

| **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.**

1. **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 logged in users.
5. Write a shell script for counting no of processes running on system

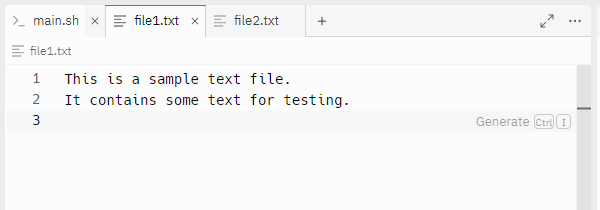
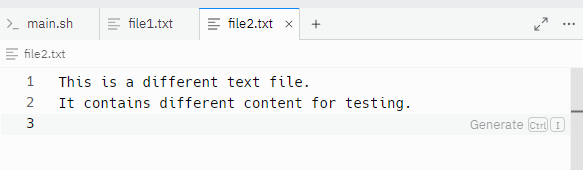
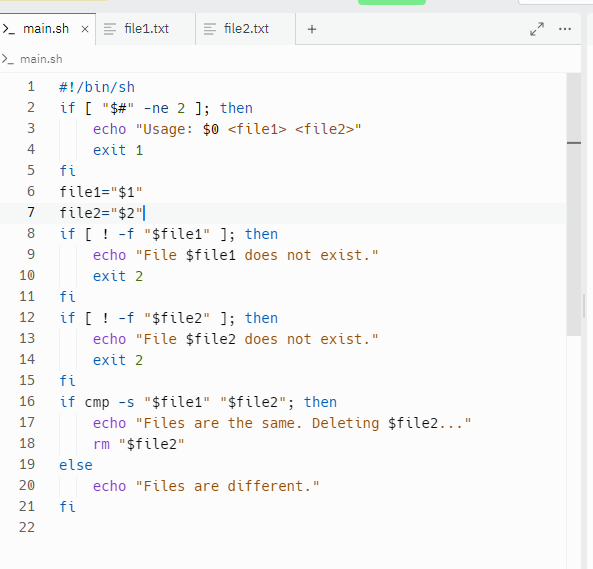
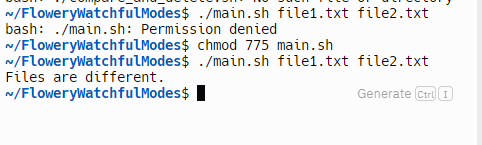
**Implementation details:** (printout of code / screen shot)

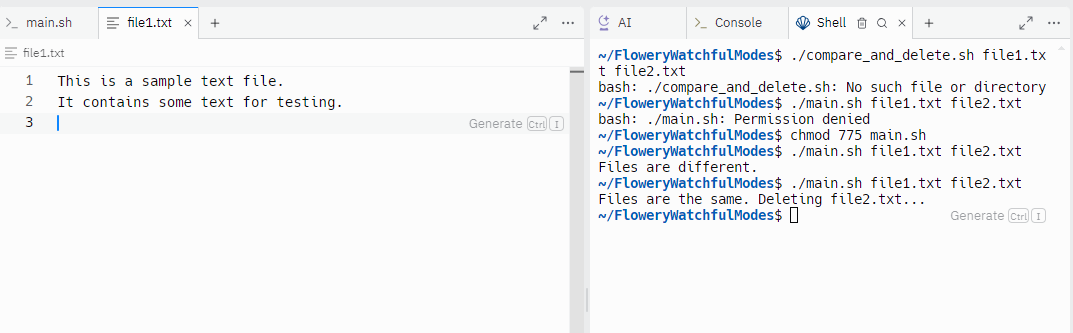
1. Write a shell Script that accepts two file names as command line arguments and

compare two file contents and check whether contents are same or not. If they are

same, then delete second file.

Files are different.

Files are same.

2. Write a shell script that accepts integer and find the factorial of number.  
  
#!/bin/sh

if [ "$#" -ne 1 ]; then

echo "Usage: $0 <integer>"

exit 1

fi

number="$1"

if ! [ "$number" -eq "$number" ] 2>/dev/null; then

echo "Error: The input must be a non-negative integer."

exit 2

fi

if [ "$number" -lt 0 ]; then

echo "Error: The factorial is not defined for negative numbers."

exit 3

fi

factorial=1

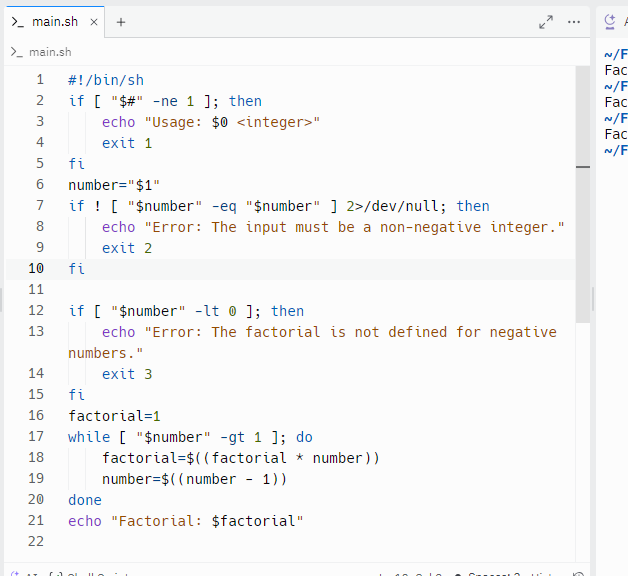
while [ "$number" -gt 1 ]; do

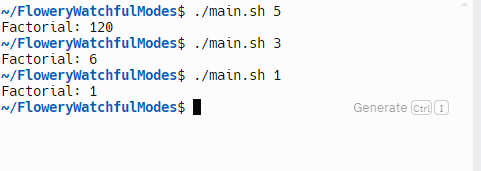
factorial=$((factorial \* number))

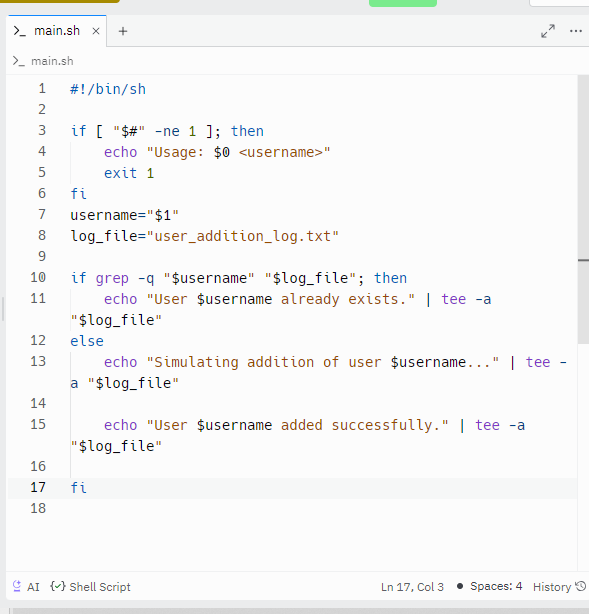
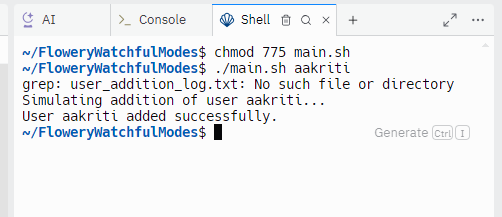
number=$((number - 1))

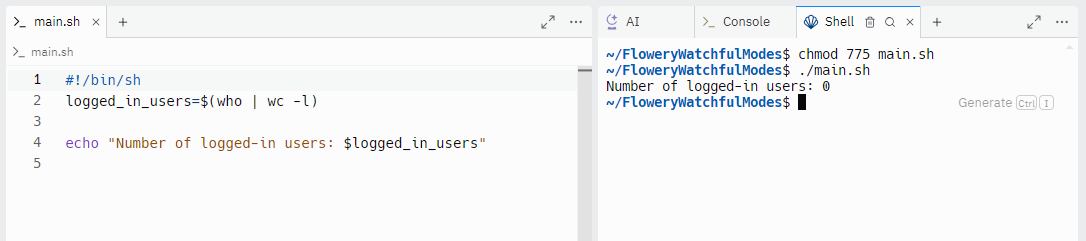
done

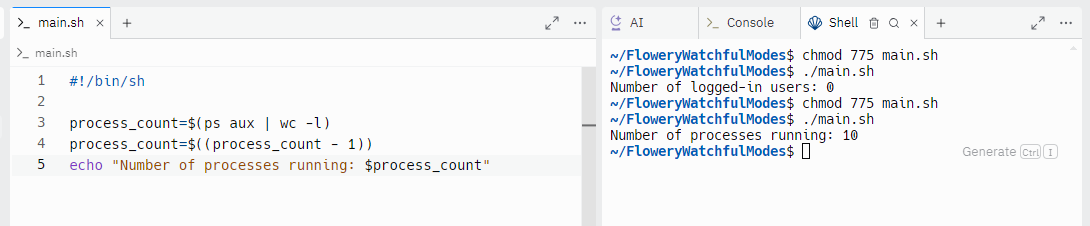
echo "Factorial: $factorial"





3. Write a shell script for adding users.  
  
  


4. Write a shell script for counting no of logged in users.  


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


**Conclusion :**

Learned and explored the shell script and implemented programs.

**Post Lab Descriptive Questions**

1. What are the different types of commonly used shells on a typical linux system?  
     
   Different types of hells used in linux system:  
   a) Bash (Bourne Again Shell)

* The default shell on many Linux distributions. It is an enhanced version of the original Bourne Shell (sh) with additional features.
* /bin/bash

b) Sh (Bourne Shell)

* The original shell written by Stephen Bourne. It is more basic compared to Bash but still widely used for scripts requiring maximum portability.
* /bin/sh

c) Zsh (Z Shell)

* Known for its powerful features, such as improved tab completion and enhanced globbing capabilities. It is highly customizable.
* /bin/zsh

d) Ksh (Korn Shell)

* Combines features of the Bourne Shell and C Shell. It includes features like job control and improved scripting capabilities.
* /bin/ksh

e) Tcsh (TENEX C Shell)

* An enhanced version of the C Shell (csh), with additional features such as command-line editing and filename completion.
* /bin/tcsh

1. How do you find out what’s your shell?  
     
   Using echo command:  
   echo $SHELL
2. List the advantages and disadvantages of shell scripting.

### Advantages of Shell Scripting

* Simplicity: Easy to write and understand for basic tasks.
* Efficiency: Automates repetitive tasks and system administration.
* Portability: Works across Unix-like systems with minimal changes.
* Integration: Easily integrates with other command-line tools.

### Disadvantages of Shell Scripting

* Limited Functionality: Not suited for complex programming tasks.
* Debugging: Can be difficult to debug and maintain.
* Security Risks: Potential for vulnerabilities if not carefully written.
* Performance: May be slower compared to compiled languages.

**Date: \_8/8/24\_ Signature of faculty in-charge**