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: Shraddha Agalawe

USN: EE24TCD002

Semester / Year: 4th Sem / 2nd Year

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.

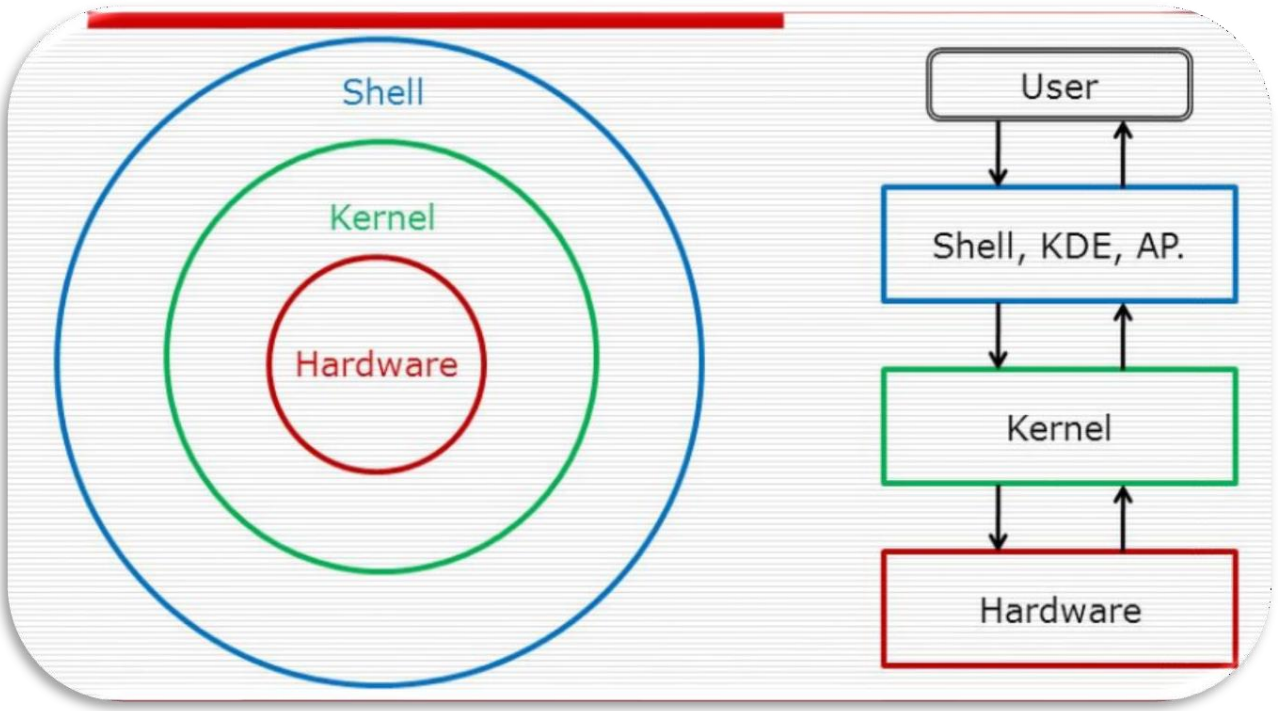
5. Menu-Driven File Management

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.**

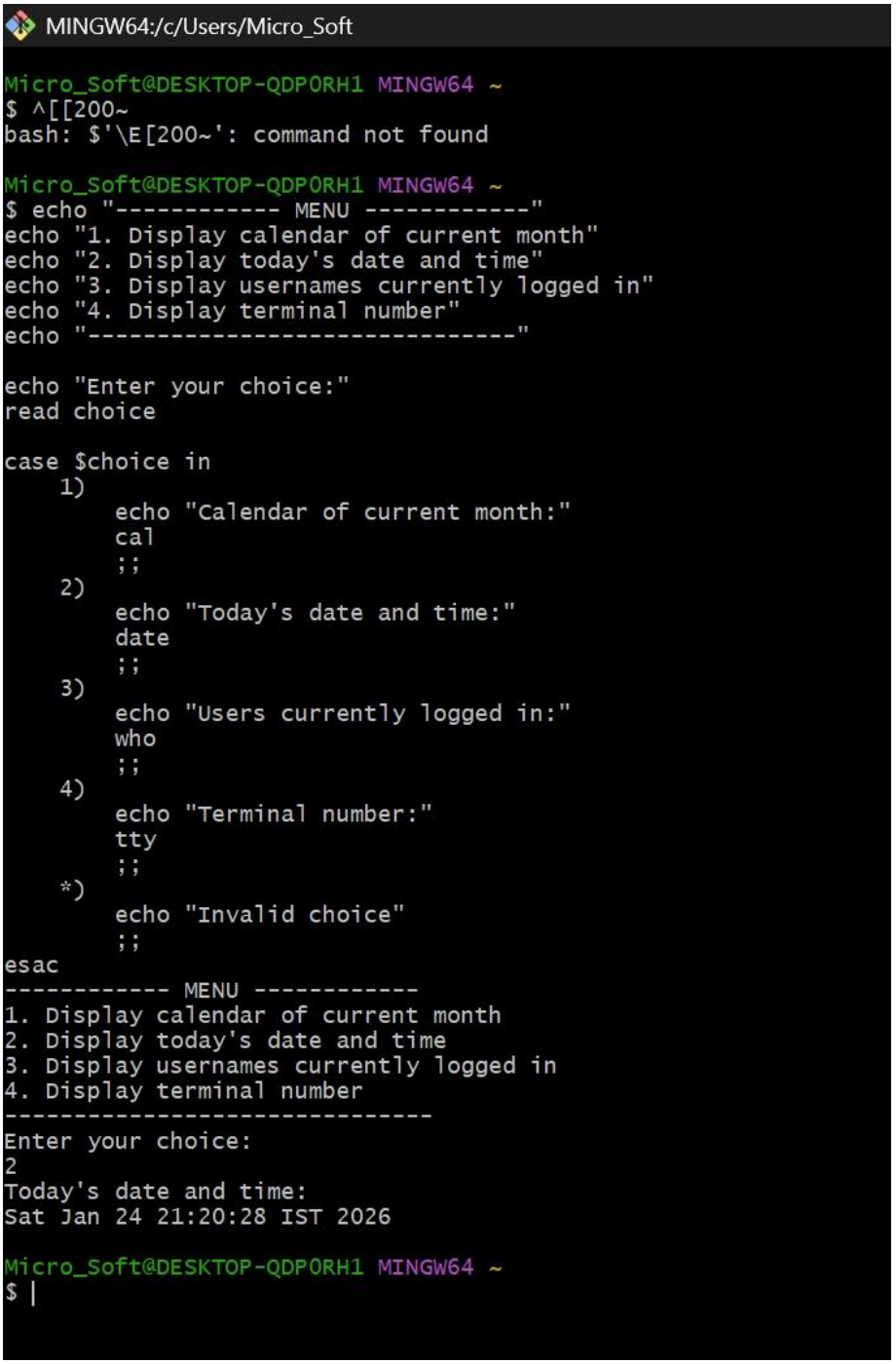
Output 1:

MINGW64:/c/Users/Micro_Soft

```
Micro_Soft@DESKTOP-QDPORH1 MINGW64 ~
$ echo "Enter marks of English"

read m1

echo "Enter marks of Maths"

read m2

echo "Enter marks of Science"

read m3

total=$((m1+m2+m3))

percentage=$((total/3))

echo "Student: Total Marks = $total"

echo "Percentage = $percentage"

if [ $percentage -ge 75 ]; then echo "Class: Distinction"

elif [ $percentage -ge 60 ]; then echo "Class: First Class"

elif [ $percentage -ge 40 ]; then echo "Class: Second Class"

elif [ $percentage -ge 35 ]; then echo "Class: Third Class"

else

echo "Class Fail"

fi
Enter marks of English
55
Enter marks of Maths
40
Enter marks of Science
80
Student: Total Marks = 120
Percentage = 40
Class: Second Class

Micro_Soft@DESKTOP-QDPORH1 MINGW64 ~
$
```

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**

Output 2:



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

Output 3:

MINGW64:/c/Users/Micro_Soft

```
Micro_Soft@DESKTOP-QDPORH1 MINGW64 ~
$ echo "Enter the value of n"
read n

a=1
b=1

echo "Fibonacci series:"

i=1
while [ $i -le $n ]
do
    echo $a
    c=$((a+b))
    a=$b
    b=$c
    i=$((i+1))
done
Enter the value of n
20
Fibonacci series:
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765

Micro_Soft@DESKTOP-QDPORH1 MINGW64 ~
$
```

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

Output 4:

MINGW64:/c/Users/Micro_Soft

```
Micro_Soft@DESKTOP-QDPORH1 MINGW64 ~
$ echo "Enter the value of n"
read n

count=0
num=2

echo "First $n prime numbers are:"

while [ $count -lt $n ]
do
    flag=0
    i=2

    while [ $i -le $((num/2)) ]
    do
        if [ $((num%i)) -eq 0 ]; then
            flag=1
            break
        fi
        i=$((i+1))
    done

    if [ $flag -eq 0 ]; then
        echo $num
        count=$((count+1))
    fi

    num=$((num+1))
done
Enter the value of n
5
First 5 prime numbers are:
2
3
5
7
11

Micro_Soft@DESKTOP-QDPORH1 MINGW64 ~
$
```

5. Write menu driven program for file handling activity
   - Creation of file.
   - Write content in the file.
   - Upend file content.
   - Delete file content

Output 5:

```
MINGW64:/c/Users/Micro_Soft

Micro_Soft@DESKTOP-QDP0RH1 MINGW64 ~
$ echo "----------- FILE HANDLING MENU -----------"
echo "1. Create a file"
echo "2. Write content into the file"
echo "3. Append content to the file"
echo "4. Delete file content"
echo "----------------------------------------"

echo "Enter your choice:"
read choice

echo "Enter file name:"
read filename

case $choice in
    1)
        touch $filename
        echo "File created successfully"
        ;;
    2)
        echo "Enter content (Press Ctrl+D to save):"
        cat > $filename
        ;;
    3)
        echo "Enter content to append (Press Ctrl+D to save):"
        cat >> $filename
        ;;
    4)
        > $filename
        echo "File content deleted"
        ;;
    *)
        echo "Invalid choice"
        ;;
esac
----------- FILE HANDLING MENU -----------
1. Create a file
2. Write content into the file
3. Append content to the file
4. Delete file content
----------------------------------------
Enter your choice:
2
Enter file name:
Operating System
Enter content (Press Ctrl+D to save):
bash: $filename: ambiguous redirect

Micro_Soft@DESKTOP-QDP0RH1 MINGW64 ~
$
```