

| **TITLE:** Shell Programming and system calls |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**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.

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

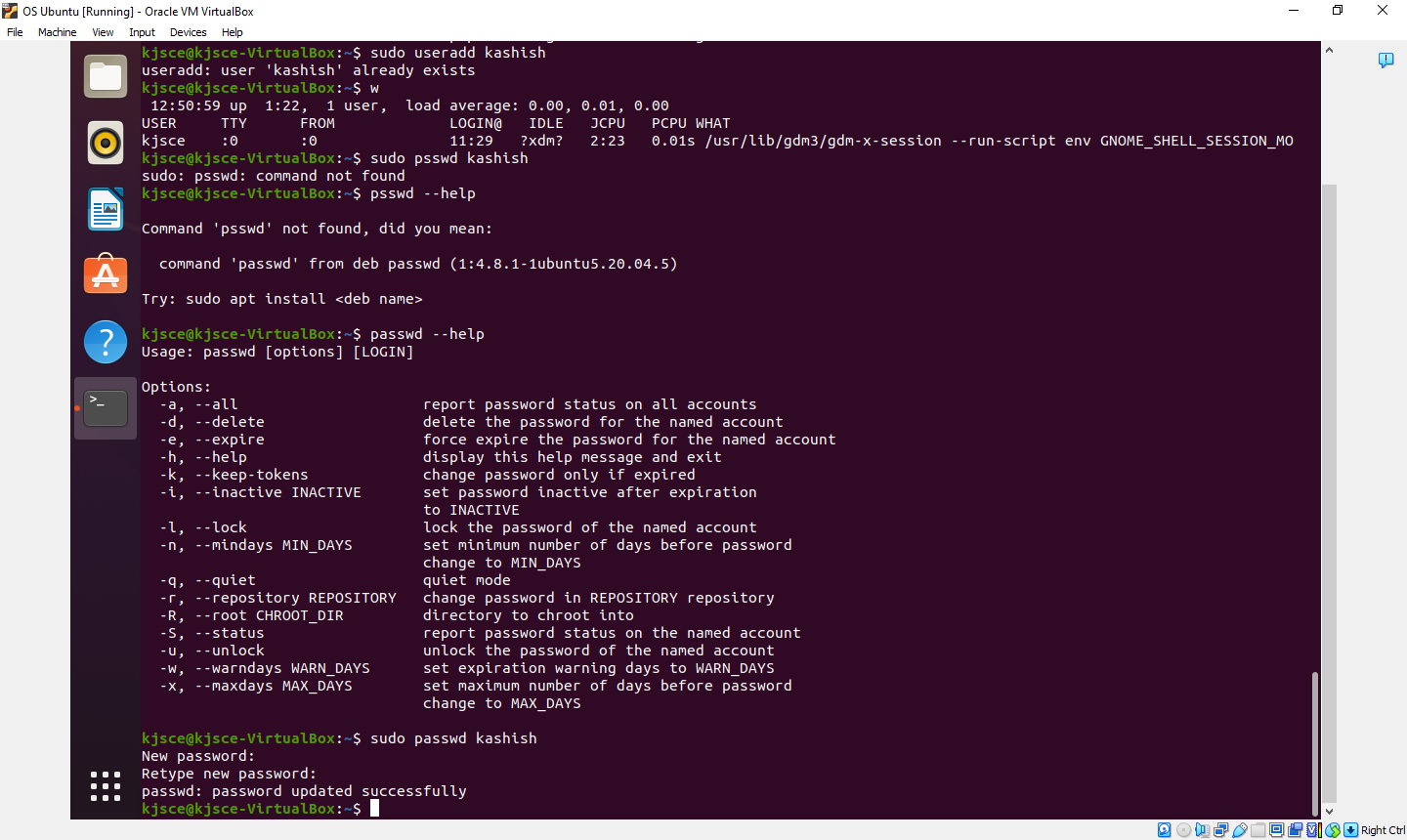
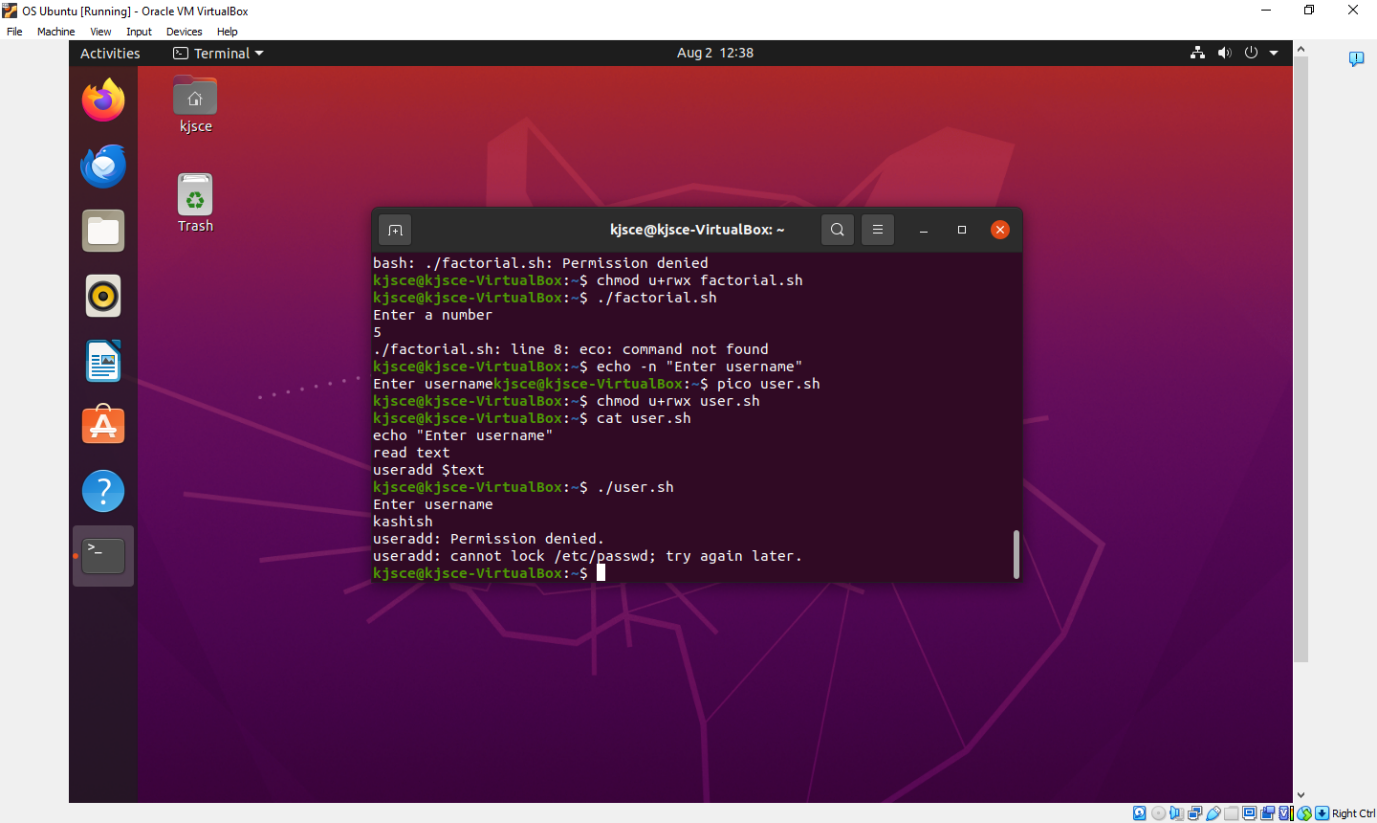
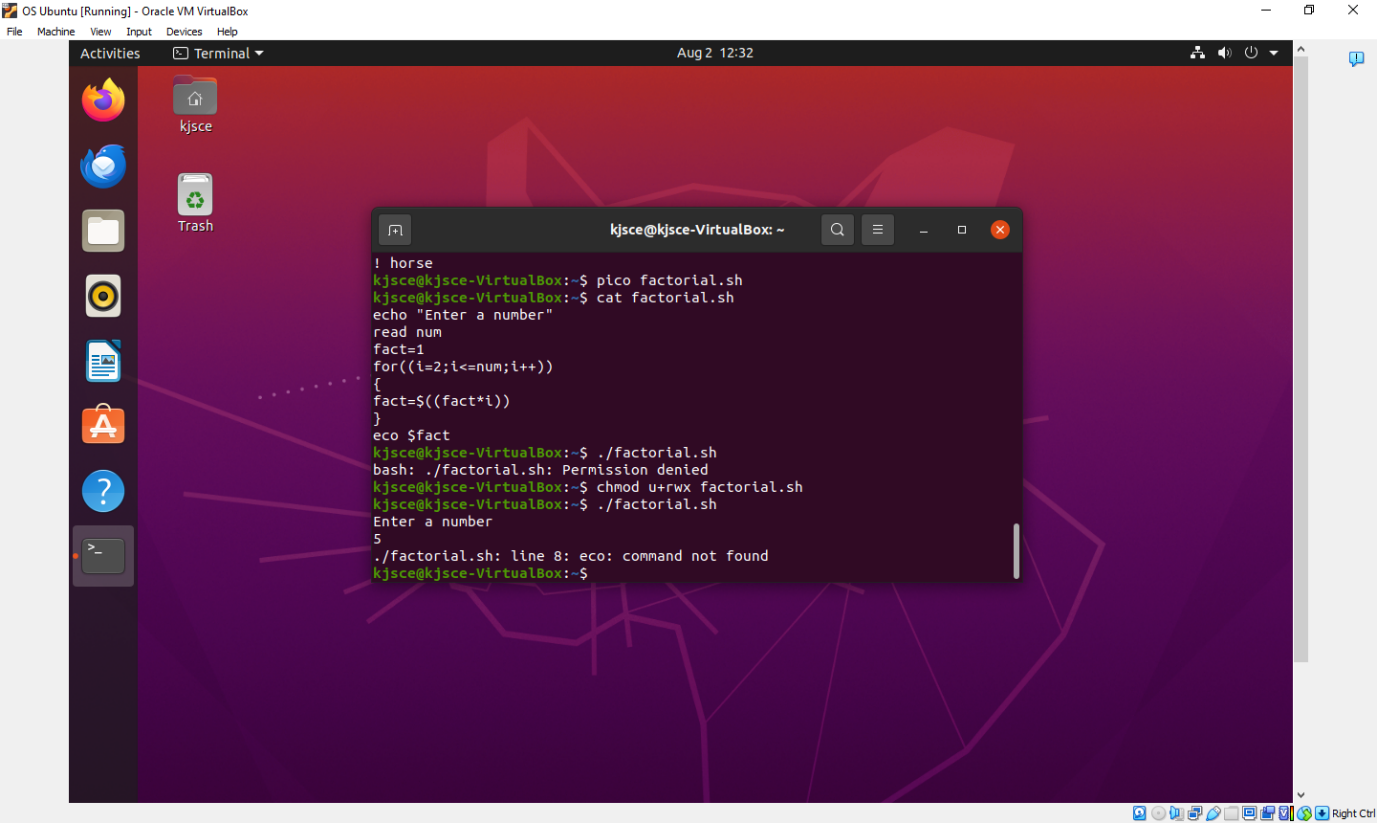
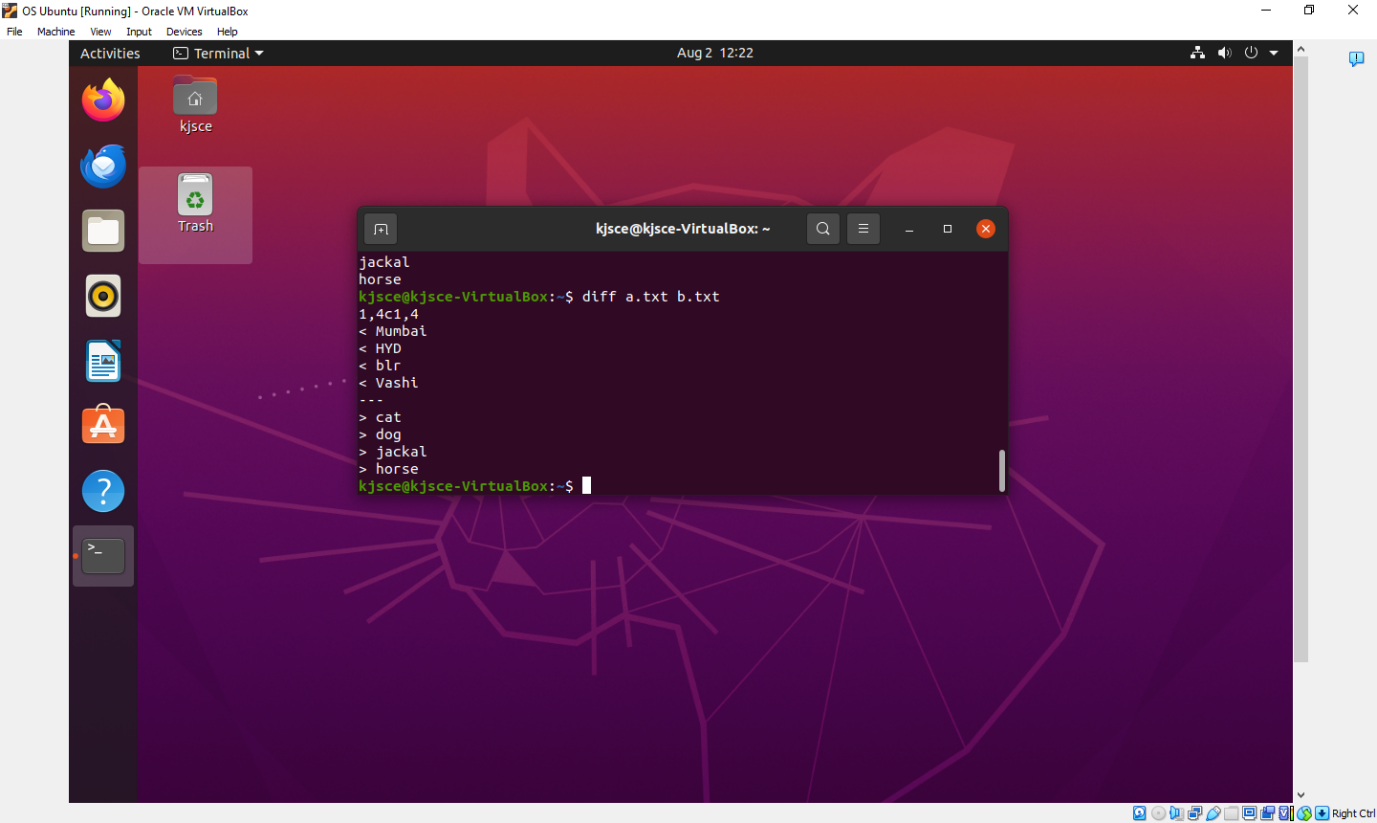
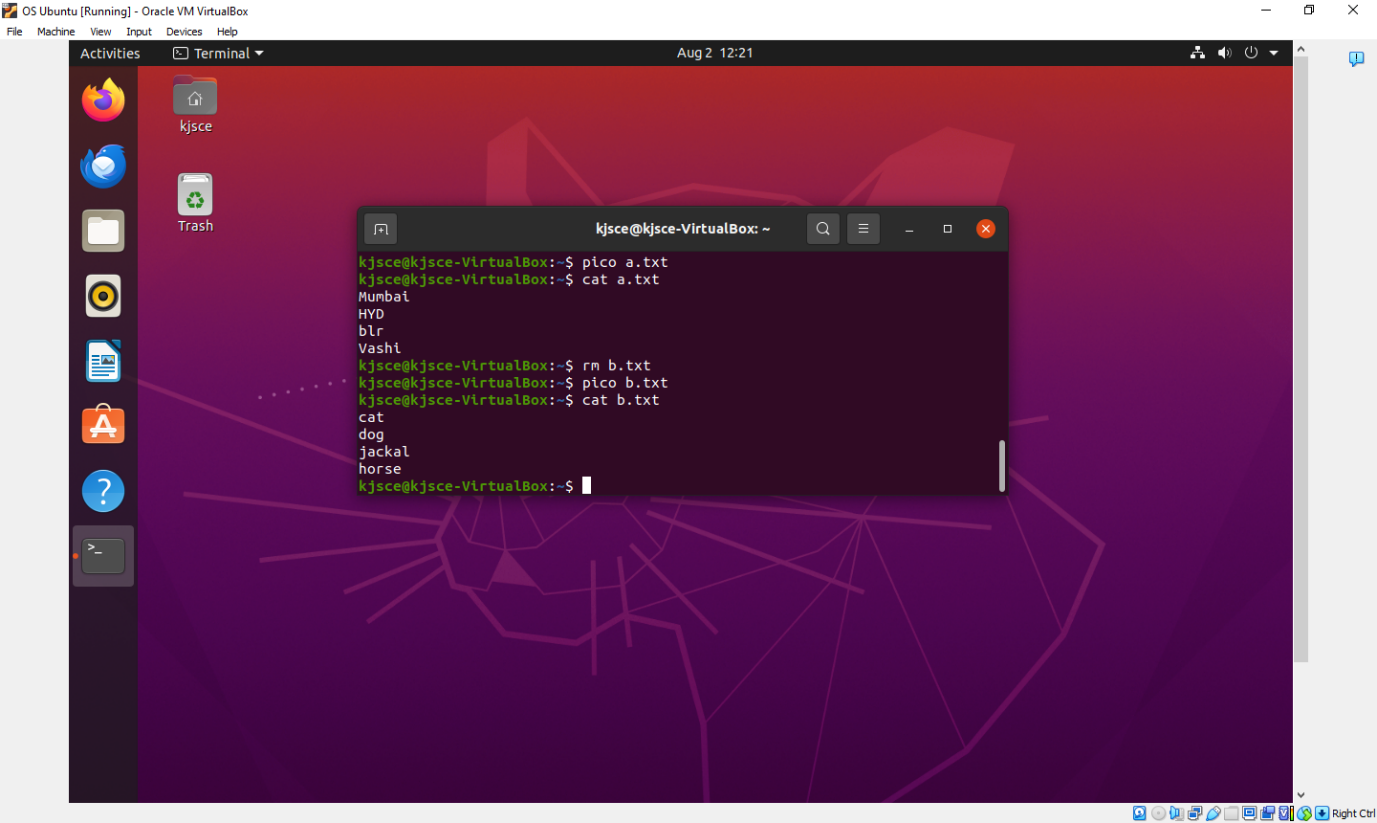
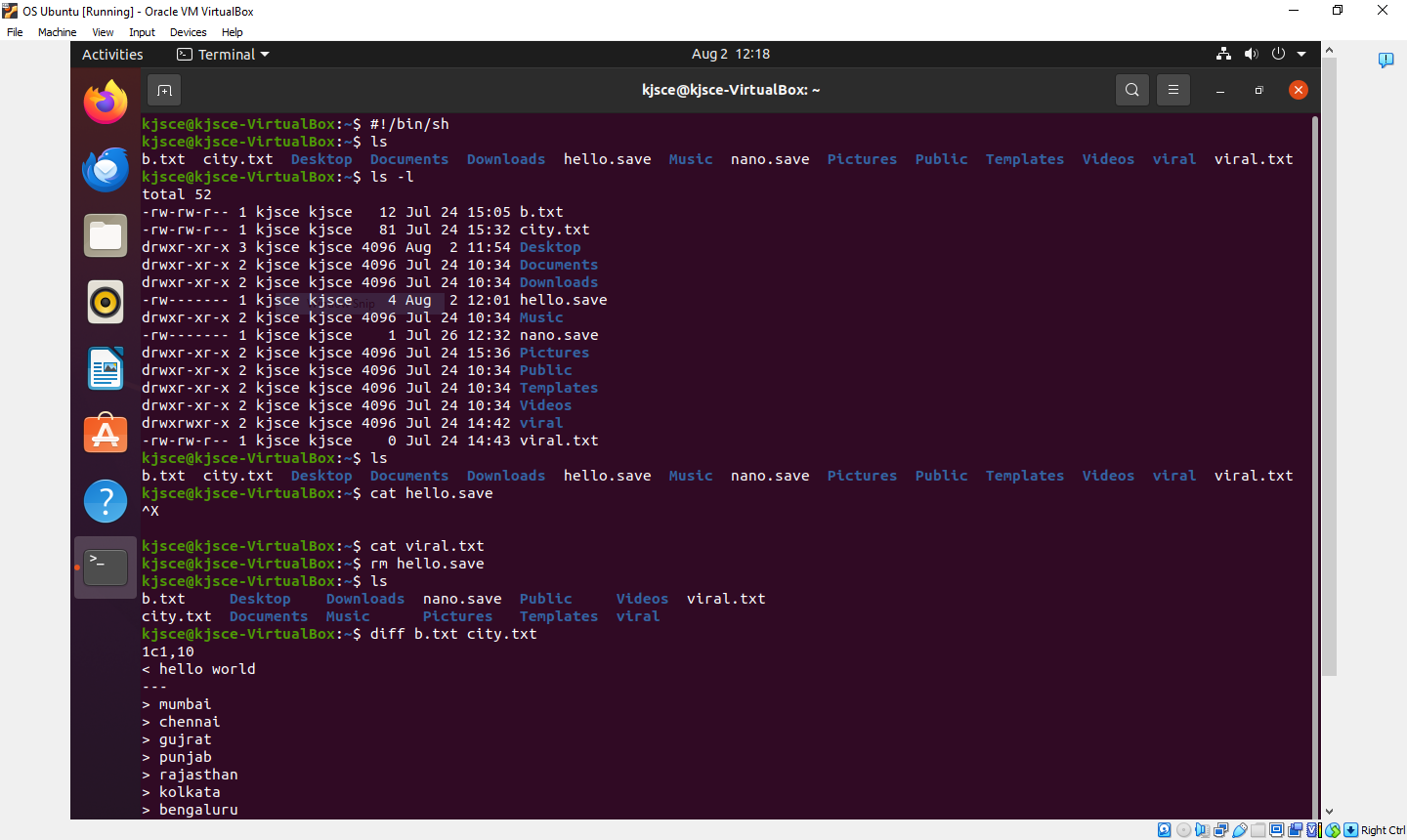
**Description of the application to be implemented**:

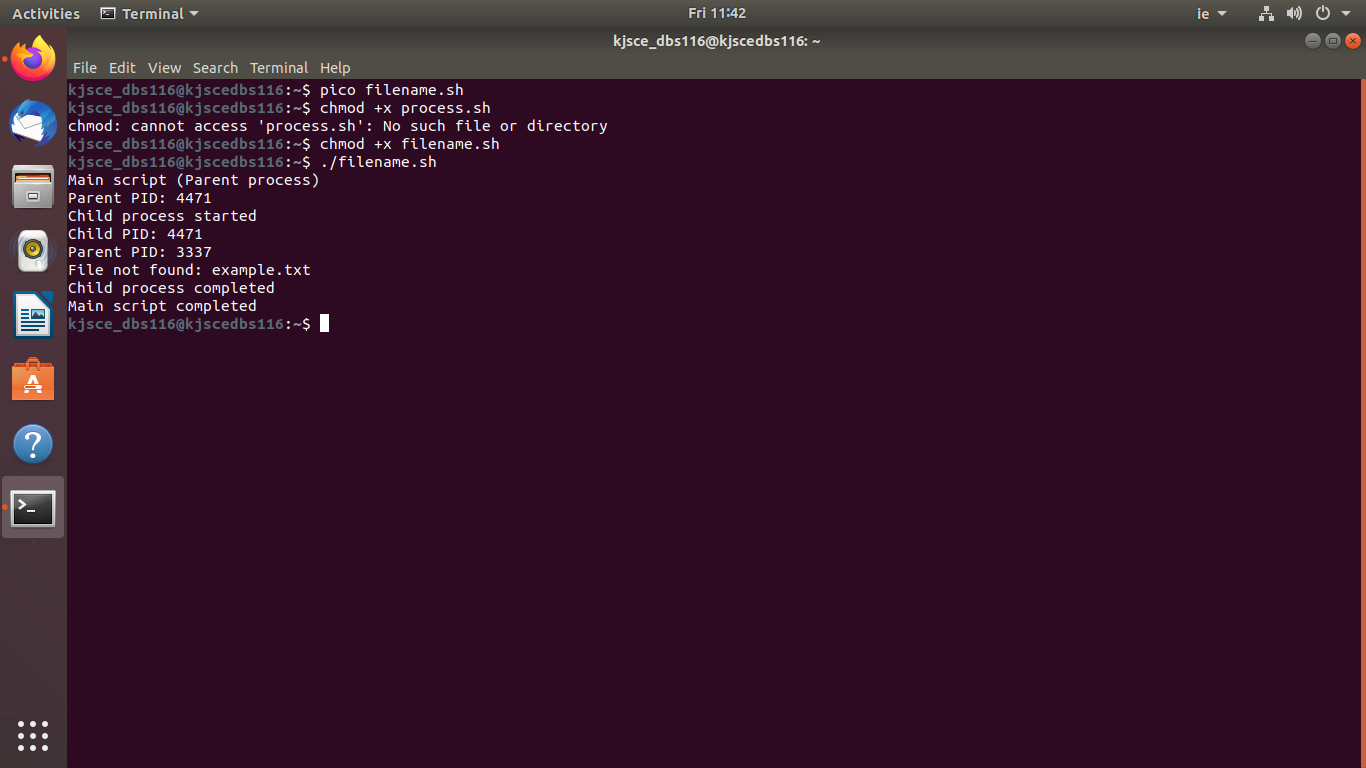
1. Write a shell Script that accepts two file names as command line arguments and compare two file contents and check whether contents are same or not. If they are same, then delete second file.
2. Write a shell script that accepts integer and find the factorial of number.
3. Write a shell script for 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

**Program for System Call:**

1. Write a Program for creating process using System call (E.g fork()) Create a child process. Display the details about that process using getpid and getppid functions. In a child process, Open the file using file system calls and read the contents and display.

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



****

**Conclusion :**

We learnt about shell scripting by implementing the give questions.

**Post Lab Descriptive Questions**

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

**Ans:**

Bash (Bourne Again Shell):

Path: /bin/bash

Features: Most widely used shell on Linux, backward-compatible with the Bourne shell (sh), includes features like command history, scripting capabilities, and job control.

Zsh (Z Shell):

Path: /bin/zsh

Features: Known for its powerful scripting, customization options, and user-friendly features like auto-completion, spelling correction, and plugin support.

Ksh (Korn Shell):

Path: /bin/ksh

Features: A superset of the Bourne shell, with scripting features like associative arrays and string manipulation.

Tcsh (TENEX C Shell):

Path: /bin/tcsh

Features: An enhanced version of the C shell (csh), includes features like command history, auto-completion, and scripting enhancements.

Fish (Friendly Interactive Shell):

Path: /usr/bin/fish

Features: Designed to be user-friendly with features like syntax highlighting, auto-suggestions, and scripting with simple syntax.

1. **How do you find out what’s your shell?**

**Ans:**

Using echo $SHELL: This command prints the path of the default shell

Using echo $0: This command gives the current shell being used

Using ps -p $$: This command shows the current shell process

1. **List the advantages and disadvantages of shell scripting.**

**Ans:**

Advantages:

Automation:

Shell scripts are ideal for automating repetitive tasks, such as backups, file management, and batch processing.

Simple and Easy to Learn:

Shell scripting is relatively easy to learn, especially for those familiar with command-line operations.

Efficiency:

Scripts can perform complex tasks in a few lines of code, often faster than running multiple manual commands.

Portability:

Shell scripts are highly portable across different UNIX-like systems without modification.

Integration:

Shell scripts can easily integrate with other command-line utilities, making them versatile for various tasks.

Disadvantages:

Performance:

Shell scripts are interpreted rather than compiled, making them slower than compiled programs for CPU-intensive tasks.

Limited Debugging:

Shell scripts lack sophisticated debugging tools, making it challenging to troubleshoot complex scripts.

Error Handling:

Shell scripting lacks advanced error-handling mechanisms, making scripts prone to failure in unpredictable scenarios.

Security Risks:

Scripts may contain commands that, if not carefully written, can cause harm (e.g., overwriting files, unintended deletions). Also, shell scripts can be vulnerable to injection attacks.

Not Suitable for Complex Applications:

Shell scripting is not ideal for developing large-scale or highly complex applications; it’s better suited for simple tasks.

**Date: 09/08/2024 Signature of faculty in-charge**