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:	Prithviraj Bele
USN:	CD24034

Semester / Year:	4th Sem / 2nd Year
Academic Session:	2025-26

Date of Performance:

Date of Submission:

Department of Emerging Technologies, SBJITMR, Nagpur.

❖ **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
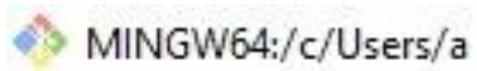- rm for deleting files

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

Diagrammatical View of Shell
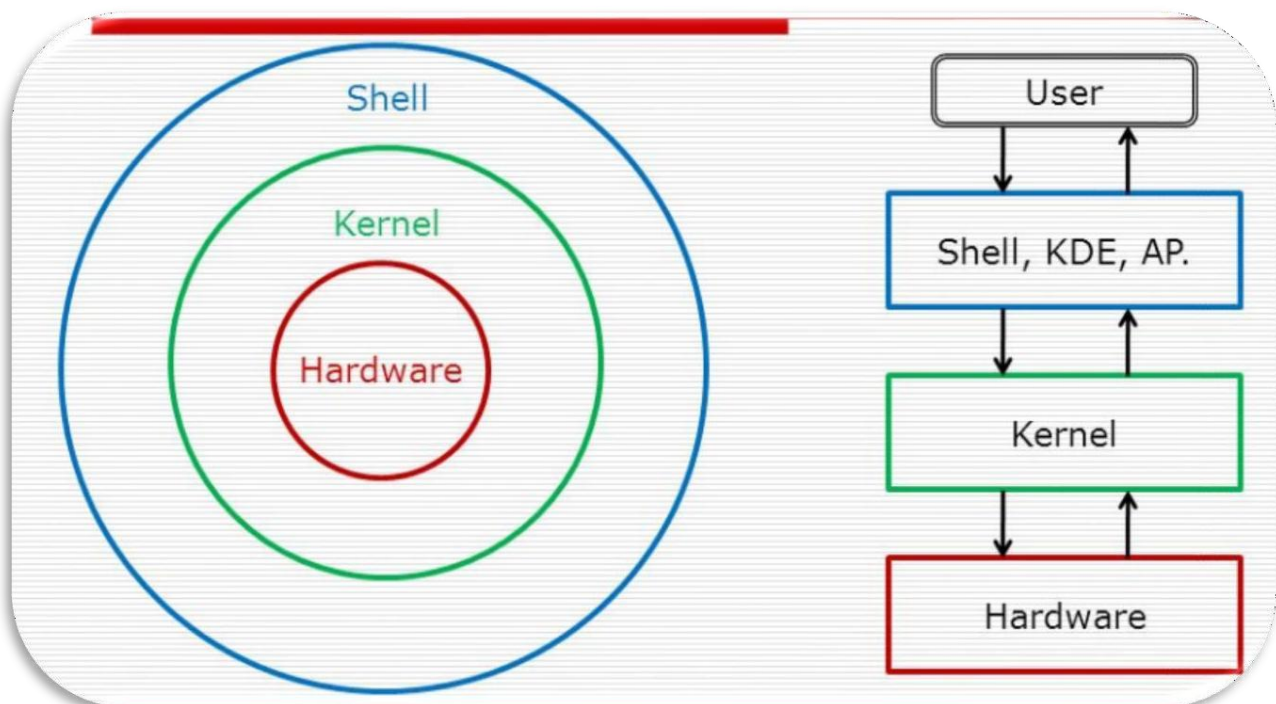
CODES:

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.

MINGW64:/c/Users/a

```
prithviraj@DESKTOP-4CLLOHE MINGW64 ~
$ echo "Enter marks of English : "
read m1
echo "Enter marks of Maths : "
read m2
echo "Enter marks of C Programming : "
read m3
echo "Enter marks of Java : "
read m4
echo "Enter marks of Python : "
read m5
total=$((m1+m2+m3+m4+m5))
per=$((total * 100 / 500))
echo "Student's Total Marks : $total"
echo "Percentage : $per"
if [ $per -ge 75 ]; then
        echo "Class : Distiction"
elif [ $per -ge 60 ]; then
        echo "Class : First Class"
elif [ $per -ge 50 ]; then
        echo "Class : Second Class"
elif [ $per -ge 40 ]; then
        echo "Class : Pass"
else
        echo "Class : Fail"
fi
```

Output 1 :

```
Enter marks of English :
98
Enter marks of Maths :
89
Enter marks of C Programming :
98
Enter marks of Java :
96
Enter marks of Python :
97
Student's Total Marks : 478
9Percentage : 95
5Class : Distiction
```

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**

```
MINGW64:/c/Users/a

prithviraj@DESKTOP-4CLL0HE MINGW64 ~
$ echo "---------------------------"
echo "           MENU"
echo "---------------------------"
echo "1. Display calendar of current month"
echo "2. Display today's date and time"
echo "3. Display usernames currently logged in"
echo "4. Display your terminal number"
echo "---------------------------"

echo "Enter your choice:"
read ch

case $ch in
    1)
        date "+%B %Y"
        ;;
    2)
        date
        ;;
    3)
        who
        ;;
    4)
        tty
        ;;
    *)
        echo "Invalid Choice"
        ;;
esac
```

Output 2:

```
----------------------------
        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
----------------------------
Enter your choice:
2
Fri Jan 23 21:49:12 IST 2026
```

```
----------------------------
        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
----------------------------
Enter your choice:
3
```

```
----------------------------
        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
----------------------------
Enter your choice:
1
January 2026
```

```
----------------------------
        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
----------------------------
Enter your choice:
4
/dev/pty0
```

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

MINGW64:/c/Users/a

```
prithviraj@DESKTOP-4CLL0HE MINGW64 ~
$ echo "Enter number of terms:"
read n

a=1
b=1

echo "Fibonacci Series:"

echo -n "$a $b "

for ((i=3; i<=n; i++))
do
    c=$((a + b))
    echo -n "$c "
    a=$b
    b=$c
done

echo
```

Output 3:

```
echo
Enter number of terms:
2
Fibonacci Series:
1 1
```

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

```
prithviraj@DESKTOP-4CLL0HE MINGW64 ~
$ echo "Enter how many prime numbers you want:"
read n

count=0
num=2

echo "Prime Numbers:"

while [ $count -lt $n ]
do
    i=2
    flag=1

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

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

    num=$((num + 1))
done

echo
```

Output 4:

```
Enter how many prime numbers you want:
7
Prime Numbers:
2 3 5 7 11 13 17
```

5. Write menu driven program for file handling activity

 Creation of file.

 Write content in the file.

 Upend file content.

 Delete file content

```
prithviraj@DESKTOP-4CLLOHE MINGW64 ~
$ echo "-----------------------------"
echo "     FILE HANDLING MENU"
echo "-----------------------------"
echo "1. Create a file"
echo "2. Write content in file"
echo "3. Append file content"
echo "4. Delete file content"
echo "-----------------------------"

echo "Enter your choice:"
read ch

echo "Enter file name:"
read fname

case $ch in
    1)
        touch $fname
        echo "File created successfully"
        ;;
    2)
        echo "Enter content (Press Ctrl+D to save):"
        cat > $fname
        echo "Content written successfully"
        ;;
    3)
        echo "Enter content to append (Press Ctrl+D to save):"
        cat >> $fname
        echo "Content appended successfully"
        ;;
    4)
        > $fname
        echo "File content deleted successfully"
        ;;
    *)
        echo "Invalid Choice"
        ;;
esac
```

Output 5:

```
------------------------------
     FILE HANDLING MENU
------------------------------
1. Create a file
2. Write content in file
3. Append file content
4. Delete file content
------------------------------
Enter your choice:
1
Enter file name:
prithvi.txt
File created successfully
```

```
           ;;
esac
------------------------------
     FILE HANDLING MENU
------------------------------
1. Create a file
2. Write content in file
3. Append file content
4. Delete file content
------------------------------
Enter your choice:
2
Enter file name:
prithvi.txt
Enter content (Press Ctrl+D to save):
hellow os!Content written successfully
```

```
--------------------------------
        FILE HANDLING MENU
--------------------------------
1. Create a file
2. Write content in file
3. Append file content
4. Delete file content
--------------------------------
Enter your choice:
4
Enter file name:
prithvi.txt
File content deleted successfully

        ''

esac
--------------------------------
        FILE HANDLING MENU
--------------------------------
1. Create a file
2. Write content in file
3. Append file content
4. Delete file content
--------------------------------
Enter your choice:
3
Enter file name:
prithvi.txt
Enter content to append (Press Ctrl+D to save):
Hellow OS!Content appended 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?**
   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(DS)
Sem: 4 / 2025-26

Department of Emerging Technologies, SBJITMR, Nagpur.