

| **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.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

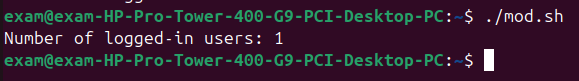
**Description of the application to be implemented**:

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 processes running on system

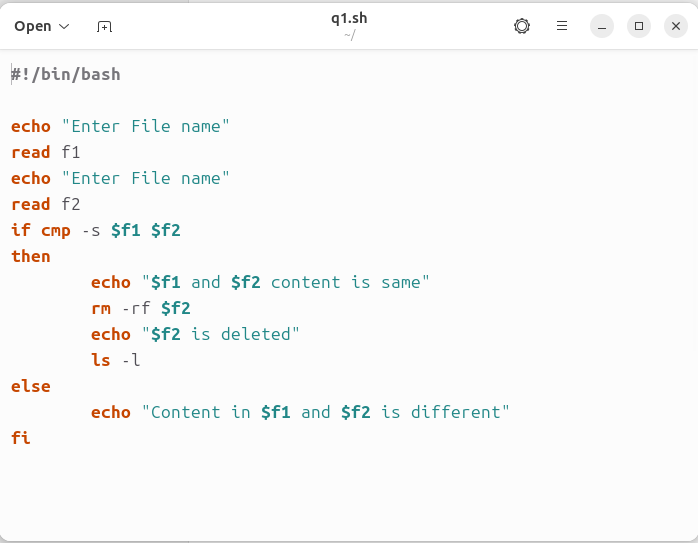
**Modification-**

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

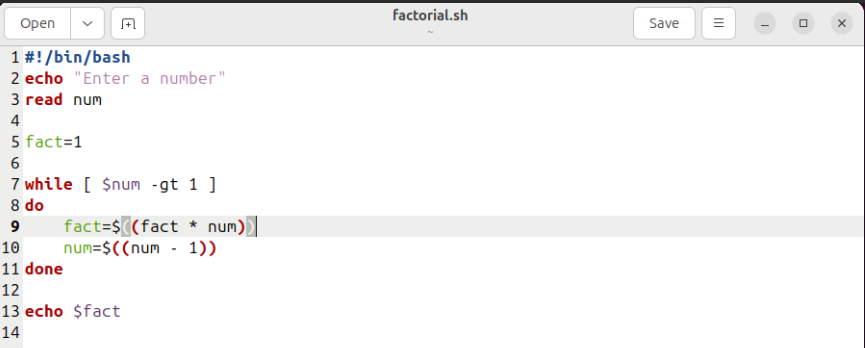




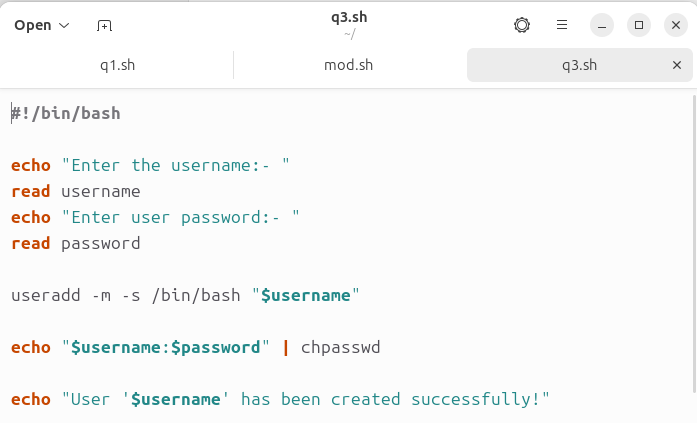
**1)**Write a shell Script that accepts two file names as command line arguments and compare two file contents and check whether the contents are the same or not. If they are the same, then delete second file.



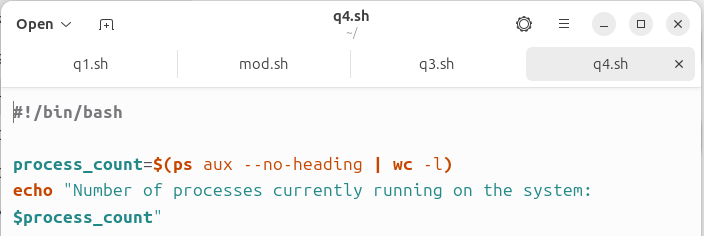
**2.)** Write a shell script that accepts integers and find the factorial of the number.



**3)** Write a shell script for adding users

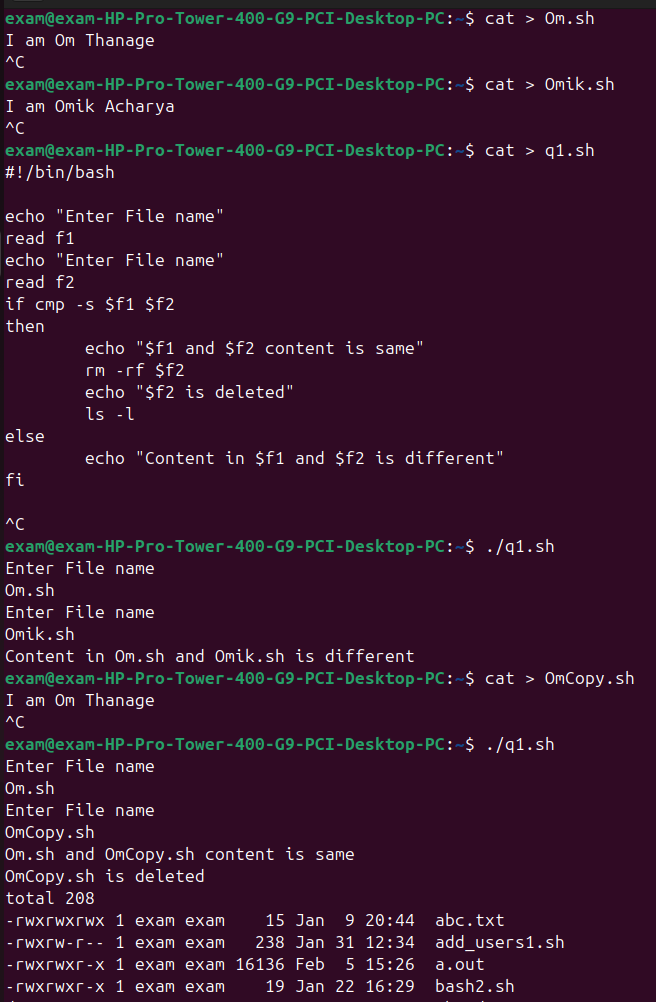
****

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

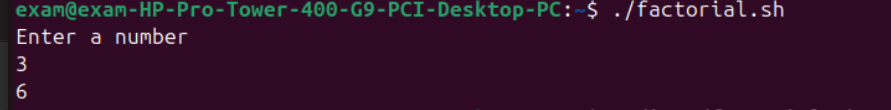


**Implementation details:** (Screen shots)

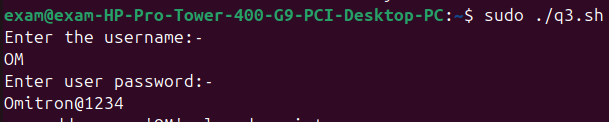
**1)**

****

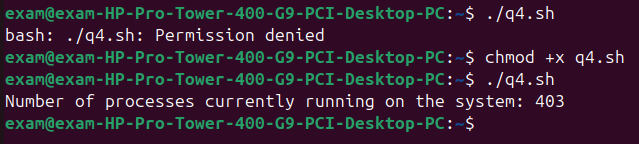
**2)**



**3)**



**4)Modified**



**Conclusion :**

In this experiment we learned to use the Bash script in linux to Do perform various task on the command line like to compare the content of 2 files and remove the second or to find the current active Task and Logged users using the command that we learned in the first experiment .

**Post Lab Descriptive Questions**

1. What are the different types of commonly used shells on a linux system?

Ans.

1. Bash (Bourne Again Shell) – The default and most widely used shell, offering advanced scripting and interactive features.
2. Sh (Bourne Shell) – A traditional Unix shell, mainly used for scripting and system administration.
3. Csh (C Shell) – A shell with C-like syntax, featuring job control and history substitution.

Ksh (Korn Shell) – A powerful shell that merges feature of sh and csh, improving scripting capabilities.

1. How do you find out the current shell that you are working on?

Ans.

We can find the current shell that you are working on by using the command “ps -p $$”

1. List the advantages and disadvantages of shell scripting.

Ans.

**Advantages of Shell Scripting**

1. **Automation** – Repetitive tasks can be automated, saving time and effort.
2. **Easy to Write & Execute** – Shell scripts are simple to write using basic commands and don’t require compilation.
3. **Flexibility** – Can be used for system administration, backups, monitoring, and task scheduling.
4. **Portability** – Shell scripts run across different Unix/Linux distributions with little to no modification.
5. **Integration** – Shell scripts can easily interact with system utilities and external programs.

**Disadvantages of Shell Scripting**

1. **Performance Issues** – Shell scripts are slower than compiled programs (e.g., C, C++).
2. **Limited Debugging** – Debugging shell scripts can be difficult due to minimal error messages.
3. **Security Risks** – Poorly written scripts may introduce vulnerabilities, such as unintended execution of commands.
4. **Complex Syntax for Large Scripts** – Managing large and complex scripts can be difficult compared to structured programming languages.
5. **Shell-Specific Differences** – Some commands behave differently across shells (e.g., Bash vs. Zsh), leading to compatibility issues.

**Date: 31/01/2025 Signature of faculty in-charge**