address

student@student-OptiPlex-3050:~$ ip addr show

20. ping command

ping is simple way to find out if you are connected to the Internet or not.

student@student-OptiPlex-3050:~$ ping google.com

Experiment No:7

Date:18-05-2022

Write a program to check whether the given number is odd or even in shell script.

Source code:

```
#!/bin/bash

Clear

echo "EVEN OR ODD IN SHELL SCRIPT"

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

Output:

---- EVEN OR ODD IN SHELL SCRIPT -----
Enter your number:71
RESULT: 71 is Odd

Experiment No:8

Date:18-05-2022

Write a shell script program 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:

Enter Num1
60
Enter Num2
53
Enter Num3
23
Greater Number is 60

Experiment No:9

Date:20-05-2022

Program to perform Arithmetic operations

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
20
13
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 20 and 13 is 33
student@student-Pegatron:~/Documents$ ./pgm5.sh
Enter Two Numbers
40
10
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 40 by 10 is 0
```

Experiment No:10

Date:20-05-2022

Write a program to check whether the 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 : 3
3 is a prime number.

Experiment No:11

Date:20-05-2022

Write a program to find the sum of digit of of a number.

Source code:

```
echo "Enter your 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 your number 256
Sum of Digit is 13

Experiment No:12

Date:24-05-2022

Write a program to find the factorial of a number.

Source code:

```
echo "Enter your 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 your number: 6
Factorial of a number is=720

Experiment No:13

Date:24-05-2022

Write a program to print even series in shell script upto n terms.

Source code:

```
clear
echo "------EVEN SERIES -----"
echo -n "Enter your 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: 10
0
2
4
6
8
10
```

Experiment No:14

Date:24-05-2022

Write a program to print odd series in shell script upto n terms.

Source code:

```
#!/bin/bash
echo "---ODD SERIES ----"
echo -n "Enter your number"
read num
checker=0
while test $checker -le $num
do
ii=`expr $checker % 2`
        if test $ii -ne 0
        then
        echo "$checker"
        fi
checker=`expr $checker + 1`
done
```

Output:

---ODD SERIES-----
Enter a number:17
1
3
5
7
9
11
13
15
17

Experiment No:15

Date:24-05-2022

Write a shell script 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
priya
Enter the Register number
1687459
Enter the Mark1
60
Enter the Mark2
47
Enter the Mark3
52
Enter the Mark4
54
Enter the Mark5
60
-----------------------
Student Mark List
-----------------------
Student Name    : priya
Register Number : 1527519
Mark1           : 60
Mark2           : 47
Mark3           : 52
Mark4           : 54
Mark5           : 60
Total          : 273
Average       : 54.6
Result          : Pass
Grade           : D
-------------------------------
```

Experiment No:16

Date:24-05-2022

Arithemetic 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
14
Enter the Second Number
7
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 14 and 7 is 21

Enter the First Number
10
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 10 and 4 is 40
```

Experiment No:17

Date:24-05-2022

Write a shell script program 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

Experiment No:18

Date:30-05-2022

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 filenm2 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 filenm 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

Experiment No:19

Date:30-05-2022

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 (Travalling 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)
06594872
Enter the Name of Employee
Anaghagesh
enter the basic salary:
30000

Empno : 1527518
Empname : Rames
Basic : 30000
HRA(House Rent Allowance): 5400
PF (Provident fund):3900
TA (Travalling Allowance): 3000
IT (Income Tax) : 4200
Gross salary : 48900
netsalary : 40800

Experiment No:20

Date:03-06-2022

Write a shell script to generate the mathematical tables.

Source code:

```
echo "Enter your number:"
read num
echo "enter a table range"
read ran
echo "Multiplication table of $n is:"
for (( i=1; i<=$ran; i++ ))
do
result=$[ $num * $i ]
echo $num "*" $i = $result
done
```

Output:

Enter your number:
10
enter a table range
10
Multiplication table of 5 is:
10 * 1 = 10
10 * 2 = 20
10 * 3 = 30
10 * 4 = 40
10 * 5 = 50
10 * 6 = 60
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100