S.B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT & RESEARCH, NAGPUR

Practical 03

Aim: Automate student marksheet generation, system information display, Fibonacci and prime number generation, and file management operations using shell scripts to enhance computational efficiency and user interaction.

Name: Prachita Ashok Mahure

USN: CM24031

Semester / Year: IV/2nd

Academic Session: 2025-2026

Date of Performance:

Date of Submission:

❖ **Aim:** Automate student marksheet generation, system information display, Fibonacci and prime number generation, and file management operations using shell scripts to enhance computational efficiency and user interaction.

❖ **Tasks to be done in this Practical.**

a) Write a shell script to generate mark- sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.
b) Write a menu driven shell script which will print the following menu and execute the given task.
   - Display calendar of current month.
   - Display today's date and time.
   - Display usernames those are currently logged in the system.
   - Display your terminal number
c) Write a shell script which will generate first n Fibonacci numbers like: 1, 1, 2, 3, 5, 13
d) Write a shell script which will accept a number b and display first n prime numbers as output.
e) Write menu driven program for file handling activity
   - Creation of file.
   - Write content in the file.
   - Upend file content.
   - Delete file content

❖ **Objectives:**

1. Automate marksheet generation with total, percentage, and class classification.
2. Develop menu-driven scripts for system information and file operations.
3. Generate Fibonacci and prime numbers for user-defined inputs.

❖ **Requirements:**

✔ **Hardware Requirements:**

- Processor: Minimum 1 GHz
- RAM: 512 MB or higher
- Storage: 100 MB free space

✔ **Software Requirements:**

- Operating System: Linux/Unix-based
- Shell: Bash 4.0 or higher
- Text Editor: Nano, Vim, or any preferred editor

❖ Theory:

Shell scripting is a powerful way to automate repetitive tasks and manage system operations efficiently. It allows users to write programs using shell commands and scripting constructs. Shell scripts are interpreted line-by-line by a shell interpreter, making them ideal for administrative tasks, file management, and system automation. This practical encompasses a variety of real-world scenarios that demonstrate the utility of shell scripting for computing tasks and resource management.

1. Marksheet Generation

This script takes input marks for three subjects, calculates the total marks, percentage, and determines the class of the student based on predefined conditions. Conditional statements (if-else) are used to classify the performance into distinction, first class, second class, or fail. This exercise emphasizes the use of arithmetic operations and decision-making constructs.
Key concepts include:

- Reading user input using read
- Arithmetic operations with $((expression))
- Conditional statements for decision-making

2. Menu-Driven Script for System Information

Menu-driven scripts enhance user interaction by presenting a list of options for performing different tasks. In this practical, options are provided to display the calendar of the current month, the current date and time, logged-in users, and the terminal number. The script utilizes looping constructs (while) and case statements for structured flow control.

Commands used:

- cal for displaying the calendar
- date for showing current date and time
- who to list logged-in users
- tty to identify the terminal

3. Fibonacci Number Generation

Fibonacci numbers are a sequence where each term is the sum of the two preceding ones. The script uses iterative constructs (for loop) to generate n terms based on user input. This practical illustrates the use of loop control and variable swapping to generate series data efficiently.

4. Prime Number Display

This script accepts an integer n and outputs the first n prime numbers. A nested loop checks divisibility to determine if a number is prime. The practical demonstrates logic building for number-theoretic operations using loops and conditionals.

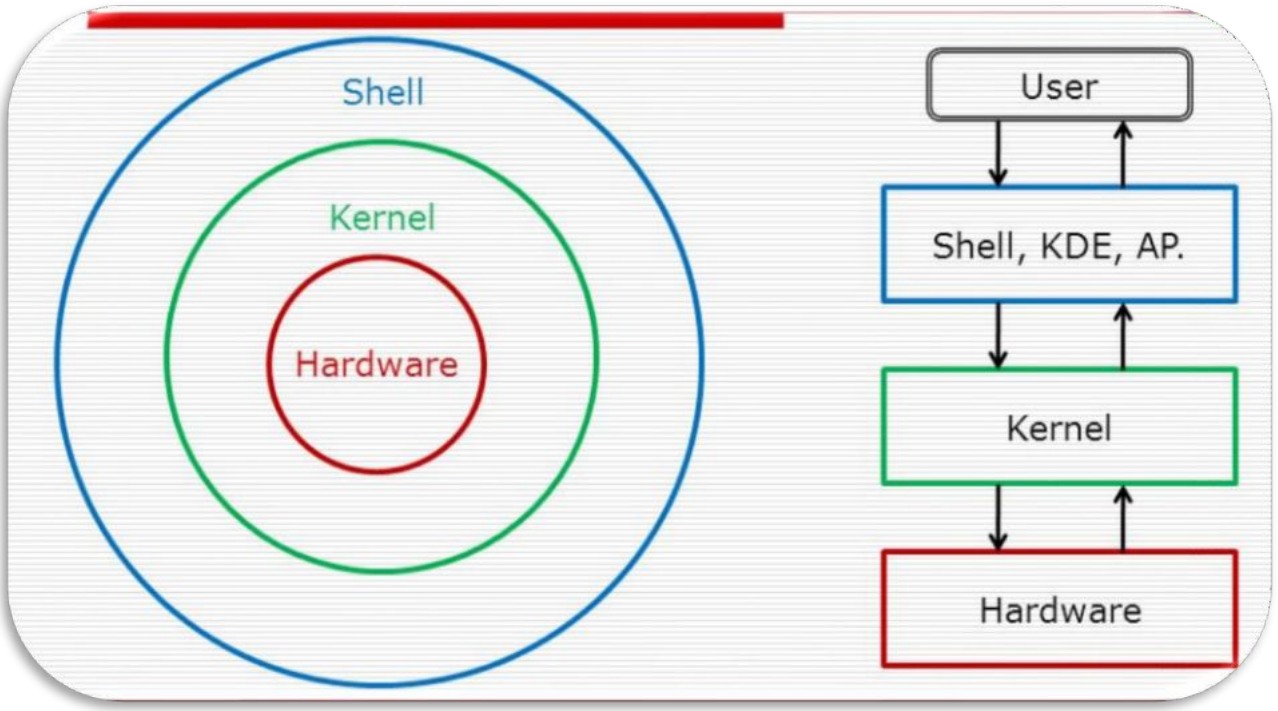
5. Menu-Driven File Management

The file handling script enables users to create, write, append, and delete file content. The case construct manages different file operations.
Commands include:

- touch to create files
- cat for writing and appending content
- rm for deleting files

This exercise emphasizes text manipulation, input handling, and file control mechanisms in Unix-like environments.

Diagrammatical View of Shell

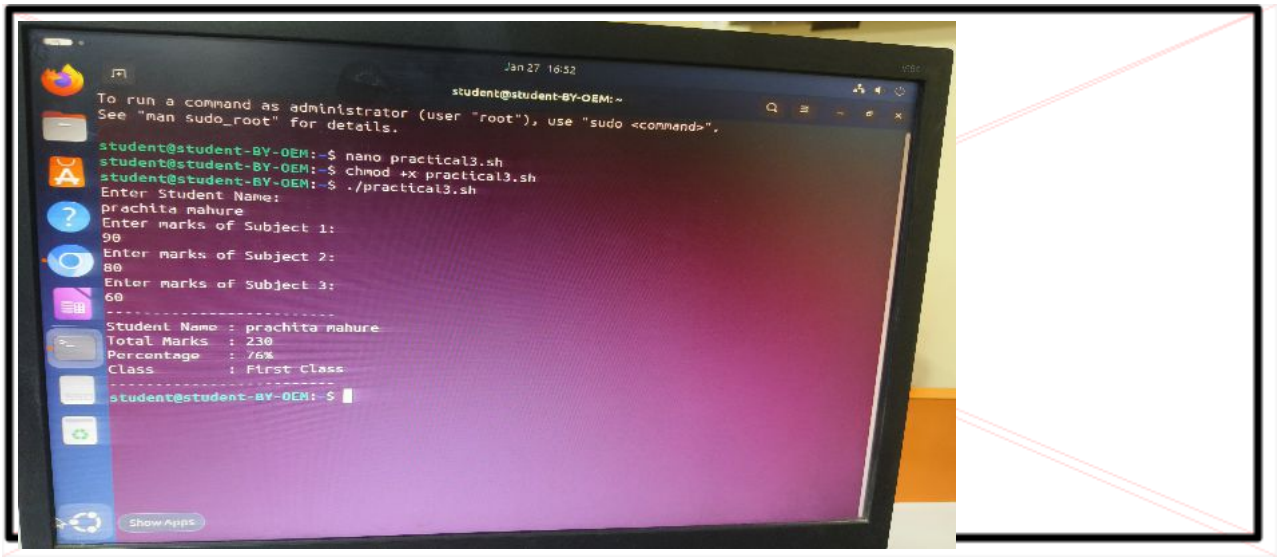


❖ **CODES**

1. **Write a shell script to generate mark- sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.**

```
#!/bin/bash

echo "Enter Student Name:"
read name

echo "Enter marks of Subject 1:"
read m1
echo "Enter marks of Subject 2:"
read m2
echo "Enter marks of Subject 3:"
read m3

total=$((m1 + m2 + m3))
percentage=$((total / 3))

if [ $percentage -ge 60 ]; then
    class="First Class"
elif [ $percentage -ge 50 ]; then
    class="Second Class"
elif [ $percentage -ge 40 ]; then
    class="Pass"
else
    class="Fail"
fi

echo "--------------------------"
echo "Student Name : $name"
echo "Total Marks  : $total"
echo "Percentage   : $percentage%"
echo "Class        : $class"
echo "--------------------------"
```

Output 1:



2. Write a menu driven shell script which will print the following menu and execute the given task.

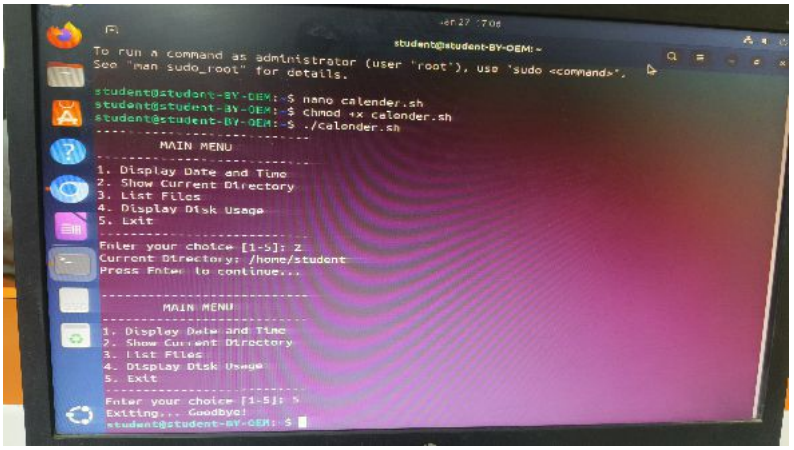
Display calendar of current month.

Display today's date and time.

Display usernames those are currently logged in the system.

Display your terminal number

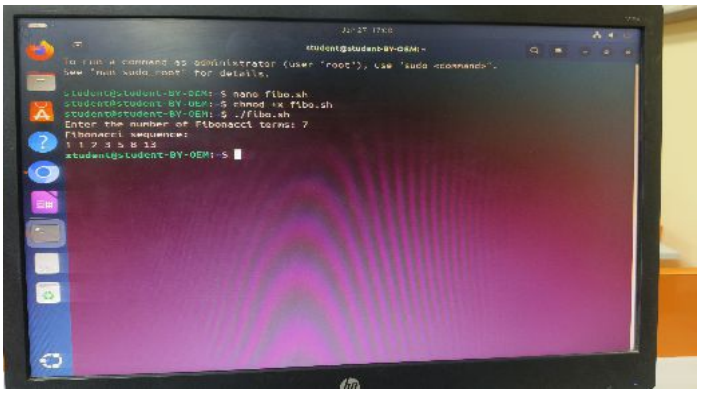
```
#!/bin/bash
while true
do
    # Display Menu
    echo "----------------------------"
    echo "        MAIN MENU"
    echo "----------------------------"
    echo "1. Display Date and Time"
    echo "2. Show Current Directory"
    echo "3. List Files"
    echo "4. Display Disk Usage"
    echo "5. Exit"
    echo "----------------------------"
    read -p "Enter your choice [1-5]: " choice
    case $choice in
        1)
            echo "Current Date and Time: $(date)"
            ;;
        2)
            echo "Current Directory: $(pwd)"
            ;;
        3)
            echo "Files in Current Directory:"
            ls -l
            ;;
        4)
            echo "Disk Usage:"
            df -h
            ;;
        5)
            echo "Exiting... Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid choice! Please select 1-5."
            ;;
    esac
    echo "Press Enter to continue..."
    read
done
```



3. Write a shell script which will generate first n Fibonacci numbers like: 1, 1, 2, 3, 5, 13

```
#!/bin/bash

# Shell Script to generate first n Fibonacci numbers

# Input n
read -p "Enter the number of Fibonacci terms: " n

# Initialize first two terms
a=1
b=1

echo "Fibonacci sequence:"

# Print first n terms
for (( i=1; i<=n; i++ ))
do
    echo -n "$a "
    # Calculate next term
    fn=$((a + b))
    a=$b
    b=$fn
done

echo ""
```

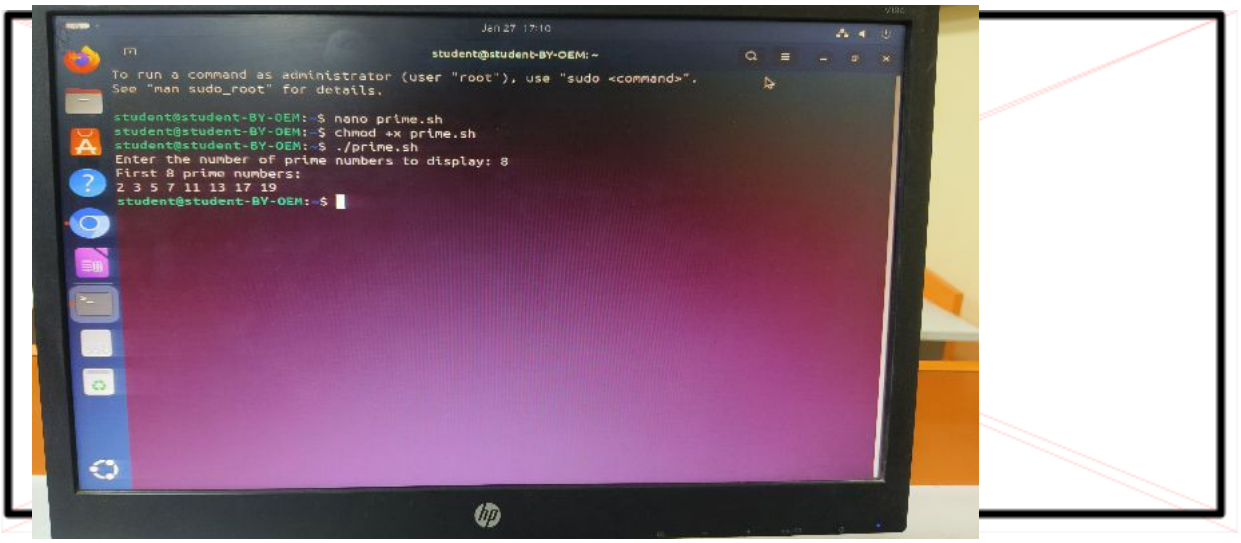
Output 3



4. Write a shell script which will accept a number b and display first n prime numbers as output.

```
#!/bin/bash

# Shell script to display first n prime numbers

# Input: number of prime numbers to display
read -p "Enter the number of prime numbers to display: " n

count=0   # To count how many primes are found
num=2     # Number to check for prime

echo "First $n prime numbers:"

while [ $count -lt $n ]
do
    is_prime=1   # Assume num is prime
    # Check if num is divisible by any number from 2 to num-1
    for (( i=2; i*i<=num; i++ ))
    do
        if (( num % i == 0 )); then
            is_prime=0
            break
        fi
    done

    if (( is_prime == 1 )); then
        echo -n "$num "
        ((count++))
    fi

    ((num++))
done

echo ""
```

Output 4:



5. Write menu driven program for file handling activity

   Creation of file.

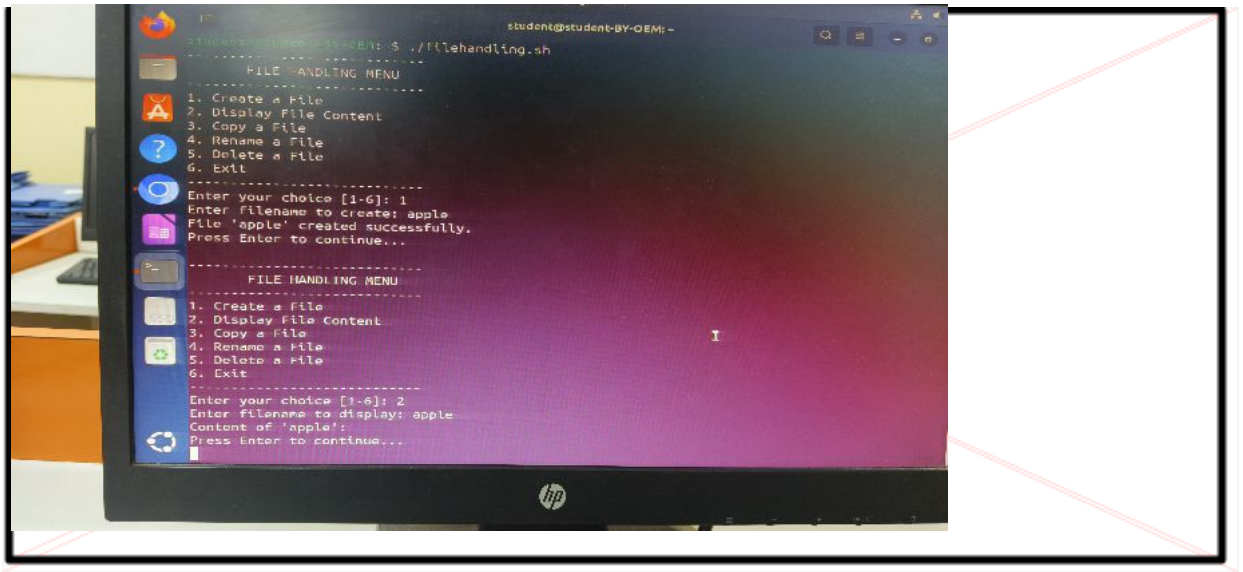
   Write content in the file.

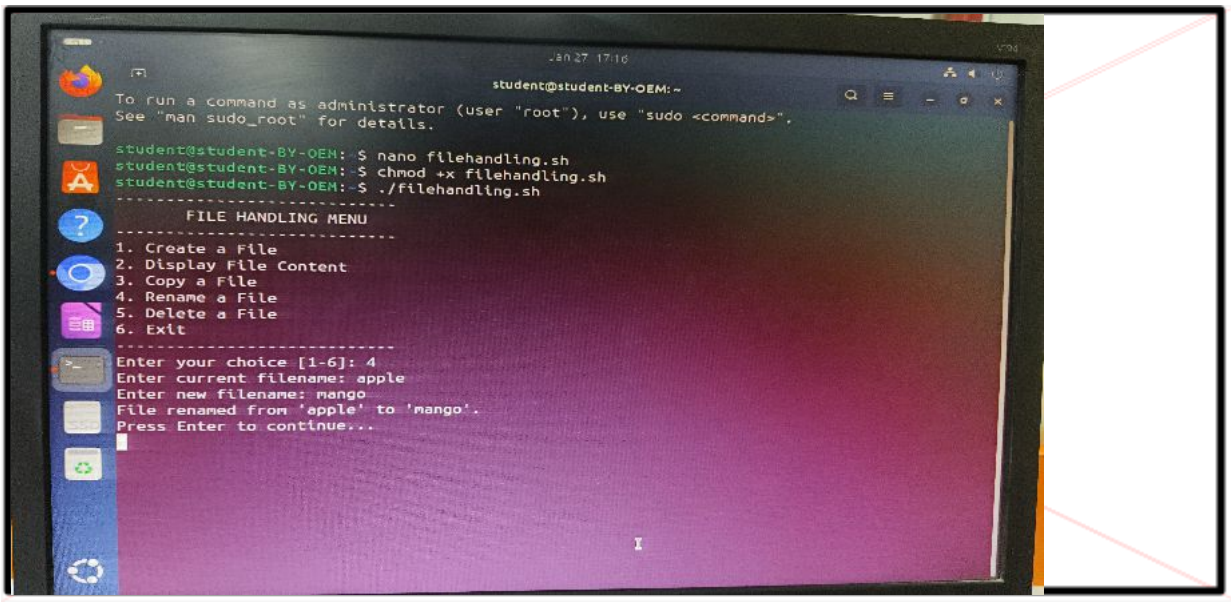
   Upend file content.

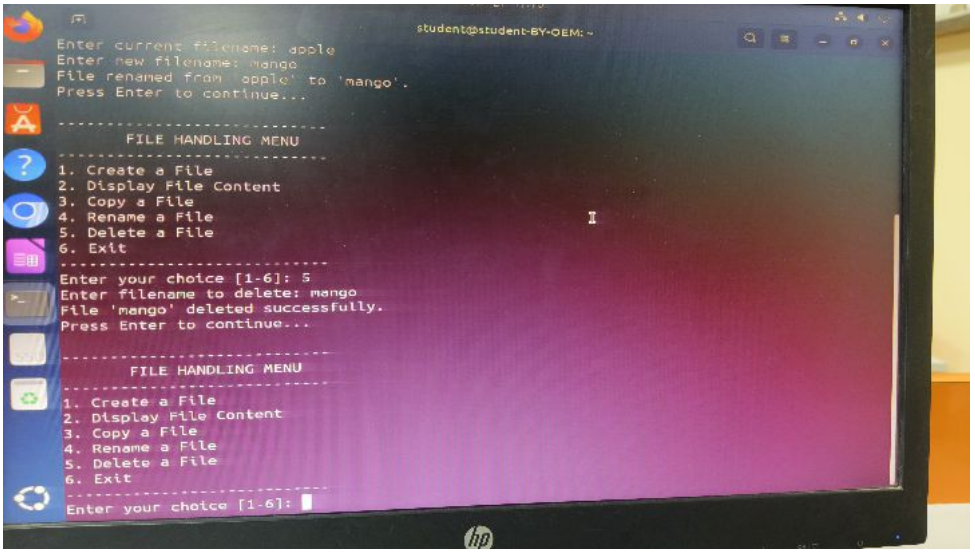
   Delete file content

```
#!/bin/bash

while true
do
    # Display Menu
    echo "----------------------------"
    echo "       FILE HANDLING MENU"
    echo "----------------------------"
    echo "1. Create a File"
    echo "2. Display File Content"
    echo "3. Copy a File"
    echo "4. Rename a File"
    echo "5. Delete a File"
    echo "6. Exit"
    echo "----------------------------"

    read -p "Enter your choice [1-6]: " choice

    case $choice in
        1)
            read -p "Enter filename to create: " filename
            touch "$filename"
            echo "File '$filename' created successfully."
            ;;
        2)
            read -p "Enter filename to display: " filename
            if [ -f "$filename" ]; then
                echo "Content of '$filename':"
                cat "$filename"
            else
                echo "File '$filename' does not exist."
            fi
            ;;
        3)
            read -p "Enter source filename: " src
            read -p "Enter destination filename: " dest
            if [ -f "$src" ]; then
                cp "$src" "$dest"
                echo "File copied from '$src' to '$dest'."
            else
                echo "Source file '$src' does not exist."
            fi
            ;;
        4)
            read -p "Enter current filename: " oldname
            read -p "Enter new filename: " newname
            if [ -f "$oldname" ]; then
                mv "$oldname" "$newname"
                echo "File renamed from '$oldname' to '$newname'."
            else
                echo "File '$oldname' does not exist."
            fi
            ;;
        5)
            read -p "Enter filename to delete: " filename
            if [ -f "$filename" ]; then
                rm "$filename"
                echo "File '$filename' deleted successfully."
            else
                echo "File '$filename' does not exist."
            fi
            ;;
        6)
            echo "Exiting... Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid choice! Please select 1-6."
            ;;
    esac

    echo "Press Enter to continue..."
    read
done
```

Output 5:


student@student-BY-OEM: ~
./filehandling.sh
FILE HANDLING MENU
1. Create a File
2. Display File Content
3. Copy a File
4. Rename a File
5. Delete a File
6. Exit
Enter your choice [1-6]: 1
Enter filename to create: apple
File 'apple' created successfully.
Press Enter to continue...
FILE HANDLING MENU
1. Create a File
2. Display File Content
3. Copy a File
4. Rename a File
5. Delete a File
6. Exit
Enter your choice [1-6]: 2
Enter filename to display: apple
Content of 'apple':
Press Enter to continue...


student@student-BY-OEM: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
student@student-BY-OEM: $ nano filehandling.sh
student@student-BY-OEM: $ chmod +x filehandling.sh
student@student-BY-OEM: $ ./filehandling.sh
FILE HANDLING MENU
1. Create a File
2. Display File Content
3. Copy a File
4. Rename a File
5. Delete a File
6. Exit
Enter your choice [1-6]: 4
Enter current filename: apple
Enter new filename: mango
File renamed from 'apple' to 'mango'.
Press Enter to continue...


student@student-BY-OEM: ~
Enter current filename: apple
Enter new filename: mango
File renamed from 'apple' to 'mango'.
Press Enter to continue...
FILE HANDLING MENU
1. Create a File
2. Display File Content
3. Copy a File
4. Rename a File
5. Delete a File
6. Exit
Enter your choice [1-6]: 5
Enter filename to delete: mango
File 'mango' deleted successfully.
Press Enter to continue...
FILE HANDLING MENU
1. Create a File
2. Display File Content
3. Copy a File
4. Rename a File
5. Delete a File
6. Exit
Enter your choice [1-6]:

❖ **Conclusion**: In this practical, we conclude that shell scripting efficiently automates tasks like marksheet generation, system information display, number computations, and file management, enhancing system operations and user interaction through command-line utilities.

❖ **Discussion Questions:**

1. **What is the purpose of using shell scripting in this practical?**

   Ans: Shell scripting in this practical automates tasks like marksheet generation, Fibonacci sequence calculation, system information display, and file management. It improves efficiency by reducing manual interventions and allows easy automation of routine tasks using simple commands and logic.

2. **Which command is used to display the current date and time?**

   Ans: The date command is used to display the current date and time. It provides a formatted output, showing the system's current time, date, and related information depending on the user's locale settings.

3. **How does the script calculate the Fibonacci sequence?**

   Ans: The Fibonacci sequence script uses a loop to iteratively calculate each number by summing the previous two numbers in the sequence, starting from 0 and 1. This continues until the specified number of terms is generated, making use of simple arithmetic operations.

4. **Which command is used to create a file in the file management script?**
   Ans: The touch command is used to create an empty file or update the timestamp of an existing file. This is a simple and effective way to initiate a file before performing further operations like writing or appending content.

5. **How does the prime number script determine if a number is prime?**
   Ans: The prime number script checks for divisibility by iterating through potential divisors up to the square root of the number. If no divisor other than 1 and the number itself is found, the number is classified as prime.

❖ **References:**

https://www.tutorialspoint.com/unix/shell_scripting.html
https://www.javatpoint.com/shell-scripting-tutorial

Date:___/___/2026

Signature
Course
Coordinator

B.Tech
CSE(AIML)