

Bash Scripting Suite for System Maintenance

Name : Piyush Kumar Sahoo

Regd. No:- 2241019096

Batch:- 2

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 |
|---------|---------------------------|---|
| File | | |
| Day1.sh | System Backup | Creates compressed system backup of /var/backups directory. |
| Day2.sh | System Update and Cleanup | Updates packages, removes unused dependencies, and cleans cache. |
| Day3.sh | Log Monitoring | Scans system logs for error messages and generates an alert file. |
| Day4.sh | Maintenance Suite | Provides a user menu to choose and execute maintenance scripts interactively. |
| Day5.sh | Menu Automated | |
| | 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:

The screenshot shows a terminal window with the following tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), PORTS, SPELL CHECKER, and POSTMAN CONSOLE. The terminal output is as follows:

```
wsl@PIYUSH:~/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@PIYUSH:~/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:

The screenshot shows a terminal window with the following tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), PORTS, SPELL CHECKER, and POSTMAN CONSOLE. The terminal output is as follows:

```
wsl@PIYUSH:~/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...
Building dependency tree...
Reading state information...
All packages are up to date.
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists...
Building dependency tree...
Reading state information...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists...
Building dependency tree...
Reading state information...
System update/clean successful.
wsl@PIYUSH:~/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:

```
ws1@PIYUSH:~/WiproProject - Copy$ ./Day3.sh
Alert file not found. Creating /tmp/alerts.txt...
Checking logs for errors...
Errors found in system logs! ->
2025-11-09T06:29:25.487028+00:00 PIYUSH systemd[1]: apport-autoreport.path - Process error reports when automatic reporting is enabled (file watch) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/autoreport).
2025-11-09T06:29:25.487032+00:00 PIYUSH systemd[1]: apport-autoreport.timer - Process error reports when automatic reporting is enabled (timer based) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/autoreport).
2025-11-09T06:29:25.487096+00:00 PIYUSH kernel: EXT4-fs (sdc): mounted filesystem with ordered data mode. Opts: discard,errors=remount-ro,data=ordered.
Quota mode: none.
2025-11-09T06:29:25.487378+00:00 PIYUSH kernel: WSL (216) ERROR: No buffer space available @telemetry.cpp:150 (StartTelemetryAgent)
ws1@PIYUSH:~/WiproProject - Copy$
```

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:

```
wsl@PIYUSH:~/NiproProject - Copy$ ./Day4.sh
1. System Backup
2. System Update and Cleanup
3. Log Monitoring
4. Exit
Enter your choice: 1
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_06-36-13.tar.gz
1. System Backup
2. System Update and Cleanup
3. Log Monitoring
4. Exit
Enter your choice: 2
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble 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
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.
1. System Backup
2. System Update and Cleanup
3. Log Monitoring
4. Exit
Enter your choice: 3
Checking logs for errors...
Errors found in system logs! ->
2025-11-09T08:29:25.487024+00:00 PIYUSH systemd[1]: apport-autoreport.path - Process error reports when automatic reporting is enabled (file watch) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/aut
2025-11-09T08:29:25.487032+00:00 PIYUSH systemd[1]: apport-autoreport.timer - Process error reports when automatic reporting is enabled (timer based) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/a
2025-11-09T08:29:25.487096+00:00 PIYUSH kernel: EXT4-fs (sdc): mounted filesystem with ordered data mode. Opts: discard,errors=remount-ro,data=ordered. Quota mode: none.
2025-11-09T08:29:25.487378+00:00 PIYUSH kernel: WSL (216) ERROR: No buffer space available @telemetry.cpp:150 (StartTelemetryAgent)
1. System Backup
2. System Update and Cleanup
3. Log Monitoring
4. Exit
Enter your choice: 4
Exiting...
wsl@PIYUSH:~/NiproProject - Copy$
```

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

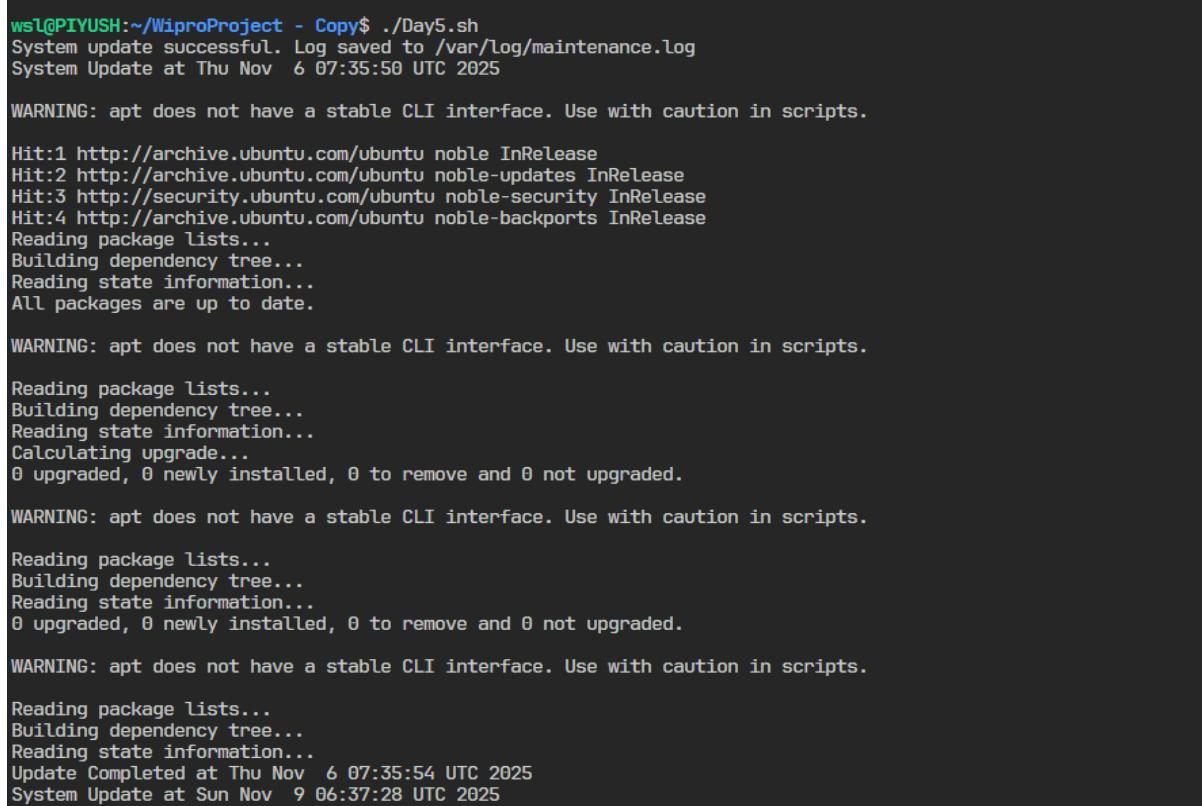
    sudo apt autoremove -y && sudo apt autoclean -y

    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:



```
wsl@PIYUSH:~/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
```

Project Workflow Summary

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

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.