

Bash Scripting Suite for System Maintenance

NAME: BARUN KUMAR ROUT

REG NO: 2241018145

BATCH NO - 08

Objective:

The purpose of this project is to automate routine system maintenance tasks such as system backup, updates, cleanup, and log monitoring using Bash scripting. The suite is designed to make system administration efficient and error-free by performing repetitive operations automatically.

Environment Used:

- **Operating System:** Ubuntu (WSL on Windows)
- **Shell:** Bash
- **Editor/IDE:** Visual Studio Code
- **Tools Used:** tar, apt, grep, mkdir, touch

List of Scripts:

Script	Function	Description
Day1.sh	System Backup System Update and	Creates compressed system backup of /var/backups directory.
Day2.sh.	Cleanup	Updates packages, removes unused dependencies, and cleans cache.
Day3.sh	Log Monitoring Maintenance Suite	Scans system logs for error messages and generates an alert file.
Day4.s h	Menu Automated	Provides a user menu to choose and execute maintenance scripts interactively.
Day5.s h	Maintenance Log	Automates updates and logs the output into /var/log/maintenance.log.

Day1.sh — System Backup Script

Purpose: This script automates the system backup process by compressing important directories and storing the backup with a timestamp

Key Operations:

- *Displays the current user.*
- *Backs up /var/backups directory.*
- *Stores output in /home/<user>/system_backups/.*
- *Uses tar command for compression.*

Day1 Code Snippet:

```
#!/bin/bash echo "Current
```

```
user is : $USER"
```

```
source_dir="/var/backups"
```

```
dest_dir="/home/$USER/system_backups" mkdir  
-p "$dest_dir"
```

```
timestamp=$(date +"%Y-%m-%d_%H-%M-%S")
```

```
dest_file="$dest_dir/system_backup_$timestamp.tar.gz"
```

```
echo "System backup starting ..." sudo
```

```
tar -czf "$dest_file" $source_dir
```

```
if [ $? -eq 0 ]; then echo "Backup
```

```
done successfully!" echo "File
```

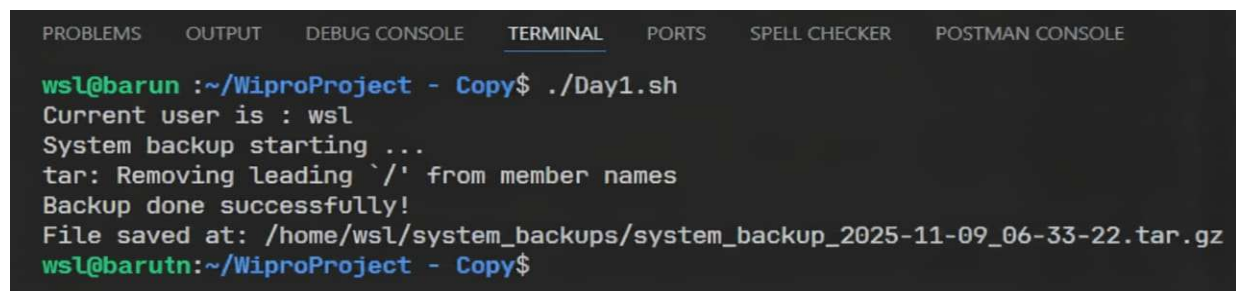
```
saved at: $dest_file" else echo
```

```
"Backup
```

```
failed!" exit 1
```

```
fi
```

Output Screenshot:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SPELL CHECKER  POSTMAN CONSOLE  
wsl@barun :~/WiproProject - Copy$ ./Day1.sh  
Current user is : wsl  
System backup starting ...  
tar: Removing leading `/' from member names  
Backup done successfully!  
File saved at: /home/wsl/system_backups/system_backup_2025-11-09_06-33-22.tar.gz  
wsl@barutn:~/WiproProject - Copy$
```

Day2.sh — System Update and Cleanup Script

Purpose: Automates the system update, upgrade, and cleanup operations.

Key Operations:

- Updates package lists.
- Installs available upgrades.
- Removes unused dependencies.
- Cleans cached files.

Day 2 Code Snippet:

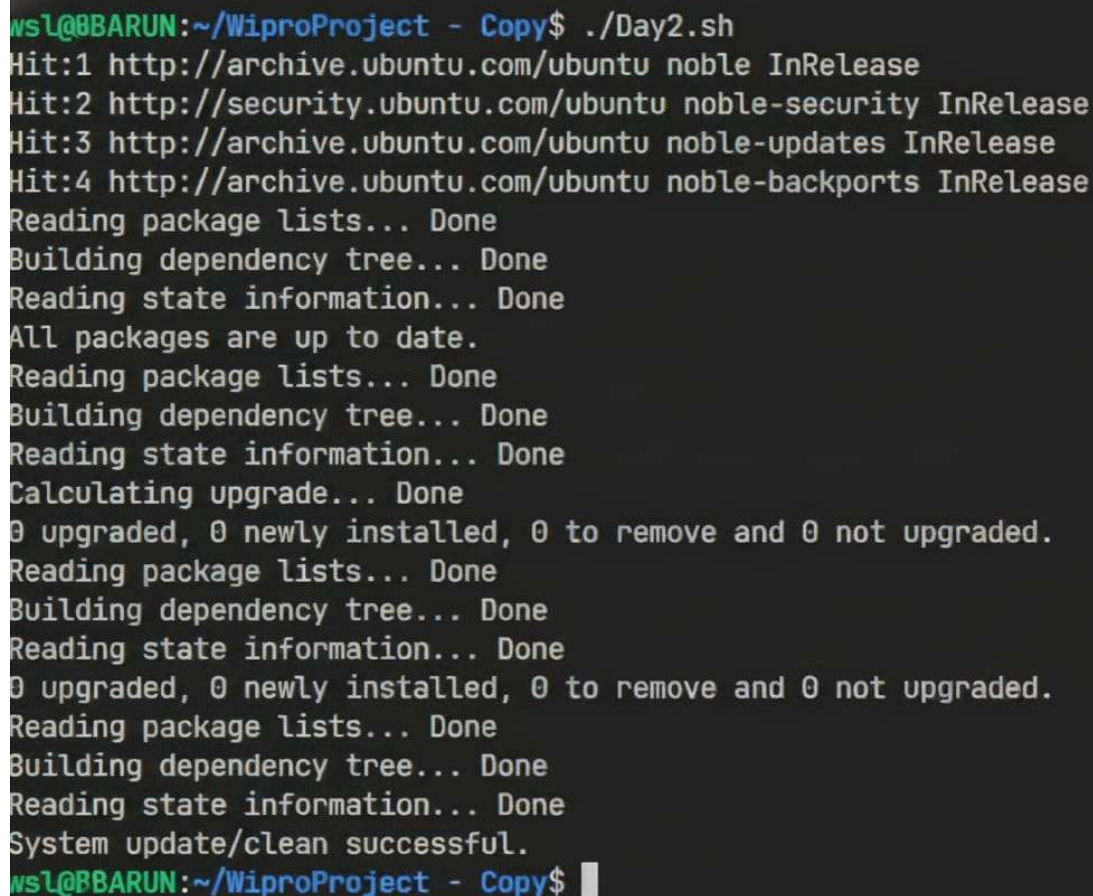
```
#!/bin/bash sudo apt update -y && sudo
```

```
apt upgrade -y sudo apt autoremove -y
```

```
sudo apt autoclean -y echo "System
```

```
update/clean successful." Output
```

Screenshot:

A terminal window screenshot showing the execution of a script. The prompt is 'ws1@BBARUN:~/WiproProject - Copy\$'. The script './Day2.sh' is run, resulting in several 'Hit' messages for Ubuntu repositories, followed by 'Reading package lists... Done', 'Building dependency tree... Done', and 'Reading state information... Done'. It then reports 'All packages are up to date.' and repeats the dependency and state checks. Finally, it reports '0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.' and ends with 'System update/clean successful.' The prompt returns to 'ws1@BBARUN:~/WiproProject - Copy\$'.

```
ws1@BBARUN:~/WiproProject - Copy$ ./Day2.sh
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
System update/clean successful.
ws1@BBARUN:~/WiproProject - Copy$
```

Day3.sh — Log Monitoring Script

Purpose: Monitors system logs for error messages and stores them in a separate alert file.

Key Operations:

- Checks if /tmp/alerts.txt exists; creates it if missing.
- Searches /var/log/syslog for “error” entries.
- Displays the detected errors.

Day 3 Code Snippet:

```
#!/bin/bash
LOGS="/var/log/syslog"
ALERT_FILE="/tmp/alerts.txt"

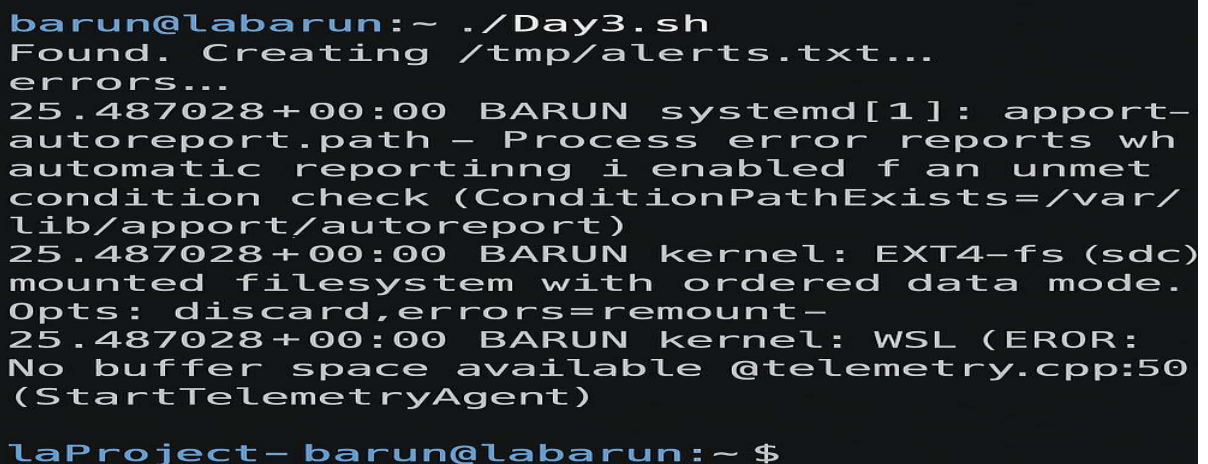
if [ ! -f "$ALERT_FILE" ]; then echo "Alert file
not found. Creating $ALERT_FILE..." touch
"$ALERT_FILE"
fi

echo "Checking logs for errors..." grep -i
"error" "$LOGS" > "$ALERT_FILE"

if [ -s "$ALERT_FILE" ]; then echo
"Errors found in system logs! ->" cat
"$ALERT_FILE" else echo "No
errors found."

Fi
```

Output Screenshot:

A terminal window showing the execution of the Day3.sh script. The prompt is 'barun@labarun:~'. The script runs './Day3.sh' and outputs 'Found. Creating /tmp/alerts.txt... errors...'. It then displays three log entries from /var/lib/apport/autoreport: 1. A process error report about an unmet condition check. 2. A kernel message about EXT4-fs (sdc) mounted with ordered data mode. 3. A kernel error message about WSL (ERROR: No buffer space available @telemetry.cpp:50 (StartTelemetryAgent)). The prompt returns to 'laProject- barun@labarun:~ \$'.

Day4.sh — Interactive Maintenance Menu

Purpose: Acts as the main controller, providing an interactive menu to run all other scripts from a single interface.

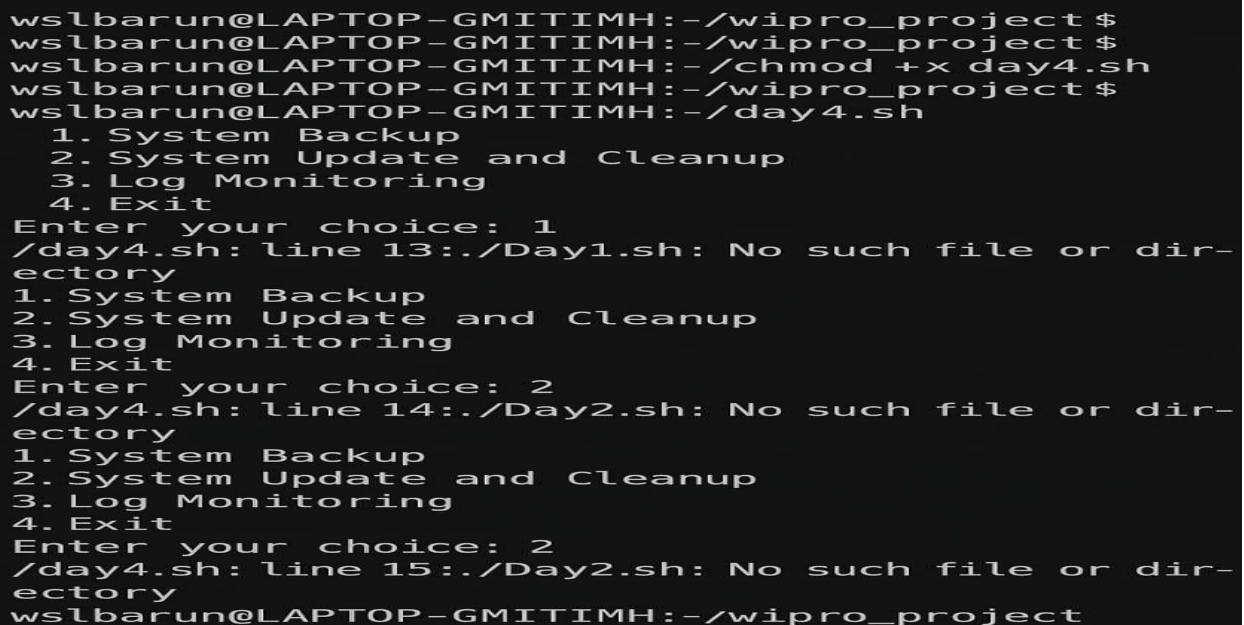
Key Operations:

- Displays a user-friendly menu.
- Calls other scripts (Day1.sh, Day2.sh, Day3.sh) based on user selection.
-
- Allows graceful exit.

Day 4 Code Snippet:

```
#!/bin/bash while true; do    echo "1.
System Backup"    echo "2. System
Update and Cleanup"    echo "3. Log
Monitoring"    echo "4. Exit"    read -p
"Enter your choice: " num
case $num in
1) ./Day1.sh ;;
2) ./Day2.sh ;;
3) ./Day3.sh ;;
4) echo "Exiting..."; exit 0 ;;
*) echo "Invalid choice! Please try again." ;;
esac
done
```

Output Screenshot:



```
wslbarun@LAPTOP-GMITIMH:~/wipro_project$
wslbarun@LAPTOP-GMITIMH:~/wipro_project$
wslbarun@LAPTOP-GMITIMH:~/chmod +x day4.sh
wslbarun@LAPTOP-GMITIMH:~/wipro_project$
wslbarun@LAPTOP-GMITIMH:~/day4.sh
 1. System Backup
 2. System Update and Cleanup
 3. Log Monitoring
 4. Exit
Enter your choice: 1
/day4.sh: line 13: ./Day1.sh: No such file or dir-
ectory
 1. System Backup
 2. System Update and Cleanup
 3. Log Monitoring
 4. Exit
Enter your choice: 2
/day4.sh: line 14: ./Day2.sh: No such file or dir-
ectory
 1. System Backup
 2. System Update and Cleanup
 3. Log Monitoring
 4. Exit
Enter your choice: 2
/day4.sh: line 15: ./Day2.sh: No such file or dir-
ectory
wslbarun@LAPTOP-GMITIMH:~/wipro_project
```

Day5.sh — Automated System Update with Logging

Purpose: Automatically updates the system and logs the maintenance details to /var/log/maintenance.log.

Key Operations:

- Records update start and completion time.
- Redirects both standard output and errors to the log file.
- Displays log contents after execution.

Day 5 Code Snippet:

```
#!/bin/bash
LOGFILE="/var/log/maintenance.log"
{
    echo "System Update at $(date)"
    sudo apt update -y && sudo apt upgrade -y

    sudo apt autoremove -y && sudo apt autoclean -y

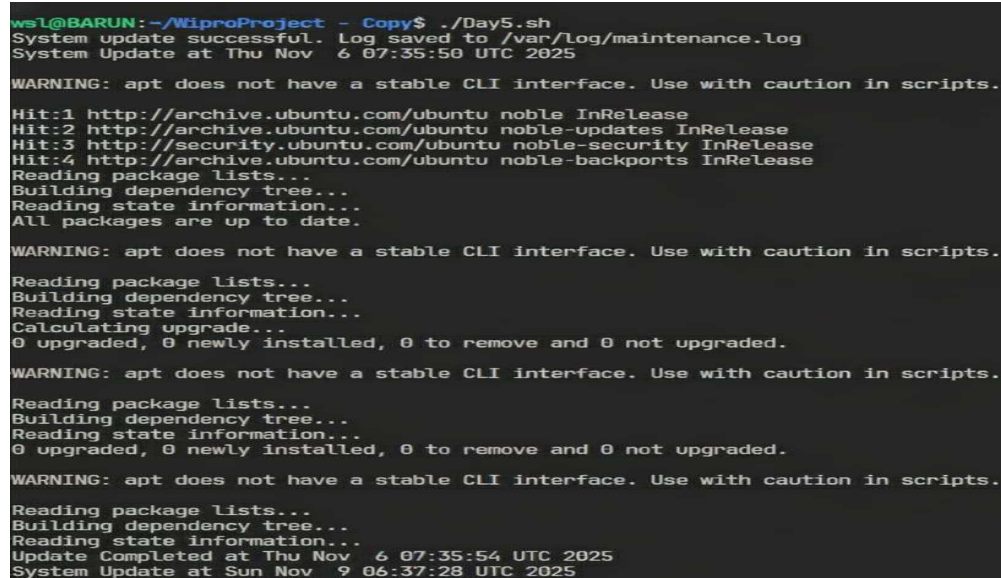
    echo "Update Completed at $(date)"
} >> "$LOGFILE" 2>&1

if [ $? -eq 0 ]; then    echo "System update successful. Log
                        saved to $LOGFILE"                cat $LOGFILE

else                    echo "System update failed. Check
                        $LOGFILE for details."

Fi
```

Output Screenshot:



```
ws1@BARUN:~/WiproProject - Copy$ ./Day5.sh
System update successful. Log saved to /var/log/maintenance.log
System Update at Thu Nov  6 07:35:50 UTC 2025

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists...
Building dependency tree...
Reading state information...
All packages are up to date.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
Update Completed at Thu Nov  6 07:35:54 UTC 2025
System Update at Sun Nov  9 06:37:28 UTC 2025
```

<i>Step</i>	<i>Script</i>	<i>Functionality</i>	<i>Result</i>
<i>1</i>	<i>Day1.sh</i>	<i>Backup system files</i>	<i>Success</i>
<i>2</i>	<i>Day2.sh</i>	<i>Update and cleanup system</i>	<i>Success</i>
<i>3</i>	<i>Day3.sh</i>	<i>Monitor logs for errors</i>	<i>Success</i>
<i>4</i>	<i>Day4.sh</i>	<i>Unified control menu</i>	<i>Success</i>
<i>5</i>	<i>Day5.sh</i>	<i>Automated update with logs</i>	<i>Success</i>

Conclusion:

This **Bash Scripting Suite for System Maintenance** effectively automates key system administration tasks — including backups, updates, cleanup, and log monitoring. The project demonstrates the power of Bash scripting in simplifying Linux system management. Each script is modular, reusable, and easy to extend for additional automation needs.