184011

27.05.21

Practical - 8

Title : To execute Shell programs

Theory:

1] Shell Scripting

⇒ Usually shells are interactive that mean they accept commands as input from user and execute them. However some time we want to execute bunch of commads, so we have to type in all commands each time in terminal. As shell can also take input from a file, we can write these commands in a file and can execute them in shell to avoid repeatative work. These files are called Shell Scripts or Shell Programs. Shell scripts are similar to the batch file in MS-DOS. Each shell script is saved with .sh file extension eg myscript.sh.

A shell script comprises of following elements :-

1) Shell keywords - if, else, break etc.
2) Shell commands - cd, ls, echo, pwd
3) control flow - if..then.., else, case, loops, function

184011

Programs.

1) Print numbers till 5 using while loop

=> vi numbers.sh

```
valid=true
count=1
while [ $valid ]
do
echo $count
if [ $count -eq 5];
then
break
fi
((count++))
done.
```

Output: $ bash numbers.sh

```
1
2
3
4
5
```

CLASSMATE
Date :
Page :

2) Sum of two numbers

⇒
```
$ vi sum.sh
echo "Enter first Number"
read num1
echo "Enter Second Number"
read num2

sum = $((($num1 + num2))
echo "the sum of $num1 and $num2 is $sum"
```

Output :
```
bash sum.sh
Enter fist Number
5
Enter second Number
5
the sum of 5 and 5 is 10
```

3] To verify if admin password is correct

⇒
```
$ vi verify.sh
echo "Enter username"
read username
echo "Enter password"
read password

if [[ ($username == "admin" && password == "secret")
    ]] ; then
```

```
echo "valid user"
else
echo "Invalid user"
fi
```

Output : bash verify.sh
Enter username
admin
Enter password
secret
valid user.

4) To give the factorial of a number

⇒ vi factorial.sh

```
echo "Enter number to find factorial"
read num
factorial=1
counter=$num
while [[ $counter -gt 0 ]]; do
  factorial=$(($factorial * $counter))
  counter=$(( $counter -1))
done
echo "Factorial of $num is $factorial"
```

Output : bash factorial.sh
Enter a number : 4
Factorial of 4 is 24

5) To show the use of switch case

⇒ $ vi case.sh

```
echo "Enter your lucky number"
read n
case $n in
10)
echo " You got 1st prize " ;;
5)
echo " You got second prize ";;
9)
echo " You got 3rd prize ";;
*)
echo " Sorry, try for the nesct time ";;
esac
```

Output : $ bash case.sh
Enter your lucky number
10
You got 1st prize

Conclusion : After performing this practical I learnt to run shell commands on linux, cygwin.