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: Aaditya S Gudhniye

USN:CM24062

Semester / Year:

Academic Session:

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.

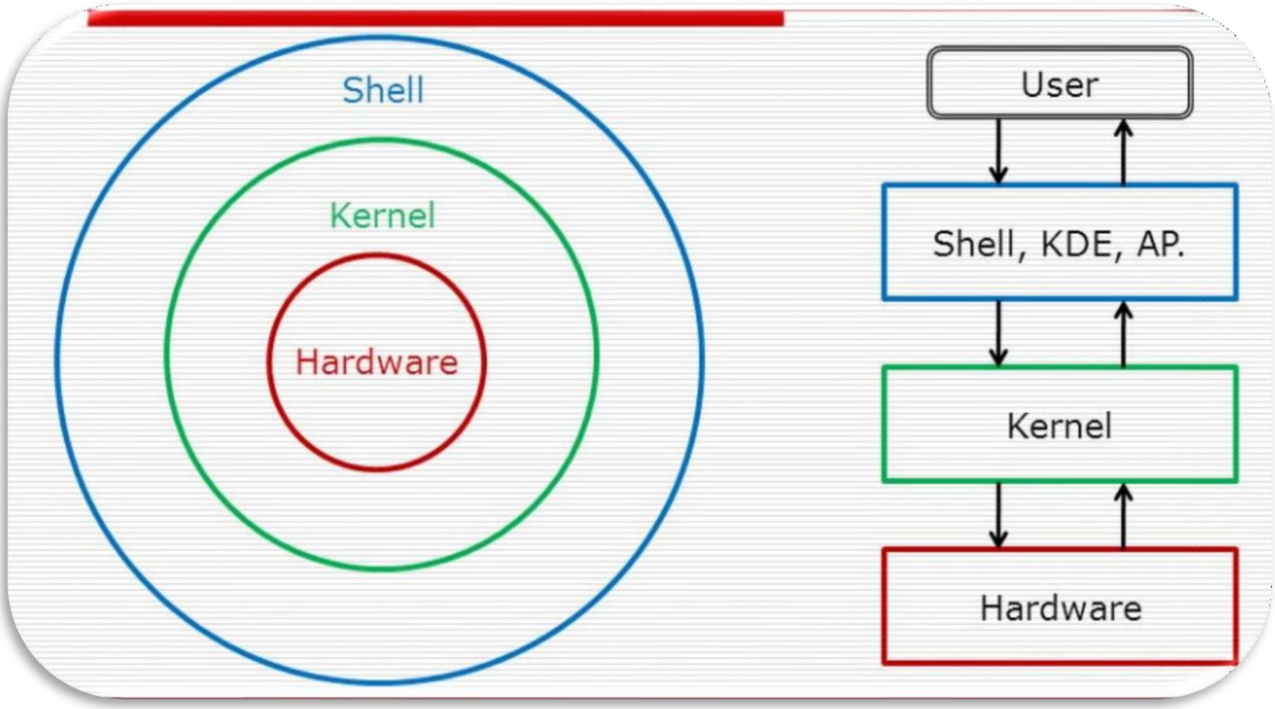
5. Menu-Driven File Management

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.

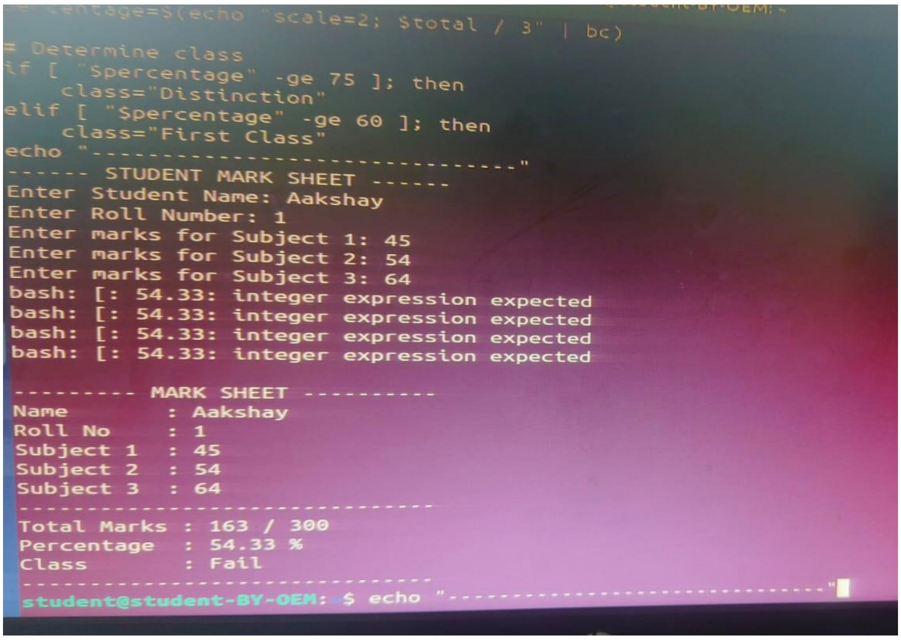
Diagrammatical View of Shell



❖ 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.**

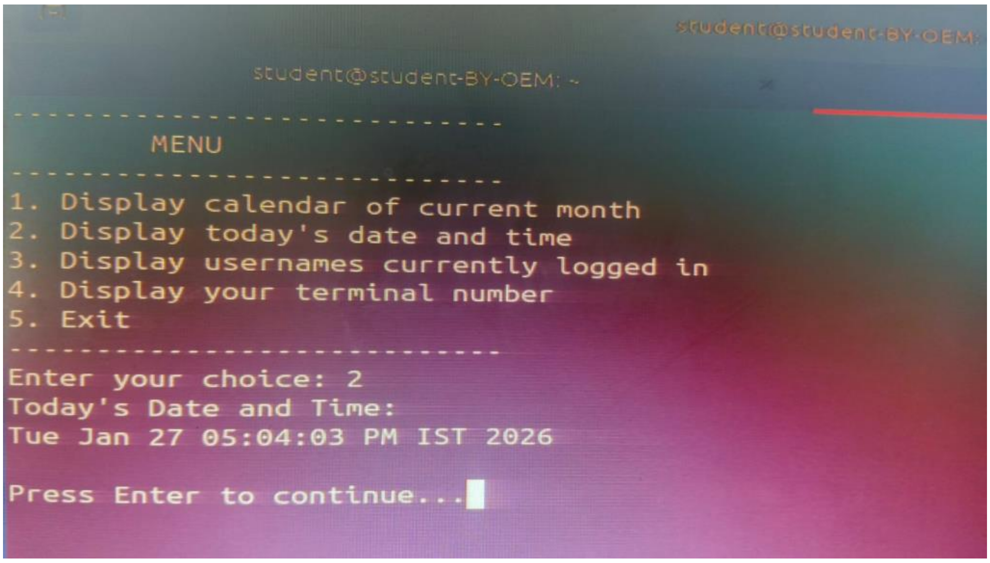
Output 1:



```
ercentage=$(echo "scale=2; $total / 3" | bc)

# Determine class
if [ "$percentage" -ge 75 ]; then
    class="Distinction"
elif [ "$percentage" -ge 60 ]; then
    class="First Class"
echo "--------------------------------"
------ STUDENT MARK SHEET ------
Enter Student Name: Aakshay
Enter Roll Number: 1
Enter marks for Subject 1: 45
Enter marks for Subject 2: 54
Enter marks for Subject 3: 64
bash: [: 54.33: integer expression expected
bash: [: 54.33: integer expression expected
bash: [: 54.33: integer expression expected
bash: [: 54.33: integer expression expected

--------- MARK SHEET ----------
Name       : Aakshay
Roll No    : 1
Subject 1  : 45
Subject 2  : 54
Subject 3  : 64
------------------------------
Total Marks : 163 / 300
Percentage   : 54.33 %
Class        : Fail
------------------------------
student@student-BY-OEM:~$ echo "------------------------------"
```

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**

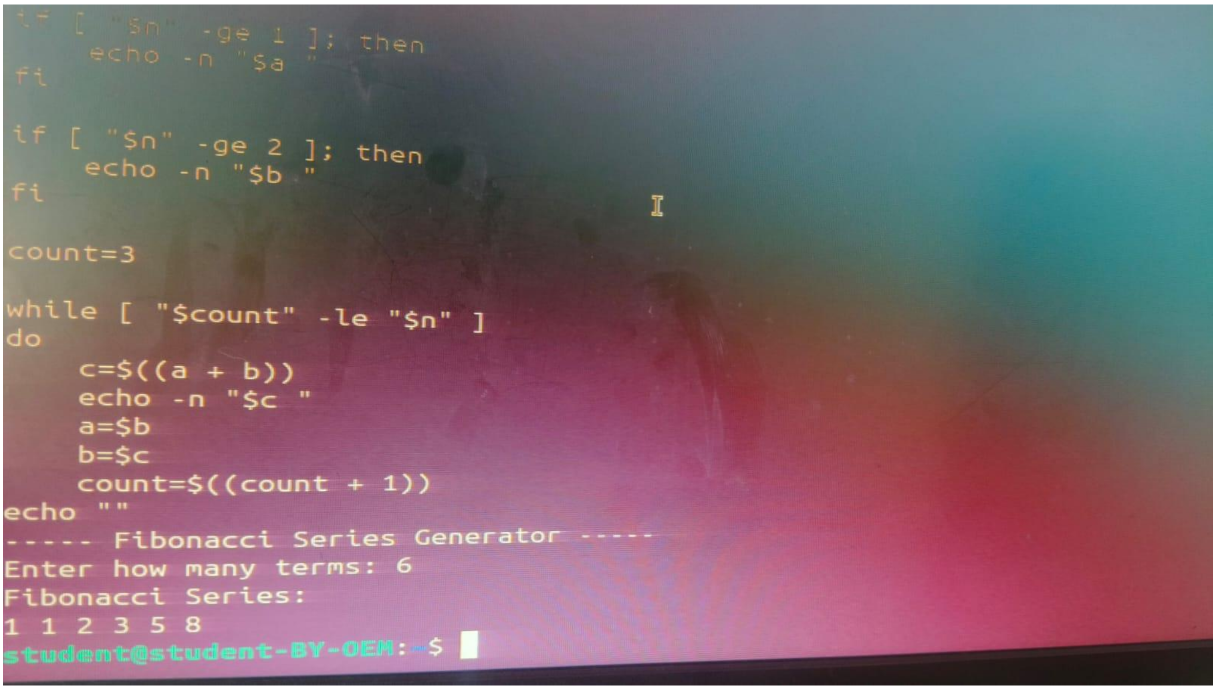
Output 2:



```
student@student-BY-OEM: ~
----------------------------
        MENU
----------------------------
1. Display calendar of current month
2. Display today's date and time
3. Display usernames currently logged in
4. Display your terminal number
5. Exit
----------------------------
Enter your choice: 2
Today's Date and Time:
Tue Jan 27 05:04:03 PM IST 2026

Press Enter to continue...
```

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

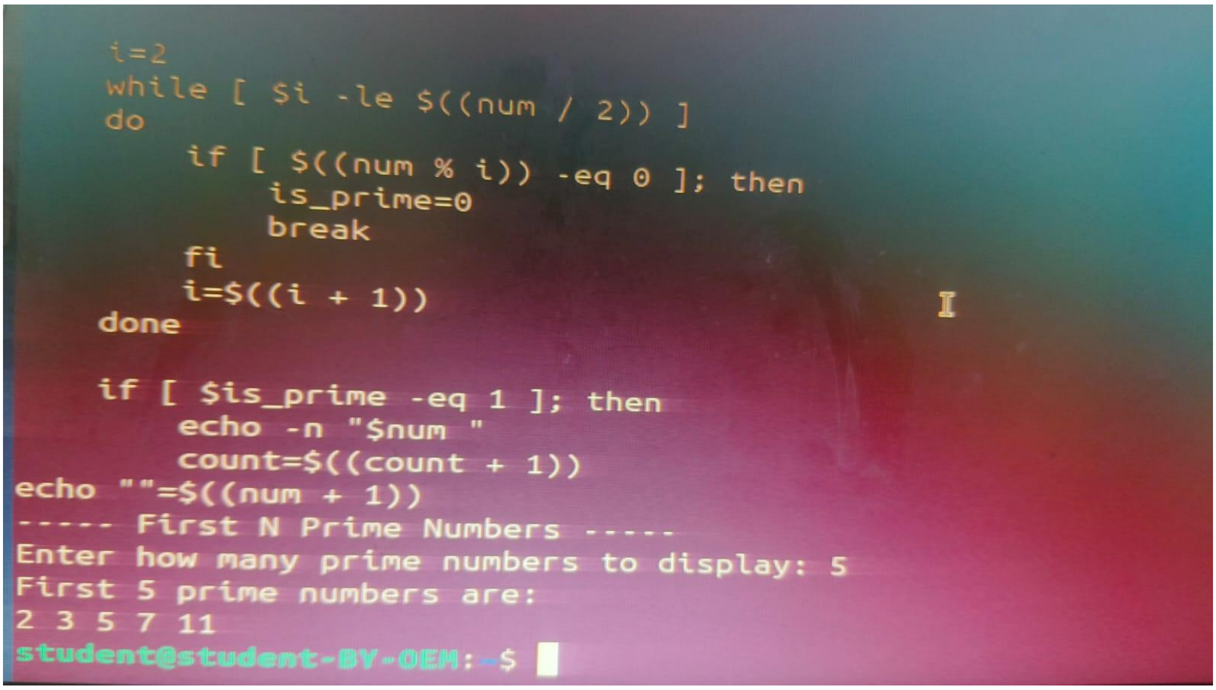
Output 3:



```
if [ "$n" -ge 1 ]; then
    echo -n "$a "
fi

if [ "$n" -ge 2 ]; then
    echo -n "$b "
fi

count=3

while [ "$count" -le "$n" ]
do
    c=$((a + b))
    echo -n "$c "
    a=$b
    b=$c
    count=$((count + 1))
echo ""
----- Fibonacci Series Generator -----
Enter how many terms: 6
Fibonacci Series:
1 1 2 3 5 8
student@student-BY-OEM:~$ 
```

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

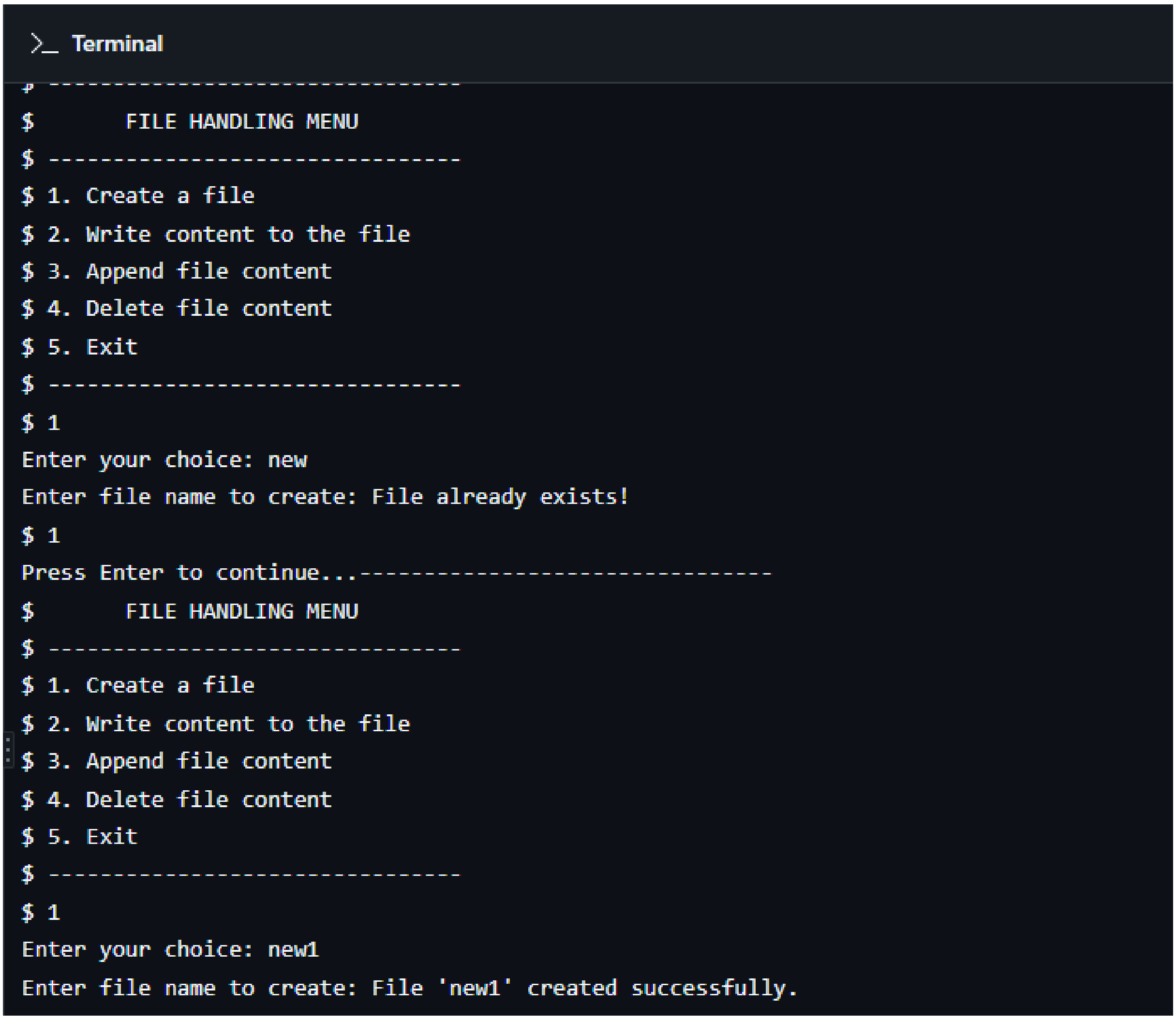
Output 4:



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

    if [ $is_prime -eq 1 ]; then
        echo -n "$num "
        count=$((count + 1))
echo ""=$((num + 1))
----- First N Prime Numbers -----
Enter how many prime numbers to display: 5
First 5 prime numbers are:
2 3 5 7 11
student@student-BY-OEM:~$ 
```

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:



```
$ --------------------------------
$       FILE HANDLING MENU
$ --------------------------------
$ 1. Create a file
$ 2. Write content to the file
$ 3. Append file content
$ 4. Delete file content
$ 5. Exit
$ --------------------------------
$ 1
Enter your choice: new
Enter file name to create: File already exists!
$ 1
Press Enter to continue...--------------------------------
$       FILE HANDLING MENU
$ --------------------------------
$ 1. Create a file
$ 2. Write content to the file
$ 3. Append file content
$ 4. Delete file content
$ 5. Exit
$ --------------------------------
$ 1
Enter your choice: new1
Enter file name to create: File 'new1' created successfully.
```

❖ **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?**
2. **Which command is used to display the current date and time?**
3. **How does the script calculate the Fibonacci sequence?**
4. **Which command is used to create a file in the file management script?**
5. **How does the prime number script determine if a number is 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)

6.