



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

### **Practical 03**

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

**Name: Rohit Kashyap**

**USN: CM24039**

**Semester / Year: 4<sup>th</sup>/2<sup>nd</sup>**

**Academic Session: 2025-26**

**Date of Performance: 27-01-26**

**Date of Submission: 3-02-26**

❖ **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. Marksheets 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**

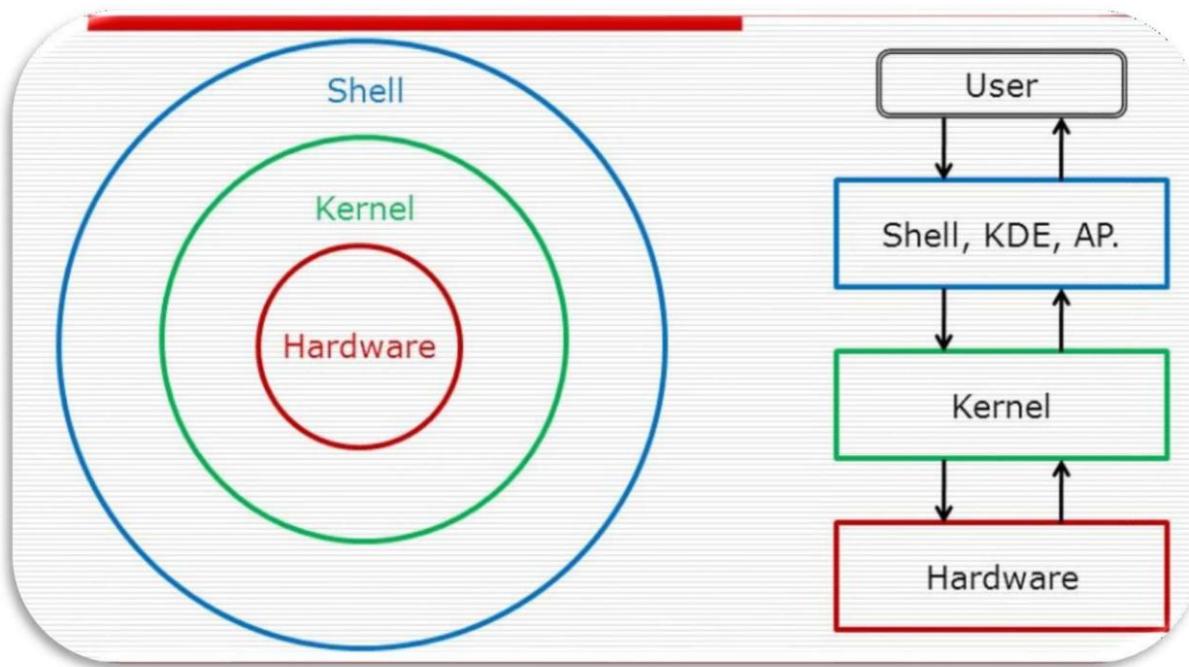
The file handling script enables users to create, write, append, and delete file content. The case construct manages different file operations.

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.

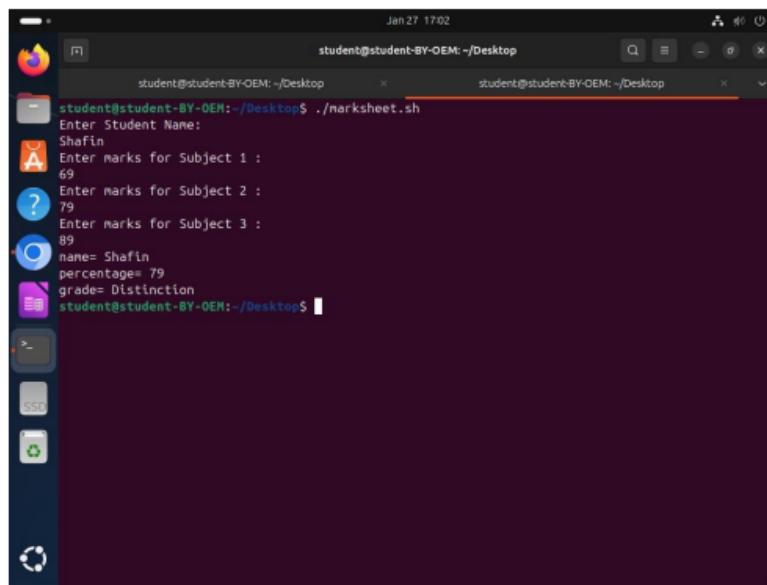
#### **Diagrammatical View of Shell**



❖ 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:**



The screenshot shows a terminal window titled "student@student-BY-OEM: ~/Desktop". The window contains the following text output from a shell script:

```
student@student-BY-OEM:~/Desktop$ ./marksheet.sh
Enter Student Name:
Shafin
Enter marks for Subject 1 :
69
Enter marks for Subject 2 :
79
Enter marks for Subject 3 :
89
name= Shafin
percentage= 79
grade= Distinction
student@student-BY-OEM:~/Desktop$
```

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:**

The screenshot shows a terminal window with a dark background and light-colored text. At the top, it displays the date and time: Feb 2 16:24. The user is running a shell script named menu.sh from the Desktop directory. The script presents a menu with five options. The user enters choice 1, which displays the calendar for February 2026. The user then enters choice 2, which shows the current date and time as Mon Feb 2 16:24:31 IST 2026. Finally, the user enters choice 3, which lists the currently logged-in users: student at seat0 and student at tty2, both connected to the login screen.

```
student@student-BY-OEM:~/Desktop$ nano menu.sh
student@student-BY-OEM:~/Desktop$ ./menu.sh
1. Display calendar of current month
2. Display today's date and time
3. Display currently logged in users
4. Display terminal number
5. Exit
Enter your choice: 1
Calendar of current month:
cal: setlocale: No such file or directory
February 2026
Su Mo Tu We Th Fr Sa
1  2  3  4  5  6  7
8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28

1. Display calendar of current month
2. Display today's date and time
3. Display currently logged in users
4. Display terminal number
5. Exit
Enter your choice: 2
Today's date and time:
Mon Feb 2 16:24:31 IST 2026

1. Display calendar of current month
2. Display today's date and time
3. Display currently logged in users
```

This screenshot shows a continuation of the terminal session from the previous one. The user has entered choice 3 again, which now displays a detailed list of currently logged-in users: student at seat0 and student at tty2, both connected to the login screen. The user then enters choice 4, which shows the terminal number as /dev/pts/0. Finally, the user enters choice 5, which exits the program.

```
student@student-BY-OEM:~/Desktop$ 
student@student-BY-OEM:~/Desktop$ Enter your choice: 2
Today's date and time:
Mon Feb 2 16:24:31 IST 2026

1. Display calendar of current month
2. Display today's date and time
3. Display currently logged in users
4. Display terminal number
5. Exit
Enter your choice: 3
Currently logged in users:
student  seat0          Feb 2 21:46 (login screen)
student  tty2          Feb 2 21:46 (tty2)

1. Display calendar of current month
2. Display today's date and time
3. Display currently logged in users
4. Display terminal number
5. Exit
Enter your choice: 4
Your terminal number:
/dev/pts/0

1. Display calendar of current month
2. Display today's date and time
3. Display currently logged in users
4. Display terminal number
5. Exit
Enter your choice: 5
Exiting program...
student@student-BY-OEM:~/Desktop$ 
```

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

**Output 3:**

Feb 2 16:29  
student@student-BY-OEM:~/Desktop\$ touch fi.sh  
student@student-BY-OEM:~/Desktop\$ nano fi.sh  
student@student-BY-OEM:~/Desktop\$ chmod +x fi.sh  
student@student-BY-OEM:~/Desktop\$ ./fi.sh  
Enter the number: 5  
Fibonacci Series:  
1 1 2 3 5  
student@student-BY-OEM:~/Desktop\$

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

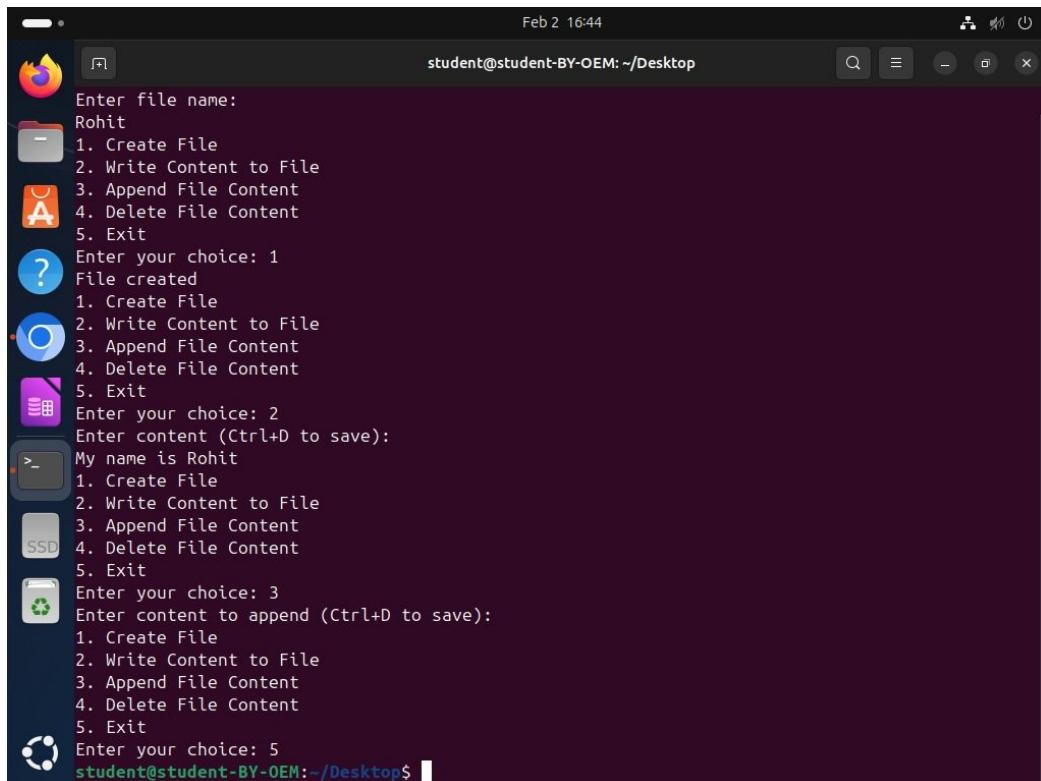
**Output 4:**

Feb 2 16:32  
student@student-BY-OEM:~/Desktop\$ touch prime.sh  
student@student-BY-OEM:~/Desktop\$ nano prime.sh  
student@student-BY-OEM:~/Desktop\$ chmod +x prime.sh  
student@student-BY-OEM:~/Desktop\$ ./prime.sh  
Enter the number: 15  
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47  
student@student-BY-OEM:~/Desktop\$

**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:**



The screenshot shows a terminal window titled "student@student-BY-OEM: ~/Desktop". The window has a dark background with light-colored text. It displays a menu-driven program for file handling. The menu options are:

- 1. Create File
- 2. Write Content to File
- 3. Append File Content
- 4. Delete File Content
- 5. Exit

The program interacts with the user through prompts and responses:

- "Enter file name: Rohit"
- "Enter your choice: 1"
- "File created"
- "Enter your choice: 2"
- "Enter content (Ctrl+D to save): My name is Rohit"
- "Enter your choice: 3"
- "Enter content to append (Ctrl+D to save):"
- "Enter your choice: 5"

The terminal prompt at the bottom is "student@student-BY-OEM:~/Desktop\$".

❖ **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.tutorialspoint.com/unix/shell_scripting.html)

<https://www.javatpoint.com/shell-scripting-tutorial>

---

**Date:**03/02/2026

**Signature**  
Course Coordinator  
B.Tech CSE(AIML)