



## **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: Prasad Hore**

**USN: CM24034**

**Semester / Year: IV/II**

**Academic Session: 2025-26**

**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. 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

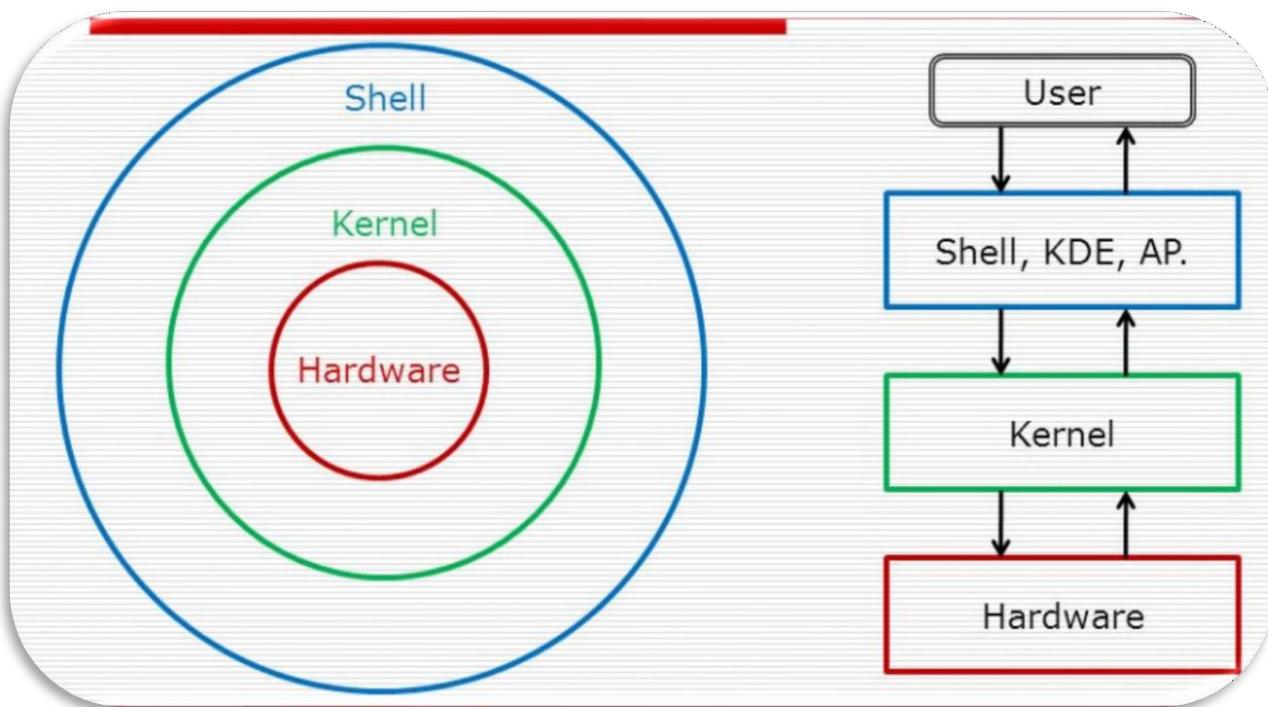
The file handling script enables users to create, write, append, and delete file content. The case construct manages different file operations.

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.

The screenshot shows a terminal window with two tabs. The left tab is titled "student@student-BY-OEM: ~" and contains the command "GNU nano 7.2". The right tab is also titled "student@student-BY-OEM: ~" and contains the script "marksheet.sh \*". The script code is as follows:

```
GNU nano 7.2
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))
echo "_____
echo "Student Name: $name"
echo "Total Marks: $total"
echo "Percentage: $percentage%"
if [ $percentage -ge 75 ]
then
    echo "Grade: A"
elif [ $percentage -ge 60 ]
then
    echo "Grade: B"
elif [ $percentage -ge 40 ]
then
    echo "Grade: C"
else
    echo "Grade: Fail"
fi
```

The screenshot shows a terminal window with three tabs. The left tab is titled "student@student-BY-OEM: ~". The middle tab is titled "student@student-BY-OEM: ~" and shows the execution of the script. The right tab is titled "student@student-BY-OEM: ~". The output of the script execution is as follows:

```
Jan 27 16:49
student@student-BY-OEM: ~
student@student-BY-OEM: ~
_____
Student Name: Narayani
Total Marks: 225
Percentage: 75%
./marksheet.sh: line 19: if[75 -ge 75]: command not found
./marksheet.sh: line 20: syntax error near unexpected token `then'
./marksheet.sh: line 20: `then'
student@student-BY-OEM:~$ nano marksheet.sh
student@student-BY-OEM:~$ chmod +x marksheet.sh
student@student-BY-OEM:~$ ./marksheet.sh
Enter Student Name:
Narayani
Enter marks of Subject 1:
80
Enter marks of Subject 2:
75
Enter marks of subject 3:
70
_____
Student Name: Narayani
Total Marks: 225
Percentage: 75%
Grade: A
student@student-BY-OEM:~$
```

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

```
GNU nano 7.2                                     ex2.sh
echo "=====Menu Program===="
echo "1. Display Current Date"
echo "2. Display Cureent Directory"
echo "3. List Files in Directory"
echo "4. Exit"
echo "====="
echo ""
echo "Enter your Choice:"
read choice

case $choice in
1) echo "Current Date and Time:"
date;;
2) echo "Current Directory:"
pwd;;
3) echo "Files in Directory:"
ls;;
4) echo "Existing program..."
exit;;
*) echo "Invalid choice"
;;
esac
```

```
student@student-BY-OEM:~$ ./ex2.sh
=====Menu Program=====
1. Display Current Date
2. Display Cureent Directory
3. List Files in Directory
4. Exit
=====

Enter your Choice:
3
Files in Directory:
demo.sh  Documents  ex1.sh  Music      Public    Templates
Desktop   Downloads  ex2.sh  Pictures   snap      Videos
student@student-BY-OEM:~$
```

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

```
Jan 27 17:16
student@student-BY-OEM: ~
student@student-BY-OEM: ~
student@student-BY-OEM: ~
student@student-BY-OEM: ~

1 2 3 4 5
Fibonacci Series:
./fibonacci.sh: line 6: ((: 1 2 3 4 5: syntax error in expression (error token is "2 3 4 5"))

student@student-BY-OEM:~$ ^C
student@student-BY-OEM:~$ nano fibonacci.sh
student@student-BY-OEM:~$ chmod +x fibonacci.sh
student@student-BY-OEM:~$ ./fibonacci.sh
Enter number of terms:
6
Fibonacci Series:
1 1 2358
student@student-BY-OEM:~$ nano fibonacci.sh
student@student-BY-OEM:~$ chmod +x fibonacci.ch
chmod: cannot access 'fibonacci.ch': No such file or directory
student@student-BY-OEM:~$ chmod +x fibonacci.sh
student@student-BY-OEM:~$ ./fibonacci.sh
Enter number of terms:
6
Fibonacci Series:
1 1 2 3 5 8
student@student-BY-OEM:~$ nano fibonacci.sh
student@student-BY-OEM:~$ chmod +x fibonacci.sh
student@student-BY-OEM:~$ ./fibonacci.sh
Enter number of terms:
6
Fibonacci Series:
```

```
Jan 27 17:15
student@student-BY-OEM: ~
student@student-BY-OEM: ~
student@student-BY-OEM: ~
student@student-BY-OEM: ~

GNU nano 7.2
fibonacci.sh
echo "Enter number of terms:"
read n

a=1
b=1
echo "Fibonacci Series:"

for (( i=1; i<=n; i++ ))
do
  echo -n "$a "
  c=$((a + b))
  a=$b
  b=$c
done

echo ""
```

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

```
GNU nano 7.2                                         ex4.sh
echo "Enter value of n:"
read n
count=0
num=2
echo "First $n prime numbers are:"
while [ $count -lt $n ]
do
flag=1
for (( i=2; i<=num/2; i++ ))
do
if [ $(($num % i)) -eq 0 ];
then flag=0
break
fi
done
if [ $flag -eq 1 ]; then echo -n "$num "
count=$((count + 1))
fi
num=$((num + 1))
done
echo ""
```

```
Enter value of n:
9
First 9 prime numbers are:
2 3 5 7 11 13 17 19 23
student@student-BY-OEM:~$
```

5. Write menu driven program for file handling activity

  - Creation of file.
  - Write content in the file.
  - Append file content.
  - Delete file content

```
student@student-BY-OEM: ~
student@student-BY-OEM: ~
student@student-BY-OEM: ~
student@student-BY-OEM: ~

GNU nano 7.2
ex5.sh
echo "File Management Operations"

touch demo.txt
echo "File Created successfully"

echo "Hello Ubuntu Shell Scripting" > demo.txt
echo "Content written to file"

echo "Reading file content:"
cat demo.txt

mv demo.txt newfile.txt
echo "File renamed"

rm newfile.txt
echo "File deleted"
```

```
Jan 27 17:13
student@student-BY-OEM:~ student@student-BY-OEM:~ student@student-BY-OEM:~ student@student-BY-OEM:~ 
student@student-BY-OEM:~ x student@student-BY-OEM:~ x student@student-BY-OEM:~ x student@student-BY-OEM:~ x 
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

student@student-BY-OEM:~$ nano ex5.sh
student@student-BY-OEM:~$ chmod +x ex5.sh
student@student-BY-OEM:~$ ./ex5.sh
File Management Operations
File Created successfully
Content written to file
Reading file content:
Hello Ubuntu Shell Scripting
File renamed
File deleted
student@student-BY-OEM:~$
```

❖ **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: \_\_\_ / \_\_\_ /2026

**Signature**

Course Coordinator  
B.Tech CSE(AIML)