Batch: A1	**Roll No.: 16010123012**
Experiment No. 02	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Shell Programming

AIM: To study the shell script and write the program using shell.

Expected Outcome of Experiment:

CO 1. To introduce basic concepts and functions of operating systems.

Books/ Journals/ Websites referred:

1. **Silberschatz A., Galvin P., Gagne G. "Operating Systems Principles", Willey Eight edition.**
2. **William Stallings "Operating Systems" Person, Seventh Edition Edition.**
3. **Sumitabha Das " UNIX Concepts & Applications", McGraw Hill Second Edition.**

Pre Lab/ Prior Concepts:

The shell provides you with an interface to the UNIX system. It gathers input from you and executes programs based on that input. When a program finishes executing, it displays that program's output.

Shell Scripts
The basic concept of a shell script is a list of commands, which are listed in the order of execution. A good shell script will have comments, preceded by a pound sign, #, describing the steps.

Steps to create a Shell Script:
create a file using any text editor say vi, gedit, nano etc
1.$ vi filename

2.Insert the script/ commands in file and save the file to execute the file we need to give execute permission to the file
3.$ chmod 775 filename
4.Now execute the above file using any of following methods:
$ sh filename
OR
$ ./filename

NOTE: Before adding anything to your script, you need to alert the system that a shell script is being started. This is done using the shebang construct. For example −
#!/bin/sh.

Description of the application to be implemented:

1. Write a shell Script that accepts two file names as command line arguments and compare two file contents and check whether contents are same or not. If they are same, then delete second file.

2. Write a shell script that accepts integer and find the factorial of number.

3. Write a shell script for adding users

4. Write a shell script for counting no of processes running on system

Modification-
Write a shell script for counting the no of logged in users.

```
  GNU nano 7.2
logged=$(who) count=$(echo "$logged" | wc -l | xargs)
echo "Number of logged in users: $count"
```

```
kjsce@kjsce-VirtualBox:~/Aaryan$ nano addusers.sh
kjsce@kjsce-VirtualBox:~/Aaryan$ chmod 775 addusers.sh
kjsce@kjsce-VirtualBox:~/Aaryan$ ./addusers.sh
Number of logged in users: 2
```

Implementation details: (Screen shots)
1.

```
  GNU nano 7.2
#!/bin/sh

echo "Enter file 1 name"
read f1
echo "Enter file 2 name"
read f2
if cmp -s $f1 $f2
then
        echo "SAME"
        rm -rf $f2
else
        echo "DIFFERENT"
fi
```

```
aaryan@Aaryan:~/Desktop/aaryan$ nano twofiles
aaryan@Aaryan:~/Desktop/aaryan$ ./twofiles.sh
bash: ./twofiles.sh: No such file or directory
aaryan@Aaryan:~/Desktop/aaryan$ ./twofiles.sh
bash: ./twofiles.sh: Permission denied
aaryan@Aaryan:~/Desktop/aaryan$ chmod 777 twofiles.sh
aaryan@Aaryan:~/Desktop/aaryan$ ./twofiles.sh
Enter file 1 name
Hello
Enter file 2 name
World
DIFFERENT
```

2.

```
  GNU nano 7.2
echo "Enter number"
read n
factorial=1
for((i=n;i>0;i--))
do
   factorial=$((factorial*i))
done
echo $factorial
```

```
kjsce@kjsce-VirtualBox:~/Aaryan$ ./exp2.sh
Enter number
5
120
kjsce@kjsce-VirtualBox:~/Aaryan$ nano exp2.sh
kjsce@kjsce-VirtualBox:~/Aaryan$ ./exp2.sh
Enter number
10
3628800
```

3.

```
  GNU nano 7.2
echo "Enter the name of user"
read name

if id "$name"&>/dev/null;then
echo "User $name  already exists"
else
        sudo useradd "$name"
        echo "User added"
fi
```

```
aaryan@Aaryan:~/Desktop/aaryan$ nano twofiles.sh
aaryan@Aaryan:~/Desktop/aaryan$ nano addusers.sh
aaryan@Aaryan:~/Desktop/aaryan$ chmod 777 addusers.sh
aaryan@Aaryan:~/Desktop/aaryan$ ./addusers.sh
Enter the name of user
Aarya
[sudo] password for aaryan:
heSorry, try again.
[sudo] password for aaryan:
Sorry, try again.
[sudo] password for aaryan:
sudo: 3 incorrect password attempts
User added
```

4.

```
  GNU nano 7.2
p=$(ps -A) count=$(echo "p" | wc -l | xargs)
echo "Number of processes: $count"
```

```
kjsce@kjsce-VirtualBox:~/Aaryan$ nano proc.sh
kjsce@kjsce-VirtualBox:~/Aaryan$ chmod 775 proc.sh
kjsce@kjsce-VirtualBox:~/Aaryan$ ./proc.sh
Number of processes: 1
```

Conclusion:

I have successfully completed the experiment and learned how to write shell scripts using the nano text editor. During the experiment, we learned how to add users, count the number of users and track the number of processes running on the system.

Post Lab Descriptive Questions

1) **What are the different types of commonly used shells on a linux system?**
   Different types of commonly used shells on a Linux system are
   1. Bourne bash shell($)
   2. Korn shell($)
   3. C shell(%)

2) **How do you find out the current shell that you are working on?**
   using echo
   echo $shell

3) **List the advantages and disadvantages of shell scripting.**
   Shell scripting is a powerful tool for automating repetitive tasks, offering platform independence on Unix/Linux systems, cost-effectiveness, and quick prototyping. It is easy to use for system administration and integrates well with other tools. However, it has limitations like poor error handling, slower performance compared to compiled languages, potential security vulnerabilities, and difficulty in maintaining and debugging complex scripts.

Date: 21 / 01 / 2025 **Signature of faculty in-charge**