



PROJECT NAME –

## **Shell Scripting Automation**

### **Project Report**

Submitted By:

Name: **Karan Kulraj Singh**

UID: **23BCA10783**

Class: **23BCA2A**

Subject: **Linux Administration Lab**

Subject Code: **23CAP-305**

Submitted To:

Name: **Mr. Rajat Patial**

Designation: **Assistant Professor**

# **Project Title: Shell Scripting Automation for System Monitoring**

## **Aim**

To design and implement a Bash shell script that automatically retrieves and displays essential system information such as uptime, CPU details, memory usage, disk status, and logged-in users — thereby providing a quick and efficient method to monitor system performance.

## **Objective**

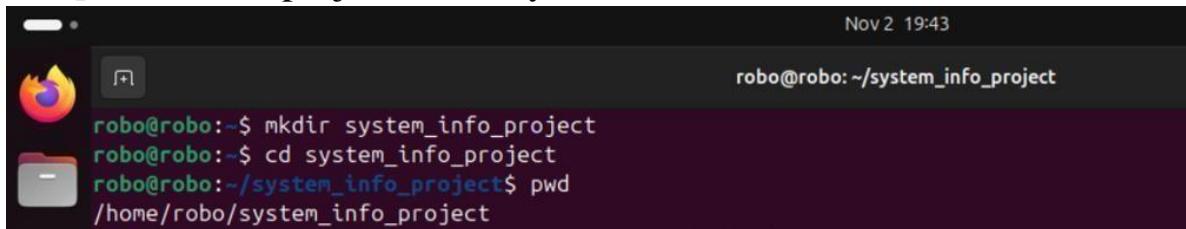
- To understand the use of Linux shell scripting for system automation.
- To create a simple and effective script that outputs important system metrics.
- To automate repetitive administrative tasks such as system resource checks.
- To demonstrate the utilization of core Linux commands for automation.

## **Commands Used and Their Purpose**

<b>Command</b>	<b>Purpose / Description</b>
#!/bin/bash	Specifies the script interpreter (Bash).
echo	Prints formatted text or messages.
uptime -p	Shows how long the system has been running.
lscpu	Displays information about the CPU architecture.
grep "Model name"	Extracts the CPU model name from lscpu output.
who	Lists all users currently logged into the system.
date	Displays the current system date and time.
chmod +x	Grants execute permission to the script file.
./system_info.sh	Runs the script from the current directory.

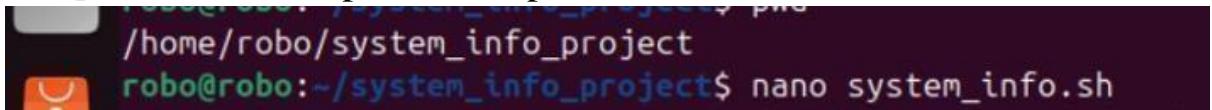
## **Working and Implementation Steps**

## Step 1: Create a project directory



```
robo@robo:~$ mkdir system_info_project
robo@robo:~$ cd system_info_project
robo@robo:~/system_info_project$ pwd
/home/robo/system_info_project
```

## Step 2: Create and open the script file



```
robo@robo:~/system_info_project$ nano system_info.sh
```

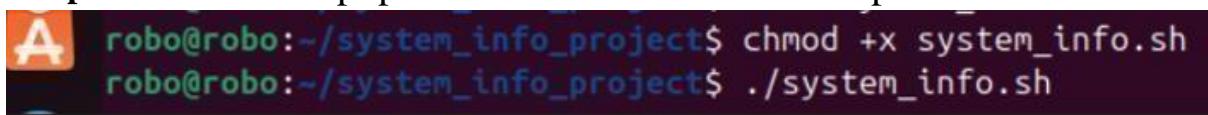
## Step 3: Write the following code:



```
GNU nano 7.2
#!/bin/bash
echo =====
echo "SYSTEM INFORMATION REPORT"
echo "Generated on: $(date)"
echo =====
echo ""
#Uptime
echo "System Uptime:"
/usr/bin/uptime -p
echo ""
#CPU Info
echo "CPU Information:"
lscpu | grep "Model name"
echo ""
#Memory Info
echo "Memory Usage:"
free -h
echo ""
#Disk Info
echo "Disk Usage:"
df -h --total | grep total
echo ""
#Logged-in User
echo "Currently Logged-in Users:"
who
echo ""
echo =====
echo "System information displayed Successfully!"
```

## Step 4: Save and exit Nano (Ctrl + O, Enter, Ctrl + X).

## Step 5: Give the script permission to run. Run the script



```
robo@robo:~/system_info_project$ chmod +x system_info.sh
robo@robo:~/system_info_project$ ./system_info.sh
```

## Output

```
- robo@robo:~/system_info_project$ nano system_info.sh
robo@robo:~/system_info_project$ chmod +x system_info.sh
robo@robo:~/system_info_project$ ./system_info.sh
=====
A SYSTEM INFORMATION REPORT
Generated on: Sun Nov  2 07:42:24 PM UTC 2025
=====

? System Uptime:
>_ up 48 minutes

CPU Information:
Model name: 12th Gen Intel(R) Core(TM) i5-1240P

Memory Usage:
total        used        free      shared  buff/cache   available
Mem:       3.8Gi     1.0Gi     1.4Gi    33Mi     1.6Gi     2.7Gi
Swap:        0B         0B         0B

Disk Usage:
total      13G  6.6G  5.3G  56% -
          2025-11-02 18:54 (login screen)
robo      tty2      2025-11-02 18:54 (tty2)

=====
System information displayed Successfully!
=====
robo@robo:~/system_info_project$
```

## **Advantages**

<b>Advantage</b>	<b>Description</b>
<b>Automation</b>	Reduces repetitive manual checks by automating system info retrieval.
<b>Simplicity</b>	Easy to write and execute — perfect for beginners.
<b>Extensible</b>	Can be easily expanded with more system monitoring features.
<b>No Dependencies</b>	Uses only built-in Linux commands.
<b>Instant Output</b>	Displays essential system details in seconds.

## **Disadvantages**

<b>Disadvantage</b>	<b>Description</b>
<b>Limited Information</b>	Does not include OS or kernel version details.
<b>Text-Only Output</b>	No graphical or visual interface.

<b>Permission Restrictions</b>	Some system data may require elevated privileges.
<b>No Historical Tracking</b>	The script only shows current status unless redirected to a log file.

## Use Cases

- System Administrators: For quick access to uptime, memory, and CPU usage.
- Developers: To verify system resource availability before running builds or applications.
- Support Engineers: For rapid troubleshooting of performance-related issues.
- Server Maintenance Teams: To automate health monitoring of Linux servers.

## Future Scope

- Enable automatic logging of output for historical tracking.
- Integrate an alert system (email or SMS) for resource threshold breaches.
- Schedule regular execution through cron jobs for periodic monitoring.
- Extend capabilities to monitor remote systems using SSH automation.
- Develop a graphical dashboard using web technologies or Python integration.

## Conclusion

This project effectively demonstrates the application of **Bash shell scripting** for system automation.

By using simple Linux commands within a structured script, system administrators can efficiently monitor key performance metrics such as uptime, CPU, memory, and disk usage.

The project serves as a foundational step toward advanced automation tools and DevOps practices, emphasizing both practicality and scalability in Linux system management.

Github Link: [karndhiman/Project\\_Based\\_Learning\\_Linux](https://github.com/karndhiman/Project_Based_Learning_Linux)

The screenshot shows the GitHub repository page for 'Project\_Based\_Learning\_Linux' owned by 'karndhiman'. The repository is public and has 4 commits. The README file is displayed, containing a shell script for system monitoring and a link to a PDF report.

**Repository Overview:**

- Code (selected)
- Issues
- Pull requests
- Actions
- Projects
- Wiki
- Security
- Insights
- Settings

**Project\_Based\_Learning\_Linux Public**

**Code** main 1 Branch 0 Tags Go to file Add file <> Code

**Commits**

Author	Message	Date	Commits
karndhiman	Add files via upload	3c167e9 · now	4 Commits
	Screenshots	Add files via upload	6 minutes ago
	Project Report.pdf	Add files via upload	now
	README.md	Update README.md	7 minutes ago

**README**

## Project\_Based\_Learning\_Linux

## 🐧 Shell Scripting Automation for System Monitoring

A simple Bash script that automatically displays key system information such as uptime, CPU details, memory usage, disk utilization, and logged-in users. This helps automate routine system monitoring tasks for Linux users and administrators.

### 🌟 Features

View uptime, CPU info, memory, and disk usage  
See currently logged-in users