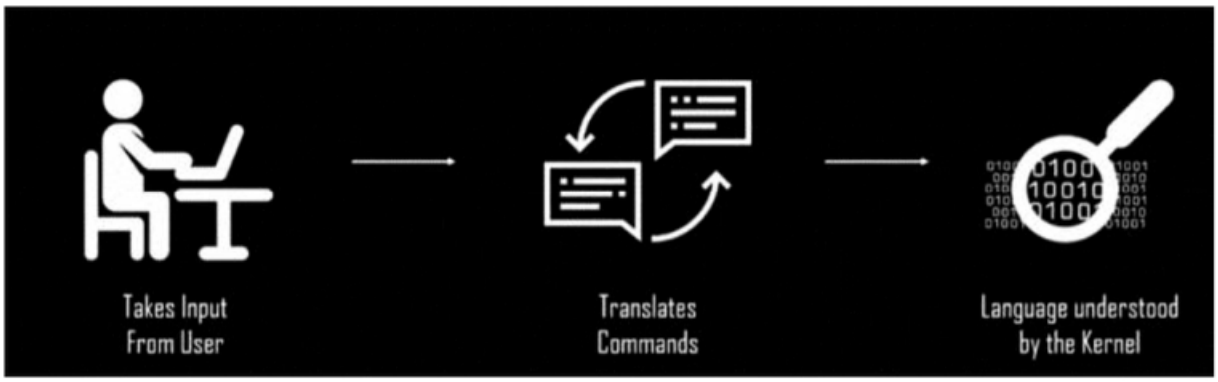
EXPERIMENT - 3 : Unix/Linux Shell Scripting

Introduction:

The shell is a command line interpreter. It translates the commands entered by the user and converts them into a language understood by the kernel. Kernel manages resource of Linux O/S. Kernel decides who will use this resource, for how long and when. It runs your programs (or set up to execute binary files).



Computer understand the language of 0's and 1's called binary language, In early days of computing, instruction are provided using binary language, which is difficult for all of us, to read and write. So, in O/s there is special program called Shell. Shell accepts your instruction or commands in English and translate it into computers native binary language.

A shell script is a computer program designed to be run by the Unix/Linux shell which could be one of the following:

- The Bourne Shell
- The C Shell
- The Korn Shell
- The GNU Bourne-Again Shell

Shell is an environment in which we can run our commands, programs, and shell scripts. There are different flavors of a shell, just as there are different flavors of operating systems. Each flavor of shell has its own.

1) Write Shell script to print your name

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano hello.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 hello.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % cat hello.sh
echo "what is your name?"
read person
echo "hello i am $person"


om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./hello.sh
what is your name?
om
hello i am om
```

2) Write Shell script to print your name

```
#!/bin/bash
echo "Enter a number: "
read num

if [ $(($num % 2)) -eq 0 ]; then
    echo "The number is even"
else
    echo "The number is odd"
fi
```

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./oddeven.sh
Enter a number:
4
The number is even
```

3) Write a script to print a table of a given number.

```
  UW PICO 5.09

#!/bin/bash
echo "Enter a number: "
read num
for i in $(seq 1 10)
do
    echo "$num x $i = $(($num * $i))"
done
```

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano printtable.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 printtable.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./printtable.sh
Enter a number:
2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

4) Write a shell script to check whether a given no. is prime or not.

```
  UW PICO 5.09

#!/bin/bash
echo "Enter a number: "
read num
if [ $num -le 1 ]; then
    echo "The number is'nt prime"
    exit
fi

for i in $(seq 2 $(($num - 1)))
do
    if [ $(($num % $i)) -eq 0 ]; then
        echo "The number is not prime"
        exit
    fi
done

echo "The number is prime"
```

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano prime.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 prime.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./prime.sh
Enter a number:
3
The number is prime
```

5) Write a shell script to find simple interest.

```
#!/bin/bash
echo " Enter the principle value: "
read p
echo " Enter the rate of interest:"
read r
echo " Enter the time period:"
read t
s=`expr $p \* $t \* $r / 100`
echo "The simple interest is "
echo $s
```

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano si.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 si.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./si.sh
 Enter the principle value:
1000
 Enter the rate of interest:
10
 Enter the time period:
1
The simple interest is
100
```

6) Write a shell script to find sum of n numbers.

```
#!/bin/bash
echo "Enter n"
read n
sum=0
for i in $(seq 1 $n)
do
    echo "Enter the number: "
    read num
    sum=$(($sum + $num))
done
echo "The sum is $sum"
```

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano nsum.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 nsum.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./nsum.sh
Enter n
3
Enter the number:
1
Enter the number:
2
Enter the number:
3
The sum is 6
```

7) Write a shell script to find largest number of 3 nos.

```
#!/bin/bash
echo "Enter 1st number"
read num1
echo "Enter 2nd number"
read num2
echo "Enter 3rd number"
read num3
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
echo $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
echo $num2
else
echo $num3
fi
```

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano lo3.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 lo3.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./lo3.sh
Enter 1st number
21
Enter 2nd number
43
Enter 3rd number
25
43
```

8) Write a menu driven shell script will point the following menu and execute the give task.

a) Display calendar of current month
b) Display today's date and time
c) Display username those are currently logged in the system
d) Display your name at given x, y position.
e) Display your terminal number.

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano q8.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 q8.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % cat q8.sh
echo "Menu"
echo "1. Display calender of current month "
echo "2. Display todays date and time"
echo "3. Display usernames those are currently logged in thesystem"
echo "4. Display your name at given x, y position"
echo "5. Display your terminal number"
echo "6. Exit"
echo "Enter your choice"
read c
case $c in
1) cal;;
2) date;;
3) who;;
4) clear
echo "Enter x, y position"
read x
read y
tput cup $x $y
whoami;;
5) tty;;
6) exit
;;esac
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./q8.sh
Menu
1. Display calender of current month
2. Display todays date and time
3. Display usernames those are currently logged in thesystem
4. Display your name at given x, y position
5. Display your terminal number
6. Exit
Enter your choice
2
Fri Feb  3 23:30:39 IST 2023
```

9) Write a shell script which will generate first n Fibonacci numbers like :1,1,2,3,5,13....\

```
#!/bin/bash
read -p "Enter the number of terms: " n
a=0
b=1
echo "The Fibonacci sequence upto $n terms is:"
for (( i=0; i<n; i++ ))
do
    echo -n "$a "
    fn=$((a+b))
    a=$b
    b=$fn
done
```

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano fib.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 fib.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./fib.sh
Enter the number of terms: 9
The Fibonacci sequence upto 9 terms is:
0 1 1 2 3 5 8 13 21 %
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab %
```

10) Write a shell script to find whether a given year is leap year or not

```
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % nano leap.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % chmod 777 leap.sh
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % cat leap.sh
#!/bin/bash
leap=$(date +"%Y")
echo taking year as $leap
if [ `expr $leap % 400` -eq 0 ]
then
echo leap year
elif [ `expr $leap % 100` -eq 0 ]
then
echo not a leap year
elif [ `expr $leap % 4` -eq 0 ]
then
echo leap year
else
echo not a leap year
fi

om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab % ./leap.sh
taking year as 2023
not a leap year
om-college@OM-M-PATEL-MACBOOK-M1-AIR os-lab %
```